Placeable and localizable elements in translation memory systems A comparative study

Dino Azzano

Placeable and localizable elements in translation memory systems
A comparative study

Inaugural Dissertation
for Conferral of the Degree of Doctor of Philosophy
by Ludwig Maximilian University of Munich
Center for Information and Language Processing

submitted by
Dino Azzano
from
Pordenone, Italy

Munich, 2011

Supervisor: Prof. Dr. Franz Guenthner

Co-supervisor: Prof. Dr. Uwe Reinke

Date of the oral defense: 22^{nd} July 2011

Contents

List of Tables

Acknowledgements

I would like to thank my supervisor, Prof. Guenthner, and my co-supervisor, Prof. Reinke for providing me with the opportunity to conduct this research and guiding its development. I am very grateful to Prof. Guenthner for his ability to grasp concepts quickly and for always thinking one step ahead. I also greatly appreciate Prof. Reinke's competence and insightfulness as well as his dedication to his profession.

I am very grateful to Michaela Geierhos for her advice on structural, typographic and administrative aspects of this thesis as well as for her informative lectures. I am also indebted to Maria Finnegan and Daniel Stein, my indefatigable native speaker advisors for English and German.

Many thanks also go to my manager Peter Kreitmeier for granting me maximum flexibility in my working hours over the last four years.

To my family in Germany and in Italy: your support has been invaluable.

Abstract

Translation memory systems (TM systems) are software packages used in computer-assisted translation (CAT) to support human translators. As an example of successful natural language processing (NLP), these applications have been discussed in monographic works, conferences, articles in specialized journals, newsletters, forums, mailing lists, etc.

This thesis focuses on how TM systems deal with placeable and localizable elements, as defined in 2.1.1.1. Although these elements are mentioned in the cited sources, there is no systematic work discussing them. This thesis is aimed at filling this gap and at suggesting improvements that could be implemented in order to tackle current shortcomings.

The thesis is divided into the following chapters. Chapter 1 is a general introduction to the field of TM technology. Chapter 2 presents the conducted research in detail. The chapters 3 to 12 each discuss a specific category of placeable and localizable elements. Finally, chapter 13 provides a conclusion summarizing the major findings of this research project.

Zusammenfassung

Dieses Kapitel ist eine Zusammenfassung der englischen Abhandlung und verschafft eine Übersicht über die durchgeführte Untersuchung. Für eine ausführliche Beschreibung wird auf die englische Version verwiesen.

I Motivation der Untersuchung

Translation-Memory-Systeme (TM-Systeme) sind Software-Applikationen, die zur Unterstützung des Übersetzungsprozesses dienen. Diese Dissertation befasst sich mit der Art und Weise, wie TM-Systeme platzierbare und lokalisierbare Elemente behandeln, siehe III. In der Fachliteratur ist eine vertiefte systematische Auseinandersetzung mit diesen Elementen noch nicht erfolgt. Das Ziel dieser Dissertation ist es, diese Lücke zu schließen sowie Verbesserungen vorzuschlagen, welche bestehende Unzulänglichkeiten lösen.

II Fachlicher Hintergrund

Die computergestützte Übersetzung unterscheidet sich von der maschinellen Übersetzung, insofern sie den Übersetzern bei der Erstellung der Übersetzung Hilfe bietet, aber nicht in der Lage ist, Übersetzungen selbst zu generieren. Eine zentrale Rolle in der computergestützten Übersetzung spielen Translation-Memorys: Das sind Datenbanken, welche Einträge in der Ausgangssprache sowie in mindestens einer Zielsprache enthalten. Zwischen den ausgangssprachlichen und den zielsprachlichen Einträgen besteht eine feste Zuordnung. Meistens sind diese Einträge Einzelsätze (Segmente).

Das Translation-Memory wird von einem TM-System ausgewertet: Dieses System führt eine Abfrage mit einem ausgangssprachlichen Text im Translation-Memory durch. Wird ein ausgangssprachlicher Treffer gefunden, kann die ihm zugeordnete zielsprachliche Entsprechung zur Weiterverarbeitung verwendet werden. Dabei ist die Suche unscharf, d.h. sie kann auch ähnliche Segmente (Fuzzy-Treffer) finden.

Eine zentrale Funktion von TM-Systemen ist die Berechnung der Ähnlichkeit. Sie erfolgt auf der Basis von Stringvergleichsverfahren, die jedoch nicht einheitlich sind. Deswegen schlagen unterschiedliche TM-Systeme auch unterschiedliche Ähnlichkeitswerte vor. Die Vergleichsverfahren berücksichtigen in jedem Fall lediglich die Oberflächenstruktur, es werden bei kommerziellen TM-Systemen fast ausnahmslos keine linguistischen oder semantischen Kriterien angewendet.

TM-Systeme sind Teil komplexerer Übersetzungsumgebungen, die weitere Funktionen bieten, wie z.B. Terminologieerkennung und -verwaltung. Während der Übersetzung eines Textes in einer Übersetzungsumgebung wird automatisch das Vorhandensein der gleichen oder ähnlicher Sätze im Translation-Memory geprüft und ggf. werden Treffer vorgeschlagen. Aus dieser knappen Beschreibung geht hervor, dass TM-Systeme im Bereich von Information Retrieval angesiedelt sind, weil sie auf vorhandene Informationen zurückgreifen und keine Texterzeugung vornehmen.

Jüngste Befragungen zeigen, dass TM-Systeme unter Übersetzern, Endkunden und Übersetzungsagenturen sehr verbreitet sind. Diese Verbreitung kann durch einige Vorteile der TM-Systeme erklärt werden. Zunächst kann die Wiederverwendung die Übersetzungskosten für Endkunden senken sowie die Produktivität der Übersetzer steigern, insbesondere wenn Texte ähnlich sind. Darüber hinaus kann eine erhöhte Konsistenz erzielt werden, weil unbeabsichtigte Übersetzungsvarianten für den gleichen Ausgangstext vermieden werden. TM-Systeme bringen allerdings auch gewisse Nachteile mit sich, z.B. Anschaffungskosten, Tendenz zur kontextgelösten Übersetzung usw. Der Bereich von TM-Systemen erlebt zurzeit tiefgreifende Entwicklungen, auf die jedoch in dieser Zusammenfassung nicht eingegangen werden kann.

III Platzierbare und lokalisierbare Elemente

Platzierbare und lokalisierbare Elemente können wie folgt definiert werden:

- Platzierbare Elemente sind Teile eines Dokuments, deren Inhalt in der Übersetzung unverändert bleibt.
- Lokalisierbare Elemente sind Teile eines Dokuments, deren Inhalt in der Übersetzung gemäß Standards oder vorgegebenen Regeln an das Gebietsschema der Zielsprache angepasst wird.

Ein Gebietsschema (oder Locale) umfasst eine Kombination aus Sprache und Region sowie weitere Vereinbarungen wie z.B. Zahlenformate. Die Anpassung platzierbarer und lokalisierbarer Elemente an das Gebietsschema erfolgt gemäß diesen allgemeinen Konventionen sowie ggf. gemäß firmenspezifischen Anweisungen. Die Position von sowohl platzierbaren als auch von lokalisierbaren Elementen im Zieltext hängt vom Satzbau der Zielsprache ab.

Platzierbare und lokalisierbare Elemente können wie folgt gegliedert werden:

- Zahlen
- Datumsangaben
- Eigennamen und Bezeichner
- URLs

- E-Mail-Adressen
- Tags
- Inline-Grafiken
- Felder
- Interpunktionszeichen

Die meisten Elemente bedürfen keiner zusätzlichen Beschreibung, bis auf Bezeichner, Tags und Felder. Bezeichner sind Namen, die Entitäten eindeutig identifizieren, wie z.B. Variablennamen und Produktcodes, welche keine Eigennamen im engeren Sinne darstellen. Tags sind Auszeichnungselemente, die Informationen zur Struktur oder zum Format eines Inhaltes vermitteln. Sie sind z.B. typisch für HTML. Felder sind Anweisungen, die unterschiedliche Auswirkungen haben können. Beispielsweise erzeugen sie Text, wie ein Datumsfeld, oder sie definieren bestimmte Aktionen für einen Text, wie ein Hyperlink.

Es ist nicht möglich, eine immer gültige, sprachunabhängige Unterscheidung zwischen platzierbaren und lokalisierbaren Elementen zu treffen. Die gleichen Elemente können entweder platzierbar oder lokalisierbar sein, je nach Zielsprache, Kontext usw. Zum Beispiel sind Zahlen in einer Übersetzung aus dem Deutschen ins Italienische in der Regel platzierbare Elemente. In einer Übersetzung aus dem Deutschen ins Englische muss jedoch beispielsweise das Komma als Dezimaltrennzeichen in einen Punkt umgewandelt werden. Darüber hinaus kann eine Umrechnung der Zahl nötig sein, wenn eine andere Maßeinheit, z.B. Meile statt Kilometer, verwendet werden soll. Weitere Beispiele könnten mit Datumsangaben, Eigennamen und allen anderen Elementen gemacht werden.

Die Erkennung platzierbarer und lokalisierbarer Elemente bringt einige allgemeine Vorteile mit sich. Wenn die Elemente erkannt und hervorgehoben werden, wird die Aufmerksamkeit der Übersetzer auf sie gelenkt. Dies ist nicht für alle Elemente notwendig bzw. sinnvoll, kann aber z.B. für Zahlen hilfreich sein. Die Übersetzungsumgebungen bieten außerdem die Möglichkeit, platzierbare und lokalisierbare Elemente in den Zieltext mittels Tastaturkürzeln zu übernehmen. Eine erleichterte Übernahme, insbesondere für platzierbare Elemente, beschleunigt die Übersetzung, weil sie Tipp- oder Diktierarbeit spart. Für diejenigen Elemente, die nicht aus reinem Text bestehen, wie z.B. Tags und Inline-Grafiken, ist diese Übernahmefunktion ohnehin notwendig. Weniger Tipp- oder Diktierarbeit bedeutet auch eine geringere Fehleranfälligkeit. Weniger Zeit muss bei der Qualitätsprüfung in die Berichtigung etwaiger Fehler investiert werden.

Die Erkennung ist umso hilfreicher, wenn sie mit automatischen Anpassungen kombiniert wird. Wenn sich ein neues Ausgangssegment vom Eintrag im Translation-Memory lediglich durch ein platzierbares oder lokalisierbares Element unterscheidet, kann dieses Element häufig automatisch angepasst werden. Automatische Anpassungen beschleunigen die Übersetzung, weil sie den Benutzer von der manuellen Anpassung mancher Fuzzy-Treffer entlasten. Ohne Erkennung wären automatische Anpassungen nicht möglich.

Die Erkennung platzierbarer und lokalisierbarer Elemente kann zudem das Retrieval verbessern. In der Regel werden für Unterschiede, die sich auf diese Elemente beziehen, niedrigere Abzüge als für rein textuale Unterschiede angewendet. Daher können in bestimmten Situationen mehr und bessere Treffer gefunden werden. Automatische Anpassungen platzierbarer und lokalisierbarer Elemente erhöhen ebenfalls den Ähnlichkeitswert, wobei aus ursprünglichen Fuzzy-Treffern 100%-Treffer werden können. Zum Schluss kann die Speicherung der Segmente in das Translation-Memory durch die Erkennung effektiver werden, weil ein einziger Platzhalter statt jedes einzelnen Elements gespeichert werden kann.

IV Durchgeführte Untersuchung

IV.i Testziele

Das Hauptziel dieser Dissertation ist es, zu untersuchen, wie kommerzielle TM-Systeme platzierbare und lokalisierbare Elemente behandeln. Da der Inhalt und die Struktur platzierbarer und lokalisierbarer Elemente sehr unterschiedlich sind, wurden verschiedene Testziele isoliert, die auf das gerade untersuchte Element abgestimmt sind.

Für alle platzierbaren und lokalisierbaren Elemente ist die Erkennung seitens der TM-Systeme eine zentrale Frage. Jedoch stellt die Erkennung für manche Elemente kein Problem dar, während sie für andere spürbar schlechter abschneidet. Aus diesem Grund ist die Erkennungsgenauigkeit das Haupttestziel für folgende Elemente: Zahlen, Datumsangaben, Eigennamen und Bezeichner, URLs sowie E-Mail-Adressen. In der vorliegenden Arbeit wurde hauptsächlich untersucht, ob die TM-Systeme sie grundsätzlich erkennen und ob alle Muster erkannt werden.

Das Retrieval und die automatischen Anpassungen stehen hingegen im Mittelpunkt für folgende Elemente: Felder, Inline-Grafiken, Tags und Interpunktionszeichen. Die Erkennung sowie die Markierung der Änderung (siehe IV.ii) wurden geprüft, der Schwerpunkt lag aber bei dem vorgeschlagenen Ähnlichkeitswert. Es wurde geprüft, ob feste Abzüge angewendet werden sowie ob die Segmentlänge, die Anzahl der Veränderungen oder die Position bzw. Art der Veränderung den Abzug beeinflussen. Schließlich wurden etwaige automatische Anpassungen bewertet.

Einige Nebenziele wurden ebenfalls berücksichtigt: Anzeige, Segmentierung, Bearbeitbarkeit bzw. Übersetzbarkeit sowie Personalisierbarkeit der platzierbaren und lokalisierbaren Elemente.

IV.ii Testmethoden

Je nach Hauptziel änderte sich die Testmethode. Wenn die Erkennung im Mittelpunkt stand, wurde ein Segment – das ein oder mehrere platzierbare und lokalisierbare Elemente enthält – im Editor des TM-Systems geöffnet. Bei manchen Systemen wurde die Erkennung markiert und konnte direkt geprüft werden. Bei anderen TM-Systemen war die Prüfung der Erkennung über einen Umweg möglich.

Wenn Retrieval und automatische Anpassungen bewertet wurden, wurde ein erstes Segment übersetzt und in das Translation-Memory gespeichert. Die weiteren Segmente, die sich mindestens durch ein platzierbares oder lokalisierbares Element vom ersten unterscheiden, wurden geöffnet, um den vorgeschlagenen Ähnlichkeitswert und etwaige automatische Anpassungen zu prüfen. Sie wurden jedoch weder übersetzt noch gespeichert, sodass das erste Segment immer als Vergleichsmaßstab galt.

IV.iii Testdaten

Die meisten Testdaten waren Teil echter Übersetzungsaufträge und kamen aus dem Bereich technischer Dokumentation. Es wurden verschiedene Quellen verwendet. Die wichtigste Quelle war eine Sammlung von vier Translation-Memorys. Die Ausgangssprache war Deutsch. Die Zielsprachen waren Englisch oder Italienisch, was jedoch für diese Untersuchung irrelevant ist. Im Deutschen waren insgesamt etwa drei Millionen Wörter vorhanden. Dieses Korpus wurde ergänzt, weil es nicht für alle Elemente eine geeignete Quelle von Testbeispielen war. Als Ergänzungen wurden ein Software-Handbuch sowie der Dump der englischen Version von Wikipedia verwendet.

Die besprochenen Korpora wurden nicht direkt getestet, sondern es wurden daraus Beispiele extrahiert, um Test-Suiten zu bilden. Die Extraktion der Beispiele erfolgte unterschiedlich. Für diejenigen Elemente, bei denen die Erkennung im Mittelpunkt steht, wurden mittels möglichst allgemeiner regulärer Ausdrücke Segmente gefunden, die ein relevantes Element beinhalteten. Die Segmente wurden anschließend geprüft, um falsche Positive zu entfernen und die verschiedenen Muster zu isolieren. Die Beispiele wurden ggf. leicht angepasst und anonymisiert, wobei diese Änderungen keinen Einfluss auf die Ergebnisse haben.

Für diejenigen Elemente, bei denen das Retrieval im Mittelpunkt steht, wurden geeignete Beispiele entweder manuell oder mit einfachen Suchmustern gesucht. Diese Beispiele dienten als Basis und wurden dann nach bestimmten Änderungstypen (Hinzufügung, Auslassung, Ersetzung, Umstellung) modifiziert. Darüber hinaus wurden eine und dieselbe Änderung in Segmente unterschiedlicher Länge sowie eine unterschiedliche Anzahl von Änderungen in das gleiche Segment eingegeben.

Für die eigentlichen Tests wurden die Beispiele entweder in MS Word- oder in HTML-Dokumenten eingebettet, um in den TM-Systemen bearbeitet zu werden. Beide Formate sind in der Übersetzungsbranche sehr verbreitet.

IV.iv Untersuchte TM-Systeme

Folgende TM-Systeme wurden getestet:

- Across Standalone Personal Edition [4.00]
- Déjà Vu X Professional [7.5.303]
- Heartsome Translation Studio Ultimate [7.0.6 2008-09-12S]

- memoQ Corporate [3.2.17]
- MultiTrans [4.3.0.84]
- SDL Trados 2007 Freelance [8.2.0.835] und SDL Trados Studio [9.1.0.0]
- Transit XV Professional [3.1 SP22 631] und Transit NXT [4.0.0.672.3]
- Wordfast [5.53] und Wordfast 6 [2.2.0.4]

Damit wurden die zur Zeit der Auswahl (Ende 2007) verbreitetsten TM-Systeme sowie ein paar damals weniger bekannte TM-Systeme berücksichtigt. Vor den Tests war es notwendig, die Einstellungen jedes TM-Systems zu prüfen und ggf. anzupassen, damit zum einen eine geeignete Vergleichsbasis geschaffen wurde, zum anderen die Möglichkeiten des TM-Systems voll ausgeschöpft werden konnten.

IV.v Verbesserungsvorschläge

Die vorgeschlagenen Verbesserungen haben zwei Hauptziele:

- Korrekte Schätzung des Benutzeraufwandes dank präziserer Abzüge.
- Verringerung des Benutzeraufwandes dank automatischer Anpassungen.

Für jedes platzierbare und lokalisierbare Element werden aus den Test-Ergebnissen Schlussfolgerungen gezogen sowie Verbesserungen vorgeschlagen. Die Art dieser Verbesserungen unterscheidet sich je nach untersuchtem Element. Ist die Erkennung das Haupttestziel, werden reguläre Ausdrücke vorgeschlagen, die eine genauere und zuverlässigere Erkennung gewährleisten. Zum Teil sind sie Eigenentwicklungen, zum Teil können bereits vorhandene wiederverwendet werden. Für alle anderen Ziele werden hingegen allgemeine Verbesserungen formuliert, die sich auf die festgestellten Unzulänglichkeiten konzentrieren.

V Ergebnisse der Untersuchung

Die Ergebnisse der Testreihen können am besten auf der Basis der vorgestellten Testziele, siehe IV.i, gegliedert werden. Allgemein lässt sich feststellen, dass in einer vertiefenden Analyse der Behandlung platzierbarer und lokalisierbarer Elemente seitens der TM-Systeme Mängel entdeckt wurden. Diese Mängel bestehen zum einen in unausgeschöpften Möglichkeiten, d.h. es fehlen nützliche Funktionen. Zum anderen sind vorhandene Funktionen unausgereift, weil z.B. die Erkennung nicht vollständig ist oder weil die Formatkonvertierung den zu übersetzenden Text nicht vollständig ermittelt. Im Grunde genommen sind die Lösungen für diese Probleme bereits vorhanden, so gibt es meist mindestens ein TM-System, das die Einzelaufgabe einwandfrei löst. Jedoch hat jedes untersuchte TM-System Stärken und Schwächen, sodass keines immer positiv abschneidet.

V.i Erkennung

Die Erkennung ist in etlichen Fällen nicht komplett zuverlässig. Bei Zahlen und Datumsangaben ist dies besonders deutlich und folgenschwer. Manche TM-Systeme bieten deren Erkennung bereits während der Übersetzung, siehe III. Zahlen werden zudem bei so gut wie allen TM-Systemen durch Qualitätsprüfungen geprüft: Wird eine Ziffer übersprungen, können Fehler in der Übersetzung unerwartet bestehen bleiben. Die Probleme bei der Erkennung liegen bei alphanumerischen Zeichenfolgen, bei Zahlen, die Dezimal- und Tausendertrennzeichen beinhalten, sowie bei Zahlen, die aus mehreren Ziffernfolgen bestehen. Eine reine Ziffernerkennung ist relativ einfach und einige TM-Systeme beschränken sich darauf. Sobald jedoch eine Zahlenerkennung angestrebt wird, sind ausgeklügelte Methoden notwendig und die Fehleranfälligkeit steigt. Andererseits bietet eine vollwertige Zahlenerkennung Funktionen, die sonst nicht möglich wären, wie z.B. automatische Zahlenumrechnungen.

Bei Eigennamen und Bezeichnern ist eine vollständige Erkennung ohne aufwändige linguistische und statistische Mittel nicht möglich. Trotzdem besteht die Möglichkeit, auf der Basis ihrer Oberflächenstruktur einen Anteil der Eigennamen und Bezeichner effizient zu erkennen. Brauchbare Muster sind komplett groß geschriebene oder gemischt geschriebene Zeichenketten, alphanumerische Zeichenketten sowie Zeichenketten, in denen Sonderzeichen vorkommen. Von dieser Möglichkeit wird jedoch nur bei zwei TM-Systemen Gebrauch gemacht, obwohl solche Elemente z.B. für eine Qualitätsprüfung relevant sein können.

Wenn URLs und E-Mail-Adressen als reiner Text vorkommen, werden sie in der Regel nicht erkannt, obwohl sie mit sehr guter Genauigkeit erkennbar sind. Sie sind zwar wesentlich seltener als Zahlen, Datumsangaben, Eigennamen und Bezeichner, aber ihre Erkennung kann im Rahmen der Qualitätsprüfung ebenfalls hilfreich sein.

V.ii Retrieval

Verschiedene TM-Systeme ermitteln unterschiedliche Ähnlichkeitswerte. Das gilt nicht nur für textuale Änderungen, wie frühere Untersuchungen bereits hervorgehoben haben, sondern auch für Änderungen, welche platzierbare und lokalisierbare Elemente betreffen. Manche TM-Systeme verwenden längenabhängige Abzüge, mit dem Ergebnis, dass kurze Segmente mit hohen Abzügen bestraft werden, obwohl der Anpassungsaufwand nicht höher als bei längeren Segmenten ist. Die Anzahl der Änderungen wird nicht von allen TM-Systemen berücksichtigt: Damit wird der Tatsache, dass der Anpassungsaufwand mit der Anzahl der Änderungen steigt, nicht Rechnung getragen.

Wenn die Änderung platzierbarer und lokalisierbarer Elemente keine Änderung von reinem Text mit sich bringt, sind feste Abzüge möglich und werden von etlichen TM-Systemen angewendet. Die Art der Änderung (Hinzufügung, Auslassung, Ersetzung oder Umstellung) spielt dabei eine untergeordnete Rolle.

Neben den bisher beschriebenen Problemen treten beim Retrieval auch Fehler auf: Es werden 100%-Treffer für Segmente angeboten, in denen sich Umstellungen gegenüber der Version im Translation-Memory ergeben haben. Andererseits werden bei minimalen

Unterschieden – z.B. bei einem geänderten Anführungszeichen – keine Treffer angeboten. Unverhältnismäßige Abzüge treten ebenfalls auf.

Der Algorithmus zur Berechnung des Ähnlichkeitswertes ist zwar nicht offen verfügbar, jedoch lassen sich bestimmte Abzüge vom Anwender personalisieren. Die Personalisierbarkeit variiert je nach TM-System stark und kann unter Umständen zu einer Falle werden. Beispielsweise können manche Abzüge auf 0% reduziert werden, obwohl damit der Ähnlichkeitswert den Anpassungsaufwand nicht mehr widerspiegelt.

V.iii Automatische Anpassungen

Im Allgemeinen funktionieren automatische Anpassungen bei Auslassungen und Ersetzungen gut, während sie bei Hinzufügungen und Umstellungen weniger erfolgreich sind. Die automatischen Anpassungen werden von den TM-Systemen unterschiedlich stark genutzt. Von einigen TM-Systemen werden sie dort nicht genutzt, wo sie von anderen erfolgreich eingesetzt werden, beispielsweise bei Unterschieden in Interpunktionszeichen.

Einige automatische Anpassungen platzierbarer und lokalisierbarer Elemente sind unvollständig, weil sie sich auf das Element beschränken. Etwa bei Auslassungen werden unter Umständen überflüssige Leerzeichen hinterlassen. Eine erfolgreiche Behandlung dieser Sonderfälle ist zwar möglich, aber nur sehr selten in den getesteten TM-Systemen zu beobachten.

Durch automatische Anpassungen können 100%-Treffer erzeugt werden. Die Tests haben jedoch gezeigt, dass falsche bzw. unvollständige automatische Anpassungen vorkommen. Aus diesem Grund sollen solche Treffer vor der Übernahme stets auf ihre Richtigkeit geprüft werden. Trotzdem sind automatische Anpassungen in den meisten Fällen erfolgreich und sparen Zeit, sodass sie zu einer Produktionssteigerung beitragen und ihr Einsatz ausgebaut werden könnte.

V.iv Unterstützung

In diesem letzten Abschnitt werden einige Themen aufgegriffen, die zu den Nebenzielen der Untersuchung gehören, aber dennoch wichtige Erkenntnisse bringen.

Die Benutzeroberfläche in einer Übersetzungsumgebung soll möglichst den persönlichen Vorlieben des Benutzers anpassbar sein. Trotzdem haben die Tests Unzulänglichkeiten der Anzeige aufgespürt, die in jedem Fall problematisch sind. Inline-Grafiken werden im Editor meistens mit einem Platzhalter angezeigt. Häufig ist dieser Platzhalter aber nicht aufschlussreich und gibt keine Information, für welches Element er steht. Dasselbe Problem betrifft auch Felder. Diverse Vorschaufunktionen ermöglichen zwar im konkreten Fall eine Prüfung, der zusätzliche Aufwand könnte aber dennoch vermieden werden.

Neben der Anzeige der platzierbaren und lokalisierbaren Elemente selbst ist auch die Anzeige der Änderungen relevant. Insbesondere in Verbindung mit Leerzeichen und Anführungszeichen kommt es vor, dass der Unterschied nicht markiert wird, obwohl er erkannt worden ist und ein Abzug angewendet wird. Andererseits werden z.B. ganze Wörter hervorgehoben, obwohl der Unterschied nur ein Anführungszeichen betrifft. Diese Probleme treten jedoch selten auf, die Änderungsanzeige ist also meistens zuverlässig.

Schwere Probleme bestehen hingegen bei den Formatfiltern, die ein Dokument in ein Format konvertieren, das im Editor der Übersetzungsumgebung bearbeitet werden kann. In diesen Tests wurden nur MS Word- und HTML-Dokumente bearbeitet, deswegen beschränken sich die Anmerkungen auf diese beiden Formate.

In HTML-Dokumenten sind gewisse Tags intern, d.h. sie kommen innerhalb eines Satzes vor. In MS Word-Dokumenten kommen Inline-Grafiken ebenfalls im Satzfluss vor. An diesen Elementen wird aber manchmal segmentiert, sodass Sätze unvollständig sind. Darüber hinaus werden manchmal Felder, die sich am Satzanfang befinden, vom Segment ausgeschlossen: Das ist ebenfalls problematisch, weil der Satzbau der Zielsprache ggf. eine andere Position des Feldes erfordert. Nicht immer können die abgeschnittenen Sätze manuell so erweitert werden, dass sie vollständig angezeigt werden. Bei manuellen Erweiterungen kann außerdem die Retrieval-Leistung nach einer Aktualisierung des Dokumentes suboptimal werden, wobei auf diesen Punkt an dieser Stelle nicht näher eingegangen werden kann.

Im HTML-Format haben gewisse Tags Attribute, deren Werte sprachabhängig sind und deswegen bearbeitbar sein müssen, um übersetzt werden zu können. Diese Bearbeitbarkeit ist wiederholt nicht gegeben. Zwar können die Formatfilter u.U. entsprechend angepasst werden, doch erfordern solche Anpassungen gute Kenntnisse und setzen voraus, dass der Fehler erkannt wird. Dies erfolgt häufig erst nach der Fertigstellung der Übersetzung und kann hohen Nachbesserungsaufwand mit sich bringen. Auch Felder in MS Word-Dokumenten zeigen ähnliche Unzulänglichkeiten, z.B. bei Hyperlinks kann nur der angezeigte Text, aber nicht das darunter liegende Ziel des Hyperlinks bearbeitet werden. Dadurch sieht der Hyperlink korrekt aus, der Sprung aber führt zum falschen Ziel. Weitere Beispiele könnten gebracht werden. Das Fazit ist, dass selbst Formatfilter für sehr verbreitete Standard-Formate nicht fehlerfrei sind und dass diese Mängel – neben weiteren Nachteilen – zu fehlerhaften Übersetzungen führen können.

Part I

Introduction

Chapter 1

Background

This chapter serves to two main aims. Firstly, it clarifies the motivational background of this thesis in section 1.1. Secondly, it provides a theoretical background in computer-assisted translation (CAT) and translation memory systems (TM systems), an understanding of which is essential in order to comprehend the conducted research. For this reason, definitions of common concepts (section 1.2) as well as a description covering TM systems in particular (section 1.3) are presented. Since this field is rapidly evolving, an account of ongoing trends is given in section 1.4.

1.1 Motivation

TM systems are well-known and widely-used tools in the translation industry, particularly for computer-assisted translation. Due to their commercial success, research in the field of translation studies has been devoted to them, resulting in surveys, manuals, articles, etc. Many of these works are cited in the bibliography and references to them are provided wherever they provide further information on topics that are just touched on briefly in this thesis.

This thesis will focus on text elements that are referred to here as "placeable and localizable elements". These are portions of a document that remain unchanged or are adapted according to specific conventions in the target language, e.g. inline graphics and numbers. For a more precise definition and more details, see 2.1.1.

The importance of these elements for the translation process and translation quality has already been recognized, see e.g. (McTait et al., 1999, 40-45), Joy (2002) and Oehmig (2006). However, no larger work has been devoted to placeable and localizable elements. This thesis aims to fill that gap. Systematic research has been conducted in order to assess existing shortcomings and determine best practices for handling these elements by means of TM systems. The results show that this subject was worth investigating: such elements are not always well-handled and there are major differences between the TM systems.

1.2 General terminology and definitions

As noted by (Somers, 2003, 1), the terminology used in the field of translation technology is rather fuzzy. The aim of this section is not to prescribe "correct" terminology, but to define the meaning of terms used later in this work, beginning with computer-assisted translation and the corresponding tools, followed by machine translation and associated techniques. This should allow misinterpretations to be excluded as far as possible. Definitions alone cannot in themselves provide insight into the many facets of the concepts described. Therefore, for further information, the reader is advised to consult the cited references. Works giving a deeper insight into the terminology "muddle" are Hutchins and Somers (1992), Bowker (2002), Reinke (2004), Quah (2006) and Lagoudaki (2008).

Computer-assisted translation (CAT): translation performed by human translators with the help of a variety of computerized tools.[1] Synonyms are machine-aided (or assisted) human translation (MAHT) and machine-aided (or assisted) translation (MAT). Sometimes human-aided (or assisted) machine translation (HAMT) is also included in CAT, see e.g. (Austermühl, 2001, 10) based on Hutchins and Somers (1992) as well as in references such as (Bowker, 2002, 4) and (Quah, 2006, 7). However, this is not deemed appropriate here because HAMT "has the machine as the principal translator – a feature that is closer to machine translation than to machine-aided human translation", (Quah, 2006, 7-8).

Translation environment [tool]: software package including a TM system as well as additional translation support systems.[2] While it always includes a TM system, other components can vary. Its function is to facilitate the translation process. Also called a CAT system, CAT tool, translator's workstation or integrated translation system.

Translation memory (TM): "a repository in which the user can store previously translated texts paired with their source text in a structured way", (Lagoudaki, 2008, 27). The repository can be a database or parallel files, see (Reinke, 2006, 63). The TM is therefore the main resource used by TM systems.

Translation memory system (TM system, TMS): (Lagoudaki, 2008, 31) provides the following definition:

> An application that links to a repository in which previous translations and their corresponding source text are stored in a structured and aligned way, so that any new text to be translated is searched for automatically and matched to the available resources associated with the system, in order for the system to be able to suggest a translation.

The TM system is the core component of a translation environment tool so that it is often used – *pars pro toto* – as a synonym. This thesis differentiates as far as possible between the two terms.

Terminology database: a repository in which monolingual or multilingual terminology entries are stored. The terminology database is distinct from the TM and is managed

[1] Definition adapted from (Bowker, 2002, 4).

[2] Definition adapted from (Lagoudaki, 2008, 27).

by a terminology management system, see (Arntz et al., 2002, 229-230). The entries have a highly customizable structure and – unlike TM entries – have to be manually added by the user.

Segment: when processing a text, TM systems split it up into units according to specific rules (segmentation). Usually, a segment corresponds to a sentence, but it can also be a paragraph.

Subsegment: a phrase, "a group of source words s that should be translated as a group of target words t", (Way, 2009, 26), also defined as a chunk.

Analysis: a breakdown that quantifies how much of a text has to be translated from scratch and how much can be reused from past translations. TM systems search the translation memory to see if parts of the document have already been translated. Different types of matches are detected, see 1.3.6 for further information. In addition, repetitions are calculated, i.e. how many segments occur more than once in the document.

Machine translation (MT): translation performed by a computer application that processes a source-language text and automatically produces a target-language text. Human intervention can be present or not, but the machine is the principal translator, see (Quah, 2006, 7-8).

Example-based machine translation (EBMT): an approach to MT that matches the source text against a repository containing paired examples in order to identify the corresponding translation segments (or subsegments) and then recombine these with the aim of creating a target text.[3] EBMT relies on probabilities for extracting symbolic translation knowledge (transfer rules), see (Carl and Way, 2003, xx).

Statistical machine translation (SMT): an "approach to MT that is characterized by the use of machine learning methods", (Lopez, 2008, 2). It "applies a learning algorithm to a large body of previously translated text [...]. The learner is then able [to] translate previously unseen sentences", (Lopez, 2008, 2). SMT systems "implement a highly developed mathematical theory of probability distribution and probability estimation", (Carl and Way, 2003, xix).[4]

Rule-based machine translation (RBMT): an approach to MT characterized by the use of linguistic rules in a series of processes that analyze input text and generate result text by means of structural conversions.[5]

[3] Definition adapted from (Lagoudaki, 2008, 26).

[4] A concise introduction to probabilistic approaches is provided by (Och, 2002, 4-9); Hearne and Way (2011) provide a detailed introduction to the principles of SMT. Within SMT, several diverging approaches are possible, see (Cancedda et al., 2009, 8-32) for an overview and Way (2010a) for a description of what is now the leading approach: phrase-based statistical machine translation. SMT does not *per se* exclude linguistic knowledge: "Although the early SMT models essentially ignored linguistic aspects, a number of efforts have attempted to reintroduce linguistic considerations [...]", (Cancedda et al., 2009, 3), as correctly predicted by Melby (2006). Recently, the difference between SMT and EBMT has been dwindling, see (Way, 2010a, 531), but the deviations that still exist between them are pointed out by (Wu, 2005, 216) as well as (Way, 2010b, 3).

[5] Definition adapted from Carl and Way (2003).

1.3 Overview of translation environment tools

The aim of this section is to give a brief overview of CAT and TM systems for novices. For the sake of clarity, some generalizations that do not account for special scenarios are unavoidable. Further reading is suggested where appropriate. Readers acquainted with the topic can proceed with 1.4.

1.3.1 Applications and modules

The basic architecture and processes of translation environment tools are described by (Lagoudaki, 2008, 38-47). Although the boundaries between **TM systems** and MT systems are becoming blurred (see 1.4.2), it is still important to make a distinction between them. Basically, TM systems cannot produce target-language text; they can only retrieve previous translations and suggest them. Translation retrieval is defined as "the process of retrieving a set of translation records from the TM which are calculated to be of potential use in translating the input [...]", (Baldwin, 2010, 196). This retrieval is flexible because it works not only for verbatim repetitions of segments, but also for similar segments. The similarity value is calculated using various means, see 1.3.6.2 for a discussion and references for further reading. A TM system does not suggest any translation for a segment for which no similar enough entry can be found in the TM. A "pure" TM system does not assemble translations and is therefore much more similar to information retrieval systems than to MT systems, see (Reinke, 2004, 58-60). On the other hand, an **MT system** is expected to produce a translation in any case. A TM system has a different purpose: it is expected to *help* the user by supporting translation in an interactive fashion with target language text suggestions that have typically been written or revised by a human translator.

A TM system is usually part of a larger software suite, the **translation environment tool**. This suite includes additional applications, see (Lagoudaki, 2009, 28). It is not necessary to devote much attention to this already well-described topic. However, suffice to say that any translation environment tool needs at least two components other than the TM system: file format filters and an editor. For most file formats (a major exception being MS Word), the translation cannot be done in the original application, e.g. Adobe InDesign, while using the TM system. Instead, the file has to be converted into another format that can be processed in the editor. For this conversion, **file format filters** – also known as format converters – are necessary. Their availability can be an important argument for purchasing one product instead of another. However, poor filter quality can be a major shortcoming, in particular (but not only) for newly developed translation environment tools, see (Geldbach, 2010c, 52 and 54) and (Geldbach, 2010b, 54). In addition, file format filters have to cope with the different versions of the same application (e.g. Adobe InDesign CS2, CS3, CS4, CS5).

In order to translate the files, an **editor** is necessary: this can be proprietary or external. Proprietary editors are part of the translation environment tool and do not exist as independent applications. In the case of external editors, a third-party application is adapted for translation purposes. The advantage of the external editor is that a familiar editing interface can be presented to the user. This is why widespread word-processing applications are used (nearly always MS Word, see Keller (2011)). The adoption of an external editor, the primary task of which is not translation, see (Chama, 2009, 37), also poses some difficulties, among others, “the reliance on a third-party tool that makes it difficult to do your own independent planning for your tool”, Zetzsche (2007a). Consequently, proprietary editors have been lately favored, see also 1.4.3.

It is of essential importance to distinguish between the translation memory and the terminology database. A **translation memory** is, as we have already seen, a collection of aligned texts (as a whole or split down to segments) that is usually updated continuously during translation and automatically queried with each new segment to be translated. This is not the case with a **terminology database**; it is a separate resource that is not automatically updated like the translation memory. Moreover, terminology database entries are usually shorter than complete sentences, yet their structure can be much more complex than a translation memory entry.

It is obvious that integrating TM systems and terminology systems yields benefits, and the two components are found in all current translation environment tools. During translation, the user accesses both at the same time. Still, they are populated and maintained in different ways. In addition, there is no automatic synchronization: the translation proposed by the terminology database is not – per se – automatically used in the translation memory. The terminology system can also be used independently, outside a translation environment tool for (even just monolingual) terminology management.

A TM system enables translation memories to be created, but does not include them. They can be either populated during translation, obtained from customers or language service providers for specific jobs, or purchased. The first method requires time and at the beginning will not yield many results. The second and the third have the disadvantage that the content is not self-generated and its quality might not meet expectations. Additionally, the third method involves also costs and is not completely free of copyright issues, see 1.4.7.

A reliable way to build up resources is to reuse one’s own past translations completed without a TM system. For this purpose, an **alignment tool** is included in translation environment tools. The source and target file are processed and a correspondence between them (usually at segment level) is established. The results of automatic alignments should be checked as errors are possible. This work can therefore be quite time-consuming. Further limitations apply e.g. to the file formats that can be processed. However, this step can prove extremely beneficial if future translations are comprised of updates, revisions or are otherwise similar to the aligned material.[6]

The management of complex translation projects involving multiple files, target languages, translators, deadlines and resources is an everyday routine at language service providers (LSPs) and language departments of large companies. Given that translation projects entail some specific aspects, full-fledged **translation management system** have been developed, Plunet BusinessManager being currently the best-known, see Sikes (2010)

[6] Translation memories can be also populated by aligning texts from the web, but this is a special type of alignment.

and Panzer et al. (2010). Many translation environment tools include a proprietary translation management system, with different levels of complexity depending on the target audience of the application version: freelance translators, LSPs or corporations. The features of these translation management systems range from relatively simple tools for freelance translators to full-fledged workflow automation solutions intended for large corporations.

When it comes to the translation of applications and texts meant for the digital world (online helps, graphical user interfaces, websites), the term localization is often used. Defining localization is difficult, see (Dunne, 2006, 1), but a working definition is:

> The process by which digital content and products developed in one locale [...] are adapted for sale and use in another locale. Localization involves: (a) translation of textual content into the language and textual conventions of the target locale; and (b) adaptation of non-textual content [...] as well as input, output and delivery mechanisms to take into account the cultural, technical and regulatory requirements of that locale. (Dunne, 2006, 4)

For more information on the term locale, see 2.1.1.1.[7] Consequently, the term **localization tool** has been introduced, see e.g. Esselink (2000), in order to refer to applications that are specifically designed for the translation of these resources. SDL Passolo and Alchemy Catalyst are among the most popular, see (Lagoudaki, 2006, 18), but others are marketed too, see Seewald-Heeg (2009). Their basic functionality is no different to that of translation environment tools. However, their distinguishing features (see also (Herrmann, 2011, 22) for an overview) affect the file formats processed because they include software source formats that are usually not supported by translation environment tools. Additionally, advanced preview functions are available and window/frame resizing can be carried out during the translation – in contrast with most translation environment tools, in which the layout can be usually adapted only after back-conversion to the source format. The quality assurance includes special checks (e.g. accelerators) that are typical of software applications. For a comprehensive description of the issues relevant to the localization process, see Esselink (2000). In fact, translation environment tools now support source formats that were typical of localization tools, e.g. SDL Trados Studio supports Java resources (.properties files). However, despite some convergence and the fact that basic versions of localization tools are sometimes included in translation environment tools, full-fledged localization tools are still separate applications.

Concordancers "allow translators to search through [...] parallel corpora", (Bowker and Barlow, 2004, 53), and output the searched string in its context as well as the corresponding passage in the target language. In this respect, they are not different from TM systems. However, the search is started manually. Consequently, concordancers serve more as a reference for translators seeking a suitable translation, usually for subsegments. A basic concordance function is implemented in existing TM systems; more advanced functionalities are provided by specific solutions as e.g. TransSearch, see Bourdaillet et al.

[7] I will not delve into the question of whether localization truly defines a separate activity or designs translation in a specific context, see Zetzsche (2010c).

(2010). A solution like Linguee makes use of the web as a parallel corpus, see Linguee (2011).

Translators use many other computer applications such as desktop searches, voice recognition software etc., which are not typical of the translation process. Therefore, although they are essential for everyday work, they are not listed here, but can be found e.g. in Austermühl (2001). A special note is necessary only for word processors because they are used as editors by some translation environment tools, see 1.4.3.

1.3.2 Market penetration

The most recent large-scale survey on the market penetration of translation environment tools dates back to 2006, see Lagoudaki (2006).[8] The overwhelming majority (90%) of the respondents were translators with 73% working as freelancers (Lagoudaki, 2006, 9). One of the key findings of this survey was that 82.5% of the respondents were users of translation environment tools (Lagoudaki, 2006, 15); this value is higher than in previous surveys with similar samples. The author identifies a correlation between this high percentage and the main specialization field of the respondents (technical documentation for 61%, (Lagoudaki, 2006, 12)) as well as their level of computer literacy (94% assessed their knowledge as good or excellent, (Lagoudaki, 2006, 11)). As pointed out in the survey itself (Lagoudaki, 2006, 6), some biases and the composition of the sample cannot be ignored, but its results are doubtlessly representative.

The success of translation environment tools is not limited to freelance translators, but also includes language service providers and companies. However, there is little large-scale, publicly available numeric data to support this assumption.[9] The survey presented by Zajontz et al. (2010) provides some valuable data regarding companies, although it is limited to the German market and does not exclusively refer to companies. 62% of the respondents report that they use translation environment tools; however, since many companies outsource translations, this figure may not account for those companies whose language service providers employ translation environment tools. Unfortunately, the presented data is not broken down between large, medium and small-sized companies and possible correlations cannot be evaluated.

1.3.3 Advantages

The use of translation environment tools affects the translation process at different stages and in several ways: a detailed account is given by (Reinke, 2004, 100-145). Therefore, it is a specific activity, different from the conventional human translation. García (2008) speaks of “TM-mediated translation”. This section is an attempt to summarize the main advantages of translation environment tools by expanding the description provided by (Esselink, 2000, 366-367).

[8] See (McBride, 2009, 46-58) for a detailed account of previous surveys.

[9] Access to the market studies of Common Sense Advisory, a translation and localization market research company, is restricted.

1.3.3.1 Productivity

Productivity gain through reuse of past translations is the best-known advantage of translation environment tools, see e.g. (Lagoudaki, 2008, 54) and Muegge (2010). Higher productivity implies that higher volumes can be processed by a single translator. The challenge is to quantify the gain and to clarify the situations where this applies.

A precise calculation of productivity gains from a translator's point of view is presented by Vallianatou (2005). However, this calculation applies only to a specific translator, to specific texts and at a specific point of time. Providing general percentages can be impressive, (Esselink, 2000, 366), but is questionable, see e.g. Macklovitch (2006) for a discussion of the not-so-trivial calculation of the productivity gains after the introduction of a translation support tool. In general, there is a broad consensus among translators on the fact that translation environment tools improve the productivity, see García (2006b), and this is presumably the key reason for their success. In addition, enhanced job satisfaction through less repetitive work has been reported, see (Lagoudaki, 2008, 56).

This productivity gain is particularly strong when repetitive texts have to be translated: "Translation memory is mainly suitable for technical documentation or content", (Esselink, 2000, 366). However, the use of translation environment tools can be beneficial also for e.g. patents, see Härtinger (2009) and Härtinger (2010). Repetitiveness is usually assessed at segment level and is particularly obvious in updates where much of the document remains the same. More controversial is the question as to whether productivity gains can also be achieved for – apparently – non-repetitive texts. This is discussed by (García, 2006b, 102-104): if the existing resources in the translation memory can be leveraged at subsegment level (see 1.4.1), productivity gains become realistic for those texts.

One of the most highlighted advantages is the reduction of translation costs. For translators, the cost reduction is measured in terms of time and effort, i.e. the productivity gain discussed above. From the point of view of end customers, cost is calculated mainly in terms of money. This reduction is achieved because existing translations lower the amount of text to be translated from scratch.

1.3.3.2 Consistency

Thanks to integration with terminology databases, greater consistency in terms of terminology is achievable, see (Esselink, 2000, 366). The prerequisite is that terminology has been previously defined, as pointed out in 1.3.1. Consistent formulations can also be obtained by exploiting the concordance search, see Muegge (2010). However, consistency alone does not guarantee quality, as bad terminology or bad formulations can be used consistently too. Still, their consistency makes them easier to correct once the error has been spotted.

Additionally, translation environment tools help users ensure better quality because they usually integrate quality checks, see Reinke (2009), e.g. checks for correct and forbidden terminology, omissions, forgotten translations, inconsistencies, repetitions, etc. In some cases, these checks are conducted interactively during the translation and contribute to quality assurance.[10]

1.3.3.3 Flexibility

The adoption of translation environment tools allows file types whose native application is not installed on the computer to be handled, see (Esselink, 2000, 366) and Muegge (2010). This means greater flexibility and reduced costs. As discussed in 1.3.1, translation environment tools come with format filters that convert the original format into a format that is suitable for the editor. However, there are some limitations: some format filters may be available only at extra cost or are not available at all, in particular for rare formats. In addition, after the translation process, a layout check (in particular for desktop publishing formats) can only be completed in the original application.

Translation memories and terminology databases are electronic knowledge resources that can be exchanged, disseminated and consulted more easily than on paper or in their original file format, see Muegge (2010). Stakeholders in the translation process can share them for a single project or on a regular basis. Moreover, this collaborative sharing can be in real time when server solutions are implemented, see also 1.4.7. The possibility to split up larger projects while ensuring a certain degree of consistency is pointed out by (Esselink, 2000, 366). This is a key advantage for jobs with high volumes and tight deadlines.

The analysis of the source text, see 1.2, is an advantage too because it assesses with precision the scope of new projects, and this was previously only possible to a limited extent, if at all. In addition, the coordination of complex projects is facilitated by translation management systems, see 1.3.1.

1.3.4 Disadvantages and solution strategies

Disadvantages are described together with some approaches used by translation environment tools to resolve them. These disadvantages are of general nature, and can be more or less serious depending on the adopted translation process or they might not apply at all to specific scenarios.

1.3.4.1 Investment

The first disadvantage for beginners is the learning curve, as discussed by García (2006b) and confirmed by (McBride, 2009, 116). This curve can be more or less steep depending on the application. The importance of the learning curve should be not played down as it is a major reason for non-adoption, see (Lagoudaki, 2006, 17), where it emerges that some translators purchased a translation environment tool but do not use it.

The necessary financial investment (with the exception of freeware and – in part – low-budget systems) can discourage adoption, particularly if not balanced by the expected

[10] Quality assurance (QA) and quality control (QC) are not synonyms: “Quality assurance is defined as the steps and processes used to ensure a final quality product, while quality control focuses on the quality of the products produced by the process”, (Esselink, 2000, 146).

benefits, see (McBride, 2009, 62). Not only the initial price but also the update and support policy play a major role: a more expensive product including support could be more valuable than a cheaper one without support. A more expensive product offering more features can be more beneficial than a cheaper one that only offers limited functionality. The best choice is highly scenario-dependent.

1.3.4.2 Missing context

One of the most commonly-cited disadvantages of working with translation environment tools is that no or little context is given for the matches coming from the TM, see (Macklovitch and Russell, 2000, 140) and (Zajontz et al., 2010, 52). This problem is addressed by some tools in different ways, and is briefly covered by Chama (2010b). On the one hand, some translation environment tools offer context-sensitive matches. These matches offer greater accuracy because the TM system recognizes that the segment being translated occurs in the source text between the same segments as in previously translated texts. Transit, on the other hand, follows a different approach because it does not save the translated segments in a database but keeps parallel texts: thus, it is possible to check whether the proposed match comes from the same or a similar context as the segment in the document being translated. MultiTrans has a similar approach to Transit because it saves all translated documents (as a whole) in a database.

The absence of context is not limited to the matches coming from the TM, but also applies to the text being translated. This problem is serious if only the portion of text to be translated (possibly a single sentence) is extracted from a document. For example, after analyzing one or more documents, SDL Trados 2007 allows source segments below a set similarity value to be exported. These segments are then passed on to the translator: thus, the text as a single unit is dismembered. This process is therefore not advisable and does not seem to be very common.

Even if the entire document is passed to translation, it is usually not displayed in a WYSIWYG fashion in the editor. Consequently, it is "difficult to see how translated text will be displayed in the final layout", (Esselink, 2000, 367), and, depending on the screen size and the translation environment tool, to see much of the text around the segment being edited. In order to overcome these disadvantages, several translation environment tools offer preview functions, e.g. in a separate window where the document being translated is displayed in real-time or on demand, the former being more helpful according to (Chama, 2009, 39). However, some limitations apply: firstly, this feature is not available for all supported formats and, secondly, it might be offered in an application (e.g. a web browser) other than the original one so that the preview layout does not reflect the final layout.

1.3.4.3 Cross-incompatibilities

The adoption of a translation environment tool can have far reaching consequences for all stakeholders involved in the translation process. If a company adopts a specific product, any LSP working for that company and the translators could be forced to adopt it too, see Geldbach (2011). This is due to the lack of standardization of exchange formats. While TMs can be exchanged between different translation environment tool thanks to the TMX standard, the XLIFF standard for documents already preprocessed by translation environment tools is not equally well supported, even though it has been gaining popularity, see 1.4.5. Translation environment tools usually convert the source file formats into a proprietary format that can be processed only in their editor, see 1.3.1.

Even the exchange of translation memories by means of TMX exports is not without loss, see (Geldbach, 2010b, 54) and (Zetzsche, 2011c, 6). The main problems arise from differences in segmentation, see the description of the SRX standard under 1.4.5, as well as from the handling of markup code.[11] (Zetzsche, 2011c, 6) quantifies this loss at 5% to 10%. On the basis of a TMX export, Zetzsche (2003) presents the different ways placeable and localizable elements are stored in the translation memory, taking formatting information and an embedded graphic as examples. For formatting, some TM systems store the concrete information, others just use placeholders. For embedded graphics, a placeholder is always saved instead of the actual graphic. Thus, the same element might be handled in different ways by different translation environment tools (or different versions of the same translation environment tool). The export to TMX level 2 – "where the standard also supports content markup [...]", (Wright, 2006, 267) – does not solve the problem of match value loss in data exchange, as highlighted by Zerfaß (2004) and Lommel (2006). (Lommel, 2006, 232) states: "At present, no tool offers a comprehensive solution to the problem of markup transfer [...]"; to my knowledge, this statement is still correct. For a more detailed description and some tests results specifically concerning the match value losses related to a TMX exchange, see Zerfaß (2004). For more information on the corresponding challenges for the developers of the TMX standard, see LISA (2005).

Vendor lock-in, i.e. dependency on a particular vendor (or even product), is a problem for different stakeholders: LSPs have to select translators depending on the translation environment tool. Freelance translators have to own and/or learn more than one translation environment tool in order to be able to collaborate with different direct customers and/or LSPs. This trend is clearly reflected in (Lagoudaki, 2006, 23): more than 50% of users worked with multiple translation environment tools. This fact magnifies some of the disadvantages discussed above: investment and learning curve. While some trends seem to make cross-platform and cross-product compatibility more realistic (e.g. standardization of exchange formats), others can potentially go in the opposite direction, see 1.4.7.

1.3.4.4 Overheads

The use of translation environment tools normally involves some management overhead. This overhead depends on the translation environment tool (and other factors): some translation environment tools require the creation of a project with several steps, others permit more straightforward procedures.

A possible overhead of the use of translation environment tools, see also (Esselink, 2000,

[11] For a definition of markup code, see 8.1.

367), is the revision required after conversion into the source format. When corrections are made, the TM has to be updated in order to reflect the correct version. However, this implies additional work. When systematic changes are necessary, e.g. to the terminology, the effort can be very high for large TMs. Additionally, it has to be considered that changes that are perfectly correct in the context of the revised document could be disadvantageous in other contexts. Proper TM maintenance ensures quality and error traceability, see (Chama, 2010b, 21), but can be time-consuming.

1.3.4.5 Price pressure

In 1.3.3.1, cost reduction thanks to the reuse of previous translations has been mentioned. However, key questions are who benefits from these reductions and what they entail. There is no single answer as different scenarios are possible. Generally, larger companies are aware of TM technology even if they outsource it to one or more LSPs. LSPs have to define discounts for high matches and pass them on to their translators.

Discounts are usually applied to higher fuzzy matches, repetitions and 100% matches, see (García, 2006b, 101-102). The discount percentage is a matter of agreement between the stakeholders in the translation process. Two drawbacks should be pointed out. Firstly, in some cases 100% matches are not supposed to be reviewed and are not paid. Some translation environment tools can lock these segments so that they cannot be edited after pretranslation, see Fairman (2010). Secondly, even more problematic are poorly-paid 100% matches and fuzzy matches coming from a poor quality TM, see also (Zetzsche, 2007b, 47). The effort needed to correct poor or out-of-context matches can outweigh the agreed discounted price. In the end, this can be compared with the problem of post-editing poor MT output described by Krings (1994). The quality of the TM is crucial, but is out of the control of an individual translator if several translators contribute to it.

To better leverage available resources, translators can build a large customer-independent TM, see (Zetzsche, 2010d, 33), but the probability of segment-level repetitions is low (putting aside for a moment further questions concerning style, terminology, etc.), see Chama (2010b). More matches can be achieved with subsegment retrieval, see Macklovitch and Russell (2000) and Chama (2010b), as this exploitation of available resources is entirely to the benefit of the translator. The availability of this functionality, see 1.4.1, could become a relevant factor in the choice of a translation environment tool.

To summarize, the translator may have to bear the main burden of cost reduction in some situations, not only in terms of discounted rates, but also as regards time spent on completing the translation. (García, 2006b, 104) also comes to a very similar conclusion.

1.3.4.6 Ownership

The question of the intellectual property and the ownership of the translation memory is much debated and affects freelance translators in particular, see (Wallis, 2006, 6). However, no definitive answer has been yet provided, see Esteves-Ferreira et al. (2009). A related disadvantage emerges when translators use remote-based TMs. In this case, they may not be allowed to retain the resources created during the translation: access to the remote resources is usually temporary and the granted rights are limited too (e.g. no import/export). Translators are effectively deprived of their work for later reuse, see (García, 2009a, 203).

1.3.4.7 Other issues

Current translation environment tools rarely incorporate linguistic knowledge. Retrieval in particular does not make use of linguistic information so that the results provided (and those disregarded) do not necessarily correlate with the human judgment, see Fifer (2007). This shortcoming is already well documented by Macklovitch and Russell (2000). However, rapid changes in this field are unlikely, see e.g. the skeptical stand taken by Benito (2009).

As in any piece of software, there are shortcomings arising from bugs or from the interaction with operating system and third-party applications. A cursory look at the discussion forums amongst translators (ProZ and TranslatorsCafé being the most popular) on specific translation environment tools reveals that such technical issues are recurrent. For example, (Esselink, 2000, 367) notes:

> Translation memory filters have not always been updated to support new versions of the file formats they process. As a result, translatable text is either not recognized, or markup is presented as translatable text.

This problem has also been observed in tests performed during the course of this study, see 8.2.3.3.

1.3.5 Similarity computation

The notion of similarity (the degree to which segments resemble each other) is of essential importance for TM systems. The commonest methods for assessing similarity are therefore presented.[12]

It is essential to stress that the comparison is always made between texts in the source language. In other words, the similarity in the target language is inferred from the similarity in the source language. The limits of this assumption are discussed by (Reinke, 2004, 267).

1.3.5.1 Comparison: macro units

Before describing the different techniques that can be used for the comparison, it is necessary to make clear that TM systems use different portions of texts (macro units, such as

[12] Extensive descriptions of the notion of similarity and of its measures are presented by Trujillo (1999), Reinke (2004) and Baldwin (2010). The descriptions in this thesis are based mostly on these three works and reference them where appropriate. In addition, a formal description of various measures of semantic similarity is given by (Manning and Schütze, 1999, 294-306). For an in-depth discussion of the concept of similarity, see in particular (Reinke, 2004, 187-269). The measure of similarity is crucial also in other areas, e.g. for automatic MT evaluation, see (Koehn, 2010, 222-228) and (Giménez and Màrquez, 2010, 201-213).

segments or strings) in order to determine similarity.[13] This distinction is closely related to segmentation. Most TM systems split texts into segments and store the segment pairs (source language segment and target language segment) in the TM. In this case, we have segment-based matching and the search for matches is limited to "the sequence of character strings of each segment", (Lagoudaki, 2009, 44). On the other hand, some TM systems store the full text. In this case, character-string-based matching can be performed in that the "similarity of equivalent continuous character strings" is calculated, (Lagoudaki, 2009, 44), see also 1.4.1. Since there is no constraint at segment level, longer sequences can be retrieved. MultiTrans uses string-based search algorithms, see (Gow, 2003, 34), unlike other translation environment tools that use segment-based search algorithms, see (Gow, 2003, 22). How the segment (for segment-based matching) or the string (for character-string-based matching) is subsequently processed is described in the following two sections.

1.3.5.2 Comparison: subunits and order sensitivity

When macro units are analyzed by the TM system, they are split into subunits. There are two possible types of subunits: characters and words.

If characters are used, the comparison is based on character-based indexing, "where the string is naively segmented into its constituent characters or character chunks of fixed size", (Baldwin, 2010, 196). Alternatively, if the macro unit is segmented into words, the comparison relies on word-based indexing, see (Baldwin, 2010, 196).

The number of subunits in common between the new source and each candidate in the TM is considered. The more subunits are shared, the higher the similarity. Additionally, this measure has to take into account the "differences in the length of the sentences being compared", (Trujillo, 1999, 62), which is known as normalization.[14]

The comparison can be order-sensitive or order-insensitive. There are two main different order-sensitive methods: Levenshtein distance and n-gram comparisons.[15] Levenshtein distance (a particular type of edit distance) measures the similarity "by the number of insertions, deletions and substitutions to convert one string into another", (Trujillo, 1999, 66). The number of operations[16] and the similarity value are inversely proportional as similar strings require less adaptations. Levenshtein distance can be applied to different types of units, ranging from single characters to whole words. In the latter case, Levenshtein distance is an order-sensitive alternative to the bag-of-words method described below. However, Levenshtein distance does not account for all situations and can be inaccurate in some cases. See (Baldwin, 2010, 205-207) for a discussion of possible enhancements.

[13] This description is mainly derived from Gow (2003) and Lagoudaki (2009).

[14] Normalization can have other meanings in natural language processing, see e.g. (Manning et al., 2009, 28).

[15] At this point it is only possible to give an overview of these techniques. For a more detailed, yet clear description, see (Trujillo, 1999, 66-67). The special technique of weighted sequential correspondence is described by (Baldwin, 2010, 205-207).

[16] (Baldwin, 2010, 203) also cites equality as a primitive edit operation. Unlike insertion, deletion and substitution, it is not associated with any cost. In addition, substitution can be seen as a composite operation, as it is the result of deletion and insertion, see (Baldwin, 2010, 203-205).

Nevertheless, most current TM systems are based on Levenshtein distance, see He et al. (2010) and Koehn and Senellart (2010).

N-gram comparisons split a macro unit into sequences of consecutive constituents, the grams (e.g. words or characters). The length n of the grams is variable and depends on the specific scenario. Two related units will have many of these n-grams in common, enabling a similarity measure. N-grams are widespread models in the field of natural language processing, see (Manning and Schütze, 1999, 192-225).

When the word order is not taken into account in the comparison, the term "bag-of-words" is used. However, according to (Baldwin, 2010, 199), "there is no TM system that does not rely on word/segment[17]/character order to some degree". This is confirmed by (Macklovitch and Russell, 2000, 139), where the effects of word-order differences are mentioned. Yet, it is not uncontroversial whether e.g. word order really improves the results: some experimental research confutes it, see Baldwin (2010). For a comparison of the results of different approaches, also considering computing speed, see Baldwin (2010), where it is suggested that the best approach might depend on the source language.

1.3.5.3 Enhancements for word-based methods

All comparison methods described so far are language-independent. If word-based methods are used, a naive approach considering words verbatim would lead to poor results for inflecting and agglutinative languages (e.g. Russian and Hungarian, respectively). Some methods are therefore required to cope with this problem. Possible enhancements are morphological analysis and the consideration of stop words, see (Fifer, 2007, 26-31).

Linguistically motivated morphological processing (lemmatization[18]) to identify roots and affixes can be applied, provided that the necessary linguistic knowledge is available, see (Trujillo, 1999, 64-65). This would allow for a more semantically-based comparison, which is not possible with the previously described methods, see (Reinke, 2004, 228). The lack of linguistic knowledge can be tackled with the use of stemming, "a process that strips off affixes and leaves [...] a stem", (Manning and Schütze, 1999, 132), by means of heuristic techniques, see (Trujillo, 1999, 64). However, for languages with frequent root changes, heuristic techniques will yield poor results.

A further enhancement is to exclude stop words from the similarity calculation, see (Trujillo, 1999, 63-64) and (Baldwin, 2010, 214). Stop words are extremely common words such as articles, prepositions and conjunctions that contribute only marginally to the semantic meaning. According to Zipf's law,[19] there is only a limited number of these words so that the effort involved in accounting for them is not as high as for morphological processing.

[17] Segment defines here what is more commonly known as a chunk.

[18] (Manning et al., 2009, 32) define lemmatization as the removal of affixes "with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the *lemma*".

[19] It is not possible here to describe Zipf's law and its fine-tuning. Please refer to (Manning and Schütze, 1999, 23-27) for further information.

Although to varying degrees, the consideration of stop words and morphological analysis require some linguistic knowledge, which is usually not available in current TM systems. There are exceptions (e.g. Similis of the manufacturer Lingua et Machina), but these cover only a limited range of languages. The above-mentioned comparison techniques that involve virtually no linguistic knowledge are therefore more adequate for tools such as TM systems which have to deal with a wide range of possible languages.

For proprietary software, it is not possible to explain how exactly similarity is calculated: "[it] is not usually made explicit in commercial systems, for proprietary reasons", (Somers, 2003, 38). Still, (Macklovitch and Russell, 2000, 139), (Bowker, 2002, 106), (Somers, 2003, 38-39) and (Lagoudaki, 2009, 43-44), among others, conclude that the calculation is performed by means of character comparison. The following statement still holds true: "As it stands, TM systems remain largely independent of the source language and of course wholly independent of the target language", (Somers, 2003, 40). Combined approaches and techniques are possible, see (Trujillo, 1999, 67) and (Fifer, 2007, 27). When it comes to their deployment, the computational cost of each method has to be considered.

1.3.6 Retrieval and matching

The following classification is based upon the match value calculated by TM systems when a source text segment is processed. The match value is a percentage value expressing the grade of similarity between the source segment in the text to be translated and the source segment in the TM.

1.3.6.1 100% matches

100% matches are also known as exact matches, see e.g. Macklovitch (2000) and O'Brien (2007), or perfect matches. However, the last term can be ambiguous because it is sometimes used as a synonym of context matches (see 1.3.6.5), which are a subset of 100% matches.

A 100% match means either that there is no difference between the segments or that these differences have been processed by the TM system by means of automatic adaptations. A 100% match might also be a segment merely containing the same set of words, in any order, see (Nübel and Seewald-Heeg, 1999b, 28). Although this notion (used by a machine translation tool investigated in that article) is criticizable, it points out that a 100% match does not necessarily reflect verbatim the segment in the TM.

100% matches should be segments that do not need any manual modification. However, in translation studies it is well known and accepted that the same utterance (segment) in the source language may have to be translated differently because of its context, see (Reinke, 2004, 236-238). TM system manufacturers have partly addressed this problem with context-sensitive solutions such as context matches, see 1.3.6.5.

1.3.6.2 Fuzzy matches

A fuzzy match is a match lower than 100% but higher than a particular threshold, the minimum similarity value.[20] The threshold can be adjusted by the TM system user. On the one hand, it should be high enough to avoid low matches for which the adaptation would entail more effort than for a translation from scratch. On the other hand, it should be low enough to maximize the retrieval of useful suggestions. If translators do not accept any of the proposed fuzzy matches, they have to translate the segment from scratch. If a fuzzy match is due to the modification of one or several placeable and localizable elements, automatic adaptations are possible depending on the TM system and on the element involved.

Fifer (2007) has empirically investigated the question as to whether the most useful match is the one with the higher fuzzy value (when more than one match is proposed). The results show that this is true approximately 60% of the time, (Fifer, 2007, 105). In other words, the machine ranking does not always correspond to the human ranking, as pointed out also by (Trujillo, 1999, 62): “The [similarity] is clearly a heuristic and therefore not guaranteed to return the sentence that a human would deem closest in meaning”. While idiosyncratic variance is possible, there are examples where the human ranking is consistently different. These findings imply that the pretranslation of fuzzy matches is not always beneficial, see (Wallis, 2006, 47).

1.3.6.3 No matches

When no match is proposed, there are usually two possibilities.

- The source text is transferred into the target language.
- The target language remains empty.

To my knowledge, no specific test has been carried out in order to assess which strategy is the best in terms of time spent and translation quality.[21]

Copying the source text has the obvious advantage that placeable and localizable elements are copied as well so that, for example, typing errors are excluded. On the other hand, the source text has to be overwritten with the translation and sentence fragments may have to be repositioned in order to reflect the target language syntax. These operations are error-prone and some translators feel uncomfortable with this way of translating. If the source text is not copied, more typing errors are possible, see (Foster, 2002, 15).

1.3.6.4 Subsegment matches

Some TM systems (e.g. Déjà Vu and MultiTrans) are able to retrieve portions of segments from the translation memory and propose its translation. The auto-completion suggestions

[20] For a brief introduction to fuzzy matching, see Sikes (2007); for a comprehensive work, see Fifer (2007).

[21] For some general considerations on translation speed depending on the match type, see O’Brien (2007).

of SDL Studio AutoSuggest and Déjà Vu AutoWrite can be considered interactive subsegment matches. This type of match can prove extremely helpful because it maximizes the retrieval, see 1.4.1 for further information.

1.3.6.5 Other matches

Some TM systems (e.g. memoQ) propose context matches – also known as in-context matches (Zerfaß, 2011, 16) – that are distinctively highlighted or are given a special match value (101%). These matches are basically 100% matches in the same context: the segment in the text to be translated comes between the same segments or at the same place as in previous documents that have already been translated and saved in the TM.

Austermühl (2001) and Lagoudaki (2008) describe the category of "full matches" and define them as follows:

> The segment in the TM differs from the source segment only in terms of variable elements, such as numbers, dates, times, currencies, measurements, and sometimes proper names. (Lagoudaki, 2008, 43)

This definition may be controversial because e.g. tags are excluded. Moreover, any full match is given a particular match value. This match value can be 100% or lower, according to the TM system penalty settings and capability of automatic adaptation, so that full matches cannot be distinguished from other matches because of their match value. In addition, a full match is a highly language-dependent notion: a change in numbers may not have any bearing on the text in English or Italian, but the same does not apply to Croatian or Russian, where the word to which the number refers takes nominative, genitive singular or genitive plural depending on the number. Consequently, this category is deemed potentially misleading because it takes for granted that other words are not affected by the modification: this can be true, but it does not have to be.

Finally, if a suggestion comes from the terminology database, there is a terminology match. It usually consists of – but is not limited to – one or few words. In this thesis, context matches and terminology matches will not be investigated as they are not relevant for the research aims.

1.3.7 Further reading

Some topics inherent or related to computer-assisted translation are not discussed in this thesis, but other works can be consulted if the reader would like to learn more.

The history of translation environment tools is very interesting, particularly because several years passed between conception and commercial implementation. The early stages are best described by Hutchins (1998). For a concise description, see (Reinke, 2004, 36-41) or (Lagoudaki, 2008, 32-37). The evolution of one commercial product is described by García (2005).

An overview of computer-assisted translation and its many tools is provided by Austermühl (2001), Bowker (2002), Somers (2003) and Quah (2006). The different components of a translation environment tool are described and some useful, not translation-specific applications (e.g. e-mail, FTP etc.) are also briefly covered.

Even though the technical details are no longer up to date, for a comprehensive comparison of commercial translation environment tools, see Massion (2005), which provides advice on the criteria to be considered when choosing a system. Similar magazine articles such as Zerfaß (2002a), Seewald-Heeg (2005) and citeKeller2011 are less extensive. Reviews of specific applications are regularly published in magazines such as MultiLingual and MDÜ, e.g. Sikes (2009), Massion (2009), García and Stevenson (2010) and Zerfaß (2011) as well as in translator newsletters as Zetzsche's Tool Box (formerly Tool Kit), and specific presentations are held at conferences, e.g. Brockmann (2009). Insights into more technical aspects are provided by McTait et al. (1999) and Reinke (2004). The theoretical discussion of translation technology in translation studies is summarized by (Quah, 2006, 22-56).

Machine translation is a topic with a vast bibliography: here it suffices to cite Hutchins and Somers (1992) as a starting point for non-statistical machine translation as well as Cancedda et al. (2009) and Koehn (2010) for statistical machine translation.

1.4 Trends

This section summarizes some trends that mainly concern translation technology.[22] For each trend, further reading is provided.

1.4.1 Subsegment retrieval

As mentioned in 1.3.3.1, see also Hunt (2003), repetition at segment level is not usual outside specific contexts such as documentation updates. It is much more frequent at subsegment level involving phrases and terminology, see Macken (2009) for an investigation supporting this statement. The challenge for TM systems is to exploit the TM in order to suggest translations concerning these repetitions.

Chama (2010b) and Zetzsche (2011a) summarize the TM systems already offering this feature, which is rather new in commercial systems. The idea was anticipated by Foster et al. (1996) and implemented in research projects such as TransType, see Foster et al. (2002), Langlais et al. (2002) and Macklovitch (2006). The basic functionality of this feature is concisely explained by (Macken, 2009, 201):

> In order to suggest matches at a sub-sentential level, the systems must be able to align source and target chunks (a non-trivial task); and must be able to identify (fuzzy) matches at sub-sentential level and have a mechanism to score multiple sub-sentential matches and select the best match.

[22] For more information, facts and figures on the translation market, Rinsche and Zanotti (2009) is recommended, albeit for the situation in the EU only.

Subsegment retrieval can be interactive or static. Interactive subsegment retrieval is particularly noticeable in auto-completion features that suggest phrases during typing. This feature – also known as typeahead – is in fact quite common in online search services, see (Puscher, 2011, 27). Static subsegment retrieval is performed when the segment is queried in the TM: along with segment matches, subsegment matches are retrieved and do not change as long as the same segment is edited.

SDL Trados (from version 2009) offers the auto-completion feature AutoSuggest, see Zetzsche (2009a). For this feature, a special dictionary of aligned phrases is extracted from an existing and adequately large TM, and these phrases are suggested to the users while they are typing, after the first letter – called "prefix" by Foster et al. (2002) – is entered. Déjà Vu (from version X2) offers the AutoWrite feature for interactive predictive translation; unlike AutoSuggest, AutoWrite works with the existing TMs. Similar to AutoSuggest and AutoWrite is the auto-completion function of Across (from version 5), see (Sikes, 2009, 18) and (Across, 2009b, 19-22). Auto-completion is also included in academic research projects, see e.g. Khadivi (2008), Koehn (2009a) and Ortiz-Martínez et al. (2010).

An example of static subsegment retrieval is the longest substring concordance function of memoQ, see (Massion, 2009, 26) and Chama (2010b). The CSB (Character String within a Bitext) approach to search and retrieval of MultiTrans, which identifies identical character strings of any length, also falls in this category, see 1.3.5, because it is not intrinsically bound to the segment level, see (Gow, 2003, 34-38). Similis is a further TM system which is able to present static subsegment matches, see Macken (2009).

To conclude, a clear improvement has been achieved with respect to the situation regarding subsegment retrieval depicted by Hunt (2003). It is likely that more and more TM systems will add subsegment retrieval in near future. The interactive variant seems to be more promising, see (García and Stevenson, 2010, 19), and preferable to solutions that present matches in a separate window, because it eliminates copy/paste overhead and is thus more effective, as pointed out by Benito (2009).

However, the drawbacks of too many or unhelpful suggestions should not be forgotten, see (García, 2006a, 107), Melby (2006), (Macken, 2009, 205) and (Macken, 2010, 134). To my knowledge, there is no extensive comparative evaluation of the functionality provided by different TM systems. (Macken, 2009, 206) reports that "a mechanism to filter out basic vocabulary words [...] would be beneficial", otherwise subsegment matches become distracting and consequently counter-productive. The importance of the domain-specificity (correlation between the training data and the text to be translated) for the completions to be useful is stressed by (Khadivi, 2008, 104).

Existing implementations can be fine-tuned in order to provide effective support and extended to languages not yet supported. To my knowledge, only statistical algorithms have been used so far: linguistic knowledge will be added only if it is found to provide much better results and is likely (or even bound) to remain limited to some major languages, see also Melby (2006) and (Zetzsche, 2011e, 26).

1.4.2 Interaction with MT

In a survey by Fulford and Granell-Zafra (2005), the usage rate of MT systems among the respondents was only 5%. It is difficult to find up-to-date representative data to quantify joint usage of TM and MT among translators. (Joscelyne and van der Meer, 2007c, 31) believe:

> As large content owners adopt and integrate MT technology into their standard TM-based translation processes, we shall also see rapid adoption of these technologies by translation practitioners as well.

The (Translation Automation User Society, 2009, 2) reports that 37% of a sample of 211 companies have adopted MT, but the representativeness of the sample could not be clearly identified.

MT is a response to the need for translations of high volumes of content that would otherwise remain untranslated and where the usefulness for the user (and not intrinsic faultlessness) is the primary objective, see Vashee (2010) and Massion (2011). Moreover, savings in terms of money and time can be achieved in comparison with other translation solutions, see Flournoy and Rueppel (2010).

TM and MT technologies can interact in two ways:

- The systems remain separate, but are used in the same process (multi-engine approach).

- The systems are fused together (hybrid approach).

The difference is clarified – despite sloppy usage of the terms elsewhere – by (Way, 2010a, 556):

> We make a distinction [...] between serial system combination (or "multi-engine MT") and truly integrated systems. In what follows, we assume that only the latter qualify for the label "hybrid".

1.4.2.1 Coexistence: multi-engine approach

The combination of the two technologies in the same process takes different shapes. The TM/MT multi-engine process usually uses SMT, see 1.2, as MT system of choice, but there are also examples of RBMT usage, see Hodász et al. (2004). The major trend seems to be towards post-editing[23] MT generated text, see (García, 2009a, 206-208) and García

[23] The concept of post-editing can be rather fuzzy. Different levels of expected quality require different levels of post-editing (partial or full), see Arenas (2010), where the challenges and tasks in an MT post-editing project are also discussed. Post-editing is usually associated with MT-generated texts, but sometimes it refers to editing of fuzzy matches from a TM, see He et al. (2010), although this use is questionable, see Zetzsche (2010a), because the modifications are applied to content generated by human translators and not by an MT system.

(2010b), although large-scale studies to back this assumption are – to my knowledge – not publicly available.[24]

The exact post-editing process can be defined in various ways, see also Kanavos and Kartsaklis (2010):[25]

- The text is completely preprocessed with MT and post-edited in a separate step.
- Post-editing is performed during interactive translation, also referred to as interactive machine translation (IMT).

The first process is reasonably transparent: after TM leveraging, the missing translations are generated by the MT system and the text is then passed to post-editing, see Massion (2011). Essentially, this solution can be implemented with any combination of TM system and MT system.

The second process is organized as follows: during interactive translation in a translation environment tool, the segments for which no match is retrieved from the TM are processed by the MT engine. This solution can be implemented by e.g. Across, memoQ, SDL Trados, Transit and Wordfast, see Zetzsche (2010b) and García (2010a).

However, more sophisticated ways of interaction are being investigated: Hewavitharana et al. (2005), Biçici and Dymetman (2008), Zhechev and van Genabith (2010) as well as Koehn and Senellart (2010) propose SMT systems that are able to process TM fuzzy matches and to "accommodate the differences with the source sentence to be translated", (Biçici and Dymetman, 2008, 454).[26] He et al. (2010) investigate the possibility of comparing fuzzy matches from the TM with MT-generated matches in order to determine which match is going to require less post-editing effort.

The combination of TM, MT and post-editing has been reported to improve speed and/or quality, see e.g. Guerberof (2008), Koehn (2009a), Joscelyne (2009b), Kanavos and Kartsaklis (2010) and García (2010a).[27] In the first two articles, however, the dependency of reported results on the quality of the raw machine translation is not discussed at all, although the findings of Krings (1994) still retain their validity: post-editing can be effective if the quality of the MT output is reasonably good. The language combination, the text, as well as the MT tool used and the available resources[28] influence the results. Customized MT systems will score better than uncustomized ("vanilla") MT systems, see Thicke (2011)

[24] Common Sense Advisory, see 1.3.2, conducted market studies in this field, see Common Sense Advisory (2011), but access to this information is restricted.

[25] This description aims at providing an overview of the multi-engine approach. In fact, the implemented process can be very complex, see e.g. Hudík and Ruopp (2011).

[26] This type of integration, in particular to what extent it improves the usability of fuzzy matches by human translators in a normal working setting, has not yet been comprehensively tested, see (Biçici and Dymetman, 2008, 455).

[27] The results of García (2010a) are controversial because of the test constraints. The test setting resembles a comparison of post-editing MT texts and translating from scratch in a traditional manner, without the full functionalities of a translation environment tool.

[28] Glossaries, data used to train the MT tool (for SMT) or the set of rules applied (for RBMT), as pointed out by Flournoy and Rueppel (2010) and Massion (2011).

and Massion (2011). The effective benefit obtainable from general purpose MT engines has been subject to some discussion and mixed reactions are reported, see e.g. Zetzsche (2010b).

It goes without saying that MT is not suitable for all text types: Circé (2005) presents some tests that support this conclusion. (García, 2006a, 106) states:

> Controlled language, natural language subjected to strict syntactical and semantic restrictions, may work well in some narrow fields (catalogues, technical specifications, meteorological reports and the like) that can then be conveyed into the other languages through machine translation with little editing.

The beneficial role of controlled language on MT post-editing has been tested and confirmed, see e.g. Thicke (2011).

A serious difficulty connected to a multi-engine TM/SMT approach where the SMT component takes charge of the translation of no matches is explained by (Simard and Isabelle, 2009, 120) and raised again by Zhechev and van Genabith (2010):

> Given that the SMT system used is presented with the “hard” translation cases (strings not seen in the TM) and is usually trained only on the data available in the TM, it tends to have only few examples from which to construct the translation, thus often producing fairly low quality output.

The risks of ineffectiveness are acknowledged by (Joscelyne, 2009b, 4): “post-processing MT can take longer than direct translation”. This uncertainty is stressed by (Arenas, 2010, 36): “There is definitely uncertainty about the gains when using MT and post-editing”. Focusing on quality, (García and Stevenson, 2011, 30-31) point out the fundamental trade-offs that choosing post-editing involves. Still, successful implementations of a post-editing process are feasible, see e.g. Plitt and Masselot (2010) and Flournoy and Rueppel (2010).

Translators’ dislike or even resistance to post-editing has been reported, see (Koehn, 2009a, 261), García (2010a) as well as Kuhn et al. (2010) (although without concrete numbers). Poor MT quality can be the most obvious reason, but also the MT system’s incapacity to learn from corrections is cited as a major drawback. The lack of match rating allowing the translator to rapidly assess the reliability of an MT-generated match – and thus the effort involved in its editing – contributes to low acceptance, see (Simard and Isabelle, 2009, 120). These scores are known as confidence estimations and specific metrics such as human-targeted translation edit rate (HTER) have been developed, see (Specia and Farzindar, 2010, 33), but are still being refined and tested, see Raybaud et al. (2011) for an up-to-date overview. Their aim is to filter out and thus prevent the post-editing of bad quality translations, which lead to the ineffectiveness mentioned above. Overall, suitable integration of MT in translation practice is still being investigated, see also Karamanis et al. (2010) and Karamanis et al. (2011).

1.4.2.2 Convergence: hybrid approach

The convergence of TM and MT can be seen in TM systems that use MT techniques to output suggestions.[29] Because of its similarity with TM, EBMT is the MT technique of choice, see (Somers and Diaz, 2004, 5-18). The main purpose of integrating EBMT is to better exploit subsegment matches, see Simard and Langlais (2001) and (Somers and Diaz, 2004, 16). Déjà Vu is a TM system that claims to already implement EBMT techniques in its Assemble feature, see (Somers and Diaz, 2004, 16-18) for a discussion of the functionality and its limitations. (García, 2006b, 103) reports the mixed reactions of users. (García and Stevenson, 2010, 19) draw a comparison with the AutoSuggest function of SDL Trados Studio – covered in 1.4.1 – and favor the latter because it is less intrusive and more efficient.

1.4.2.3 Outlook

(Zetzsche, 2007c, 30) states that "the gaps between [TM and MT] will dissolve" and explains the reasons for this merge. Nogueira and Semolini (2010) foresee that post-editing MT output will be an increasingly important activity for translators. (García, 2009b, 29-30) states that the convergence and integration of TM and MT should be part of future translation technology research and points out:

> The key question for localization now is when, under what conditions, and for what type of task, controlled language plus TM plus MT plus post-editing will produce equal quality faster and more cheaply than the current TM model. (García, 2009b, 30)

Finally, confidentiality issues connected with the use of publicly available online MT services will play an important role, too, in the evolution of TM/MT interaction, see Nogueira and Semolini (2010) and Zetzsche (2010b). Confidentiality is one of the reasons why large companies deploy their own MT services, see (Porsiel, 2009, 424) and (Porsiel, 2011, 36).

1.4.3 Editors

As already described in 1.3.1, translation environment tools use two different types of editor: external or proprietary (internal). The trend is moving towards abandoning external editors in favor of internal ones and it was already recognized by Zetzsche (2007a), where some reasons are provided. Mainly, from the point of view of software developers, proprietary editors give more control and more flexibility with respect to product development, product distribution and supported features. SDL Trados Studio 2009 and Wordfast 6 confirmed this trend: MS Word can no longer be used as an editor. Interestingly, both are still equipped with the previous versions that support MS Word. MultiTrans, at least up to version 4.4, also provides both possibilities: along with MS Word, an XLIFF editor can be used.

[29] MT systems that integrate TM leveraging are not covered in this thesis, see 2.1.2.

Chama (2010b) points out that the trend away from MS Word (or similar word processors) will eventually go towards having the editor integrated in a web browser. However, so far, browser-based solutions do not seem to be very popular.[30] Appearance and functionality can vary significantly and customizability for the individual user is an open issue.[31] For further information on translation environment tools and the web, see also 1.4.7.

1.4.4 Formats

(Lagoudaki, 2006, 12) provides an overview of the most widespread source file formats translated with translation environment tools. Although some formats are reported to be gaining ground, it is in fact difficult to find up-to-date statistical evidence for these trends. For example, (Chama, 2010a, 17) claims that markup formats have been on the rise, but does not provide any figures. (Dunne, 2006, 3) states that the spread of electronic devices and of the Internet "fueled the proliferation of an ever-expanding variety of 'content' to be localized in an ever-increasing number of formats [...]". Unfortunately, the formats involved are not discussed further. (Nagel et al., 2009, 19) and (Jüngst, 2010, 3) claim that audiovisual translation (with a strong focus on subtitling and dubbing, but including also game localization, animations, videos and – to some extent – apps for smartphones and tablet PCs) has been gaining popularity, but without numerical evidence.[32] The shift from text to video particularly for user training is described by Deschamps-Potter (2010).

However, it is misleading to think that new applications and products generate per se new formats to be dealt with. Regarding the localization of computer games, (Dietz, 2006, 121) states:

> The texts to be translated might be included in plain text files, MS Word documents, MS Excel spreadsheets, MS Access databases, HTML code, source code, or be part of bit-mapped graphics.

In fact, nearly all of these formats are not new at all. A similar picture emerges in (Deschamps-Potter, 2010, 39), where again MS Word is used for nearly all written texts of a multimedia project.[33] A further example: the localization of apps for the mobile operating system Android is based on XML files, see (Junginger, 2011, 22).

Even absolute growth (more content in a particular format is being translated) does not necessarily indicate relative growth: the share of that format may still be shrinking because of its slower growth compared to other formats.

To summarize, it is impossible to provide an accurate overview of the trends concerning formats without a specific diachronic survey. Very broadly speaking, a trend towards XML-based formats is recognizable:

[30] It is difficult to find representative usage data.

[31] The importance of customizability is stressed e.g. by (Chama, 2009, 39).

[32] The difficulty of gathering data in these two fields is pinpointed by Riggio (2010).

[33] This fact is confirmed by (Lagoudaki, 2008, 166), where game-specific formats as well as subtitling files are seldom processed and do not feature strongly in the list of formats for which users report a need for support (Lagoudaki, 2006, 12).

- XML-based formats replace older ones, see e.g. the DOCX format introduced with MS Word 2007.
- It is possible to export the text content of a document into an XML-based format, see e.g. IDML for Adobe InDesign.

The reason for the popularity of XML is its versatility, confirmed by (Lagoudaki, 2008, 165):

> Using XML as a gateway format to convert to/from any other format can be the solution, since many applications nowadays can export their files into an XML-based format which is translatable.

Translation environment tools currently have to support many file formats. If the XLIFF exchange format, see 1.4.5, were to become a standard export format for different applications, see Mateos (2011), file format filters would be no longer necessary. However, it is unlikely that this will take place for the most common formats and it can be almost certainly excluded for exotic formats.

1.4.5 Standards

In the translation market, different XML-based standard formats have been developed for facilitating the exchange of data and avoiding vendor lock-in, see 1.3.4.3. As pointed out by (Lommel, 2006, 228), the information contained in the TM or in the terminology database is "the core of a company's information assets (at least in the multilingual/multinational arena)" and therefore its availability is of crucial importance. This point is stressed also by (Wright, 2006, 267). In addition, (Lommel, 2006, 230) estimates:

> [...] the value of the information contained in a TM database exceeds the cost of the tools themselves by several orders of magnitude, a fact that makes independence from a specific tool highly desirable.

The most important standards are:[34]

- TMX (Translation Memory eXchange): exchange format for TMs.
- XLIFF (XML Localization Interchange File Format): exchange format for documents to be translated or just translated.
- TBX (TermBase eXchange): exchange format for terminology databases.

[34] (Krenz, 2008, 206-281) probably provides the most detailed account of XML-based standards. For more concise accounts including, but not limited to, XML-based standards, see Lommel (2006), (Kleinophorst, 2010, 38-42) and Gough (2010). (Lommel, 2006, 230) also draws an important distinction between vendor-neutral and de facto standards. A de facto standard is a format used by virtually everyone, but controlled by a specific company. Those presented here are exclusively vendor-neutral standards. Wright (2006) provides an overview of standards relevant for, but not specific to, the translation field, e.g. Unicode.

- SRX (Segmentation Rules eXchange): exchange format for segmentation rules used by translation environment tools when processing documents.[35]

Some of these standard formats are well established, notably TMX and, more recently, XLIFF.[36] They are supported by almost all translation environment tools as well as most localization tools, see Mateos (2011), but the quality of the implementations is not always satisfactory, see e.g. (Lieske, 2011, 51) for a discussion of XLIFF. Others are occasionally supported, namely TBX[37] and SRX, see e.g. (Zerfaß, 2011, 19-20). An – at the time of writing – up-to-date account of the support of open standards declared by translation environment systems is provided by Gough (2010), where also the availability of open interfaces in form of application programming interfaces is covered.

It is likely that the support of standard formats will increase, see (Massion, 2009, 26) and Gough (2010). Furthermore, technologies are evolving and newer versions of the standards will be released to address known shortcomings and new requirements, see e.g. (Lieske, 2011, 52). Nevertheless, even when standard exchange formats are used, there is no guarantee that the exchange is without loss, see (McBride, 2009, 27) and 1.3.4.3. Seamless data exchange as well as interoperability between translation environment tools remain important targets.

However, it would be simplistic to envisage a translation market ruled by standardized formats. Other types of vendor lock-in that are becoming prevalent, see 1.4.7, constitute a move in the direction of non-standardization and non-interoperability.

1.4.6 Integration with other software

In 1.3.1, it has been pointed out that current translation environment tools package different applications. Still, some applications do not belong to the core of translation environment tools.

Authoring tools, for example, are intended to ease the creation of documentation. Therefore, they are not primarily intended to be used by translators. One of the basic ideas behind the integration of authoring tools and translation environment tools is reusing what has been written before (so that less content has to be translated) and keeping new formulations as close as possible to previous ones (so that fuzzy matches are likely to be produced). The integrated tools offered e.g. by Across,[38] SDL Trados and Star are called authoring memory systems or authoring memories. It is not clear if additional translation environment tools are going to offer similar products, see (García and Stevenson, 2010,

[35] The impact of differences concerning segmentation should not be underestimated because they "can result in the loss of several percentage points of effectiveness in TM leverage", (Lommel, 2006, 232). Consequently, "these few percentage points' worth of lost 100% matches could translate into a direct cost in the tens of thousands of dollars [...]", (Lommel, 2006, 233).

[36] Anastasiou (2010) presents a small survey on the level of awareness concerning XLIFF.

[37] Some reasons for the slow adoption of TBX are provided by (Lommel, 2006, 234) and (Wright, 2006, 269).

[38] For more information on the specific integration in Across of authoring tool and TM system, see Rösener (2010).

19). However, this integration seems crucial for products aimed at large-scale enterprises, in particular when combined with the adoption of controlled language (see Geldbach (2009) for a discussion of controlled language checkers, applications that allow for sophisticated linguistic checks).

When considering the means of inputting a translation, it is natural to think about the keyboard. However, the use of speech recognition is nothing new. (Fulford and Granell-Zafra, 2005, 9) note that "few respondents made use of voice recognition software", but give no percentage. Zetzsche (2007a) pleads for a better integration between translation environment tool editors and speech recognition applications and criticizes that the latter "are often less than optimal". Interestingly, a more recent survey also reports that "discussions relating to the use of voice recognition software (VRS) and translation environment tools were not extremely common [...]", (McBride, 2009, 110). The last few years do not seem to have witnessed any significant shift in this field. However, it is at least conceivable that a major breakthrough in man/machine interaction using voice technology will occur in the next decade.

Ongoing research aimed at closer integration going beyond simple dictation is described by Vidal et al. (2006): "the human translator determines acceptable prefixes of the suggestions made by the system by reading (with possible modifications) parts of these suggestions", (Vidal et al., 2006, 942). A similar research activity is presented by Khadivi (2008).[39] The integration of speech recognition and statistical machine translation for MAHT is also under research, see Reddy and Rose (2010). So far, no commercial translation environment tool offers similar features.

The users of translation environment tools frequently use applications that perform optical character recognition (OCR), see (Bowker, 2002, 26-30). OCR tools are a commonly requested plug-in for translation environment tools, see (Lagoudaki, 2008, 98), because they are believed to enhance translator productivity (Lagoudaki, 2008, 180). Interestingly, the findings presented by McBride (2009) show that OCR software is not debated that frequently in translator forums. The trend towards digital content is complete in several areas (e.g. technical documentation) and it is likely that it will continue to embrace nearly the totality of the documents to be translated. On the other hand, as long as documents are made available in electronic form, but in non-editable formats (e.g. sometimes PDF), OCR will still remain an interesting option for some projects.

Process management tools are often integrated into translation environment tools, particularly into corporate versions, see Sikes (2009). However, there are also external solutions that serve the same purpose, see (Reinke, 2009, 175-177). The main drawback of internal solutions is that they cannot be used in conjunction with other translation environment tools. As long as only one translation environment tool is used, this is not a problem. But many LSPs employ several: in such cases, a third-party application, which is capable of interacting with all translation environment tools, may be preferable. It is therefore unlikely that external process management tools will become obsolete.

A similar situation applies to quality check tools. Translation environment tools in-

[39] See (Khadivi, 2008, 53-54) for an account of earlier exploratory studies in this field.

tegrate many check functions and can apply them during translation: when an error is spotted, it is immediately reported, e.g. by highlighting the segment and providing a description of the error in question. For example, memoQ and SDL Trados support this type of quality assurance. QA Distiller and ErrorSpy can be cited among the third-party quality control tools that can be used after the translation.

The QA/QC components of current translation environment tools usually do not support the quality measurement and assessment provided by third-party solutions, see (Reinke, 2009, 174). As translation environment tools improve and extend their QA/QC components, it is likely that third-party solutions will focus on special checks, e.g. compliance with company-specific rules where linguistic knowledge is necessary. While translation environment tools can integrate such tools, similar to Across in its crossAuthor Linguistic module for authoring purposes, see (Sikes, 2009, 18), it is unlikely that they will eventually merge.

1.4.7 Translation environment tools and the web

1.4.7.1 Enabling technologies

Translation environment tools are no longer available only as desktop applications. New software distribution models have been developed, e.g. Software as a Service (SaaS), which is defined as follows:

> In the software as a service model, the application, or service, is deployed from a centralized data center across a network [...] providing access and use on a recurring fee basis. (Software and Information Industry Association, 2001, 4)

The SaaS concept has at least two peculiarities, see Kreckwitz (2007):

- The user does not purchase the software, but is charged for utilization.
- Overheads (installations, updates, maintenance, administration) take place on the server side.

Different licensing models can be applied, and some SaaS applications are free of charge. For a general introduction to the topic, see Software and Information Industry Association (2001), where the benefits of this model for vendors and customers are also discussed.

SaaS solutions use cloud computing, see (Wyld, 2010, 45). Cloud computing can be difficult to define, see (Baun and Kunze, 2010, 111), but the main idea behind it is that software and hardware resources are located in large data centers and can be accessed remotely by users. Thus, “cloud computing enables a new platform and location-independent perspective”, (Wyld, 2010, 44).

(Baun and Kunze, 2010, 111) list the advantages (flexibility, scalability and reliability) along with the open issues of interoperability and vendor lock-in, see (Baun and Kunze, 2010, 116) and (Born, 2010, 19). For a discussion of cloud computing from the point of view of the translation market, see García and Stevenson (2009b).

(Software and Information Industry Association, 2001, 14) states:

> The combination of increasing accessibility and declining costs of bandwidth allows a hosted solution delivered over the Net or a thin-client to become both a technologically and financially feasible process.

Since 2001, when this paper was written, the evolution of this technology has accelerated significantly. At the time of writing, it is still ongoing, although it has not advanced to the same degree worldwide due to the digital divide.

1.4.7.2 From remote resources to remote services

While traditionally located on the translator's workstation, TMs are increasingly located on a remote computer or server. Multi-user access is possible through a local area network or through the Internet. This is referred to as TM sharing (real-time sharing), but it should not be confused with the exchange of TM resources, see 1.4.9. This trend was initially described by Levitt (2003). Two years later, "online access to databases as a common and well-used feature", (Zetzsche, 2005, 32), was considered a midterm development. The feature was available but its usage was still limited. A further two years later, the availability of this feature seemed to be more widespread, see (García, 2007, 62), and (Zetzsche, 2007b, 47) who added that "in this question we find ourselves in a twilight zone between eras". This trend is likely to continue, see Zetzsche (2011d).

With respect to remote TMs and translation environment tools, it is necessary to distinguish at least three possibilities. On the one hand, standard desktop solutions can be connected to a server TM instead of a local TM, i.e. the locally-installed application can work with both local and remote TMs. On the other hand, there are local clients that cannot be used with local TMs but only in a client-server architecture. Finally, there are browser-based solutions (e.g. Lingotek, see García and Stevenson (2006), XTM Cloud and Wordfast Anywhere) similar to clients, but with the major difference that no software needs to be installed on the local machine because the editor is integrated in the web browser. Both clients and browser-based solutions are examples of server-based computing:

> An application is run on a server, but the user interface is presented to a thin client to the end user. Users can access the output [...] via a special client program or within a browser. (Software and Information Industry Association, 2001, 9)

Kreckwitz (2007) predicts that browser-based applications will eventually become dominant, although in the near feature a mixed scenario is conceivable. Consequently, the tendency towards the use of several translation environment tools (be it traditional desktop solutions, clients or web-based applications) by the same translator, already noted by (Lagoudaki, 2006, 23), is probably going to strengthen.

1.4.7.3 Advantages and disadvantages of remote services

With web-based translation nevertheless advancing more slowly than expected,[40] it is important to discuss the advantages and disadvantages, see also (García, 2007, 63). The advantages can be summarized as follows:

- Enhanced leveraging of resources.
- Control over the translation process and the resources used (for LSPs/customers).
- Negligible or no purchase costs compared to traditional desktop solutions (for translators).
- Potentially easier management (for LSPs/customers).

For browser-based solutions, it is advantageous that no local installation is necessary. Platform-independence is a major advantage too.

The main disadvantages,[41] see also García (2007) and García (2009a), are:

- The database can be accessed exclusively through a specific application and for a given period of time (for translators).
- Web access downtimes and server downtimes can prevent the job from being completed.
- The job cannot be completed with a different application and the application cannot be used for projects created with a different translation environment tool.
- Own databases cannot be used in parallel (for translators).
- The translation cannot be added to a personal database and reused for other projects (for translators).
- Constraints on the internal management, e.g. no subcontracting (for translators).
- Less features with clients and web-based applications than with full desktop solutions.

From the lists above it emerges that advantages can become disadvantages (or vice versa) depending on one's perspective: e.g. a disadvantage for the translator might be an advantage for the LSP, see (García and Stevenson, 2007, 24).

[40] (García, 2007, 64) states that "the push for web-interactive translation is coming from language vendors, not freelance translators".

[41] The description of the disadvantages contains some generalizations that do not apply to every single solution. For example, Wordfast Anywhere offers a time-unlimited access, permits the user to download the built TM, does not exploit the user's TM, which is strictly personal, allows the use of other resources, etc. Thus, it better meets the needs of freelance translators. The list of disadvantages is not exhaustive, but other glitches are not inherent to the software architecture.

Among the more serious disadvantages mentioned is that translators no longer have the freedom to choose which software to use, although they would wish to do so, see (McBride, 2009, 134) and (Zetzsche, 2011e, 25). This could be, however, mitigated if the software comes for free. For browser-based solutions a further prerequisite is crucial to their growth: the widespread availability of fast and affordable Internet connections. Access speed is a major issue in acceptance, see (García, 2009a, 203), in addition to the seamless availability of the servers where the resources are stored.

As the disadvantages mainly affect translators, it is not surprising that "anecdotal evidence already suggests many [translators] are not happy [...]", (García, 2009b, 28). However, translator-friendly solutions, such as Wordfast Anywhere, are less likely to be met with skepticism.

So far, general pro and cons have been discussed. However, project-related issues must also be considered. Kreckwitz (2007) points out that the balance between benefits and disadvantages depends on the specific scenario.

García (2007) predicts long-term de-skilling of translators, in particular as regards technical issues. This can be seen as focusing on the translation task, however translators would be "restricted to the narrow mechanics of de-contextualised segments" and "de-skilling has the potential to transform them from valued localization partners to anonymous and easily interchangeable components", (García, 2007, 66).

A discussion of web-based applications would not be complete without some remarks on the Google Translator Toolkit, an example of a browser-based translation tool. At the time of writing, the Google Translator Toolkit is still a semi-professional translation memory, far from being full-fledged from a translator's point of view, see Zetzsche (2009b), García and Stevenson (2009a) and García (2010a).

1.4.8 Crowd-sourcing

The basic idea that a translation is not provided as a service but on a voluntary basis has been defined in many different ways e.g. hive translation (see e.g. (García and Stevenson, 2009b, 3)), open source translation, crowd-sourced translation, cooperative translation, community translation and user-generated translation (see e.g. (Perrino, 2009, 62)).

This idea is not new; open source software has had to rely on it for localization, see (García, 2009a, 209) and (Geldbach, 2010a, 39). For example, the browser Firefox "is available in over 70 languages, thanks to the contributions from Mozilla community members around the world", Mozilla Foundation (2010). The Linux distribution Debian is another example of software translated by volunteers, see Gonzalez-Barahona and Peña (2008). Volunteer translations have also been provided to humanitarian and non-governmental organizations.

The difference is that this approach has now been taken by commercial companies: the most well-known and successful example is Facebook, see e.g. García and Stevenson (2009b) and Geldbach (2010a). Will volunteer translation increasingly substitute commercial translation? Despite the hype and interest from other businesses, see (García, 2009a, 210), this model relies on large, motivated multilingual communities and needs coordinated effort as well as ad-hoc tools, see (Geldbach, 2010a, 41-42). Systems and workflows are currently not yet well-established: "It's debatable whether this saved Facebook money, as setting up the clever system to collect the translations, peer-evaluate and then publish them must have been costly", (García and Stevenson, 2009b, 30). In addition, "users are only interested in translating customer-facing content [...] and (unsurprisingly) are not interested in translating technical or legal documentation [...]", (Carson-Berndsen et al., 2009, 60). Under these constraints, the feasibility of crowd-sourcing translation projects will remain limited even if specific workflows and tools are eventually optimized.

1.4.9 Translation corpora and exchange of TM resources

Large amounts of multilingual data are increasingly available, see Zetzsche (2010d). These large data collections are provided by translators (e.g. VLTM), companies (e.g. TAUS) or public bodies (e.g. DGT-TM) sharing their linguistic assets.[42] However, like TAUS, they are not always freely available. The availability of such resources is also limited as regards language pairs and specialization fields. In addition to these limitations, there are some general concerns, pointed out by (Zetzsche, 2010d, 33): while they can be helpful for reference purposes (typically, concordance search), such resources can be distracting if different terminology and style are used or if quality is not adequate. While quality can be addressed with a ranking system based on reuse, see (García, 2009b, 29), terminology is more difficult to cope with (although properly filled fields, if available, can be used to filter the results).

The basic issue is, however, that the primary aim of many of these collections (except for e.g. VLTM) is to feed statistical machine translation engines, (Zetzsche, 2010d, 33) and (García, 2009a, 206). They are not primarily intended as TM resources despite the fact that they include TM material. Therefore, while being helpful in some situations, their importance for MAHT should not be overestimated. On the other hand, a direct lookup from within the TM systems could be of interest to translators, see (Kübler and Aston, 2010, 512). In any case, the availability of large corpora of multilingual data is likely to

[42] VLTM stands for Very Large Translation Memory and is a project from Wordfast. The server-based TMs are accessible over the Internet through the Wordfast desktop application. There is a public VLTM, however, private ones can be defined too. Workgroups can be set up to share a specific VLTM in real time. The data is made publicly available only if the user explicitly allows it. The main aim of this project, which is driven and supported by freelance translators, is better leveraging. For more information, see Wordfast (2010a).

TAUS stands for Translation Automation User Society and defines itself as "a think tank for the translation industry, undertaking research for buyers and providers of translation services and technologies", Translation Automation User Society (2010). The TAUS Data Association (TDA) is a spinoff "providing a neutral and secure platform for sharing language data", TAUS Data Association (2010). Language data refers to translation memories. This project is driven by private companies and focuses on training MT engines with the collected data, see (Joscelyne, 2009a, 2) and (Kreimeier, 2010, 46).

Finally, DGT-TM is the translation memory of the Directorate-General for Translation of the European Commission. This multilingual TM collects the Acquis Communautaire, i.e. the body of European legislation, in 22 official languages of the European Union. See Directorate-General for Translation (2010) for more information.

increase.

In this context, platforms for translators wishing to sell/buy TMs should be mentioned. However, the copyright issue is unclear, see also Esteves-Ferreira et al. (2009) and 1.3.4.6. Some end customers and LSPs require translators to sign non-disclosure agreements that prohibit the pooling of translated texts. An unequivocal answer to the question of whether exchanging TMs is legally permissible is not possible, not least because national copyright regulations differ.

Chapter 2

Conducted research

This chapter introduces the research conducted on placeable and localizable elements in this thesis. Section 2.1 explains the concept of placeable and localizable elements and highlights the reasons why proper software support is beneficial, see in particular 2.1.1.2. It also clarifies the research scope of this thesis and outlines issues that are more comprehensively covered in other works. Section 2.2 focuses on the objectives of the research and distinguishes the main objectives from the corollary ones. Section 2.3 covers the methodology of the research by presenting the guidelines of the evaluation, its workflow and its parameters. Section 2.4 concentrates on the data used for the tests, introduces the TM systems tested and outlines how the suggested improvements will be formulated. Finally, section 2.5 explains the standard structure of the subsequent chapters where the tests are presented in detail.

2.1 Scope

The main target of this thesis is to assess how TM systems handle placeable and localizable elements.[1] Recognition (see 2.2.1), retrieval performance (see 2.2.2) as well as further aspects relating to usability and functionality (see 2.2.3) will be discussed. The suggested improvements are geared towards resolving the identified flaws and have two main aims:

- Reliable assessment of user effort through more accurate penalty values.
- Reduction of user effort through a combination of recognition and automatic adaptations of placeable and localizable elements.

As (Lagoudaki, 2009, 18) puts it:

> TM systems are translator-support tools operating in the constant interaction with the user with the aim of facilitating her work. Therefore, the main focus

[1] This chapter presupposes acquaintance with the concepts of translation technology discussed in chapter 1.

> is probably misplaced if it is not on how to advance this interaction and how to identify ways of supporting the translator in a more effective way.

This work focuses on translators because placeable and localizable elements are first and foremost relevant for them during translation. However, some issues discussed later on, e.g. the reliability of quality control checks in 3.2.2 and the editability of some tag attributes in 8.2.3.3, are also of concern for project managers.

2.1.1 Placeable and localizable elements

2.1.1.1 Definition

Placeable and localizable elements can be defined as follows:

> **Placeable elements** are portions of a document which – in the translation – do not change in content.
>
> **Localizable elements** are portions of a document which – in the translation – are adapted in content to a target locale according to standard or given rules.

A locale represents "a specific combination of language, region, and character encoding", (Esselink, 2000, 1). However, a locale also embraces formal conventions regarding, for example, typesetting and number formats. The adaptation of localizable elements in the target language conforms to these standard conventions and, sometimes, to corporate-specific instructions (e.g. corporate style). In other words, these elements "are treated in translation in a rather transparent manner, either not translated at all, or subject to specific conventions [...]", (Somers and Diaz, 2004, 14). In translation, both placeable and localizable elements may need to occur at a different position compared to the source text in order to fit the target text syntax and formulation.

Placeable and localizable elements can be grouped as follows:[2]

- Numbers
- Dates
- Proper nouns and identifiers
- URLs
- E-mail addresses
- Tags
- Inline graphics

[2] This list is not necessarily exhaustive, but concentrates on those elements that are deemed most frequent in the translation activity.

- Fields
- Punctuation marks and word delimiters

Placeable and localizable elements can be completely extra-textual (certain types of tags), can improve the formatting and readability of texts (punctuation marks, other types of tags) or can be part of the text (numbers, dates, URLs, e-mail addresses, proper nouns and identifiers). Section 2.4.1.5 provides some statistics on the frequency of these elements.

Some errors affecting placeable and localizable elements can be detected automatically, see (Russell, 2005, 3 and 8-9), and are proofed by existing quality check tools, see (Reinke, 2009, 173) and e.g. Yamagata Europe (2011).

Placeable or localizable?

It is not possible to make a clear, language-independent distinction that defines which elements are placeable and which are localizable. In fact, the same group of elements can be either placeable or localizable depending on the target language, on the individual element, on the context, etc.

Numbers, for example, are usually placeable elements in a translation from German into Italian: the decimal separator in both languages is a comma. However, when translating from German into English, the decimal separator has to be a point. Additionally, conversion of the number may be necessary if a different measuring unit, e.g. mile versus kilometer, is used.[3] The same applies to dates that may require a different order of day and month, e.g. German DD.MM.YYYY becomes in US English MM-DD-YYYY (or MM.DD.YYYY).

Proper nouns are a complex topic; they are sometimes translated, sometimes left unchanged. A particular type of adaptation is transliteration, which may be adopted when the target language uses an alphabet other than the source language (e.g. the Cyrillic alphabet in Russian and Serbian versus the Latin alphabet of most Western European languages), see Pouliquen and Steinberger (2009) for more information and examples.

If URLs and e-mail addresses do not have to change in the target text, they are placeable elements. However, localization might be necessary, e.g. if a different website or a different contact person exists for the target locale.

Inline graphics are usually placeable elements if they are language-independent. However, if they include translatable text, localization is usually necessary. In addition, pictures without text might have to be replaced with another version that is specific to the target locale.[4]

While formatting tags are placeable most of the time, sometimes it may be necessary to duplicate or to delete them in the target language. Consider the following German real-life example along with its Polish translation:

- Abhängig von der Anzahl der Kontakte wird die <b>**Suche angeboten**<\b>.

[3] There are standards prescribing the measuring units that have to be used in a specific market, see Speer (2010).

[4] For an overview of graphics localization, see (Esselink, 2000, 347-358).

- W zależności od ilości kontaktów, <b>**udostępniana**<\b> jest funkcja <b>**wyszukiwania**<\b>.

In the German source text, the words in bold are neighboring. In Polish, reformulation and syntax reordering are required: the corresponding words are separated as a result and the translator has to duplicate the bold tag in order to mark them. This adaptation is not intrinsic to the element itself, but is due to the target language text. Other types of tags are localizable because the content of their attributes is language-dependent, see 8.2.3.3 for some examples.

Fields encompass different types of elements, see 10.1, e.g. they insert text automatically or contain language-dependent content. The treatment of fields depends on the specific type.

In many language combinations, punctuation marks and word delimiters do not need any adaptation because the same characters are used. However, there are plenty of exceptions. For example, the question mark in Greek, the full stop in Chinese and Japanese and the semicolon in Arabic are not the characters used in English or German. Moreover, the usage of the punctuation mark itself can be different.[5]

To summarize, virtually any element can be either placeable or localizable depending on the textual and extra-textual context.

Alternative terms and concepts

Several TM systems, e.g. Across, SDL Trados and Wordfast, define most of these elements as "placeables". Other TM systems adopt a different terminology, e.g. memoQ uses "non-translatables" (for abbreviations, acronyms and proper nouns) and "auto-translatables" (for numbers). Déjà Vu uses the term "embedded codes", but this only refers to tags. Several other terms are cited below. These terms have been grouped according to the reason why they were not deemed suitable.

1. Some terms do not take into account the possibility that modifications *may* be necessary:

   The term "placeables" is used by several TM systems (see above) as well as by (Bowker, 2002, 98). In this thesis, a distinction is made between placeable and localizable because the concept of placeable may imply that these elements are only to be placed at the correct position, but do not require any adaptation, which is not always the case.

   The term "non-translatables" is used by Macklovitch and Russell (2000): as placeables, it implies that these elements are invariable across languages. As already

[5] Unlike for other placeable and localizable elements, the source text may be irrelevant for the usage of punctuation marks and word delimiters in the target text, e.g. in German the comma is used according to rules that do not exist in English or Italian. However, the usage of punctuation marks and word delimiters is often determined by the source text, e.g. when in a software manual quotes are used to mark strings coming from the graphical user interface. Therefore, punctuation marks and word delimiters have been tested too.

discussed, this does not hold true for all elements and for all language combinations. It suffices to list the examples provided in Macklovitch and Russell (2000) for non-translatables (“proper names, dates or other types of numerical expressions”) to see that adaptations can be necessary in the target language.

“DNT (do-not-translate) units” is used by Groves (2008) and refers to “HTML tags, formatting information, filenames, URLs, placeholders etc.”. “Constants” is used by Commission of the EU (1996). Both of these terms are misleading as they exclude any modification of the elements in question.

2. Some terms are already used with a different meaning in other contexts:

   Many placeable and localizable elements are referred to as “paralinguistic expressions” in (Russell, 2005, 8).[6] In fact, the term “paralinguistic” is usually used with the following meaning: “of or belonging to paralanguage; relating to or designating non-phonemic speech features”, Oxford University Press (2009). “Paralanguage” is defined as “the non-phonemic but vocal component of speech, such as tone of voice, tempo of speech, and sighing, by which communication is assisted. Sometimes also taken to include non-vocal factors such as gesture and facial expression”, Oxford University Press (2009).

   “Named entities” is a common term in the field of information retrieval and denotes “continuous fragments of texts [...] which refer to information units such as persons, geographical locations, names of organizations, dates, percentages, amounts of money, locations”, (Graliński et al., 2009, 88). Whilst there is some overlap with the concept of placeable and localizable elements, the focus of named entities is clearly shifted towards proper nouns. Equally important elements such as fields and tags are excluded altogether.

3. Some terms are blurred:

   “Numerical and special expressions” are cited by (Mikheev, 2003, 208), but this includes only email addresses, URLs, numbers, dates as well as other items which are not relevant here (such as book citations).

   “Extra-linguistic elements” (Piperidis et al. (1999)) refer, among others, to “dates, abbreviations, acronyms, list enumerators, numbers”. This definition arbitrarily excludes the named elements from linguistic utterances.

   The term “special elements” is used by (Zerfaß, 2002a, 11), but refers only to abbreviations and acronyms and does not specify in what respect these elements are special.

   Gaussier et al. (1992) use the term “transwords” (cited by Somers and Diaz (2004)), but excludes e.g. tags.

[6] In Russell (2005), the author is well aware of this unconventional usage and quotes the term “paralinguistic” when he first uses it.

4. (Trujillo, 1999, 68-68) and (Hudík and Ruopp, 2011, 47) cite several types of placeable and localizable elements (numbers, dates, product names, graphics and formatting), recognize that they can be grouped together, but fail to provide a term embracing them all.

In the end, the term placeable and localizable elements has been deemed preferable to all alternatives because it best encompasses all tested elements and best accounts for their behavior in translation.

2.1.1.2 Advantages

Extensive recognition, see 2.2.1, and correct support of placeable and localizable elements, see 2.2.2 and 2.2.3, bring some general advantages. For clarity's sake, they are presented separately, but they are in fact interrelated. These advantages do not apply equally to all placeable and localizable elements.

Translation throughput and quality

The first advantage of recognition accompanied by proper highlighting is that attention is drawn to the placeable and localizable elements. This is not necessary or even significant in all cases, but can be useful e.g. for numbers. Translation environment tools allow recognized placeable and localizable elements to be transferred into the target text by means of shortcuts, see e.g. 3.2.2. Easy transfer – particularly of placeable elements that do not need any adaptation – speeds up translation because it saves typing, or dictation if speech recognition software is used.[7] In fact, as elements such as tags and inline graphics are not plain text, a transfer function is indispensable.

Less typing/dictation also reduces the likelihood of errors, even though e.g. misplacements are still possible. As a result, a certain level of translation quality is guaranteed before general quality checks. This results in faster translation because less time has to be spent on correcting errors.[8]

Recognition becomes more beneficial if combined with automatic adaptations, see 2.2.2. If a new source segment differs from the entry in the TM only because of a placeable or localizable element, often this element can be automatically adapted. Automatic adaptations speed up translation because they relieve the user of some of the effort that is otherwise necessary to adapt fuzzy matches. These adaptations would not be possible if placeable or localizable elements were not specifically recognized. In general, automatic adaptations function well when placeable and localizable elements have to be deleted or replaced; later tests, see e.g. 10.2.3.1 and 10.2.3.4, show that additions and transpositions pose more problems.

Further assistance is provided if placeable and localizable elements, e.g. numbers, are automatically converted to the conventions of the target locale. This conversion requires

[7] This option is not useful if the source text is automatically copied into the target when no match has been found. However, this is not the default setting in most TM systems, see 1.3.6.3.

[8] Further advantages in terms of quality assurance/quality control are discussed in 5.3.

precise recognition of the placeable and localizable elements as well as knowledge of the target locale conventions. This functionality is available only in few TM systems and is not covered in this thesis. Nevertheless, it is worth citing because many users would like to find it in translation environment tools, see (Lagoudaki, 2008, 144).

Retrieval and memorization

The recognition of placeable and localizable elements can enhance retrieval in different ways. Firstly, lower penalties are usually applied to the differences concerning these elements compared with textual differences. As a result, more and higher matches can be retrieved in some situations. Automatic adaptations applied to placeable and localizable elements increase the match value too, sometimes converting fuzzy matches into 100% matches. Finally, the recognition of placeable and localizable elements can make the memorization of segments in the TM more efficient.

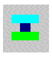
Later tests will provide plenty of examples for higher matches. The advantages for memorization can be seen in this ad-hoc test similar to Zetzsche (2003).

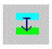
The following sentences included in an MS Word 2003 file were tested:

1. Press the button 1 and select a setting.

2. Press the button 3 and select a setting.

3. Press the button 25 and select a setting.

4. Press the button  and select a setting.

5. Press the button  and select a setting.

These sentences were processed with a subset of the TM systems tested later:[9]

- Heartsome Translation Studio Ultimate (7.0.6 2008-09-12S)

- SDL Trados 2007 Freelance (8.2.0.835)

- Wordfast (5.53)

With SDL Trados and Wordfast, MS Word was used as an editor. With Heartsome, the proprietary editor was used. Each sentence was translated and confirmed individually. With SDL Trados, the translation was repeated with different settings using two separate translation memories. The option Placeable differences penalty under Options > Translation Memory Options > Penalties was set to 2% in both cases. However, in the first case, the option Apply placeable penalty also when source tags differ was deactivated (SDL Trados 1), in the second case it was activated (SDL Trados

[9] Unless otherwise specified, the settings of the translation environment tools are the same as in 9.2.1.2.

2). Changing this setting does not affect the general handling of placeable and localizable elements but it does affect the match values and the entries saved in the TM.

Table 2.1 provides an overview of how the tested translation environment tools handle placeable and localizable elements. Wordfast and SDL Trados treat both numbers and inline graphics as placeable elements. Heartsome treats numbers as plain text, whereas it converts inline graphics into tags.

	Numbers	**Inline graphics**
Heartsome	plain text	tag
SDL Trados	placeable	placeable
Wordfast	placeable	placeable

Table 2.1: Handling of placeable and localizable elements

The match results and the presence of automatic adaptations, indicated by an [a], are summarized in table 2.2. For the first segment there is obviously no match. These results are of relevance because match values, automatic adaptation and memorization are correlated.

	2	**3**	**4**	**5**
Heartsome	95	93	92	100[a]
SDL Trados (1)	100[a]	100[a]	89	100[a]
SDL Trados (2)	100[a]	100[a]	89	98
Wordfast	100[a]	100[a]	100[a]	100[a]

Table 2.2: Match values

Table 2.3 provides an overview of the number of entries in the translation memory.

	No. of entries
Heartsome	5
SDL Trados (1)	2
SDL Trados (2)	3
Wordfast	1

Table 2.3: Number of saved TM entries

The way the segments are saved in the translation memory differs.[10]

- Heartsome saves all segments. Numbers are in plain text. Inline graphics are lengthy placeholders between <ph> and </ph>; their content differs depending on the inline graphic.

[10] For Wordfast, the translation memory is available as a plain text document. For Heartsome and SDL Trados, export to the TMX format was performed.

- SDL Trados 1 saves two segments. The first segment is saved "as is". The second contains a placeholder for the inline graphic (<ph type="image">{\pict}<\ph>), which is *different* from the placeholder used in Heartsome.
- SDL Trados 2 saves 3 segments. One is the first segment as above. Both subsequent segments include a placeholder for the inline graphics. Interestingly, the placeholder is exactly the same in both cases so that the TM contains the same segment twice.
- Wordfast saves only the first segment "as is", with the number "1" in plain text.

This small test shows that the recognition of placeable and localizable elements allows for a more efficient memorization of similar segments in the TM. The specific handling depends on the placeable or localizable element itself and on the translation environment tool and its settings.

The fact that match values differ from TM system to TM system is known, see Reinke (1999), Seewald-Heeg (2007) and Guillardeau (2009), and also applies to machine translation systems incorporating a translation memory functionality, see (Nübel and Seewald-Heeg, 1999b, 20). Discrepancies and unreliable penalty values are also discussed in translator forums, see e.g. ProZ (2010a) and ProZ (2010b).

2.1.2 Research scope limitations

Translation environment tools are complex software packages consisting of several applications, see 1.3.1. As the focus of this research is on placeable and localizable elements, only TM systems will be discussed. For a comprehensive comparison of different translation environment tools, see e.g. Massion (2005).

Some MT systems, e.g. Systran and PROMT, include TM functionalities as well. However, they will not be considered because the research scope is limited to translation environment tools that assist the user in interactive translation.

The TM systems will not be compared on the basis of their retrieval of primarily textual segments with few or no placeable and localizable elements. This comparison can be found e.g. in Fifer (2007). Textual modifications play a secondary role in this thesis. The retrieval speed will not be covered either: the limited size of the translation memories used do not permit any conclusion in this regard. A similar research scope restraint can be found in (Nübel and Seewald-Heeg, 1999b, 21). Refer to (Trujillo, 1999, 67-68) and (Manning and Schütze, 1999, 532-534) for more information on speed issues in information retrieval.

Usability and interface design will be touched upon in several chapters. However, the research will not focus on these topics so that their discussion is cursory and restricted to obvious shortcomings.

2.2 Objectives

This section aims at providing an overview of the overall test objectives. However, since the content and the structure of placeable and localizable elements varies significantly, the test aims were tailored to the element investigated in each case.

For all placeable and localizable elements, the crucial question is whether they are recognized by TM systems. Some elements are recognized seamlessly, but others are not. Therefore, placeable and localizable elements are split up into two groups.

1. Elements for which recognition difficulties were experienced: the main test objective is recognition, see 2.2.1.
2. Elements that are generally correctly recognized: the main test objective is retrieval, see 2.2.2.

Some additional issues emerged during the tests, see 2.2.3. They are not relevant for all placeable and localizable elements and will therefore only be introduced when appropriate.

2.2.1 Recognition

The following elements, which constitute semantic units, should be recognized by TM systems as placeable or localizable elements, also when they occur as plain text:

- Numbers
- Dates
- Proper nouns and identifiers
- URLs
- E-mail addresses

This thesis will investigate whether TM systems normally recognize these elements, whether all patterns are found and whether the entire string is recognized as one single element.[11] The focus is on accuracy, defined as "the provision of right or agreed results or effects", see ISO 9126 as cited by (Höge, 2002, 89). Automatic conversions, see 2.1.1.2, are not assessed.

2.2.2 Retrieval and automatic adaptations

The placeable and localizable elements listed below are usually correctly recognized. They are either not plain text (fields, inline graphics and tags) or are orthographic markers (punctuation marks and word delimiters) and – generally – without semantic meaning.

- Fields
- Inline graphics

[11] This is particularly relevant for elements consisting of several building blocks, such as dates.

- Tags
- Punctuation marks and word delimiters

The tests concentrate on the impact of modifications of such placeable and localizable elements (see 2.3.2 for a description of the test procedure), and are organized around three objectives:

- Recognition of the modification and appropriate highlighting.
- The match value proposed by TM systems. The penalties are compared and underlying rules are subsumed.
  - Is a fixed penalty applied?
  - Does the segment length affect the penalty?
  - Does the number of modifications affect the penalty?
  - Does the position or type of the modification affect the penalty?
- The presence and reliability of automatic adaptations.

The focus is on suitability, defined as "the presence and appropriateness of a set of functions for specified tasks", ISO 9126 cited in (Höge, 2002, 89).

2.2.3 Other objectives

The following issues only affect some placeable and localizable elements:

- Are they clearly displayed?
- Is their segmentation correct?
- Are they correctly editable or translatable?
- Is their handling customizable?

Clear display means that the placeable or localizable element is recognizable for the users. General customizable aspects such as color, size etc. are not considered.[12]

The test data used in this thesis is available in standardized source formats, see 2.4.1. This should help ensure proper segmentation and editability by the built-in file format filters of the TM systems.[13] Some placeable and localizable elements are not plain text, e.g. tags and fields, but may need to be modified or contain text that has to be translated.

[12] The role played by the user interface in the eyes of TM system users is stressed by (Lagoudaki, 2009, 175), but is outside the scope of this thesis, see 2.1.2.

[13] For example, XML would not be a suitable source format because, apart from some standards, see Pelster (2011), its elements can be defined freely and the user has to ensure that they are properly processed, see Chama (2010a).

Finally, the handling of placeable and localizable elements often depends on particular settings. The customizability of these settings enhances the flexibility of TM systems and is therefore of particular interest in this study. Customizability is also frequently cited among the needs of TM system users, see (Lagoudaki, 2009, 177).

2.3 Methodology

The definition of evaluation methods and evaluation metrics for translation environment tools has been the focus of several papers and works, which serve as a basis for the methodology of this thesis.[14]

2.3.1 Modeling the evaluation

The tests performed within the scope of this study were conceived using the black box testing approach: "the M(A)T system is seen as a black box whose operation is treated purely in terms of its input-output behaviour, without regard for its internal operation", (Trujillo, 1999, 256). Black box testing is suitable for end-users, see (Quah, 2006, 138), and is applied in most publicly available evaluations, e.g. (Seewald-Heeg, 2007, 562). Although the evaluation is conducted from a translator's point of view, the results can also be of interest for developers of translation environment tools.

In particular for match values, if the settings governing them were not accessible/visible, the tests were constructed to "reveal the strategies and rules used by these systems", (Way, 2010a, 555), which is typical of reverse engineering. In software engineering, reverse engineering involves examining and analyzing software systems in order to recover information, particularly functional specifications (adapted from (Wills, 1996, 7)).

The tests are task-oriented and examine "whether and the extent to which a piece of software offers functions to perform specific tasks", (Höge, 2002, 138). In accordance with the peculiarities of task-oriented testing, some tasks were repeated when problems were encountered or different settings had to be tested.

Additionally, taking a closer look at some of the features of the TM systems is typical of feature inspection, defined as "describing the technical features or a system in detail. The purpose is to allow an end-user to compare the system with other systems of similar kind", (Quah, 2006, 144).

As these tests compare several TM systems, the use of different methods (task-oriented testing and feature inspection) was considered appropriate, see (White, 2003, 235).

[14] Examples include: Commission of the EU (1996), Reinke (1999), Whyman and Somers (1999), Rico (2001), Höge (2002), Gow (2003), Reinke (2004), Massion (2005), Seewald-Heeg (2005) and Lagoudaki (2008).

2.3.2 Test procedures

The test procedures differ depending on the main objective for the placeable and localizable element group: recognition, see 2.2.1, or retrieval, see 2.2.2. This section is therefore organized in two corresponding subsections (2.3.2.1 and 2.3.2.2, respectively). The corollary objectives, see 2.2.3, are evaluated in both cases along the way.

2.3.2.1 Recognition

A segment containing one or more relevant placeable and localizable elements was opened for editing in the editor of the TM system. With some TM systems, the recognized elements were highlighted and recognition could be assessed directly. For other TM systems, a workaround was sometimes necessary, see 3.2.2 and 4.2.2 for further details. It was not necessary to provide any translation because recognition assessment only needs the source language.

2.3.2.2 Retrieval and automatic adaptations

To assess retrieval and automatic adaptations, a first segment was translated and saved in the translation memory. The subsequent segments that differed from the first one were opened in order to ascertain the proposed match value, but not translated. Therefore, the match value was calculated and the automatic adaptation applied with respect to the first segment.

2.3.3 Metrics

No global evaluation metric or scoring system was developed because the test objectives concentrated on the identification of the problems related to placeable and localizable elements irrespective of the TM system. If a global ranking were required, it would be necessary to assign scores to all tested features and to develop a weighting system. In fact, it would be first necessary to check whether and how these features can be measured, see Höge (2002) and Gow (2003). Throughout this work, several features are discussed that are difficult to express as one single value, e.g. the display of fields in chapter 10. Which display suits the user's way of working best is largely a matter of preference. A global metric would be of little help and the weighting of the different components (recognition, retrieval, display, etc.) would be extremely arbitrary. Specific metrics and criteria, however, were applied in order to better assess recognition and retrieval performance and can be integrated into existing frameworks for future evaluations.[15]

[15] Evaluation frameworks are defined as "general guidelines or procedures designed [...] as the basis for more detailed, customized evaluations", (Quah, 2006, 129).

2.3.3.1 Recognition assessment

In chapters 3, 4 and 5, specific metrics were necessary to interpret recognition results more easily.[16] In order to better understand those metrics, the basic measures of precision and recall are introduced. Precision and recall are typical evaluation parameters in information retrieval.

Precision is defined as “the ratio of relevant items retrieved over all the items retrieved”, (Trujillo, 1999, 63).

$$precision = \frac{tp}{tp + fp} \tag{2.1}$$

tp stands for true positives, *fp* for false positives.

On the other hand, recall is the “proportion of the target items that the system selected”, (Manning and Schütze, 1999, 269). (Trujillo, 1999, 63) adapts the definition of recall to TM systems as follows: “ratio of the relevant items retrieved over all relevant items in TM”.

$$recall = \frac{tp}{tp + fn} \tag{2.2}$$

tp stands for true positives, *fn* for false negatives.

When recognizing e.g. numbers, TM systems select tokens[17] that are deemed to be numbers. True positives are correctly recognized numbers, whilst false positives are recognized elements that are not numbers. Precision assesses whether incorrect recognition is common: a score of 1 means no errors. However, TM systems can miss numbers and produce false negatives, which are not accounted for in the precision result. This is the reason why recall is used: again, a score of 1 means no errors.

Generally speaking, precision and recall tend to be inversely correlated (as later findings will confirm), so that a trade-off is inevitable if the values of both measures have to be approximately equal, see (Manning and Schütze, 1999, 269). In order to obtain a measure of the overall performance, and assuming equal weighting[18] of precision and recall, the F measure is used:

$$F = 2 \times \frac{precision \times recall}{precision + recall} \tag{2.3}$$

Can these measures be applied to test results? For the purposes of this study, a test suite was built (see 2.4.1.4) so that, from a methodological point of view, screening of the input had already taken place. Consequently, precision could not be reliably assessed, as false positives were underrepresented in the selected examples. More telling is the recall measure, as the examples of the test suite were intentionally selected in order to generate true positives.

[16] Similar metrics could have been used in chapters 6 and 7, but this was not necessary because the results are self-explanatory.

[17] “A *token* is an instance of a sequence of characters [...] that are grouped together as a useful semantic unit for processing”, (Manning et al., 2009, 22).

[18] For the formula that allows for different weighting, see (Manning and Schütze, 1999, 269).

The evaluation of the results suggests that a graduate score is necessary in order to accurately assess recognition. The details of this score are described in 3.2.2, 4.2.2, and 5.2.2, respectively. A discussion of the specific ranking derived from these metrics is provided in 13.2.4.

2.3.3.2 Retrieval assessment

Chapters 8 to 12 concentrate on retrieval performance. In order to assess and compare the results provided by TM systems, two criteria were adopted: the number of retrieval errors and the number of automatic adaptations. Relative figures are presented in 13.3.3 and 13.3.4, which also detail the rules used to quantify them, while 13.3.5 provides the resulting specific ranking.

2.4 Test framework

Notes on terminology

Letters and digits are collectively referred to as "alphanumeric characters". This term often embraces only A-Z (upper and lower case) and 0-9 from the ASCII encoding.[19] In this thesis, however, it is used in a broader sense and includes the Unicode properties "Letter" `\p{L}` and "Number" `\p{N}`. Unicode properties are explained in 2.4.4.1.

Throughout the tests, characters that are neither letters nor digits are labeled as "non-alphanumeric characters". Non-alphanumeric characters are referred to by their official Unicode names. Their corresponding Unicode codes are listed in appendix A. The term "special characters" is avoided because it is confusing: generally, it indicates characters that cannot be entered with normal keyboard keys and also includes letters with diacritic marks. In addition, special contexts such as HTML further restrict the scope of special characters.

Finally, the term "symbols" is used to identify the characters of the Unicode property "Symbol" `\p{S}` and some characters from the Unicode property "Punctuation, other" `\p{Po}`, see chapter 5 for more details.

2.4.1 Test data

2.4.1.1 Requirements and preliminary considerations

For the selection of the source data,[20] two general requirements were defined:

- The data is taken from translation jobs (and therefore provides real-life examples).

[19] ASCII stands for American Standard Code for Information Interchange. It consists of control (or non-printable) characters and printable (or graphic) characters. Graphic characters "include 26 uppercase letters, 26 lowercase letters, and 10 digits", (Korpela, 2006, 119), as well as 32 other characters such as solidus, see (Korpela, 2006, 117-119) for more information.

[20] The general procedure and issues of building a written corpus are described by Nelson (2010).

- The data consists of technical texts.

The first requirement derives from the assumption that the performance of TM systems in conjunction with placeable and localizable elements is best assessed using texts usually processed by TM system users. The second requirement takes into account the dominance of technical translations in the translation market, see (Lagoudaki, 2006, 12).

Since the tests involved different placeable and localizable elements and focused on different objectives, see 2.2, several data sources were necessary in order to construct the appropriate tests. The availability of suitable existing corpora was assessed. However, when the research began in 2007, no public corpus was available which met the requirements.[21] Common corpora, see Lee (2010), focus on other specialization fields and have been built for different purposes. Therefore, ad hoc data was necessary.

2.4.1.2 Main source

The main data source was a collection of four translation memories created with SDL Trados. A translation memory is a unidirectional aligned parallel corpus, see (Kenning, 2010, 487-488), and does not consist of complete texts, but of a collection of single segments. This limitation makes this corpus unsuitable for several linguistic analyses, see (Lemnitzer and Zinsmeister, 2010, 40-41), but does not affect the investigation conducted in this thesis. As an alternative to translation memories, the original documents could have been used. However, the significantly greater effort necessary to (re-)build the corpus was not deemed justified. For the purposes of this study, which was not primarily linguistic, it was not necessary to obtain the context information provided by the complete documents.

The translation memories, available in the SDL Trados 2007 proprietary format (TMW), were exported into TMX format. German was the source language and English or Italian the target languages. The target language had no bearing on the choice of examples to be tested. In fact, the translation memory was used as a monolingual corpus in the source language. The TMX metafields (e.g. “creationdate” and “creationid”) were not relevant either. The total size of the TMX translation memories for the source language was slightly less than 3 million words[22] and was sufficient for a specialized corpus “designed to answer specific research questions”, (Koester, 2010, 71). In addition, the manageable corpus size permitted more careful extraction of the examples, see (Koester, 2010, 67).

Most of the segments came from technical handbooks, but marketing material (technical brochures, newsletters) was also included and accounted for approximately 20% of the total. The source formats of the original documents were mostly MS Word and HTML; MS Excel and MS PowerPoint texts had been added in isolated cases. It was not possible to determine the exact ratio. However, this was not necessary for the aims of this thesis.

The representativeness of the selected corpus was ensured by the following criteria, adopted when sampling the translation memories. Each translation memory contained

[21] The TDA corpus, see 1.4.9, would now be a possibility to consider.

[22] For more statistics on the test corpus, see 2.4.1.5.

texts from a large company and covered different products. In addition, all selected companies generate high volumes of technical translation every year, often in a variety of target languages. The sectors they operate in (hardware and software solutions for telecommunication, packaging, printing and power supply) also vary to a relatively large extent.[23]

Because of non-disclosure agreements, the corpus is not publicly available and the examples tested were selected on the basis of their neutrality. Company names do not occur. Explicit anonymization was performed for URLs and e-mail addresses, a common and recommended practice in major corpora, see (Nelson, 2010, 61). The unavailability of the corpus as a whole does not affect the repeatability of the tests because they focus only on the examples presented.

2.4.1.3 Additional sources

The described corpus was not suitable for tags, inline graphics and fields (see chapters 8, 9 and 10 respectively), because the documents containing these items are needed before conversion using the file format filter of a translation environment tool, see 1.3.1. Consequently, additional sources were selected as follows:

- Tags: original documents preferably written in a markup language.
- Inline graphics and fields: original documents.

For tags, a dump from the 7th of July 2009 of the English version of Wikipedia was chosen because of the possibility of finding the necessary examples with minimum effort as it constitutes a large and easily available corpus.[24] This choice was a trade-off with respect to the requirements under 2.4.1.1 because its content does not have to be translated. However, not meeting the requirements in this case did not affect the tests or their results.

For inline graphics and fields, the English user manual of Wordfast available in MS Word format was selected: when learning how to use this TM system, it became evident that several issues concerning fields and inline graphics were present, despite the limited size of this manual. It also fulfilled the other requirements for the source data.

Because of the different sources, the examples tested are in German or English. German examples will not pose a problem for readers who do not speak German because the meaning is not relevant for the discussion. It could even prove beneficial as it allows the reader to focus exclusively on the placeable and localizable elements. Unless otherwise specified, the source language did not impact on the test results.

[23] The corpus was not intended to encompass as many sectors as possible. If constructed for other purposes, different requirements as regards sampling strategies would have been required, see Nelson (2010) and (Palmer and Xue, 2010, 259-260).

[24] This dump file did not contain history, discussions or images of the articles. The HTML code was cross-checked in the corresponding online article.

2.4.1.4 Test corpus and test suite

In this context, it is necessary to distinguish between test corpus and test suite. A test corpus is "essentially a collection of texts which attempts to represent naturally occurring linguistic data", (Quah, 2006, 137). The use of a corpus – if adequately large – has the advantage of covering most of the phenomena of interest, in other words, it is "more likely to reflect the real-world complexity", (Prasad and Sarkar, 2000, 11).[25] On the other hand, a test suite is defined as a "carefully constructed set of examples that represent some pre-determined 'linguistic phenomena'", (Quah, 2006, 136), and is used as "a range of controlled examples to discover where errors occur", (Quah, 2006, 137).

The translation memories, the Wordfast manual and Wikipedia constituted the corpora. Through the extraction of relevant examples from the corpora, a test suite was created, see 2.4.1.6 for details.

2.4.1.5 Test corpus statistics

Some figures about the frequency of placeable and localizable elements in the test corpus enlighten their role in translation tasks. Statistics were calculated for the placeable and localizable elements extracted from the main source, see 2.4.1.2, usually by means of regular expressions presented later, see 2.4.4.1. Such statistics are not meaningful either for tags – as Wikipedia does not reflect a typical translation job – or for inline graphics and fields, as a single source document was used, see 2.4.1.3.

The main source consists of 22,144,469 characters (printable characters and whitespaces, excluding control characters), 2,842,806 words[26] and 331,519 segments. The counts of placeable and localizable elements (except for punctuation marks) are summarized in table 2.4. It goes without saying that these figures cannot be assumed for different corpora.

Element		**Count**	**%**
Numbers and numeric dates		388,465	13.66
Proper nouns and identifiers		547,959	19.27
thereof:	symbols	192,046	6.75
	completely capitalized strings	103,466	3.64
	mixed-case strings	9,862	0.35
	alphanumeric strings	242,585	8.53
URLs		2,065	0.07
E-mail addresses		480	0.02

Table 2.4: Counts of placeable and localizable elements

For more information on the categorization of proper nouns and identifiers, see 5.1.

[25] For more information about advantages and disadvantages of test corpus and test suite as well as about the convenience of a combined methodology, see Prasad and Sarkar (2000).

[26] Word counts are by their own nature approximative because the notion of "word" is fuzzy, see (Manning and Schütze, 1999, 124-131). The figure has therefore to be taken as a rough reference.

While the figure for numbers and numeric dates is not surprising, at least in a corpus containing technical texts, the figure for proper nouns and identifiers is unexpectedly high. It is due to the fact that the translation memories included parts catalogs and spare parts lists with product codes and that these codes occur in plain text and as values of the translatable alt attribute used in img elements.[27]

For punctuation marks and word delimiters, a different approach is necessary. Many punctuations marks mark the end of a sentence and are exploited by TM systems for text segmentation, see 1.2. It is therefore interesting to assess how many segments end with a punctuation mark: 172,051 (out of 331,519), i.e. 51.9%. However, if the global frequency of punctuation marks is assessed, only 779,253 characters (out of 22,144,469) are punctuation characters, i.e. 3.52%.

2.4.1.6 Construction of the test suites

In order to extract the examples for the test suites from the corpora, different strategies were applied depending on the placeable and localizable elements involved.

Recognition evaluation

Numbers, dates, proper nouns and identifiers, URLs and e-mail addresses (chapters 3 to 7) were identified using simple Perl scripts. The search patterns were very general in order to prevent false negatives and focused only on the recognition of one or more distinctive characters occurring in the placeable or localizable elements as follows:[28]

- Numbers and dates: at least one digit
- Proper nouns and identifiers
  - Words containing symbols, e.g. AAC+
  - Words in uppercase, e.g. DVB-T
  - Words in mixed-case, e.g. MySQL
  - Words consisting of letters and digits, e.g. DDR2
- URLs: www, http or ftp
- E-mail addresses: commercial at

The matched segments were checked and raw lists including only placeable or localizable elements were compiled. Afterwards, the raw lists were manually filtered in order to:

[27] For more information on these elements, see 8.1.

[28] These Perl scripts were relatively basic and are therefore only briefly mentioned. The regular expressions suggested in the "Possible improvements" section of chapters 3 to 7 are much more complex because they are aimed at automatically recognizing the complete placeable and localizable element and excluding any false positive.

- remove false positives (if any) and
- extract examples which showed unique patterns.

For the purposes of this study, it would not have been effective to test examples sharing the same pattern, e.g. a simple digit sequence as in “5230” and “1271”.

The lists of the remaining true positives for each type of placeable and localizable element were compiled in TXT format, but they were not processed directly, see 2.4.2.

Retrieval evaluation

For tags, inline graphics, fields, punctuation marks and word delimiters (chapters 8 to 12), a different approach was applied. Fields and inline graphics were manually identified in the Wordfast manual. The Wikipedia dump was searched for specific tags according to the classification presented in 8.1. Punctuation marks and word delimiters were searched for in the translation memory corpus according to the classification presented in 11.1 and 12.1. The search did not encompass the complete corpus, but was interrupted as soon as a sufficient number of suitable examples had been found.

The extracted examples were used as a basis: the tested modifications, see 2.2.2, were manually inserted according to the test design described below. As pointed out by (Nübel and Seewald-Heeg, 1999a, 120), the evaluation of the match function requires classification of the possible modification types. For this study, the categories defined by (Reinke, 2004, 169) have been adopted: additions (or insertions), deletions and replacements (or substitutions). They are the typical operations considered in the calculation of the Levenshtein distance, see 1.3.5. However, they have been supplemented by a fourth one derived from Nübel and Seewald-Heeg (1999b): transpositions. These four modifications are known as “primitive operations”, (Abekawa and Kageura, 2008, 2002).

These operations are not uncontroversial. Substitutions can be seen as the result of the deletion of one element and the addition of another. This decomposition is adopted e.g. by (Baldwin, 2010, 203). Transpositions can be seen as the sum of the deletion and the addition of the same element in another position. However, decomposition would not have been of help in the presentation of the tests or their results. Substitutions and transpositions are therefore specifically tested and discussed.

The evaluation of the match function takes into account further aspects in addition to the modifications described above. Firstly, the context of the modification, i.e. the segment affected, is relevant too: in particular, it may be the case that the match value for the same modification is calculated based on the segment length. The same type of modification is therefore evaluated with segments of different length. Secondly, the number of modifications in the same segment is investigated too, with respect to the match value: at first, this might seem unnecessary, but this will help illustrate the different strategies employed by the TM systems.

2.4.2 Source formats of the tested documents

The examples extracted from the translation memories and from the Wordfast manual were inserted into MS Word documents (MS Word 2003 SP 3). There are two reasons for this:

- MS Word is currently the most frequently-used source format in the translation sector, see Lagoudaki (2006), and also the most discussed, see (McBride, 2009, 121).
- This format is directly supported by virtually all translation environment tools.[29]

HTML – a widespread format in the translation industry, see (Lagoudaki, 2008, 234) – was used for the tests with tags. Thus, the typical "context of use", (Rico, 2000, 36), of TM systems is ensured.

2.4.3 TM systems

2.4.3.1 Products and versions

The TM systems tested are listed below. Their version number(s) are in square brackets, their manufacturers in parentheses.

- Across Standalone Personal Edition [4.00] (Across)
- Déjà Vu X Professional [7.5.303] (Atril)
- Heartsome Translation Studio Ultimate [7.0.6 2008-09-12S] (Heartsome)
- memoQ Corporate [3.2.17] (Kilgray)
- MultiTrans [4.3.0.84] (MultiCorpora)
- SDL Trados 2007 Freelance [8.2.0.835] and SDL Trados Studio [9.1.0.0] (SDL)
- Transit XV Professional [3.1 SP22 631] and Transit NXT [4.0.0.672.3] (STAR)
- Wordfast [5.53] and Wordfast 6 [2.2.0.4] (Wordfast)

The selection had to balance two separate requirements: on the one hand, the selection had to be as representative as possible of the available TM systems; on the other hand, the selection had to be limited given the impossibility of investigating all available TM systems. The selection was made according to two criteria:

- Including popular TM systems.

[29] With the exception of Heartsome, which requires a conversion into the DOCX format, first introduced with MS Word 2007.

- Including (at the time of selection) less common TM systems that offered interesting features or the potential to gain importance – admittedly a highly subjective judgement.

Popular TM systems at the time of selection (end of 2007) were and are Across, Déjà Vu, MultiTrans, SDL Trados, Transit and Wordfast. Less common were Heartsome Studio and memoQ. In the meantime, memoQ has proven to be innovative and has stood its ground. Heartsome was to some extent a precursor of the ongoing trend of reliance on open standards, see 1.4.5.

If carried out today, the selection would probably be different, see e.g. the selection made by Keller (2011). This limitation applies to all comparative tests: after some time, tested applications may be taken off the market, become obsolete, merge, etc. The market is anything but static: magazines such as MultiLingual and MDÜ regularly publish articles on new applications, new releases etc., see e.g. García and Stevenson (2010), Geldbach (2010c) and Zerfaß (2010).

2.4.3.2 General TM system settings

The TM system settings are important for the repeatability of tests. The standard settings were sometimes modified in order to fully exploit the functionalities offered by the TM systems. However, customization was limited to activating/deactivating existing functions. Some TM systems allow for deeper customization, e.g. memoQ can be enriched with regular expressions, see also 2.4.4.1. Despite its usefulness, this feature goes beyond the presumable knowledge of many users – an assumption confirmed by (Zerfaß, 2010, 17) – and the results do not depend on the TM system, but solely on the user's skills. Any adaptation of this kind would bias the comparability of the results and was consequently avoided. Nonetheless, customizability is an aspect of the evaluation, see 2.2.3, and the customizability options applied when testing placeable or localizable elements will be discussed in the section "Customizability" of each chapter.[30]

The settings that were relevant for all tests are presented below. Further settings, which were relevant only for some test subsets, are described in the section "TM system settings" of each chapter.

Minimum match value

The lowest possible minimum match value was set for all TM systems. This setting was necessary in order to evaluate as many suggestions as possible, but is generally not advisable during translation because it causes unhelpful fuzzy matches to be suggested. The following list includes all TM systems tested along with their default minimum match value, their lowest possible minimum match value, and the path to this setting.

[30] The same considerations and methodology can be found in Makoushina (2008), albeit the test object (quality assurance tools) is different.

- **Across**: default 50%; min. 50%; TOOLS > PROFILE SETTINGS > CROSSDESK > CROSSTANK > CROSSTANK SEARCH SETTINGS > MINIMUM MATCHING DEGREE.
- **Déjà Vu**: default 75%; min. 30%; TOOLS > OPTIONS > GENERAL > DATABASE LOOKUP > MINIMUM SCORE.
- **Heartsome**: default 70%; min. 35%; OPTIONS > SET MINIMUM MATCH PERCENTAGE.
- **memoQ**: default 30%[31]; min. 0%; TOOLS > OPTIONS > TM DEFAULTS, USER INFO > THRESHOLDS AND PENALTIES > MATCH THRESHOLD.
- **MultiTrans**: default 75%; min. 0%; OPTIONS > FUZZY MATCHING > FUZZY FACTOR.
- **SDL Trados 2007**: default 70%; min. 30%; OPTIONS > TRANSLATION MEMORY OPTIONS > GENERAL > MINIMUM MATCH VALUE.
- **SDL Trados Studio**: default 70%; min. 30%; TOOLS > OPTIONS > LANGUAGE PAIRS > ALL LANGUAGE PAIRS > TRANSLATION MEMORY AND AUTOMATED TRANSLATION > SEARCH > TRANSLATION > MINIMUM MATCH VALUE.
- **Transit XV**: default 70%; min. 0%; OPTIONS > PROFILE > SETTINGS > FUZZY INDEX > MIN. QUALITY FOR FUZZY MATCHES (%).
- **Transit NXT**: default 70%; min. 0%; USER PREFERENCE > DUAL FUZZY > SOURCE LANGUAGE > MINIMUM QUALITY (%).
- **Wordfast 5**: default 75%; min. 50%; SETUP > GENERAL > FUZZY THRESHOLD.
- **Wordfast 6**: default 75%; min. 40%; EDIT > PREFERENCES > TRANSLATION MEMORY > FUZZY MATCH THRESHOLD IN (%).

Other settings

Across: under TOOLS > SYSTEM SETTINGS > GENERAL > CROSSTANK, the default option USE RICH TRANSLATION MEMORY is activated. Rich translation memory (rich TM) means that "formatting, inline objects and other characters (control characters, special characters and spaces) are saved in crossTank", Across (2009a). The configuration is carried out under TOOLS > PROFILE SETTINGS > CROSSTANK > STORING, where the options STORE FORMAT RANGES, STORE INLINE OBJECTS and STORE SPECIAL CHARACTERS are active.

Déjà Vu: the options AutoAssemble and AutoPropagate under TOOLS > OPTIONS > ENVIRONMENT are active. If no match can be found for the current source segment, AutoAssemble assembles the translation from several target language fragments coming from

[31] Value set in Kilgray (2008).

dictionaries, terminology databases or translation memories. This functionality depends also on other settings, see (ATRIL, 2003, 97).

If a source segment repeats in a translation project, after confirming its translation when it first occurs, AutoPropagate automatically inserts the translation as a 100% match into all other occurrences. This feature is relevant because of its interaction with automatic adaptations.

memoQ: when creating a translation memory, the option STORE FORMATTING is active. Formatting information as e.g. font attributes (bold, italics etc.) as well as formatting tags are saved in the translation memory.

2.4.4 Suggested improvements

Each chapter presents possible improvements based on the test results. These improvements are either regular expressions or general remarks.

2.4.4.1 Regular expressions

Regular expressions are patterns describing text and are far more flexible than literal strings. For a general and comprehensive introduction to regular expressions, see Friedl (2006), Wiedl (2006), Goyvaerts (2007) and Goyvaerts and Levithan (2009).

Regular expressions are suggested in chapters 3 to 7. The idea of using regular expressions in order to enhance the functionality of TM systems is not new, see e.g. Gintrowicz and Jassem (2007).[32]

Seven regular expressions will be presented for the recognition of:

- Numbers
- Alphanumeric strings
- Strings containing symbols
- Completely capitalized strings
- Mixed-case strings
- URLs
- E-mail addresses

[32] Gintrowicz and Jassem (2007) discuss certain date and time formats (for English and Polish), while currencies, metric expressions, numbers and e-mail addresses are mentioned briefly.

Foregoing considerations

When writing regular expressions, it is necessary to choose a flavor, i.e. a particular regular expression engine and regular expression syntax, see (Goyvaerts and Levithan, 2009, 2-5). For this thesis, the regular expression flavor used by Perl 5 was chosen and the code snippets can be integrated into Perl code.[33] The choice of Perl is motivated by the fact that it is the most popular regular expression flavor (Goyvaerts, 2007, 3-4), and that it did not pose any limitations when engineering the regular expressions.

For some placeable and localizable elements, adequate regular expressions have already been proposed in the reference works cited above. These expressions were analyzed in order to determine which one suits best in a TM system. If necessary, they were adapted. For other placeable and localizable elements, new regular expressions were created. RegexBuddy was used both for checking existing regular expressions and for creating new ones.[34]

When crafting regular expressions, a distinction has to be drawn between matching specific items and validating them. This is particularly clear for dates: 30/02/2010, for example, is not a valid date, although its format is correct. In this thesis, the focus was placed on matching for several reasons. Firstly, validation is not a primary goal within a TM system. Secondly, the regular expressions should be as language-independent as possible. Thirdly, validation generally requires more complex regular expressions and more specific constraints. Furthermore, “there’s often a trade-off between what’s exact, and what’s practical”, (Goyvaerts, 2007, 73), as the discussion of e-mail addresses will show, see 7.4. The regular expressions presented follow the non-validating approach.

Although the regular expressions were crafted with broad language support in mind, see the use of Unicode properties below, they have been tested only on English and German source data and might be unsuitable for other languages.

It was not possible to implement and test the regular expressions in commercial TM systems because it is not possible to obtain the necessary access to the source code. Consequently, further aspects that would arise during implementation, e.g. handling overlaps between matches, could not be covered.

Structure and general characteristics

The match of all regular expressions is included in parentheses because parentheses have a capturing function: the matched text is saved in a variable and can be used later in the regular expression (backreferencing) or in further program code. The latter is what is necessary in a TM system, where the placeable or localizable element is highlighted and can be further processed.

Most regular expressions presented make use of Unicode properties and sub-properties, which are well supported by the Perl flavor, see (Friedl, 2006, 125). Unicode properties are

[33] Note that “regex flavors do not correspond one-to-one with programming languages”, (Goyvaerts and Levithan, 2009, 3).

[34] For more information on RegexBuddy, see Goyvaerts (2010a).

hierarchical categories that group Unicode code points according to their characteristics (e.g. letter, number, etc.).[35] The use of Unicode properties prevents more ambiguous notation. For example, the word character \w can have a very different meaning depending on the regular expression flavor, see (Goyvaerts and Levithan, 2009, 32 and 42-43). In addition, Unicode properties clearly specify character categories, e.g. all characters for currencies are included in \p{Sc}. They allow for more flexibility in the regular expression and keep it more readable, e.g. switching between uppercase and lowercase letters is as simple as \p{Lu}\p{Ll}. Unicode properties do pose some problems, e.g. the attribution of a character to a property can be confusing, in particular if the character itself is ambiguous: for example, the hyphen-minus is included in the punctuation property \p{P} although it can be also used as a mathematical symbol. Such difficulties will be discussed as appropriate.

It has been assumed that the input for the regular expression is a whole segment and that no tokenization[36] has been performed. Consequently, in several of the regular expressions suggested, word boundary anchors (\b) are used. They mark a "location where there is a 'word character' on one side, and not on the other", (Friedl, 2006, 133). This strategy can be problematic because word characters are defined by the ambiguous \w discussed above. However, in Perl, \w reliably includes \p{L}, \p{N} and the low line.

Special attention has been devoted to the problem of combining marks:

> In Unicode, a character with a diacritic mark can often be represented in two ways. You can express é as a *precomposed* character or as a *decomposed*—i.e., as a character pair consisting of "e" and a combining acute accent. Both representations are possible for a large number of commonly used characters [...]. (Korpela, 2006, 225)

The resulting visual form of the character (glyph) is exactly the same. The use of combining marks – still poorly supported in fact, see (Korpela, 2006, 225) – has to be accounted for in the regular expressions. For example, \p{L} recognizes the entire character (é) if precomposed, but recognizes only the base (e) if decomposed. The adopted solution is to allow optional combining marks, which are defined by the Unicode property \p{M}.

In order to increase readability and to add comments easily, the regular expressions are written and presented in free-spacing mode, denoted by /x. For more information, see (Friedl, 2006, 111) and (Goyvaerts and Levithan, 2009, 83-84). As a segment passed over from the TM system can obviously contain multiple instances of the element, /g is used, i.e. the search does not stop after the first match in the segment.

[35] A list of the general categories can be found in (Friedl, 2006, 122), a complete discussion in (Korpela, 2006, 209-291).

[36] (Manning et al., 2009, 22) define tokenization as follows: "Given a character sequence [...], tokenization is the task of chopping it up into [...] tokens [...]", see 2.3.3.1 for a definition of the term token.

2.4.4.2 Retrieval-related remarks

In particular for chapters 8 to 12, improvements to retrieval are designed to prevent the shortcomings that have been pointed out in the tests. Given the impossibility of accessing the source code, the improvements are formulated as remarks and could not be implemented and tested. In addition, sometimes it was not possible to ascertain the exact causes of shortcomings, e.g. whether a specific feature is buggy or missing entirely. Reverse engineering techniques, see also 2.3.1, could have been applied, but – along with further considerations – the effort required in the investigation would not have been sustainable and would have exceeded the scope of this thesis, which does not include debugging the TM systems tested.

The remarks pertain to the objectives defined in 2.2.2: recognition and proper highlighting of the modification, adequate match values and helpful automatic adaptations.

2.4.4.3 Other remarks

The improvements concerning the corollary objectives (display, segmentation, editability, customizability), see 2.2.3, have been formulated as remarks. Display preferences are highly individual. Consequently, more acceptable alternatives, mostly taken from other TM systems, are only presented for overt display shortcomings. As regards issues arising from incorrect segmentation and a lack of editability, brief descriptions of correct support of the source format are provided. Finally, some notes on proper TM system settings are made where unsuitable customization would result in questionable handling of placeable and localizable elements.

2.5 Structure of test descriptions

Chapter design

The chapters containing the test descriptions have a consistent structure. Each chapter is divided into the following sections:

1. An introduction describing the analyzed elements.
2. A test section including:
    - A subsection “TM system settings” that describes the versions and customizability options.
    - An optional subsection “General remarks on TM system behavior” that provides information that allows better interpretation of test set(s).
    - One or more subsections describing the tests and results.
3. Conclusions focusing on the comparison of the results.
4. Possible improvements.

Table conventions

In the tables presenting the recognition results, the recognized character sequence is shown in gray and the following standardized abbreviations are used:

- yes+: full recognition comprising additional elements
- yes: full recognition
- no: no recognition
- p: partial recognition
- ex: over-recognition

In the tables presenting the retrieval results, the percent sign in the match values is omitted.

Part II

Tests

Chapter 3

Numbers

3.1 Introduction

Numbers[1] are common placeable or localizable elements that vary according to language-specific or company-specific rules. Numbers may have to be converted if used in conjunction with measuring units that vary from the source to the target language (cf. imperial system and metric system).

3.2 Tests

The tests are aimed at checking the following issues:

- Are numbers fully recognized?
- Are numbers recognized as a unit or in digit sequences?

The aim of the tests was to assess the performance of number recognition, regardless of how numbers are processed afterwards. Subsequent automatic adaptations will not be tested. As explained in 2.2, the retrieval performance and the match values applied to differences concerning numbers are not covered. For some background on this topic, see (Nübel and Seewald-Heeg, 1999b, 21–22). Recognition assessment does not require that target text be entered, see 2.3.2.1.

Further aims were only relevant for specific test subsets and (if any) are listed at the beginning of the relevant subset. Recognition was assessed at different stages of the translation process (translation or quality assurance/quality control), depending on the TM system, see 3.2.2.

For testing purposes, an approximate semantic classification was made:

- Prices

[1] This chapter covers only numbers consisting of digits, not number words ("three", "ten", etc.).

- Time specifications
- Measuring units
- Telephone numbers
- Standards and versions
- Other numbers

The section “Other numbers” collects miscellaneous examples, classified according to their patterns, see 3.2.3.6. This pattern-based classification could have been used for the preceding sections, but semantic classification was preferred because of its clarity. The recognition of dates will be covered in chapter 4. These tests include some typographic characters that will be dealt with more extensively in chapter 12.

Although only the numbers are presented and discussed, in the test files they had to be accompanied by some “dummy text” in each paragraph. Otherwise they would have been skipped by some TM systems, e.g. Wordfast. In addition, they could not be at the beginning of the paragraph if their format was <number><full stop>. Otherwise they would have been interpreted as part of a numbered list and skipped as well, e.g. by SDL Trados.

3.2.1 TM system settings

3.2.1.1 Versions

TM system	**Version**
Across Standalone Personal Edition	4.00
Déjà Vu X Professional	7.5.303
Heartsome Translation Studio Ultimate	7.0.6 2008-09-12S
memoQ Corporate	3.2.17
MultiTrans	4.3.0.84
SDL Trados 2007 Freelance	8.2.0.835
STAR Transit XV Professional	3.1 SP22 631
Wordfast	5.53

Table 3.1: TM systems used in the tests

3.2.1.2 Customizability

Firstly, the customizability options for number recognition are discussed. Additionally – although the specific feature was not under test – the automatic conversion of numbers is briefly covered. For some TM systems (Déjà Vu, MultiTrans, memoQ and Transit), this feature indirectly confirms that they recognize numbers even though they do not treat them as placeable or localizable elements during translation, see 3.2.2.

Across: under Tools > System settings > General > Language settings >Locale settings > Language sets > Standard language set > Languages > [specific language] > Format > Number and Time, language-dependent number and time formats are specified. They are editable and new ones can be added. In addition, under Standard language set > Format, language-independent number and time formats are specified. They can be modified and supplemented. However, the standard settings were retained.

Across offers automatic conversions for numbers if the option Use autochanges under Tools > System Settings > General > crossTank is activated (default setting) and a Rich TM is used, see 2.4.3.2 for more information.

Déjà Vu: numbers are automatically "taken over from the source to the target", (ATRIL, 2003, 196). This feature works together with AutoPropagate, see 2.4.3.2. In other words, if numbers are the only difference between two segments, the translation will be inserted and the number(s) will be replaced automatically. Numbers are also transferred when AutoAssemble is performed, see 2.4.3.2. This proves that number recognition is available, but it is not customizable.

Déjà Vu offers automatic conversions if the option Allow decimal conversions is activated under Tools > Options > General > Conversions.

Heartsome: the tested version does not provide any number recognition. Neither is a quality assurance component available. Consequently, Heartsome will not be included in these tests. These functions are planned for future versions.

memoQ: the feature Adjust fuzzy hits and inline tags under Tools > Options > TM defaults, user info > Adjustments enables automatic adaptations. It is activated by default and enables "on-the-fly adjustment of numbers [...] within translation memory hits with less than 100% match rate", Kilgray (2008). However, the default pattern for number recognition cannot be viewed.

Additionally, personalized patterns can be defined as auto-translatables. Under Tools > Options > Auto-translatables > Auto-translation rules, the user can define some rules through regular expressions that enable e.g. the recognition and transfer of numbers. However, this requires some knowledge that cannot be taken for granted for all users. Moreover, recognition can only be as good as the regular expression itself, in other words the performance no longer depends on the software. Therefore, this feature was not used, see 2.4.3.2.

MultiTrans: segments can be automatically adapted if they differ only by a number. The old number is replaced by the new one and a 100% match is generated.[2] However, the default pattern used for number recognition cannot be viewed or modified. This automatic adaptation feature is built-in and cannot be deactivated.

SDL Trados: under File > Setup > Substitutions > Numbers and Measurements, it is possible to "define [...] numbers [...] as variables rather than normal words in

[2] Information obtained in an e-mail exchange with MultiCorpora.

Translator's Workbench. [...] Translator's Workbench recognises designated variables and treats them as non-translatable or placeable elements during translation", SDL (2007). This option is activated by default. However, it is not possible to view or modify the patterns used for recognition.

In addition, under Options > Translation Memory Options > Substitution Localisation > Numbers and Measurements, it is possible to define "if, and how, Translator's Workbench should automatically adapt the format of variable elements to suit the target language [...]", SDL (2007). The digit grouping symbol and the decimal symbol can be specified.

Transit: in the automatic pretranslation, "if a source-language sentence in the reference material and a source-language sentence in the current project differ only in terms of numbers and/or formatting, [...] the numbers and tags are taken from the current source-language file", STAR (2005). However, the default pattern used for number recognition cannot be viewed or modified and no conversion option is provided.

Wordfast: numbers are by default considered as placeables, defined as "[...] typically untranslatable items (like numbers, fields, tags) [...]", Champollion (2008b). The default pattern used for number recognition cannot been viewed. No conversion option is provided.

3.2.2 General remarks on TM system behavior

Display

For numbers, TM systems have to be divided into two groups. Some recognize and highlight numbers in the source text during editing, while others do not, see table 3.2. When numbers are recognized, they can be placed into the target segment by means of a shortcut. In the TM system documentation, they are referred to as "placeables" for this reason. Failing to display numbers as placeables does not imply that they are always treated as plain text, see 3.2.1.2 for more information.

	Numbers highlighted?	**Shortcut**
Across	yes	yes
Déjà Vu	no	-
Heartsome	no	-
memoQ	no	-
MultiTrans	no	-
SDL Trados	yes	yes
Transit	no	-
Wordfast	yes	yes

Table 3.2: Highlighting of numbers

Table 3.3 shows how recognized numbers are displayed.

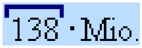
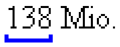
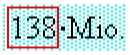
	Display
Across	 138 Mio.
SDL Trados	 138 Mio.
Wordfast	 138 Mio.

Table 3.3: Display of recognized numbers

Across: recognized numbers are marked with a blue overline. The current element can be transferred with the shortcut Ctrl+Alt+0.

SDL Trados: recognized numbers are marked with a blue underline. The current element can be transferred with the shortcut Ctrl+Alt+↓.

Wordfast: the current recognized number is marked with a red square. The current element can be transferred with the shortcut Ctrl+Alt+↓.

Recognition assessment

For TM systems that highlight numbers in their editor (Across, SDL Trados and Wordfast), assessing what has been recognized is relatively straightforward. For all other TM systems, it is necessary to adopt a different strategy. This strategy consists of using the quality check function for numbers that is available for almost all TM systems, see table 3.4. This function reveals which numbers are recognized, and in nearly all cases, also indicates whether recognition covers numbers as units or splits them.

	Quality check tool
Across	Quality Management
Déjà Vu	Terminology Check
Heartsome	(planned)
memoQ	QA
MultiTrans	QA add-on
SDL Trados	QA checker (with TagEditor)
Transit	Check formatting codes
Wordfast	Quality Check

Table 3.4: Quality check tools

Déjà Vu: Déjà Vu does not provide any specific transfer function for numbers (unlike for tags, with the shortcut F8). The system supports users with the functions AutoPropagate and AutoAssemble, which copy numbers into the target segment, see 2.4.3.2 for more information. However, these functions are not suitable in order to ascertain whether Déjà Vu recognizes some numbers as a unit or splits them into several digit sequences. Therefore, a different strategy was adopted.

With Déjà Vu it is possible to verify numbers (Check Numerals). If the numbers in the source and in the target text differ, the digits will be displayed in red and the current segment will be marked with a red exclamation mark. During the tests, numbers were deliberately changed and successively verified. By changing one digit, it is possible to check whether the number as a unit is red or only some of its digits. An example is provided by segment 4 in 3.2.3.2, where 8:00 is recognized as two numbers separated by the colon.

memoQ: memoQ does not provide any standard transfer function for numbers, but a quality assurance function for verifying numbers was used for test purposes. The QA check in memoQ indicates only if there is a general number difference between the source and target segment. A general error description is provided, e.g. "Non-standard number format in the target side" or "Numbers in source and target segment do not match", however it is not always easy to ascertain the exact error because the diverging digits are not marked (as e.g. in Déjà Vu). Fortunately, recognition of decimal points and thousand separators as part of the number – particularly relevant for these tests – can be checked: if the decimal point or the thousand separator is modified, the QA checker recognizes the modification.

MultiTrans: MultiTrans does not provide any standard transfer function for numbers. An add-on, the QA Agent, can be used to verify whether numbers in the source and target text correspond. Because of copyright regulations, it was not possible to obtain the necessary license. Therefore, no recognition assessment could be made and MultiTrans is not included in these tests.

Transit: Transit does not provide any standard transfer function for numbers, but it is possible to verify them with the option Options > Check Formatting Codes > Settings > Numbers. If the numbers in the source and in the target text differ, a warning is displayed quoting the altered number.

Others: for Across and Wordfast, the recognition results in the editor were cross-checked with the quality control function. No divergences were ascertained, i.e., digits that were not recognized were not checked either. In the case of SDL Trados, since MS Word was used as the editor, no number check was available.

Scoring system

For a better comparison of the performance of TM systems, a scoring system was introduced. The reasons why usual precision and recall were not suitable have already been explained in 2.3.3.1. As regards precision, false positives are virtually absent in the test suites. Exceptions occur only for Wordfast, which applies extended alphanumeric recognition. These examples will be discussed, but do not in any way offer a basis broad enough for a sound numeric conclusion. As regards recall, a more elaborate metric provides a more accurate account of the performance of the TM systems, see 3.3.2.

A score was calculated for each instance of recognition according to the following rules:

- 3 points were assigned when all digits were recognized.

- If digits were separated by non-standard separators (such as a solidus e.g. in “1/4”), they were treated as independent sequences.
- 0.5 points were assigned for each correctly recognized thousand or decimal separator, only if preceded or followed by digits (e.g. in “0.70”).
- Bonus of 0.5 points was assigned when ranges consisting of two or more independent sequences were recognized (e.g. in “0.20-0.40”). Partial recognitions were excluded. No extra bonus was applied to the non-alphanumeric character (usually a hyphen-minus sign) used to define the range.
- Bonus of 0.25 points was assigned for each correctly recognized additional character (e.g. a mathematical operator such as the plus sign in “+10%”). When a sequence of the same character type was recognized (e.g. alphabetic characters), the bonus was applied only once.
- Penalty of 0.5 points was applied in the case of over-recognition (e.g. the word “Seiten”, meaning pages, in “A4-Seiten”).
- Minor misrecognitions (hyphen-minus sign interpreted as a minus character when used in a range) were ignored (no bonus, no penalty).

A baseline[3] was calculated: it assumes complete recognition of all digits and of all separators (excluding non-standard separators such as a solidus, etc.).

In order to prevent bonuses compensating for poor digit recognition – which is deemed the most serious problem – low bonus values were set. Standard separators were included in the minimum match value because, in particular for measures and to some extent times, correct recognition of these elements is a prerequisite for any automatic adaptation.

Here are some score examples (the recognized characters are printed in gray, as in the tables):

1.234,5 = 3 (all digits recognized) + 0.5 (one separator (comma) correctly recognized). The baseline is 3+0.5*2=4.

30 - 250 V = 3 + 3 (all digits recognized, 2 sequences) + 0.25 (V recognized) + 0.5 (range recognized). The baseline is 3+3=6.

In order to make the score more transparent, each table indicates how the score was calculated and which elements were considered. The elements are referred to with the following abbreviations:

- D: digit(s)
- S: separator
- R: range

[3] The notion of baseline, see (Resnik and Lin, 2010, 279-280), is very close to that of golden score, see (Manning et al., 2009, 152).

- A: additional character(s)
- O: over-recognition

3.2.3 Test suite

3.2.3.1 Prices

The following examples were used:

	Text	**Baseline**	**Calculation**
1	95.-	3	D
2	95.–	3	D
3	3'000	3.5	D+S
4	12.––	3	D
5	0.20	3.5	D+S
6	2,-/Tag	3	D
7	250.00	3.5	D+S
8	-.10	3	D
9	138 Mio.	3	D
10	CHF 202.60	3.5	D+S
11	0.20-0.40	7 (3.5x2)	(D+S)x2
12	320.00, 50.00 und 100.00	10.5 (3.5x3)	(D+S)x3
13	1'550.00	4	D+(Sx2)
14	0.70*	3.5	D+S
		57	**Total**

Table 3.5: Prices: test examples

Results

Across: see table 3.6. Across recognizes the digits but not the separators. This shortcoming is due to the standard settings under Tools > System settings > General > Language settings > Locale settings > Language sets > Standard language set > Languages > [Deutsch] > Format > Number. For each language, Across has specific requirements concerning what a number looks like. If the number has a different structure, e.g. because another decimal separator is used, recognition is not possible.[4] This limitation may cause problems because number formats may not be consistent within a language or even within the same document. This is the case with this test set, where German texts used the English standard (most likely because of a company-specific convention). With a different source language (English), recognition would have included the decimal separators in the examples.

[4] This is confirmed by the Across online help: "If the source segment contains an 'incorrect' number format – i.e. a number format that does not correspond to the particular language setting [...] the number will not be changed automatically", Across (2009a).

Déjà Vu: see table 3.7. Déjà Vu can recognize several separators if they are followed or preceded by numbers. The apostrophe is an exception.

memoQ: see table 3.8. memoQ recognizes all separators as part of the number if followed or preceded by digits.

SDL Trados: see table 3.9. The hyphen-minus sign before a digit is interpreted as a minus character (example 11).

Transit: see table 3.10. Separators are recognized as part of the number if they are followed or preceded by numbers. However, not all of them are recognized correctly, e.g. the apostrophe. Moreover, a hyphen-minus sign before a digit is interpreted as a minus character (example 11).

Wordfast: see table 3.11. Wordfast recognizes all numbers and it is the only system that can recognize ranges correctly (example 11). It correctly handles decimal and thousand separators. But there is over-recognition in example 6.

	Recognition	**Result**	**Score**	**Calculation**
1	yes	95.-	3	D
2	yes	95.–	3	D
3	p	3'000	3	D
4	yes	12.––	3	D
5	p	0.20	3	D
6	yes	2,-/Tag	3	D
7	p	250.00	3	D
8	yes	-.10	3	D
9	yes	138 Mio.	3	D
10	p	CHF 202.60	3	D
11	p	0.20-0.40	6 (3x2)	Dx2
12	p	320.00, 50.00 und 100.00	9 (3x3)	Dx3
13	p	1'550.00	3	D
14	p	0.70*	3	D
			51	**Total**

Table 3.6: Prices: Across

	Recognition	**Result**	**Score**	**Calculation**
1	yes	95.-	3	D
2	yes	95.–	3	D
3	p	3'000	3	D
4	yes	12.——	3	D
5	yes	0.20	3.5	D+S
6	yes	2,-/Tag	3	D
7	yes	250.00	3.5	D+S
8	yes	-.10	3	D
9	yes	138 Mio.	3	D
10	yes	CHF 202.60	3.5	D+S
11	yes	0.20-0.40	7 (3.5x2)	(D+S)x2
12	yes	320.00, 50.00 und 100.00	10.5 (3.5x3)	(D+S)x3
13	p	1'550.00	3.5	D+S
14	yes	0.70*	3.5	D+S
			56	**Total**

Table 3.7: Prices: Déjà Vu

	Recognition	**Result**	**Score**	**Calculation**
1	yes	95.-	3	D
2	yes	95.–	3	D
3	yes	3'000	3.5	D+S
4	yes	12.——	3	D
5	yes	0.20	3.5	D+S
6	yes	2,-/Tag	3	D
7	yes	250.00	3.5	D+S
8	yes	-.10	3	D
9	yes	138 Mio.	3	D
10	yes	CHF 202.60	3.5	D+S
11	yes	0.20-0.40	7 (3.5x2)	(D+S)x2
12	yes	320.00, 50.00 und 100.00	10.5 (3.5x3)	(D+S)x3
13	yes	1'550.00	4	D+(Sx2)
14	yes	0.70*	3.5	D+S
			57	**Total**

Table 3.8: Prices: memoQ

	Recognition	**Result**	**Score**	**Calculation**
1	yes	95.-	3	D
2	yes	95.–	3	D
3	yes	3'000	3.5	D+S
4	yes	12.−−	3	D
5	yes	0.20	3.5	D+S
6	yes	2,-/Tag	3	D
7	yes	250.00	3.5	D+S
8	yes	-.10	3	D
9	yes	138 Mio.	3	D
10	yes	CHF 202.60	3.5	D+S
11	yes	0.20-0.40	7 (3.5x2)	(D+S)x2
12	yes	320.00, 50.00 und 100.00	10.5 (3.5x3)	(D+S)x2
13	yes	1'550.00	4	D+(Sx2)
14	yes	0.70*	3.5	D+S
			57	**Total**

Table 3.9: Prices: SDL Trados

	Recognition	**Result**	**Score**	**Calculation**
1	yes	95.-	3	D
2	yes	95.–	3	D
3	p	3'000	3	D
4	yes	12.−−	3	D
5	yes	0.20	3.5	D+S
6	yes	2,-/Tag	3	D
7	yes	250.00	3.5	D+S
8	yes	-.10	3	D
9	yes	138 Mio.	3	D
10	yes	CHF 202.60	3.5	D+S
11	yes	0.20-0.40	7 (3.5x2)	(D+S)x2
12	yes	320.00, 50.00 und 100.00	10.5 (3.5x3)	(D+S)x3
13	p	1'550.00	3.5	D+S
14	yes	0.70*	3.5	D+S
			56	**Total**

Table 3.10: Prices: Transit

	Recognition	Result	Score	Calculation
1	yes	95.-	3	D
2	yes	95.–	3	D
3	yes	3'000	3.5	D+S
4	yes	12.——	3	D
5	yes	0.20	3.5	D+S
6	ex	2,-/Tag	2.5	D-O
7	yes	250.00	3.5	D+S
8	yes	-.10	3	D
9	yes	138 Mio.	3	D
10	yes+	CHF 202.60	3.75	D+S+A
11	yes+	0.20-0.40	7.5 (3.5x2+0.5)	(D+S)x2+R
12	yes	320.00, 50.00 und 100.00	10.5 (3.5x3)	(D+S)x3
13	yes	1'550.00	4	D+(Sx2)
14	yes	0.70*	3.5	D+S
			57.25	**Total**

Table 3.11: Prices: Wordfast

3.2.3.2 Time specifications

In addition to the overall aims (see 3.2), this test subset investigates the following question:

- Do alphanumeric strings prevent the recognition of numeric time specifications?

The following examples were used:

	Text	Baseline	Calculation
1	3 Monate gratis	3	D
2	1x jährlich	3	D
3	in 3-5 Jahren	6 (3x2)	Dx2
4	von 8:00 bis 17:00	7 (3.5x2)	(D+S)x2
5	>3 Minuten	3	D
6	7x24h	6 (3x2)	Dx2
7	30min.	3	D
8	2 $\frac{1}{2}$ Std.	6 (3x2)	Dx2
9	1.50/h	3.5	D+S
		40.5	**Total**

Table 3.12: Time specifications: test examples

These examples present several difficulties: there are unusual constructions where the number and the time specification are not separated by a space. A fraction ($\frac{1}{2}$) also occurs.

Results

Across: see table 3.13. As already noted in 3.2.3.1, Across does not recognize ranges (example 3) and recognizes only some separators (example 4 vs. example 9). In alphanumeric strings (examples 2 and 6), only the numeric part is recognized. The fraction is not recognized (example 8).

Déjà Vu: see table 3.14. Recognition is mainly limited to digits. The hour is not recognized as a unit (example 4). The presence of alphabetic characters can hinder digit recognition (example 6). The fraction is not recognized (example 8).

memoQ: see table 3.15. The hour is not recognized as a unit (example 4). The fraction is not recognized (example 8).

SDL Trados: see table 3.16. Recognition fails for some alphanumeric sequences (examples 2 and 6) as well as for the fraction (example 8). Separators (including the colon) are recognized, but ranges are not (the hyphen-minus sign is interpreted as a minus sign, see example 3). The "min" abbreviation is recognized.

Transit: see table 3.17. Recognition fails only with an alphanumeric sequence (example 6). The colon is not recognized as a separator and the hyphen-minus sign in ranges is interpreted as a minus sign. The fraction is recognized.

Wordfast: see table 3.18. Recognition is successful for every digit, except for the fraction (example 8). Alphanumeric sequences are recognized too, even with a solidus (example 9). Ranges (example 3) and separators are interpreted correctly.

	Recognition	**Result**	**Score**	**Calculation**
1	yes	3 Monate gratis	3	D
2	yes	1x jährlich	3	D
3	yes	in 3-5 Jahren	6 (3x2)	Dx2
4	yes	von 8:00 bis 17:00	7 (3.5x2)	(D+S)x2
5	yes	>3 Minuten	3	D
6	yes	7x24h	6 (3x2)	Dx2
7	yes	30min.	3	D
8	p	2 $\frac{1}{2}$ Std.	3	D
9	p	1.50/h	3	D
			37	**Total**

Table 3.13: Time specifications: Across

	Recognition	**Result**	**Score**	**Calculation**
1	yes	3 Monate gratis	3	D
2	yes	1x jährlich	3	D
3	yes	in 3-5 Jahren	6 (3x2)	Dx2
4	p	von 8:00 bis 17:00	6 (3x2)	Dx2
5	yes	>3 Minuten	3	D
6	p	7x24h	3	D
7	yes	30min.	3	D
8	p	2 $\frac{1}{2}$ Std.	3	D
9	yes	1.50/h	3.5	D+S
			33.5	**Total**

Table 3.14: Time specifications: Déjà Vu

	Recognition	**Result**	**Score**	**Calculation**
1	yes	3 Monate gratis	3	D
2	yes	1x jährlich	3	D
3	yes	in 3-5 Jahren	6 (3x2)	Dx2
4	p	von 8:00 bis 17:00	6 (3x2)	Dx2
5	yes	>3 Minuten	3	D
6	yes	7x24h	6 (3x2)	Dx2
7	yes	30min.	3	D
8	p	$2\ \frac{1}{2}$ Std.	3	D
9	yes	1.50/h	3.5	D+S
			36.5	**Total**

Table 3.15: Time specifications: memoQ

	Recognition	**Result**	**Score**	**Calculation**
1	yes	3 Monate gratis	3	D
2	no	1x jährlich	0	
3	yes	in 3-5 Jahren	6 (3x2)	Dx2
4	yes	von 8:00 bis 17:00	7 (3.5x2)	(D+S)x2
5	yes	>3 Minuten	3	D
6	no	7x24h	0	
7	yes+	30min.	3.25	D+A
8	p	$2\ \frac{1}{2}$ Std.	3	D
9	yes	1.50/h	3.5	D+S
			28.75	**Total**

Table 3.16: Time specifications: SDL Trados

	Recognition	**Result**	**Score**	**Calculation**
1	yes	3 Monate gratis	3	D
2	yes	1x jährlich	3	D
3	yes	in 3-5 Jahren	6 (3x2)	Dx2
4	p	von 8:00 bis 17:00	6 (3x2)	Dx2
5	yes	>3 Minuten	3	D
6	no	7x24h	0	
7	yes	30min.	3	D
8	yes	$2\ \frac{1}{2}$ Std.	6 (3x2)	Dx2
9	yes	1.50/h	3.5	D+S
			33.5	**Total**

Table 3.17: Time specifications: Transit

	Recognition	Result	Score	Calculation
1	yes	3 Monate gratis	3	D
2	yes+	1x jährlich	3.25 (3+0.25)	D+A
3	yes+	in 3-5 Jahren	6.5 (3x2+0.5)	(Dx2)+R
4	yes	von 8:00 bis 17:00	7 (3.5x2)	(D+S)x2
5	yes	>3 Minuten	3	D
6	yes+	7x24h	6.5 (3x2+0.25x2)	(Dx2)+(Ax2)
7	yes+	30min.	3.25 (3+0.25)	D+A
8	p	2 $\frac{1}{2}$ Std.	3	D
9	yes+	1.50/h	4 (3+0.5+0.25x2)	D+S+(Ax2)
			39.5	**Total**

Table 3.18: Time specifications: Wordfast

3.2.3.3 Measures

In addition to the overall aims (see 3.2), this test subset investigates the following questions:

- Is additional information (e.g. the measuring unit) recognized besides the numeric measure?
- Are further characters recognized (e.g. hyphen-minus sign)?

The abbreviations of measuring units are often standardized and language-independent. So it can be useful to recognize them together with the number. Depending on the subject of the corpus (physics, chemistry, etc.), the frequency of measuring units will vary significantly.

The following examples were used:

	Text	**Baseline**	**Calculation**
1	360°	3	D
2	10°C	3	D
3	25 °C bis + 70 °C	6 (3x2)	Dx2
4	10 KB	3	D
5	10KB	3	D
6	7,05 cm	3.5	D+S
7	25m	3	D
8	25ms	3	D
9	1 1/4"	9 (3x3)	Dx3
10	20A	3	D
11	230V	3	D
12	3x400V	6 (3x2)	Dx2
13	0.60m - 3.00m	7 (3.5x2)	(D+S)x2
14	13,7km	3.5	D+S
15	-48 V	3	D
16	29,124,000/25=1,165kH	10.5 (4+3+3.5)	(Dx3)+(Sx3)
17	2x34 Mbps	6 (3x2)	Dx2
18	64/128/192/256 kbit/s	12 (3x4)	Dx4
19	44.1kHz	3.5	D+S
20	<50 Ohm	3	D
21	J.41 32 kHz	6 (3x2)	Dx2
22	1'000Mpbs	3.5	D+S
23	125x71.6x18.7 mm	10 (3+3.5x2)	(Dx3)+(Sx2)
24	170 x 170 x 68 mm	9 (3x3)	Dx3
25	0,3 m bis 1,8 m	7 (3.5x2)	(D+S)x2
		132.5	**Total**

Table 3.19: Measures: test examples

Results

Across: see table 3.20. Measuring units are not recognized as part of the measure. The hyphen-minus sign is recognized only if not separated by a space from the number. The degree sign, the less-than sign and the plus sign are not recognized. If the separator does not comply with the standard settings, numbers are split (example 23).

Déjà Vu: see table 3.21. Measuring units, the hyphen-minus sign, the degree sign, the less-than sign as well as the plus sign are not recognized. In alphanumeric sequences only the first number is recognized (examples 12, 17 and 23). Certain thousand separators are not recognized (example 22).

memoQ: see table 3.22. Measuring units, the hyphen-minus sign, the degree sign, the less-than sign as well as the plus sign are not recognized. All separators are correctly recognized. Only digits are recognized in alphanumeric sequences.

SDL Trados: see table 3.23. Measuring units are recognized in several cases, even if they are separated by a space from the digits. However, this is not always the case (examples 8 and 18), because they are not among the supported measurement units, see SDL (2007). The hyphen-minus sign is recognized if not separated by a space. The degree sign is included in recognition. The less-than sign as well as the plus sign are not recognized. Alphanumeric constructs are not recognized (examples 12 and 16); SDL Trados skips them altogether.

Transit: see table 3.24. Measuring units are excluded from recognition. The hyphen-minus sign is recognized if not separated by a space. The degree sign, the less-than sign and the plus sign are not recognized. Complex alphanumeric sequences are not recognized completely (examples 12 and 17). The apostrophe is not dealt with correctly, but the solidus is.

Wordfast: see table 3.25. Measuring units are recognized only if they are not separated by a space from the digits, i.e. if they build an alphanumeric sequence. The hyphen-minus sign, the degree sign, the less-than sign as well as the plus sign are not recognized as part of the number. All separators are included in the number recognition.

	Recognition	**Result**	**Score**	**Calculation**
1	yes	360°	3	D
2	yes	10°C	3	D
3	yes	25 °C bis + 70 °C	6 (3x2)	Dx2
4	yes	10 KB	3	D
5	yes	10KB	3	D
6	yes	7,05 cm	3.5	D+S
7	yes	25m	3	D
8	yes	25ms	3	D
9	yes	1 1/4"	9 (3x3)	Dx3
10	yes	20A	3	D
11	yes	230V	3	D
12	yes	3x400V	6 (3x2)	Dx2
13	p	0.60m - 3.00m	6 (3x2)	Dx2
14	yes	13,7km	3.5	D+S
15	yes+	-48 V	3.25 (3+0.25)	D+A
16	p	29,124,000/25=1,165kH	9.5 (3+3+3.5)	(Dx3)+S
17	yes	2x34 Mbps	6 (3x2)	Dx2
18	yes	64/128/192/256 kbit/s	12 (3x4)	Dx4
19	p	44.1kHz	3	D
20	yes	<50 Ohm	3	D
21	yes	J.41 32 kHz	6 (3x2)	Dx2
22	p	1'000Mpbs	3	D
23	p	125x71.6x18.7 mm	9 (3x3)	Dx3
24	yes	170 x 170 x 68 mm	9 (3x3)	Dx3
25	yes	0,3 m bis 1,8 m	7 (3.5x2)	(D+S)x2
			128.75	**Total**

Table 3.20: Measures: Across

	Recognition	**Result**	**Score**	**Calculation**
1	yes	360°	3	D
2	yes	10°C	3	D
3	yes	25 °C bis + 70 °C	6 (3x2)	Dx2
4	yes	10 KB	3	D
5	yes	10KB	3	D
6	yes	7,05 cm	3.5	D+S
7	yes	25m	3	D
8	yes	25ms	3	D
9	yes	1 1/4"	9 (3x3)	Dx3
10	yes	20A	3	D
11	yes	230V	3	D
12	p	3x400V	3	D
13	yes	0.60m - 3.00m	7 (3.5x2)	(D+S)x2
14	yes	13,7km	3.5	D+S
15	yes	-48 V	3	D
16	yes	29,124,000/25=1,165kH	10.5 (4+3+3.5)	(Dx3)+(Sx3)
17	p	2x34 Mbps	3	D
18	yes	64/128/192/256 kbit/s	12 (3x4)	Dx4
19	yes	44.1kHz	3.5	D+S
20	yes	<50 Ohm	3	D
21	yes	J.41 32 kHz	6 (3x2)	Dx2
22	p	1'000Mpbs	3	D
23	p	125x71.6x18.7 mm	3	D
24	yes	170 x 170 x 68 mm	9 (3x3)	Dx3
25	yes	0,3 m bis 1,8 m	7 (3.5x2)	(D+S)x2
			119	**Total**

Table 3.21: Measures: Déjà Vu

	Recognition	**Result**	**Score**	**Calculation**
1	yes	360°	3	D
2	yes	10°C	3	D
3	yes	25 °C bis + 70 °C	6 (3x2)	Dx2
4	yes	10 KB	3	D
5	yes	10KB	3	D
6	yes	7,05 cm	3.5	D+S
7	yes	25m	3	D
8	yes	25ms	3	D
9	yes	1 1/4"	9 (3x3)	Dx3
10	yes	20A	3	D
11	yes	230V	3	D
12	yes	3x400V	6 (3x2)	Dx2
13	yes	0.60m - 3.00m	7 (3.5x2)	(D+S)x2
14	yes	13,7km	3.5	D+S
15	yes	-48 V	3	D
16	yes	29,124,000/25=1,165kH	10.5 (4+3+3.5)	(Dx3)+(Sx3)
17	yes	2x34 Mbps	6 (3x2)	Dx2
18	yes	64/128/192/256 kbit/s	12 (3x4)	Dx3
19	yes	44.1kHz	3.5	D+S
20	yes	<50 Ohm	3	D
21	yes	J.41 32 kHz	6 (3x2)	Dx2
22	yes	1'000Mpbs	3.5	D+S
23	yes	125x71.6x18.7 mm	10 (3+3.5x2)	(Dx3)+(Sx2)
24	yes	170 x 170 x 68 mm	9 (3x3)	Dx3
25	yes	0,3 m bis 1,8 m	7 (3.5x2)	(D+S)x2
			132.5	**Total**

Table 3.22: Measures: memoQ

	Recognition	**Result**	**Score**	**Calculation**
1	yes+	360°	3.25 (3+0.25)	D+A
2	yes+	10°C	3.5 (3+0.25x2)	D+(Ax2)
3	yes+	25 °C bis + 70 °C	7 (3x2+0.25x4)	(Dx2)+(Ax4)
4	yes+	10 KB	3.25 (3+0.25)	D+A
5	yes+	10KB	3.25 (3+0.25)	D+A
6	yes+	7,05 cm	3.75 (3.5+0.25)	D+S+A
7	yes+	25m	3.25 (3+0.25)	D+A
8	no	25ms	0	
9	yes	1 1/4"	9 (3x3)	Dx3
10	yes+	20A	3.25 (3+0.25)	D+A
11	yes+	230V	3.25 (3+0.25)	D+A
12	no	3x400V	0	
13	yes+	0.60m - 3.00m	7.5 (3.5x2+0.25x2)	(D+S+A)x2
14	yes+	13,7km	3.75 (3.5+0.25)	D+S+A
15	yes+	-48 V	3.5 (3+0.25x2)	D+(Ax2)
16	p	29,124,000/25=1,165kH	4	D+(Sx2)
17	no	2x34 Mbps	0	
18	yes	64/128/192/256 kbit/s	12 (3x4)	Dx4
19	yes+	44.1kHz	3.75 (3.5+0.25)	D+S+A
20	yes	<50 Ohm	3	D
21	p	J.41 32 kHz	3.25 (3+0.25)	D+A
22	p	1'000Mpbs	0	
23	no	125x71.6x18.7 mm	0	
24	yes+	170 x 170 x 68 mm	9.25 (3x3+0.25)	(Dx3)+A
25	yes+	0,3 m bis 1,8 m	7.5 (3.5x2+0.25x2)	(D+S+A)x2
			100.25	**Total**

Table 3.23: Measures: SDL Trados

	Recognition	**Result**	**Score**	**Calculation**
1	yes	360°	3	D
2	yes	10°C	3	D
3	yes	25 °C bis + 70 °C	6 (3x2)	Dx2
4	yes	10 KB	3	D
5	yes	10KB	3	D
6	yes	7,05 cm	3.5	D+S
7	yes	25m	3	D
8	yes	25ms	3	D
9	yes+	1 1/4"	9.25 (3x3+0.25)	(Dx3)+A
10	yes	20A	3	D
11	yes	230V	3	D
12	p	3x400V	3	D
13	yes	0.60m - 3.00m	7 (3.5x2)	(D+S)x2
14	yes	13,7km	3.5	D+S
15	yes+	-48 V	3.25 (3+0.25)	D+A
16	yes+	29,124,000/25=1,165kH	10.75 (4+0.25+3+3.5)	(Dx3)+(Sx3)+A
17	p	2x34 Mbps	3	D
18	yes+	64/128/192/256 kbit/s	12.5 (3x4+0.5)	(Dx4)+R
19	yes	44.1kHz	3.5	D+S
20	yes	<50 Ohm	3	D
21	yes	J.41 32 kHz	6 (3x2)	Dx2
22	p	1'000Mpbs	3	D
23	p	125x71.6x18.7 mm	3	D
24	yes	170 x 170 x 68 mm	9 (3x3)	Dx3
25	yes	0,3 m bis 1,8 m	7 (3.5x2)	(D+S)x2
			120.25	**Total**

Table 3.24: Measures: Transit

	Recognition	Result	Score	Calculation
1	yes	360°	3	D
2	yes+	10°C	3.5 (3+0.25x2)	D+(Ax2)
3	yes	25 °C bis + 70 °C	6 (3x2)	Dx2
4	yes	10 KB	3	D
5	yes+	10KB	3.25 (3+0.25)	D+A
6	yes	7,05 cm	3.5	D+S
7	yes+	25m	3.25 (3+0.25)	D+A
8	yes+	25ms	3.25 (3+0.25)	D+A
9	yes+	1 1/4"	9.25 (3x3+0.25)	(Dx3)+A
10	yes+	20A	3.25 (3+0.25)	D+A
11	yes+	230V	3.25 (3+0.25)	D+A
12	yes+	3x400V	6.5 (3x2+0.25x2)	(D+A)x2
13	yes+	0.60m - 3.00m	7.5 (3.5x2+0.25x2)	(D+S+A)x2
14	yes+	13,7km	3.75 (3.5+0.25)	D+S+A
15	yes	-48 V	3	D
16	yes+	29,124,000/25=1,165kH	11.25 (4+3+3.5+0.5+0.25)	(D+S)x3+R+A
17	yes+	2x34 Mbps	6.25 (3x2+0.25)	(Dx2)+A
18	yes+	64/128/192/256 kbit/s	12.5 (3x4+0.5)	(Dx4)+R
19	yes+	44.1kHz	3.75 (3.5+0.25)	D+S+A
20	yes	<50 Ohm	3	D
21	yes+	J.41 32 kHz	6.5 (0.25x2+3x2)	(Ax2)+(Dx2)
22	yes	1'000Mpbs	3.5	D+S
23	yes+	125x71.6x18.7 mm	10.5 (3+3.5x2+0.5)	(Dx3)+(Sx2)+R
24	yes	170 x 170 x 68 mm	9 (3x3)	Dx3
25	yes	0,3 m bis 1,8 m	7 (3.5x2)	(D+S)x2
			138.5	**Total**

Table 3.25: Measures: Wordfast

3.2.3.4 Telephone numbers

Telephone numbers usually consist of several digit sequences as well as some non-numeric characters such as the solidus, the plus sign and the hyphen-minus sign.

The following examples were used:

	Text	**Baseline**	**Calculation**
1	Gratisnummer 0800 888 888	3	D
2	Telefon 044/888 88 88	6 (3x2)	Dx2
3	Telefon +41 88 888 88 88	3	D
4	Telefon 0800 44 44 44 + 4444	6 (3x2)	Dx2
5	Telefon +44-44-1234567	9 (3x3)	Dx3
		27	**Total**

Table 3.26: Telephone numbers: test examples

When the separator is a space, a single sequence should be recognized. However, TM systems that fail to do so are not penalized. With other separators, several sequences are possible.

Results

Across: see table 3.27. Telephone numbers are never recognized as one single unit; each non-numeric character is treated as a separator. The plus sign is recognized if it is not separated from the digits by a space.

Déjà Vu: see table 3.28. Telephone numbers are never recognized as one single unit; each non-numeric character is treated as a separator. The plus sign is never recognized.

memoQ: see table 3.29.[5] Telephone numbers are never recognized as one single unit; each non-numeric character is treated as a separator. The plus sign is never recognized.

SDL Trados: see table 3.30. Some numeric sequences separated by a space are recognized as one unit (examples 1 and 3), but never include a complete telephone number. Some numeric sequences are not recognized at all (examples 2 and 4). The plus sign is recognized only if it is not separated from the digits by a space. The hyphen-minus sign is interpreted as a minus.

Transit: see table 3.31. When the digit sequences that constitute the telephone numbers are separated by a space or a solidus, the number is recognized as one sequence (example 1 to 3). With the plus sign in example 4, the number is split into several sequences. The plus sign is recognized only if it is not separated from the digits by a space. The hyphen-minus sign is interpreted as a minus and splits the number into several sequences (example 5).

[5] If the option Check for number format on the target side is active under Project Settings > Project QA Settings > Numbers and the numbers of the source segment are copied into the target segment, the following error message is displayed for the segments 1 and 5: “Non-standard number format in the target side”. This is most likely because number format rules concerning separators are erroneously applied to telephone numbers as well. On the other hand, it is not clear why this error is not displayed for the segments 2, 3 and 4.

Wordfast: see table 3.32. Telephone numbers are not recognized as a single unit if digit sequences are separated by spaces. The solidus is recognized as part of the number (example 2), but not the plus sign. The hyphen-minus sign is included in the number (example 5), which is recognized as one sequence.

	Recognition	Result	Score	Calculation
1	yes	Gratisnummer 0800 888 888	3	D
2	yes	Telefon 044/888 88 88	6 (3x2)	Dx2
3	yes+	Telefon +41 88 888 88 88	3.25 (3+0.25)	D+A
4	yes	Telefon 0800 44 44 44 + 4444	6 (3x2)	Dx2
5	yes+	Telefon +44-44-1234567	9.25 (3x3+0.25)	(Dx3)+A
			27.5	**Total**

Table 3.27: Telephone numbers: Across

	Recognition	Result	Score	Calculation
1	yes	Gratisnummer 0800 888 888	3	D
2	yes	Telefon 044/888 88 88	6 (3x2)	Dx2
3	yes	Telefon +41 88 888 88 88	3	D
4	yes	Telefon 0800 44 44 44 + 4444	6 (3x2)	Dx2
5	yes	Telefon +44-44-1234567	9 (3x3)	Dx3
			27	**Total**

Table 3.28: Telephone numbers: Déjà Vu

	Recognition	Result	Score	Calculation
1	yes	Gratisnummer 0800 888 888	3	D
2	yes	Telefon 044/888 88 88	6 (3x2)	Dx2
3	yes	Telefon +41 88 888 88 88	3	D
4	yes	Telefon 0800 44 44 44 + 4444	6 (3x2)	Dx2
5	yes	Telefon +44-44-1234567	9 (3x3)	Dx3
			27	**Total**

Table 3.29: Telephone numbers: memoQ

	Recognition	Result	Score	Calculation
1	p	Gratisnummer 0800 888 888	0	
2	p	Telefon 044/888 88 88	3	D
3	yes+	Telefon +41 88 888 88 88	3.25 (3+0.25)	D+A
4	yes	Telefon 0800 44 44 44 + 4444	3	D
5	yes+	Telefon +44-44-1234567	9.25 (3x3+0.25)	(Dx3)+A
			18.5	**Total**

Table 3.30: Telephone numbers: SDL Trados

	Recognition	Result	Score	Calculation
1	yes+	Gratisnummer 0800 888 888 (one sequence)	3.5 (3+0.5)	D+R
2	yes+	Telefon 044/888 88 88 (one sequence)	6.5 (3x2+0.5)	(Dx2)+R
3	yes+	Telefon +41 88 888 88 88 (one sequence)	3.75 (3+0.5+0.25)	D+R+A
4	yes	Telefon 0800 44 44 44 + 4444	6 (3x2)	Dx2
5	yes+	Telefon +44-44-1234567	9.25 (3x3+0.25)	(Dx3)+A
			29	**Total**

Table 3.31: Telephone numbers: Transit

	Recognition	Result	Score	Calculation
1	yes	Gratisnummer 0800 888 888	3	D
2	yes+	Telefon 044/888 88 88	6.25 (3x2+0.25)	(Dx2)+A
3	yes	Telefon +41 88 888 88 88	3	D
4	yes	Telefon 0800 44 44 44 + 4444	6 (3x2)	Dx2
5	yes+	Telefon +44-44-1234567 (one sequence)	9.5 (3x3+0.5)	(Dx3)+R
			27.75	**Total**

Table 3.32: Telephone numbers: Wordfast

3.2.3.5 Standards and versions

Standards and versions have different structures and include non-numeric characters, but represent one unit. Alphabetic characters are specifically dealt with in 3.2.3.6.

The following examples were used:

	Text	Baseline	Calculation
1	ISO 9001-2000	6 (3x2)	Dx2
2	ISO9001-2000/14001	9 (3x3)	Dx3
3	Release 2.1.0	4	D+(Sx2)
4	Norm 802.11	3.5	D+S
		22.5	**Total**

Table 3.33: Standards: test examples

As in 3.2.3.4, TM systems that fail to recognize standards and versions as single sequences are not penalized.

Results

Across: see table 3.34. Only single numeric sequences are recognized. Non-alphanumeric characters are excluded from recognition.

Déjà Vu: see table 3.35. Only single numeric sequences are recognized. Non-alphanumeric characters are excluded from recognition. Moreover, numbers are not always recognized (example 2).

memoQ: see table 3.36. Numbers containing more than one separator are split. Non-alphanumeric characters are excluded from recognition. For example 2, the error message

"Non-standard number format in the target side" is displayed even though the source segment has been copied without any modification into the target segment.

SDL Trados: see table 3.37. The hyphen-minus sign is always interpreted as a minus sign (example 1). The alphanumeric sequence in examples 2 and 3 is not recognized. The whole sequence in example 4 is recognized.

Transit: see table 3.38. Some sequences are recognized as a single unit (examples 3 and 4). However, the alphanumeric sequence in example 2 is not recognized. The hyphen-minus is always interpreted as a minus sign.

Wordfast: see table 3.39. Recognition is always complete and correct.

	Recognition	**Result**	**Score**	**Calculation**
1	yes	ISO 9001-2000	6 (3x2)	Dx2
2	yes	ISO9001-2000/14001	9 (3x3)	Dx3
3	p	Release 2.1.0	3	D
4	p	Norm 802.11	3	D
			21	**Total**

Table 3.34: Standards: Across

	Recognition	**Result**	**Score**	**Calculation**
1	yes	ISO 9001-2000	6 (3x2)	Dx2
2	p	ISO9001-2000/14001	6 (3x2)	Dx2
3	p	Release 2.1.0	3	D
4	p	Norm 802.11	3	D
			18	**Total**

Table 3.35: Standards: Déjà Vu

	Recognition	**Result**	**Score**	**Calculation**
1	yes	ISO 9001-2000	6 (3x2)	Dx2
2	yes	ISO9001-2000/14001	9 (3x3)	Dx3
3	p	Release 2.1.0	3.5 (3+0.5)	D+S
4	yes	Norm 802.11	3.5 (3+0.5)	D+S
			22	**Total**

Table 3.36: Standards: memoQ

	Recognition	**Result**	**Score**	**Calculation**
1	yes	ISO 9001-2000	6 (3x2)	Dx2
2	p	ISO9001-2000/14001	6 (3x2)	Dx2
3	no	Release 2.1.0	0	
4	yes	Norm 802.11	3.5 (3+0.5)	D+S
			15.5	**Total**

Table 3.37: Standards: SDL Trados

	Recognition	Result	Score	Calculation
1	yes	ISO 9001-2000	6 (3x2)	Dx2
2	p	ISO9001-2000/14001	6.25 (3x2+0.25)	(Dx2)+A
3	yes	Release 2.1.0	4 (3+0.5x2)	D+(Sx2)
4	yes	Norm 802.11	3.5 (3+0.5)	D+S
			19.75	**Total**

Table 3.38: Standards: Transit

	Recognition	Result	Score	Calculation
1	yes+	ISO 9001-2000	6.5 (3x2+0.5)	(Dx2)+R
2	yes+	ISO9001-2000/14001	9.75 (3x3+0.5+0.25)	(Dx3)+R+A
3	yes	Release 2.1.0	4 (3+0.5x2)	D+(Sx2)
4	yes	Norm 802.11	3.5 (3+0.5)	D+S
			23.75	**Total**

Table 3.39: Standards: Wordfast

3.2.3.6 Other numbers

Some numeric patterns were not classified semantically but according to their structure. Handling of the following elements was investigated:

- Co-occurrence of non-numeric and numeric characters
- Percent sign
- Mathematical operators
- Other numerals

In addition to the overall aims (see 3.2), this test subset investigates the following questions:

- Are non-numeric characters recognized along with digits?
- Are mathematical symbols recognized as part of the number?

Several non-numeric characters already occurred in the previous sections. They are tested further here.

Co-occurrence of non-numeric and numeric characters

The following examples were used:

	Text	Baseline	Calculation
1	13.Monatslohn	3	D
2	2008™	3	D
3	N4/20	6 (3x2)	Dx2
4	20xE1	6 (3x2)	Dx2
5	RJ45	3	D
6	PHP 4	3	D
7	3er	3	D
8	25polig	3	D
9	802.11g-Standard	3.5	D+S
10	4-Port	3	D
11	111B-11-11	9 (3x3)	Dx3
12	Fit-4-Fun	3	D
13	A4-Seiten	3	D
14	Tippen Sie *111#	3	D
		54.5	**Total**

Table 3.40: Alphanumeric strings: test examples

These examples show that some sequences are placeable elements (e.g. examples 3, 4 and 11), while others need to be (partially) translated. Accurate recognition separates the translatable part (e.g. “Seiten” in example 13), but this is not always possible (e.g. in example 8). See chapter 5 for more details.

Results

Across: see table 3.41. Recognition is limited to digits, which are always correctly recognized. Non-numeric characters are excluded.

Déjà Vu: see table 3.42. Recognition is limited to digits, non-numeric characters are excluded. However, alphanumeric sequences are problematic: sometimes their digits are skipped (examples 3, 4, 5 and 13). As a result, digit recognition is not complete.

memoQ: see table 3.43. Recognition is limited to digits, which are always correctly recognized. Non-numeric characters are excluded.

SDL Trados: see table 3.44. In example 9, recognition includes a letter (probably interpreted as the abbreviation “g” for grams). The number sign is recognized in example 14. Several alphanumeric patterns are not recognized at all (examples 3, 4, 5 and 13). As a result, digit recognition is not complete.

Transit: see table 3.45. Recognition is limited to digits only, non-numeric characters are excluded. Some alphanumeric sequences are problematic: their digits are skipped (examples 3, 4, 5 and 13). As a result, digit recognition is not complete.

Wordfast: see table 3.46. Recognition also includes letters and some non-alphanumeric characters such as a solidus. Digit recognition is complete, but there are several cases of over-recognition (examples from 8 to 13). It was tested whether over-recognition prevented the quality check: numbers were properly verified.

	Recognition	Result	Score	Calculation
1	yes	13.Monatslohn	3	D
2	yes	2008™	3	D
3	yes	N4/20	6 (3x2)	Dx2
4	yes	20xE1	6 (3x2)	Dx2
5	yes	RJ45	3	D
6	yes	PHP 4	3	D
7	yes	3er	3	D
8	yes	25polig	3	D
9	p	802.11g-Standard	3	D
10	yes	4-Port	3	D
11	yes	111B-11-11	9 (3x3)	Dx3
12	yes	Fit-4-Fun	3	D
13	yes	A4-Seiten	3	D
14	yes	*111#	3	D
			54	**Total**

Table 3.41: Alphanumeric strings: Across

	Recognition	Result	Score	Calculation
1	yes	13.Monatslohn	3	D
2	yes	2008™	3	D
3	p	N4/20	3	D
4	p	20xE1	3	D
5	no	RJ45	0	
6	yes	PHP 4	3	D
7	yes	3er	3	D
8	yes	25polig	3	D
9	yes	802.11g-Standard	3.5 (3+0.5)	D+S
10	yes	4-Port	3	D
11	yes	111B-11-11	9 (3x3)	Dx3
12	yes	Fit-4-Fun	3	D
13	no	A4-Seiten	0	
14	yes	*111#	3	D
			42.5	**Total**

Table 3.42: Alphanumeric strings: Déjà Vu

	Recognition	Result	Score	Calculation
1	yes	13.Monatslohn	3	D
2	yes	2008™	3	D
3	yes	N4/20	6 (3x2)	Dx2
4	yes	20xE1	6 (3x2)	Dx2
5	yes	RJ45	3	D
6	yes	PHP 4	3	D
7	yes	3er	3	D
8	yes	25polig	3	D
9	yes	802.11g-Standard	3.5 (3+0.5)	D+S
10	yes	4-Port	3	D
11	yes	111B-11-11	9 (3x3)	Dx3
12	yes	Fit-4-Fun	3	D
13	yes	A4-Seiten	3	D
14	yes	*111#	3	D
			54.5	**Total**

Table 3.43: Alphanumeric strings: memoQ

	Recognition	Result	Score	Calculation
1	yes	13.Monatslohn	3	D
2	no	2008™	0	
3	p	N4/20	3	D
4	no	20xE1	0	
5	no	RJ45	0	
6	yes	PHP 4	3	D
7	no	3er	0	
8	no	25polig	0	
9	yes+	802.11g-Standard	3.75 (3+0.5+0.25)	D+S+A
10	yes	4-Port	3	D
11	yes	111B-11-11	9 (3x3)	Dx3
12	yes	Fit-4-Fun	3	D
13	no	A4-Seiten	0	
14	yes+	*111#	3.25 (3+0.25)	D+A
			31	**Total**

Table 3.44: Alphanumeric strings: SDL Trados

	Recognition	Result	Score	Calculation
1	yes	13.Monatslohn	3	D
2	yes	2008™	3	D
3	p	N4/20	3	D
4	p	20xE1	3	D
5	no	RJ45	0	
6	yes	PHP 4	3	D
7	yes	3er	3	D
8	yes	25polig	3	D
9	yes	802.11g-Standard	3.5 (3+0.5)	D+S
10	yes	4-Port	3	D
11	yes	111B-11-11	9 (3x3)	Dx3
12	yes	Fit-4-Fun	3	D
13	no	A4-Seiten	0	
14	yes	*111#	3	D
			42.5	**Total**

Table 3.45: Alphanumeric strings: Transit

	Recognition	Result	Score	Calculation
1	ex	13.Monatslohn	2.5 (3-0.5)	D-O
2	yes	2008™	3	D
3	yes+	N4/20	6.5 (3x2+0.25x2)	(Dx2)+(Ax2)
4	yes+	20xE1	6.25 (3x2+0.25)	(Dx2)+A
5	yes+	RJ45	3.25 (3+0.25)	D+A
6	yes	PHP 4	3	D
7	ex	3er	2.5 (3-0.5)	D-O
8	ex	25polig	2.5 (3-0.5)	D-O
9	ex	802.11g-Standard	3.25 (3+0.5+0.25-0.5)	D+S+A-O
10	ex	4-Port	2.5 (3-0.5)	D-O
11	yes+	111B-11-11	9.75 (3x3+0.5+0.25)	(Dx3)+R+A
12	ex	Fit-4-Fun	2 (3-0.5x2)	D-(Ox2)
13	ex	A4-Seiten	2.5 (3-0.5)	D-O
14	yes	*111#	3	D
			52.5	**Total**

Table 3.46: Alphanumeric strings: Wordfast

Percent sign

The following examples were used:

	Text	Baseline	Calculation
1	Beitrag von 1%	3	D
2	bis zu 10 %	3	D
3	+10%	3	D
		9	**Total**

Table 3.47: Percent sign: test examples

The percent sign can come immediately after the number or be separated by a space or no-break space, see also (Korpela, 2006, 384). The plus sign occurs in one example: other mathematical symbols will be discussed in 3.2.3.6.

Results

Across: see table 3.48. The percent sign is not included in the number recognition. The plus sign is included, however.

Déjà Vu: see table 3.49. The percent sign and the plus sign are not included in the number recognition.

memoQ: see table 3.50. The percent sign and the plus sign are not included in the number recognition.

SDL Trados: see table 3.51. The percent sign and the plus sign are always included in recognition.

Transit: see table 3.52. The percent sign is not included in the number recognition. The plus sign is included, however.

Wordfast: see table 3.53. The percent sign and the plus sign are not included in the number recognition.

	Recognition	Result	Score	Calculation
1	yes	Beitrag von 1%	3	D
2	yes	bis zu 10 %	3	D
3	yes+	+10%	3.25 (3+0.25)	D+A
			9.25	**Total**

Table 3.48: Percent sign: Across

	Recognition	Result	Score	Calculation
1	yes	Beitrag von 1%	3	D
2	yes	bis zu 10 %	3	D
3	yes	+10%	3	D
			9	**Total**

Table 3.49: Percent sign: Déjà Vu

	Recognition	Result	Score	Calculation
1	yes	Beitrag von 1%	3	D
2	yes	bis zu 10 %	3	D
3	yes	+10%	3	D
			9	**Total**

Table 3.50: Percent sign: memoQ

	Recognition	Result	Score	Calculation
1	yes+	Beitrag von 1%	3.25 (3+0.25)	D+A
2	yes+	bis zu 10 %	3.25 (3+0.25)	D+A
3	yes+	+10%	3.5 (3+0.25x2)	D+(Ax2)
			10	**Total**

Table 3.51: Percent sign: SDL Trados

	Recognition	Result	Score	Calculation
1	yes	Beitrag von 1%	3	D
2	yes	bis zu 10 %	3	D
3	yes+	+10%	3.25 (3+0.25)	D+A
			9.25	**Total**

Table 3.52: Percent sign: Transit

	Recognition	Result	Score	Calculation
1	yes	Beitrag von 1%	3	D
2	yes	bis zu 10 %	3	D
3	yes	+10%	3	D
			9	**Total**

Table 3.53: Percent sign: Wordfast

Mathematical operators

It is important to define mathematical operators properly. In fact, for simple mathematical operations, polysemic characters are used instead of the unequivocal ones defined by the Unicode standard, see table 3.54 and appendix A. For more information on polysemic characters, see Korpela (2006).

Operation	Polysemic character	Unequivocal character
Subtraction	hyphen-minus sign	minus sign
Multiplication	letter “x”	multiplication sign
Proportion	colon	ratio
Division	solidus	division slash

Table 3.54: Juxtaposition of polysemic and unequivocal mathematical operators

There are many more mathematical symbols than those covered in the examples. However, they have been either already described or they are highly specialized and less common. The hyphen-minus sign was covered in previous test sets and is therefore not included here.

The following examples were used:

	Text	Baseline	Calculation
1	1+1 Gerät	6 (3x2)	Dx2
2	240x320 Pixel	6 (3x2)	Dx2
3	1 / 6 x 100	9 (3x3)	Dx3
4	Ergebnis 0:6	6 (3x2)	Dx2
5	Promotion 2006+	3	D
		30	**Total**

Table 3.55: Mathematical operators: test examples

The polysemic characters were used. The only exception is the unequivocal plus sign, which has no polysemic counterpart.

Results

Across: see table 3.56. Apart from the colon (example 4), mathematical operators are excluded from recognition.

Déjà Vu: see table 3.57. Mathematical operators are always excluded from recognition. In example 2, recognition is incomplete.

memoQ: see table 3.58. Mathematical operators are always excluded from recognition, but all digits are recognized.

SDL Trados: see table 3.59. The presence of mathematical operators can prevent recognition (examples 1 and 5). Where recognition is successful, mathematical operators are excluded.

Transit: see table 3.60. Mathematical operators are always excluded from recognition. In example 2, recognition is incomplete.

Wordfast: see table 3.61. Mathematical operators are sometimes included in recognition, depending on their position. For example, the plus sign is included in example 1, but excluded in example 5. Nevertheless, digits are always correctly recognized.[6]

	Recognition	**Result**	**Score**	**Calculation**
1	yes	1+1 Gerät	6 (3x2)	Dx2
2	yes	240x320 Pixel	6 (3x2)	Dx2
3	yes	1 / 6 x 100	9 (3x3)	Dx3
4	yes+	Ergebnis 0:6	6.25 (3x2+0.25)	(Dx2)+A
5	yes	Promotion 2006+	3	D
			30.25	**Total**

Table 3.56: Mathematical operators: Across

	Recognition	**Result**	**Score**	**Calculation**
1	yes	1+1 Gerät	6 (3x2)	Dx2
2	p	240x320 Pixel	3	D
3	yes	1 / 6 x 100	9 (3x3)	Dx3
4	yes	Ergebnis 0:6	6 (3x2)	Dx2
5	yes	Promotion 2006+	3	D
			27	**Total**

Table 3.57: Mathematical operators: Déjà Vu

	Recognition	**Result**	**Score**	**Calculation**
1	yes	1+1 Gerät	6 (3x2)	Dx2
2	yes	240x320 Pixel	6 (3x2)	Dx2
3	yes	1 / 6 x 100	9 (3x3)	Dx3
4	yes	Ergebnis 0:6	6 (3x2)	Dx2
5	yes	Promotion 2006+	3	D
			30	**Total**

Table 3.58: Mathematical operators: memoQ

[6] A separate test with the segment “Unterschied -6”, where the minus sign is used (not the hyphen-minus sign), showed that Wordfast also recognizes this mathematical operator.

	Recognition	Result	Score	Calculation
1	no	1+1 Gerät	0	
2	no	240x320 Pixel	0	
3	yes	1 / 6 x 100	9 (3x3)	Dx3
4	yes	Ergebnis 0:6	6 (3x2)	Dx2
5	no	Promotion 2006+	0	
			15	**Total**

Table 3.59: Mathematical operators: SDL Trados

	Recognition	Result	Score	Calculation
1	yes	1+1 Gerät	6 (3x2)	Dx2
2	p	240x320 Pixel	3	D
3	yes	1 / 6 x 100	9 (3x3)	Dx3
4	yes	Ergebnis 0:6	6 (3x2)	Dx2
5	yes	Promotion 2006+	3	D
			27	**Total**

Table 3.60: Mathematical operators: Transit

	Recognition	Result	Score	Calculation
1	yes+	1+1 Gerät	6.25 (3x2+0.25)	D+A
2	yes+	240x320 Pixel	6.25 (3x2+0.25)	(Dx2)+A
3	yes	1 / 6 x 100	9 (3x3)	Dx3
4	yes+	Ergebnis 0:6	6.25 (3x2+0.25)	(Dx2)+A
5	yes	Promotion 2006+	3	D
			30.75	**Total**

Table 3.61: Mathematical operators: Wordfast

Other numerals

Arabic digits are the most frequent in Western European languages. However, Roman numerals can also be found. They are commonly written as a sequence of capital letters, instead of using the Unicode characters assigned to them (included in the range from U+2160 to U+2188), see (Korpela, 2006, 429).[7] The segment "Layer III", written as a sequence of three capital i's, was taken as an example. No TM system recognized it.

In other languages, further numerals can be found, e.g. Indian numerals in Hindi. However, due to the tester's lack of specific linguistic knowledge, no test was performed.

3.3 Conclusions

The conclusions are divided into two sections. Firstly, an overview of the different characters and elements recognized is given in section 3.3.1. Secondly, the quality of recognition is discussed in section 3.3.2, where the totaled scores of the TM systems are compared.

3.3.1 Recognized elements

Thousand and decimal separators pose some difficulties, in particular if they do not correspond to the typical language or locale-dependent conventions. Some TM systems handle only some of them correctly, e.g. Across, Déjà Vu and Transit do not recognize the apostrophe.

Wordfast is the only TM system that recognizes ranges separated by a hyphen-minus sign. Other TM systems sometimes split them and sometimes interpret the second element as a negative number. The same applies to digit sequences such as telephone numbers. With the exception of Wordfast, TM systems split these up, and the building blocks vary from TM system to TM system.

The only TM system that recognizes measuring units is SDL Trados. Its recognition is based on a list of units and is also possible when the unit is separated from the digit by a space. However, this list is partial. Wordfast recognizes measuring units only if there is no space between the number and the unit; the string is then treated as an alphanumeric sequence.

Alphanumeric strings can cause digit recognition to fail: Déjà Vu, SDL Trados[8] and Transit sometimes skip digits. Particularly problematic are sequences starting with a capital letter followed by digits. Across and memoQ limit their recognition to the digits contained in the alphanumeric strings, but work reliably. In the case of Wordfast, alphanumeric strings are recognized, but recognition sometimes embraces too many characters. Particularly prone to over-recognition are sequences where digits and letters are separated only by a hyphen-minus sign.

[7] Consequently, a regular expression covering only unequivocal Unicode characters would rarely be successful.

[8] For Trados, under-recognition is also described by Joy (2002).

The overview table 3.62 is helpful in order to summarize which non-numeric characters are included in recognition and to stress the differences between TM systems. The table does not account for occasional recognition failures.

Most separators, the hyphen-minus sign and the plus sign are usually recognized by the majority or at least by half of the TM systems. They are inherently the most common characters. The degree sign and percent sign are recognized only by SDL Trados, fractions only by Transit. Several characters are never recognized: the less-than sign, greater-than sign and plus sign (unless immediately followed by a digit).

Déjà Vu and memoQ strictly limit their recognition to digits and some separators. Across and Transit also include basic mathematical operators. SDL Trados and Wordfast recognize more non-numeric characters (in conjunction with digits) than other TM systems.

To summarize, recognition is incomplete in several cases, but usually correct (high precision). Incomplete recall does not imply that recognition is not useful during the text editing or quality assurance/quality control. As Macklovitch (1995) states (though referring to a quality assurance tool):

> Having a text checked by such a system offers no guarantee that it is fully accurate and correct; on the other hand, whatever errors the system does manage to automatically detect will still contribute to improving the quality of the final text.

Automatic adaptations of measurement units, dates and time formats are considered part of the user's wish list, see Lagoudaki (2009), but were not specifically tested here. However, if recognition is incomplete, adaptation will be prone to error. In addition, for successful automatic adaptations, recognition must increase in complexity since it is necessary to semantically discriminate the numbers (e.g. dates from other numeric expressions): this can only be done with more specialized, language-specific regular expressions, see e.g. Gintrowicz and Jassem (2007), possibly with the help of linguistic resources.[9]

[9] The use of appropriate local grammars can enhance recognition; for a brief introduction to this topic and some examples involving numbers, see Gross (1997).

	Across	**Déjà Vu**	**memoQ**	**SDL Trados**	**Transit**	**Wordfast**
Comma (as a separator)	no	yes	yes	yes	yes	yes
Full stop (as a separator)	no	yes	yes	yes	yes	yes
Apostrophe (as a separator)	no	no	yes	yes	no	yes
Colon (as a separator)	yes	no	no	yes	no	yes
Alphanumeric strings	no	no	no	no	no	yes
Hyphen-minus sign (as a minus sign)	yes	no	no	yes	yes	no
Ranges	no	no	no	no	no	yes
Fraction	no	no	no	no	yes	no
Degree sign	no	no	no	yes	no	no
Less-than sign	no	no	no	no	no	no
Greater-than sign	no	no	no	no	no	no
Plus sign (separated)	no	no	no	no	no	no
Plus sign (not separated)	yes	no	no	yes	yes	no
Percent sign (separated)	no	no	no	yes	no	no
Percent sign (not separated)	no	no	no	yes	no	no
Measuring units (separated)	no	no	no	yes	no	no
Measuring units (not separated)	no	no	no	yes	no	no
Recognized	**3**	**2**	**3**	**11**	**5**	**6**

Table 3.62: Overview: recognition of non-numeric characters

3.3.2 Ranking

This section aims to provide an overview of the results obtained throughout the chapter. Table 3.63 summarizes the scores obtained by the TM systems for each test subset. At the end of each row the total score is given. Wordfast outperforms the baseline because it earns several bonuses for its recognition of ranges and other characters. memoQ and Across are just under the baseline mainly because they sometimes fail to recognize thousand or decimal separators. Moreover, the results for Across are related to its source-language settings, see 3.2.3.1. The other TM systems repeatedly fail to recognize digits. This poses a serious problem if quality checks or automatic conversions are to be carried out on the basis of this recognition. The mean of the total scores is 341.92, which is clearly below the baseline value (373). The standard deviation corresponds to 33.71. Based on this figure, Wordfast over-performs while SDL Trados under-performs.

In order to be able to calculate the recall according to formula 2.2 and using the baseline, the bonuses had to be first deducted from these scores. Otherwise, the recall values would have been biased. Table 3.64 presents the scores less bonuses and table 3.65 shows the recall calculated accordingly.[10] Only memoQ and Wordfast achieve almost complete recall, other TM systems fall more or less significantly below the baseline. An additional consideration looking at table 3.64 is that most TM systems earn very few bonuses, if at all. Exceptions are Wordfast and SDL Trados; the latter, however, also shows the lowest recall.

[10] Note that recall values refer to the test suite, not to all number tokens in the test corpus.

	Prices	Times	Measures	Phone numbers	Standards	Mixed[§]	Percent	Symbols	**Total**
Baseline	57	40.5	132.5	27	22.5	54.5	9	30	**373**
Across	51	37	128.75	27.5	21	54	9.25	30.25	**358.75**
Déjà Vu	56	33.5	119	27	18	42.5	9	27	**332**
memoQ	57	36.5	132.5	27	22	54.5	9	30	**368.5**
SDL Trados	57	28.75	100.25	18.5	15.5	31	10	15	**276**
Transit	56	33.5	120.25	29	19.75	42.5	9.25	27	**337.25**
Wordfast	57.25	39.5	138.5	27.75	23.75	52.5	9	30.75	**379**

Table 3.63: Overview: test subset scores

	Prices	Times	Measures	Phone numbers	Standards	Mixed[§]	Percent	Symbols	**Total**
Baseline	57	40.5	132.5	27	22.5	54.5	9	30	**373**
Across	51	37	128.5	27	21	54	9	30	**357.5**
Déjà Vu	56	33.5	119	27	18	42.5	9	27	**332**
memoQ	57	36.5	132.5	27	22	54.5	9	30	**368.5**
SDL Trados	57	28.5	94.5	18	15.5	30.5	9	15	**268**
Transit	56	33.5	119	27	19.5	42.5	9	27	**333.5**
Wordfast	56.5	37.5	132.5	27	22.5	50.5	9	30	**365.5**

Table 3.64: Overview: test subset scores excluding bonuses

§: "Mixed" stands for strings where digits and other characters occur together, see tests under 3.2.3.6.

Rank	TM system	Recall value
0	Baseline	1
1	memoQ	0.99
2	Wordfast	0.98
3	Across	0.96
4	Transit	0.89
5	Déjà Vu	0.89
6	SDL Trados	0.72

Table 3.65: Overview: recall

3.4 Possible improvements

So that different number formats that coexist in one language or vary from language to language are recognized, recognition should not be bound to specific patterns. The list of separators should be as complete as possible. Additionally, ranges should ideally be recognized, although this is not a prerequisite.

Several solutions for recognizing numbers are available, see (Goyvaerts and Levithan, 2009, 323-346), but they have specific purposes and limit recognition to rather specific patterns. This is not a suitable strategy in a TM system where – as demonstrated by the examples presented – many variants are possible.

3.4.1 Prices

```
m/
(
(?:
    (?:
        \p{Sm}\p{Zs}?
        |
        \p{Pd}[,.]?
    )?

    \p{N}+
        (?:
        \p{P}[\p{N}\p{Pd}]*?\p{N}+
        |
        \p{P}\p{Pd}+(?!\p{L})
        |
        \p{Zs}\p{N}+
        )*
)+

(?:
    \p{Zs}[\p{Lu}\p{M}]+\b
    |
23     \p{Zs}?\p{Sc}
24 )?
25 )
26 /gx
```

The regular expression is designed to recognize numbers and some number sequences. It is divided into three parts:

- An optional leading part (lines 4 to 8) for non-numeric characters that can precede a number.

- A main part (lines 10 to 17) for the compulsory numeric part, allowing for complex structures with separators.

- An optional closing part (lines 20 to 24) for symbols or letters to be expected only at the end of a number.

The leading part consists of two possibilities: either a mathematical symbol followed by an optional space (line 5) or a dash followed by an optional dot or comma (line 7).

The main part requires at least one digit \p{N} (line 10), which is the only compulsory element of the whole regular expression. After the digit(s), three optional possibilities are considered. The first one (line 12) is designed to recognize complex numbers containing separators (included in the \p{P} Unicode property). The second one (line 14) is limited to the special case where the zero after the decimal separator is substituted by one or two dashes. The negative lookahead limits the recognition of strings as e.g. “30%-solution” to “30%”. The third option (line 16) allows for a space as a separator, but requires at least one digit to follow in order to avoid over-recognition.

The closing part consists of two possibilities: either a currency symbol (line 23) or, as a heuristic mechanism for some measuring units, a completely uppercase word separated by a space (line 21, note the \b word boundary).[11] These units, particularly if standardized in the International System of Units, are likely to remain the same through different languages, with the main exception of texts from or for the U.S. market.

3.4.2 Time specifications

In the face of the test results, the following improvements would be helpful. Firstly, mathematical symbols including the greater-than sign should be recognized. Secondly, special numeric characters such as $\frac{1}{2}$ should be considered too. Thirdly, alphanumeric sequences are likely to be non-translatable elements, although exceptions are possible.

The first two issues have already been addressed in the regular expression under 3.4.1: mathematical symbols are identified through the Unicode property \p{Sm} and all numeric characters are included in the Unicode property \p{N}. For alphanumeric sequences, a separate regular expression is presented in 5.4.4.

[11] The recognition of measuring units can be further improved with a list of standard abbreviations, but this possibility is not exploited here.

3.4.3 Measures

Different issues require improvements to the regular expression. Firstly, grades indicating angles and temperatures should be recognized. Also the quotation mark for inches could be included in recognition, but its ambiguity due to its predominant use in quotations makes false positives too likely. Therefore, it has been ignored.

Secondly, to achieve the recognition of measuring units written together with numbers, the pattern for alphanumeric expressions, see 5.4.4, could be used. However, that pattern is less flexible when it comes to recognizing more complex numbers. On the contrary, with some additions to the optional closing part (lines 20 to 24), the regular expression presented in 3.4.1 can also recognize measuring units attached to a number.

```
20 (?:
21     [\p{L}\p{M}]+\b
22     |
23     \p{Zs}[\p{Lu}\p{M}]+\b
24     |
25     \p{Zs}?\p{Sc}
26     |
27     \p{Zs}?°[CF]?
28 )?
```

Recognition has been extended to letters that immediately follow the number (line 21) as well as to temperatures and angles (line 27). The \b word boundary is used in line 21 to avoid partial recognition of strings that are best handled by the regular expressions for alphanumeric strings. Minor overlapping is still present: the sequence in which the regular expressions are applied is important. In general, the regular expression for alphanumeric strings should be applied before the regular expression for numbers; however, more sophisticated strategies can be applied too. In any case, it is important to check which one yields the best (i.e. the longest) match or to merge the matches if they overlap.

The sequence of the four variants has been crafted to be more effective when testing a string: the variant with letters immediately following the number is checked first.

3.4.4 Telephone numbers

Basically, two improvements are relevant to telephone numbers. Firstly, if possible, they should be recognized as a single unit, which makes them faster and easier to transfer. On the other hand, telephone numbers do not have to be identified as such, but simply as numeric sequences in broader sense. For discriminative recognition, highly specialized regular expressions are necessary, see e.g. (Goyvaerts and Levithan, 2009, 219-226). Secondly, telephone numbers should also include non-numeric characters.

The regular expression needs no modification to achieve complete recognition of all presented telephone numbers, including their non-numeric characters. However, in one case, the number is split into two sequences: 0800 44 44 44 and + 4444. Solving this issue was not deemed crucial and in the end disregarded in order to keep the regular expression uncluttered.

3.4.5 Standards and versions

Similar to telephone numbers, two improvements are relevant. Firstly, if possible, standards and versions should be recognized as a single unit. Secondly, they should also include non-numeric characters, particularly alphabetic characters.

Most standards and versions are correctly recognized by the regular expression defined above. The only problem is represented by sequences starting with a letter (example ISO9001-2000/14001): ISO9001 is recognized by the regular expression for alphanumeric sequences, 9001-2000/14001 by the regular expression for numbers. However, such examples are seldom (in this case resulting from a typing error) so that they were not addressed specifically. Furthermore, adequate strategies (e.g. match merging as discussed in 3.4.3) can provide an effective solution.

3.4.6 Other numbers

The first improvement is again the recognition of alphanumeric patterns, possibly as single sequences, but over-recognition should be avoided. Secondly, mathematical signs are to be treated as integral part of numbers or number sequences. Finally, when digits other than Arabic digits are used, they should be handled as well.

Non-Arabic digits are included in the Unicode property \p{N}, including e.g. Roman numerals. However, they have to be written with their non-ambiguous Unicode character and this is very rarely the case. On the other hand, regular expressions are available for the recognition of Roman numerals written in alphabetic letters, see (Goyvaerts and Levithan, 2009, 344), but it is questionable whether their use pays off. For the tested data, the answer is no, therefore they are not discussed.

The regular expression defined for numbers and alphanumeric strings shows precise and complete recognition of the examples tested: the only examples of over-recognition – unavoidable if the TM system does not have appropriate linguistic knowledge – are “25polig” and “3er”. Otherwise, if the number is separated from the letters (as in 13.Monatslohn, 4-Port, Fit-4-Fun, A4-Seiten), recognition stops before the separator.

Non-alphanumeric characters such as ™, * and # have been excluded, as is the case with all TM systems, but their presence does not prevent recognition. Slight modification of the regular expression is necessary instead in order to include the percent sign (line 25).

```
20 (?:
21     [\p{L}\p{M}]+\b
22     |
23     \p{Zs}[\p{Lu}\p{M}]+\b
24     |
25     \p{Zs}?[\p{Sc}%]
26     |
27     \p{Zs}?°[CF]?
28 )?
```

3.4.7 Assembling the regular expression

The regular expressions for numbers, including the improvements, is presented below. Several decisions and trade-offs had to be made. In addition, the regular expression was constructed on the basis of the test data. Despite all efforts to keep it as general as possible, it is likely that it will need some adaptations when applied to different test data.

```
m/
(
(?:
    (?:
        \p{Sm}\p{Zs}?
        |
        \p{Pd}[,.]?
    )?

    \p{N}+
        (?:
        \p{P}[\p{N}\p{Pd}]*?\p{N}+
        |
        \p{P}\p{Pd}+(?!\p{L})
        |
        \p{Zs}\p{N}+
        )*
)+

(?:
    [\p{L}\p{M}]+\b
    |
    \p{Zs}[\p{Lu}\p{M}]+\b
    |
    \p{Zs}?[\p{Sc}%]
    |
    \p{Zs}?°[CF]?
)?
)
/gx
```

Chapter 4

Dates

4.1 Introduction

Because of their peculiarities, dates were not considered as a subset of numbers in chapter 3. There are two types of dates:

- Numeric dates, e.g. 02-02-2010.
- Alphanumeric dates, e.g. 2nd February 2010.

Alphanumeric dates can be recognized by TM systems only if linguistic knowledge is available. Ideally, this knowledge should be available for each supported language (and most TM systems support virtually any language).

4.2 Tests

The tests are aimed at checking the following issues:

- Are numeric dates recognized as one single unit?
- Are alphanumeric dates recognized as one single unit?

As in chapter 3, the automatic conversion of dates from the source language format (e.g. 26.01.2011 in German) into the target language format (e.g. 01-26-2011 in US English) offered by several TM systems is not tested.[1] Recognition assessment does not require that target text be entered, see 2.3.2.1.

In the examples given in this chapter, dates always occur as plain text. If dates have been inserted in a document as fields (e.g. in MS Word 2003 with INSERT > FIELD > DATE), they are processed in a different way. See chapter 10 for further information.

[1] (Trujillo, 1999, 68) states that “dates can be automatically reformatted and the names of the month and day reliably translated”. Whilst it is often true, it implies some basic linguistic knowledge for all supported languages.

4.2.1 TM system settings

4.2.1.1 Versions

TM system	Version
Across Standalone Personal Edition	4.00
Déjà Vu X Professional	7.5.303
Heartsome Translation Studio Ultimate	7.0.6 2008-09-12S
memoQ Corporate	3.2.17
MultiTrans	4.3.0.84
SDL Trados 2007 Freelance	8.2.0.835
STAR Transit XV Professional	3.1 SP22 631
Wordfast	5.53

Table 4.1: TM systems used in the tests

4.2.1.2 Customizability

The customizability options mainly determine whether date-specific patterns can be modified and whether automatic conversion is performed.

Across: under Tools > System settings > General > Language settings >Locale settings > Language sets > Standard language set > Languages > [specific language] > Format > Date long and Date short, language-dependent date formats are specified. They are editable and new ones can be added. In addition, under Standard language set > Format, language-independent date formats are specified. They are editable as well, and can be supplemented. However, only numeric dates are taken into account.

Across offers automatic conversion for dates if the option Use autochanges under Tools > System Settings > General > crossTank is activated (default setting) and a Rich TM is used, see 2.4.3.2 for more information.

Déjà Vu, Transit, Wordfast: no specific setting for the recognition of dates is available. Numeric dates are treated as numbers, see 3.2.1.2.

Heartsome: the tested version does not provide any date recognition. Neither is a quality assurance component available. Consequently, Heartsome will not be included in these tests. These functions are planned for future versions.

memoQ: no embedded function for recognizing or converting dates is available. For this purpose, the function Auto-translatables described in 3.2.1.2 could have been used, but entails writing regular expressions.[2] As defined in 2.4.3.2, any customizing that goes beyond activating/deactivating functions is not considered.

[2] Auto-translatables rules specify “how a portion of the source text is converted to its equivalent in target text”, Kilgray (2008).

MultiTrans: as long as dates are numeric only, the same remarks as in 3.2.1.2 apply. MultiTrans does not support number recognition and a test by means of the QA Add-on was not possible due to license restrictions, see 3.2.2. No specific options are available for other types of dates.

SDL Trados: under File > Setup > Substitutions > Dates and Times, it is possible to "define [...] dates [...] as variables rather than normal words in Translator's Workbench. [...] Translator's Workbench recognises designated variables and treats them as non-translatable or placeable elements during translation", SDL (2007). This option is activated by default. However, it is not possible to see or modify the patterns used.

In addition, under Options > Translation Memory Options > Substitution Localisation > Dates and Times, it is possible to define "if, and how, Translator's Workbench should automatically adapt the format of variable elements to suit the target language [...]", SDL (2007).

4.2.2 General remarks on TM system behavior

In terms of display and recognition assessment, dates are handled as numbers, see 3.2.2. A scoring system was used for dates too, and is described in the following section.

Scoring system

A score was calculated for each instance of recognition according to the following rules:

- 3 points were assigned when the date as a whole (including separators) was recognized. Mere digit recognition was not given any point if it excluded separators (e.g. the full stops in "21.1.2007") or months written out in full (e.g. "Juli 2008").
- Bonus of 0.5 points was assigned when time intervals consisting of two dates (e.g. "1.4.07 - 31.3.08") were recognized.

A baseline assuming complete recognition of all dates (excluding the bonuses for interval recognition) was calculated. The scores achieved by the TM systems and the baseline are compared in 4.3.

In order to make the score more transparent, each table indicates how the score was calculated and which elements were considered. The elements are referred to with the following abbreviations:

- D: digit(s)
- I: interval

Recognition assessment

The remarks under 3.2.2 on page 71 also pertain to date recognition.

4.2.3 Test suite

The examples contain minor formal errors, e.g. missing spaces in 3 and 17, taken literally from the source corpus. They have been retained in order to ascertain whether recognition incorporates fault tolerance. The sequences that should be recognized are printed in gray. The following examples were used:

	Text	**Baseline**	**Calculation**
1	Juni und Juli 2008	3	D
2	vom 1. bis 30. Juni 2007	3	D
3	1.April 2007	3	D
4	ab 1. Mai 2008	3	D
5	ab 06. Mai 2007	3	D
6	ab 13. Mai 07	3	D
7	ab 1. Dez. 2006	3	D
8	Freitag, 21.09.2007	3	D
9	ab 06.03.2008	3	D
10	Dauer 20.-22.9.2007	3	D
11	Dauer 29.6-1.7.07	6 (3+3)	D+D
12	26.05 bis 30.06.07	6 (3+3)	D+D
13	Zeit 9.-11. Juni 2006	3	D
14	vom 11.-14. Juli	3	D
15	bis 20./21. September 2007	3	D
16	1 März - 31. Juli 2007	6 (3+3)	D+D
17	Von Juli bis Sept.05	3	D
18	am 1.5.2008	3	D
19	am 1.7.08	3	D
20	am 21.1.2007	3	D
21	Stand 12.2004	3	D
22	Daten 1.4.07 - 31.3.08	6 (3+3)	D+D
23	am 1. Januar	3	D
24	für den 30. April, 2. und 3. Mai	6 (3+3)	D+D
25	vom 21.4. bis 25.4.	6 (3+3)	D+D
		93	**Total**

Table 4.2: Dates: test examples

Results

Only complete date recognition is highlighted: simple digits are not highlighted because this recognition has already been dealt with in chapter 3.

Across: see table 4.3. Only examples 8 and 9 as well as 30.06.07 in example 12 are recognized as a single sequence.

Across can recognize numeric dates only if their format conforms to those formats defined under Tools > System settings > General > Language settings >Locale settings > Language sets > Standard language set > Languages > [Deutsch] > Format > Date long or Date short. The default formats for German are DD.MM.YY and DD.MM.YYYY. If dates have another format, they are not recognized, unless these formats are added manually, see 4.2.1.2. Alphanumeric dates and intervals are not recognized.

Déjà Vu: see table 4.4. The standard segmentation is problematic because many dates are segmented if the full stop is followed by a space and an uppercase letter or a digit. Adding two exceptions (.^w^# and .^w^A) under Tools > Options > Delimiters > Exceptions prevents this error from occurring and was used only for this test set.

Déjà Vu does not support the recognition of intervals and alphanumeric dates. On the other hand, numeric dates are recognized, e.g. 21.09.2007 in example 20. This recognition can be checked as follows: if one digit is changed (e.g. to 22.09.2007) and Check Numerals is activated, the whole date is marked in red.

memoQ: see table 4.5. Numeric dates are recognized only if they have the pattern [0−9]+\.[0−9]+, which is not date-specific. Dates such as 21.09.2007 are recognized only in part. This recognition can be checked as follows: if the first point (between 22 and 09) is changed and the QA check is run, an error message is displayed. If the second point (between 09 and 2007) and the QA check is run, no error message is displayed. Alphanumeric dates and intervals are not recognized.

SDL Trados: see table 4.6. Alphanumeric dates are generally recognized. However, if the month is abbreviated, there is no recognition.

Patterns of numeric dates are recognized: DD.MM.YYYY, DD.MM.YY as well as DD.MM, which is, however, not specific for dates. Intervals with a start and an end date are not recognized.

Transit: see table 4.7. The recognition of intervals and alphanumeric dates is not supported. On the other hand, numeric dates are recognized. This recognition can be checked as follows: if one of its digits is modified (e.g. 22.09.2007 instead of 21.09.2007) and a number check is carried out, the error message “Number "[...]" not found” contains the whole date.

Wordfast: see table 4.8. The recognition of alphanumeric dates is not supported. On the other hand, numeric dates are recognized.

If intervals of time are defined by two dates separated by a hyphen-minus sign (e.g. examples 10 and 11), they are recognized as a single unit, but only if there is no space. Otherwise, two separated dates are recognized (e.g. example 22).

	Recognition	Result	Score	Calculation
1	no	Juni und Juli 2008	0	
2	no	vom 1. bis 30. Juni 2007	0	
3	no	1.April 2007	0	
4	no	ab 1. Mai 2008	0	
5	no	ab 06. Mai 2007	0	
6	no	ab 13. Mai 07	0	
7	no	ab 1. Dez. 2006	0	
8	yes	Freitag, 21.09.2007	3	D
9	yes	ab 06.03.2008	3	D
10	no	Dauer 20.-22.9.2007	0	
11	no	Dauer 29.6-1.7.07	0	
12	p	26.05 bis 30.06.07	3	D
13	no	Zeit 9.-11. Juni 2006	0	
14	no	vom 11.-14. Juli	0	
15	no	bis 20./21. September 2007	0	
16	no	1 März - 31. Juli 2007	0	
17	no	Von Juli bis Sept.05	0	
18	no	am 1.5.2008	0	
19	no	am 1.7.08	0	
20	no	am 21.1.2007	0	
21	no	Stand 12.2004	0	
22	no	Daten 1.4.07 - 31.3.08	0	
23	no	am 1. Januar	0	
24	no	für den 30. April, 2. und 3. Mai	0	
25	no	vom 21.4. bis 25.4.	0	
			9	**Total**

Table 4.3: Dates: Across

	Recognition	**Result**	**Score**	**Calculation**
1	no	Juni und Juli 2008	0	
2	no	vom 1. bis 30. Juni 2007	0	
3	no	1.April 2007	0	
4	no	ab 1. Mai 2008	0	
5	no	ab 06. Mai 2007	0	
6	no	ab 13. Mai 07	0	
7	no	ab 1. Dez. 2006	0	
8	yes	Freitag, 21.09.2007	3	D
9	yes	ab 06.03.2008	3	D
10	yes	Dauer 20.-22.9.2007	3	D
11	yes	Dauer 29.6-1.7.07	6 (3+3)	D+D
12	yes	26.05 bis 30.06.07	6 (3+3)	D+D
13	no	Zeit 9.-11. Juni 2006	0	
14	no	vom 11.-14. Juli	0	
15	no	bis 20./21. September 2007	0	
16	no	1 März - 31. Juli 2007	0	
17	no	Von Juli bis Sept.05	0	
18	yes	am 1.5.2008	3	D
19	yes	am 1.7.08	3	D
20	yes	am 21.1.2007	3	D
21	yes	Stand 12.2004	3	D
22	yes	Daten 1.4.07 - 31.3.08	6 (3+3)	D+D
23	no	am 1. Januar	0	
24	no	für den 30. April, 2. und 3. Mai	0	
25	yes	vom 21.4. bis 25.4.	6 (3+3)	D+D
			45	**Total**

Table 4.4: Dates: Déjà Vu

	Recognition	**Result**	**Score**	**Calculation**
1	no	Juni und Juli 2008	0	
2	no	vom 1. bis 30. Juni 2007	0	
3	no	1.April 2007	0	
4	no	ab 1. Mai 2008	0	
5	no	ab 06. Mai 2007	0	
6	no	ab 13. Mai 07	0	
7	no	ab 1. Dez. 2006	0	
8	no	Freitag, 21.09.2007	0	
9	no	ab 06.03.2008	0	
10	no	Dauer 20.-22.9.2007	0	
11	p	Dauer 29.6-1.7.07	3	D
12	p	26.05 bis 30.06.07	3	D
13	no	Zeit 9.-11. Juni 2006	0	
14	no	vom 11.-14. Juli	0	
15	no	bis 20./21. September 2007	0	
16	no	1 März - 31. Juli 2007	0	
17	no	Von Juli bis Sept.05	0	
18	no	am 1.5.2008	0	
19	no	am 1.7.08	0	
20	no	am 21.1.2007	0	
21	yes	Stand 12.2004	3	D
22	no	Daten 1.4.07 - 31.3.08	0	
23	no	am 1. Januar	0	
24	no	für den 30. April, 2. und 3. Mai	0	
25	yes	vom 21.4. bis 25.4.	6 (3+3)	D+D
			15	**Total**

Table 4.5: Dates: memoQ

	Recognition	Result	Score	Calculation
1	no	Juni und Juli 2008	0	
2	yes	vom 1. bis 30. Juni 2007	3	D
3	no	1.April 2007	0	
4	yes	ab 1. Mai 2008	3	D
5	yes	ab 06. Mai 2007	3	D
6	yes	ab 13. Mai 07	3	D
7	no	ab 1. Dez. 2006	0	
8	yes	Freitag, 21.09.2007	3	D
9	yes	ab 06.03.2008	3	D
10	no	Dauer 20.-22.9.2007	0	
11	p	Dauer 29.6-1.7.07	3	D
12	yes	26.05 bis 30.06.07	6 (3+3)	D+D
13	no	Zeit 9.-11. Juni 2006	0	
14	no	vom 11.-14. Juli	0	
15	yes	bis 20./21. September 2007	3	D
16	p	1 März - 31. Juli 2007	3	D
17	no	Von Juli bis Sept.05	0	
18	yes	am 1.5.2008	3	D
19	yes	am 1.7.08	3	D
20	yes	am 21.1.2007	3	D
21	yes	Stand 12.2004	3	D
22	yes	Daten 1.4.07 - 31.3.08	6 (3+3)	D+D
23	no	am 1. Januar	0	
24	no	für den 30. April, 2. und 3. Mai	0	
25	yes	vom 21.4. bis 25.4.	6 (3+3)	D+D
			57	**Total**

Table 4.6: Dates: SDL Trados

	Recognition	**Result**	**Score**	**Calculation**
1	no	Juni und Juli 2008	0	
2	no	vom 1. bis 30. Juni 2007	0	
3	no	1.April 2007	0	
4	no	ab 1. Mai 2008	0	
5	no	ab 06. Mai 2007	0	
6	no	ab 13. Mai 07	0	
7	no	ab 1. Dez. 2006	0	
8	yes	Freitag, 21.09.2007	3	D
9	yes	ab 06.03.2008	3	D
10	yes	Dauer 20.-22.9.2007	3	D
11	yes	Dauer 29.6-1.7.07	6 (3+3)	D+D
12	yes	26.05 bis 30.06.07	6 (3+3)	D+D
13	no	Zeit 9.-11. Juni 2006	0	
14	no	vom 11.-14. Juli	0	
15	no	bis 20./21. September 2007	0	
16	no	1 März - 31. Juli 2007	0	
17	no	Von Juli bis Sept.05	0	
18	yes	am 1.5.2008	3	D
19	yes	am 1.7.08	3	D
20	yes	am 21.1.2007	3	D
21	yes	Stand 12.2004	3	D
22	yes	Daten 1.4.07 - 31.3.08	6 (3+3)	D+D
23	no	am 1. Januar	0	
24	no	für den 30. April, 2. und 3. Mai	0	
25	yes	vom 21.4. bis 25.4.	6 (3+3)	D+D
			45	**Total**

Table 4.7: Dates: Transit

	Recognition	**Result**	**Score**	**Calculation**
1	no	Juni und Juli 2008	0	
2	no	vom 1. bis 30. Juni 2007	0	
3	no	1.April 2007	0	
4	no	ab 1. Mai 2008	0	
5	no	ab 06. Mai 2007	0	
6	no	ab 13. Mai 07	0	
7	no	ab 1. Dez. 2006	0	
8	yes	Freitag, 21.09.2007	3	D
9	yes	ab 06.03.2008	3	D
10	yes	Dauer 20.-22.9.2007	3	D
11	yes	Dauer 29.6-1.7.07	6.5 (3+3+0.5)	D+D+I
12	yes	26.05 bis 30.06.07	6 (3+3)	D+D
13	no	Zeit 9.-11. Juni 2006	0	
14	no	vom 11.-14. Juli	0	
15	no	bis 20./21. September 2007	0	
16	no	1 März - 31. Juli 2007	0	
17	no	Von Juli bis Sept.05	0	
18	yes	am 1.5.2008	3	D
19	yes	am 1.7.08	3	D
20	yes	am 21.1.2007	3	D
21	yes	Stand 12.2004	3	D
22	yes	Daten 1.4.07 - 31.3.08	6 (3+3)	D+D
23	no	am 1. Januar	0	
24	no	für den 30. April, 2. und 3. Mai	0	
25	yes	vom 21.4. bis 25.4.	6 (3+3)	D+D
			45.5	**Total**

Table 4.8: Dates: Wordfast

4.3 Conclusions

As shown in table 4.9, only SDL Trados supports the recognition of alphanumeric dates. This requires linguistic knowledge. However, its recognition is not always successful, e.g. abbreviated German months are not identified. This test is limited to German: other source languages would need to be tested separately.

On the other hand, nearly all TM systems (except for memoQ) recognize formats that are specific to numeric dates. However, this recognition can fail, e.g. in Across the date format has to conform to a specific pattern, which limits the number of possible recognitions. Finally, only Wordfast recognizes some intervals defined by a start and an end date.

	Alphanumeric dates?	**Date-specific formats?**	**Intervals?**
Across	no	yes	no
Déjà Vu	no	yes	no
memoQ	no	no	no
SDL Trados	yes	yes	no
Transit	no	yes	no
Wordfast	no	yes	p

Table 4.9: Overview: recognition

The modest date recognition scores achieved by the TM systems tested are listed in table 4.10. The recall values, see 2.3.3 and 3.3.2 for more information on the calculation, show that reliable recognition of numeric dates enables a recall of about 0.5 to be achieved.[3] A significant improvement is possible if linguistic knowledge is used to recognize alphanumeric dates.

Rank	**TM system**	**Total score**	**Recall value**
0	Baseline	93	1
1	SDL Trados	57	0.61
2	Wordfast	45,5	0.48§
3	Déjà Vu	45	0.48
3	Transit	45	0.48
5	memoQ	15	0.16
6	Across	9	0.10

Table 4.10: Overview: recall

§: The bonus of 0.5 points had to be deducted before calculating the recall value, see also 3.3.2.

To summarize, the recognition of dates is often not optimal. Alphanumeric dates can be

[3] This holds true only for this specific test set and cannot be generalized.

recognized only with the help of linguistic knowledge, which is seldom available. Numeric dates have many possible formats that require flexible recognition.

4.4 Possible improvements

Numeric date formats and general number formats cannot always be distinguished. The regular expression developed for numbers, see 3.4.7, can be applied. All numeric dates are correctly and completely recognized, also when an interval is specified (examples 10, 11, 13, 14 and 22). The only incomplete recognition is found in example 15: because of the dot after 20, two sequences (20 and 21) are recognized. In order to avoid over-recognition in other contexts, the regular expression is left unchanged. The regular expression also recognizes dates with formats such as 01/01/2009 or 01-01-2009, which are not among the examples.

(Goyvaerts and Levithan, 2009, 226-234) present different solutions for recognizing dates, ranging from lax to strict ones. If it is necessary to distinguish a date from other numbers and to validate it, the regular expression will incorporate many more constraints: simplicity vs. accuracy is thus a matter of the intended purpose and source data, see (Goyvaerts and Levithan, 2009, 228). For example, if the dates conform to a specific convention, e.g. leading zeros are always used (01-01-2009 instead of 1-1-2009), the regular expression is simpler because it does not need to account for variations. If the source data is clean and accurate, validation is superfluous because only valid dates are to be expected. The validation itself can take different levels of complexity: while e.g. 99/04/2009 can never be valid because no month has 99 days (supposing that the day precedes the month), 31/04/2009 is invalid too, but only because the specific month (April) does not have 31 days. Leap days in February are a further element of complexity.

The use of linguistic knowledge within the TM systems would improve alphanumeric date recognition. However, this would require considerable effort as all languages along with their locales would need to be considered. Additionally, different calendar systems (e.g. the Hebrew calendar and the Islamic calendar) should be borne in mind.

For English, a grammar that would be able to recognize date expressions is presented by (Karttunen et al., 1996, 311-312). If invalid dates (e.g. April 31) are to be distinguished from valid ones, the grammar gains in complexity. This requirement would not be crucial in a TM system scenario. However, implementation in a parser would be necessary in either case.[4]

As already pointed out in 2.4.4.1, there is no single solution for all applications. In a TM system, if automatic conversion is not a requirement, there is no need to adopt strict variants that are crafted specifically for dates.

[4] For more information on the proposed solution, see Karttunen et al. (1996).

Chapter 5

Proper nouns and identifiers

5.1 Introduction

Proper nouns[1] can refer to the following, see (Huddleston and Pullum, 2002, 515-518):

- persons and animals
- places
- institutions, companies and organizations
- products
- titles of written works, movies, plays, paintings, magazines, etc.
- others (e.g. days of the week)

In English and other languages, proper nouns are recognizable because their first letter is capitalized.[2] However, e.g. German capitalizes all nouns, and in other languages capitalization does not exist, e.g. Chinese and Arabic.

Some proper nouns do not need to be translated (names of persons, companies and products), others are often translated (titles of works, movies, paintings, names of institutions and places). This is admittedly an oversimplification that allows for many exceptions. A brief discussion of the behavior of proper nouns in translation can be found in Russell (2005).

Complete recognition of proper nouns (or named entities, as they are also known, see (Pouliquen and Steinberger, 2009, 65-66)) is complex and is beyond the scope of this chapter and of a TM system. For translation purposes, it would be helpful if some proper nouns *that*

[1] This chapter does not consider the distinction between "proper names" and "proper nouns" made by e.g. Huddleston and Pullum (2002).

[2] Still, ambiguities are not seldom and robust recognition is a complicated matter. For a largely language-independent approach, see e.g. Mikheev (1999). For a language-dependent alternative, see e.g. Maynard et al. (2001).

need not to be translated could be recognized and transferred into the target language as placeable elements. This recognition should be based on string patterns (although proper nouns do not conform to relatively stringent patterns as URLs and e-mail addresses), and do without linguistic knowledge.

In many languages, there are some patterns indicating *possible* proper nouns:

- Always capitalized first letter
- Capitalization of the whole word
- Mixed uppercase and lowercase (except for the first letter)
- Mixed letters and digits
- Presence of some symbols[3] (#,=,+,%,&,_)

The capitalized first letter is not taken into account because this would mean considering the word position within the segment and recognizing many proper nouns that do require translation.

The remaining patterns hardly apply to some categories of proper nouns: persons, animals, places and titles in general, although exceptions are possible. On the other hand, companies and products can show the patterns indicated. Completely capitalized acronyms and abbreviations are often proper nouns.[4] Additionally, the concept of "identifiers" is introduced in order to include, e.g. product codes, variables[5], classes, keyboard shortcuts, etc.

5.2 Tests

The tests are aimed at checking the following issues:

1. Which patterns are recognized?
2. Is recognition complete?

Recognition assessment does not require that target text be entered, see 2.3.2.1. The examples were anonymized. This did not affect the results since the patterns were not modified.

In the test discussion, Across, Déjà Vu, Heartsome, memoQ, MultiTrans and Transit do not appear because they do not offer any recognition, neither by default, nor as an activatable option.

[3] For a definition of "symbol", see 2.4.

[4] Not covered here are capitalized abbreviations in which every capital letter is followed by a full stop, e.g. "U.S.A.", see (Müller et al., 1980, 51) for more examples. (Zerfaß, 2002a, 14) describes the recognition and automatic substitution of acronyms offered by some TM systems.

[5] Variables play an extremely important role in some translation jobs, see Wittner (2011).

5.2.1 TM system settings

5.2.1.1 Versions

TM system	Version
Across Standalone Personal Edition	4.00
Déjà Vu X Professional	7.5.303
Heartsome Translation Studio Ultimate	7.0.6 2008-09-12S
memoQ Corporate	3.2.17
MultiTrans	4.3.0.84
SDL Trados 2007 Freelance	8.2.0.835
STAR Transit XV Professional	3.1 SP22 631
Wordfast	5.53

Table 5.1: TM systems used in the tests

5.2.1.2 Customizability

Although Déjà Vu and memoQ do not recognize any of the patterns, they offer some helpful features.

Déjà Vu: "any word of two or less characters is an acronym [...] and [Déjà Vu] will thus take it over from the source text", (ATRIL, 2003, 196). This default behavior cannot be customized. However, it could not be reproduced in the tests below (which do not fulfill the requirements) or in an additional test set (designed to match those requirements). Atril confirmed that the feature was not working properly in the tested version and that it was going to be fixed.

memoQ: although there is no built-in option for recognizing proper nouns, they can be specified under TOOLS > OPTIONS > NON-TRANSLATABLES. However, according to the constraints specified in 2.4.3.2, this was not done to preserve the comparability of the results.

SDL Trados: under FILE > SETUP > SUBSTITUTIONS, the option ACRONYMS was activated in order to "define [...] acronyms as variables rather than normal words in Translator's Workbench. [...] Translator's Workbench recognises designated variables and treats them as non-translatable or placeable elements during translation", SDL (2007). However, it is not possible to see or modify the patterns used for recognition.

In addition, proper nouns can be manually specified under FILE > SETUP > SUBSTITUTIONS > VARIABLES. However, similar to memoQ and according to the constraints specified in 2.4.3.2, this was not done to preserve the comparability of the results.

Wordfast: to exploit all functionalities, three options of the Pandora's Box[6] were activated under WORDFAST > SETUP > SETUP > PB:

[6] The Pandora's Box is a container of options, enhancements and additional functions for Wordfast. All can be activated/deactivated and some are customizable.

- PlaceableContains #=+/\-
- Placeable=AllCap
- Placeable=MixedCase

The first option applies to words containing the non-alphanumeric characters indicated (the list is customizable). The second option applies to completely capitalized words. The third option applies to words with mixed-case, except for a capitalized first letter. The matches are then recognized as placeable elements.

The option Placeable=FirstCap in the Pandora's Box was not activated during the tests: it would instruct "Wordfast to consider words with a capitalised first letter [...] as placeables", (Champollion, 2008b, 63). This option is disadvantageous in German, as described in 5.1.

5.2.2 General remarks on TM system behavior

Scoring system

A score was calculated for each instance of recognition according to the following rules:

- 3 points were assigned when the proper noun or identifier as a whole was recognized.
- 2 points were assigned when the proper noun or identifier was recognized only in part.
- 1 point was deducted if over-recognition occurred.

A baseline assuming complete recognition of all proper nouns and identifiers was calculated. The scores achieved by the TM systems and the baseline are compared in 5.3. Unlike chapter 3 and 4, some of the examples shown are not supposed to be recognized. These examples are not assigned any point.

In order to make the score more transparent, each table indicates how it was calculated and which elements were considered. The elements are referred to with the following abbreviations:

- N: proper noun/identifier
- Np: proper noun/identifier (partial)
- O: penalty for over-recognition

5.2.3 Test suite

5.2.3.1 Symbols

The following examples were used:

	Text	**Baseline**	**Calculation**
1	T+T Kabel	3	N
2	ALT+TAB drücken	3	N
3	&Support		
4	Bereich Marketing&Sales		
5	Audio-Codec AAC+/MPEG-4	6 (3+3)	N+N
6	Tastenkombination Ctrl+D	3	N
7	Hallo #NICK#	3	N
8	BA_A4_Manual_GMX220_O8	3	N
9	%s aus Kontakte löschen?	3	N
10	Deinstallation von %APP% %VERSION%	6 (3+3)	N+N
11	ad+pgfcont=1	3	N
		33	**Total**

Table 5.2: Symbols: test examples

Several symbols were tested: plus sign, hyphen-minus sign, ampersand, number sign, low line and percent sign. The ampersand should not be recognized because it does not identify proper nouns/identifiers in the examples. This, however, does not necessarily apply to test suites derived from other corpora.

Results

SDL Trados: see table 5.3. Recognition is limited to capitalized strings that consist of at least three letters and do not contain non-alphanumeric characters. Some characters (e.g. the number sign) are excluded, others (e.g. the percent sign) seem to hamper recognition (example 10).

Wordfast: see table 5.4. Some non-alphanumeric characters are included in the recognized string (examples 1 and 2). However, if an alphanumeric character occurs at the beginning or at the end of the string, it is excluded from the placeable (examples 3, 7 and 10).

Some over-recognitions occur (examples 4 and 5). In example 4, this is due to the mixed-case of the word, see 5.2.3.3. In example 5, this is due to the presence of a hyphen-minus sign, which is often used in German (and English) to join two words.[7]

[7] Note that the hyphen-minus sign can be removed from the list of symbols in PLACEABLECONTAINS.

	Recognition	Result	Score	Calculation
1	no	T+T Kabel		
2	no	ALT+TAB drücken		
3	no	&Support		
4	no	Bereich Marketing&Sales		
5	p	Audio-Codec AAC+/MPEG-4	4 (2+2)	Np+Np
6	no	Tastenkombination Ctrl+D		
7	p	Hallo #NICK#	2	Np
8	no	BA_A4_Manual_GMX220_O8		
9	no	%s aus Kontakte löschen?		
10	no	Deinstallation von %APP% %VERSION%		
11	no	ad+pgfcont=1		
			6	**Total**

Table 5.3: Symbols: SDL Trados

	Recognition	Result	Score	Calculation
1	yes	T+T Kabel	3	N
2	yes	ALT+TAB drücken	3	N
3	no	&Support		
4	ex	Bereich Marketing&Sales	-1	O
5	ex	Audio-Codec AAC+/MPEG-4	5 (3+3-1)	N+N-O
6	yes	Tastenkombination Ctrl+D	3	N
7	p	Hallo #NICK#	2	Np
8	yes	BA_A4_Manual_GMX220_O8	3	N
9	no	%s aus Kontakte löschen?		
10	p	Deinstallation von %APP% %VERSION%	4 (2+2)	Np+Np
11	yes	ad+pgfcont=1	3	N
			25	**Total**

Table 5.4: Symbols: Wordfast

5.2.3.2 Completely capitalized strings

The following examples were used:

	Text	Baseline	Calculation
1	Wählen Sie FALSE	3	N
2	Empfangstechnik DVB-T	3	N
3	USB-Schnittstelle	3	N
4	IEEE 802.11g	3	N
5	RC.STRING.EXCEEDED_LIMIT	3	N
6	FAQ zu diesem Thema	3	N
7	WICHTIG:		
		18	**Total**

Table 5.5: Completely capitalized strings: test examples

These examples illustrate that capitalization can have different semantic meanings:

- Abbreviation/acronym (“DVB-T”, “USB”, “IEEE”, “FAQ”)
- Important information (“WICHTIG”)
- Special content, such as a GUI element (example 1) or source code (example 5)

Important information should not be recognized, but this is not easily achievable.

Results

SDL Trados: see table 5.6. Recognition in SDL Trados has some limitations: characters other than letters are never included. This prevents Trados from recognizing the full expression “DVB-T”. Example 5 is not recognized, not even in its individual parts (RC - STRING - EXCEEDED - LIMIT). Finally, it is unclear why “IEEE” is not recognized although it has the same structure as “FAQ”.

Wordfast: see table 5.7. Recognition in Wordfast has two problems: in one case, the inclusion of non-alphanumeric characters results in over-recognition (“USB-Schnittstelle”), as observed in 5.2.3.1. On the other hand, example 5 is not recognized.

	Recognition	Result	Score	Calculation
1	yes	Wählen Sie FALSE	3	N
2	p	Empfangstechnik DVB-T	2	Np
3	yes	USB-Schnittstelle	3	N
4	no	IEEE 802.11g		
5	no	RC.STRING.EXCEEDED_LIMIT		
6	yes	FAQ zu diesem Thema	3	N
7	no	WICHTIG:		
			11	**Total**

Table 5.6: Completely capitalized strings: SDL Trados

	Recognition	Result	Score	Calculation
1	yes	Wählen Sie FALSE	3	N
2	yes	Empfangstechnik DVB-T	3	N
3	ex	USB-Schnittstelle	2 (3-1)	N-O
4	yes	IEEE 802.11g	3	N
5	no	RC.STRING.EXCEEDED_LIMIT		
6	yes	FAQ zu diesem Thema	3	N
7	no	WICHTIG:		
			14	**Total**

Table 5.7: Completely capitalized strings: Wordfast

5.2.3.3 Mixed-case strings

The following examples were used:

	Text	Baseline	Calculation
1	xDSL-Technologie	3	N
2	ThinkPad, Acer TravelMate, MacBook, BlackBerry	12 (3*4)	N*4
3	JavaScript muss zugelassen sein.	3	N
4	MySQL Datenbank	3	N
5	HiFi-Anlage	3	N
6	InterviewpartnerIn		
		24	**Total**

Table 5.8: Mixed-case strings: test examples

While mixed case is a reliable indicator for proper nouns or identifiers, exceptions due to special spelling conventions (example 6) are possible.

Results

SDL Trados: mixed-case strings are never recognized.

Wordfast: see table 5.9. Recognition is mostly correct. However, as the hyphen-minus sign is considered part of the word, over-recognitions (examples 1 and 5) are unavoidable, see also 5.2.3.1. It is worth pointing out the over-recognition in example 6: it would be necessary to perform a linguistic analysis to exclude the mixed-case word from recognition. In fact, the German ending “In” with upper-case “i” means that the person can be a man or a woman (Interviewpartner or Interviewpartnerin).

	Recognition	Result	Score	Calculation
1	ex	xDSL-Technologie	2 (3-1)	N-O
2	yes	ThinkPad, Acer TravelMate, MacBook, BlackBerry	12 (3*4)	N*4
3	yes	JavaScript muss zugelassen sein.	3	N
4	yes	MySQL Datenbank	3	N
5	ex	HiFi-Anlage	2 (3-1)	N-O
6	ex	InterviewpartnerIn	-1	O
			21	**Total**

Table 5.9: Mixed-case strings: Wordfast

5.2.3.4 Alphanumeric strings

The following examples were used:

	Text	Baseline	Calculation
1	DDR2 SDRAM	3	N
2	42-48F1	3	N
3	Modell W280	3	N
4	A4-Seiten	3	N
5	03.A11.101	3	N
6	1x12ABCD	3	N
		18	**Total**

Table 5.10: Alphanumeric strings: test examples

Results

SDL Trados: alphanumeric strings are never recognized.

Wordfast: see table 5.11. As the hyphen-minus sign is considered part of the word, over-recognition in example 4 is unavoidable. Otherwise, recognition is correct and complete.

	Recognition	Result	Score	Calculation
1	yes	DDR2 SDRAM	3	N
2	yes	42-48F1	3	N
3	yes	Modell W280	3	N
4	ex	A4-Seiten	2 (3-1)	N-O
5	yes	03.A11.101	3	N
6	yes	1x12ABCD	3	N
			17	**Total**

Table 5.11: Alphanumeric strings: Wordfast

5.3 Conclusions

The recognition of invariable proper nouns and identifiers can speed up the translation process. If no match is proposed for the target segment and the source segment is not copied into the target, a simple transfer feature can save typing and prevent typos. Furthermore, recognition can be used for quality assurance/quality control because, for example, it would be possible to check whether these proper nouns and identifiers have been changed by mistake.[8] However, only SDL Trados and Wordfast try – by default – to heuristically recognize proper nouns and identifiers. Tables 5.12 and 5.13 summarize their scores and recall values.

	Baseline	**SDL Trados**	**Wordfast**
Symbols	33	6	25
Completely capitalized strings	18	11	14
Mixed-case strings	24	0	21
Alphanumeric strings	18	0	17
Total	**93**	**17**	**77**

Table 5.12: Overview: recognition

Rank	**TM system**	**Total score**	**Recall value**
0	Baseline	93	1
1	Wordfast	77	0.83
2	SDL Trados	17	0.18

Table 5.13: Overview: recall

In the case of SDL Trados, recognition is not customizable and is limited to completely capitalized alphabetic strings that are longer than two characters. Moreover, even if the prerequisite is fulfilled, recognition does not always work properly. Non-alphabetic characters are always ignored.

Wordfast is more powerful and can be customized. It can recognize words with non-alphabetic characters. However, not all non-alphanumeric default characters are suitable: for example, the ampersand and the hyphen-minus sign also occur in compound words that have to be translated and cause several cases of over-recognition.[9] Non-alphanumeric characters are included in the placeable only when they occur in the middle of the word. Strings with mixed-case and alphanumeric strings are recognized accurately.

To summarize, it is unavoidable that formal recognition of proper nouns and identifiers sometimes results in incorrect or unsuitable matches. For example, the semantic ambiguity of the capitalization of whole words can cause errors. It is not possible to filter out

[8] For a different and more systematic approach to quality assurance for proper nouns and user interface elements, see (Massion, 2010, 43).

[9] Wordfast provides the possibility to customize the list of symbols so that over-recognition can be avoided.

irrelevant occurrences without linguistic knowledge or sophisticated statistical methods. Nevertheless, formal recognition can be powerful, as Wordfast's recall value in table 5.13 demonstrates, and it can be improved if customizable.

It is difficult to determine whether capitalized acronyms and abbreviations should be generally included in recognition. Presumably it depends on the source text and language combination. If recognition is counter-productive, it should be deactivatable.

5.4 Possible improvements

The following regular expressions cover only a limited range of proper nouns and identifiers. However, they do not require any linguistic knowledge and apply to several European languages.

5.4.1 Symbols

```
1  m/
2  (
3      (?:
4          [\p{S}#%_]+
5          [\p{L}\p{N}\p{M}]+
6      |
7          [\p{L}\p{N}\p{M}]+
8          [\p{S}#%_]+
9      )
10     (?:
11         [\p{L}\p{M}\p{N}\p{S}\p{Pc}\p{Po}]*
12         [\p{L}\p{M}\p{N}\p{S}\p{Pc}#%]
13     )?
14 )
15 /gx
```

The main question to be settled before crafting this regular expression is which symbols should be considered. Some symbols (e.g. the plus sign) are included in the Unicode property \p{S}. However, e.g. the number sign and the percent sign belong to the Unicode property "Other punctuation" \p{Po}, which – as a whole – is not suitable for the purpose of recognition in this case. They therefore have to be included individually. The list of symbols considered is not exhaustive. However, it is not guaranteed that any addition would be suitable in all contexts. For example, the asterisk was first included but later rejected because of the many cases of standard strings either starting or ending with it. The ampersand, too, has been intentionally ignored to prevent over-recognition.

The regular expression is divided into a compulsory main part (lines 3 to 9) and an optional closing part (lines 10 to 13). The main part consists of two alternatives:

- It starts with at least one symbol, followed by at least one letter or digit.

- It starts with at least one letter or digit, followed by at least one symbol.

At least one digit or letter is always required, i.e. a string consisting only of symbols is not recognized.

The optional part is very flexible, but punctuation marks at the end are not allowed (line 12): this is a trade-off to avoid over-recognition.[10]

5.4.2 Completely capitalized strings

```
1  m/
2  \b
3  (
4      [\p{Lu}\p{Lt}]\p{M}?\p{Lu}\p{M}?[\p{Lu}\p{M}]*
5      (?:
6          [\p{Pd}\p{Po}\p{Pc}]
7          [\p{Lu}\p{M}]+
8      )*
9  )
10 \b
11 /gx
```

The regular expression is structured in two parts:

- A main part (line 4).
- An optional closing part (lines 5 to 8).

In the main part, at least two uppercase (or titlecase[11] followed by uppercase) letters are needed. This prevents recognition of e.g. the article “A” at the beginning of a sentence.

In its optional closing part, the regular expression allows for more complex structures. Some non-alphanumeric characters are allowed (line 6), but over-recognition – particularly after the hyphen-minus sign – is avoided in that those characters have to be followed by one or more uppercase letters (line 7), i.e. they cannot be the last character of the word. A sequence of non-alphanumeric characters is not allowed.

5.4.3 Mixed-case strings

```
1  m/
2  \b
```

[10] Unlike the other regular expressions in this chapter, word boundaries (\b) are not used because they would prevent the recognition of some symbols that are not considered as word characters, e.g. the number sign.

[11] “Titlecase refers to a character used at the start of a word written with a capital initial [...]. For most characters, titlecase is the same as uppercase. However, for some letters that are originally ligatures, only the first component is in uppercase version in the titlecase form”, (Korpela, 2006, 244).

```
3  (
4      (?:
5          [\p{Lu}\p{Lt}]\p{M}?
6          \p{Ll}\p{M}?[\p{Ll}\p{M}]*
7          \p{Lu}\p{M}?
8      |
9          [\p{Lu}\p{Lt}]\p{M}?\p{Lu}\p{M}?[\p{Lu}\p{M}]*
10         \p{Ll}\p{M}?
11     |
12         \p{Ll}\p{M}?[\p{Ll}\p{M}]*
13         \p{Lu}\p{M}?
14     )
15     [\p{L}\p{M}]*
16 )
17 \b
18 /gx
```

The regular expression has to prevent over-recognition due to the hyphen-minus sign. For this purpose, recognition is limited to expressions consisting of letters only. If digits are present, the regular expression presented in 5.4.4 applies.

The regular expression takes into account three possibilities.

- The first possibility (lines 5 to 7) starts with one uppercase letter or one titlecase letter (line 5). If followed by lowercase letters (line 6), it is not necessarily a mixed-case word because of the capitalization. Consequently, a further uppercase letter must follow (line 7).

- The second possibility (lines 9 and 10) covers a word beginning with at least 2 uppercase letters (or titlecase followed by uppercase). It then suffices to check whether a lowercase letter follows.[12]

- The third possibility (lines 12 and 13) recognizes at least one lowercase letter followed by at least one uppercase letter.

After these three options, any combination of letters can follow (line 15).

The previous version of the regular expression can be made more compact, although it becomes less readable.

```
1  m/
2  \b
3  (
4      (?:
5          (?:[\p{Lu}\p{Lt}]\p{M}?)?
6          \p{Ll}\p{M}?[\p{Ll}\p{M}]*
7          \p{Lu}\p{M}?
8      |
9          [\p{Lu}\p{Lt}]\p{M}?\p{Lu}\p{M}?[\p{Lu}\p{M}]*
10         \p{Ll}\p{M}?
11     )
```

[12] Words consisting only of uppercase letters are recognized by the regular expression under 5.4.2.

```
    [\p{L}\p{M}]*
)
\b
/gx
```

5.4.4 Alphanumeric strings

```
1  m/
2  \b
3  (
4      (?:
5          (?>
6          \p{N}+
7          (?:\p{P}\p{N}+)*
8          [\p{L}\p{M}]+
9          )
10     |
11         (?>
12         [\p{L}\p{M}]+
13         \p{N}+
14         )
15     )
16     [\p{L}\p{M}\p{N}]*
17 )
18 \b
19 /gx
```

The regular expression consists of two alternatives.

1. The first (lines 6 to 8) starts with at least one number, optionally followed by a punctuation mark and further number(s) and closed by at least one letter.

2. The second (lines 12 and 13) consists of at least one letter followed by at least one number.

Both variants can be optionally followed by letters and/or numbers in any sequence (line 16).

The regular expressions correctly recognize all examples but one: in “03.A11.101” only “A11” is recognized. The following modification in line 7 would allow complete recognition: (?:\p{P}\p{N}*)* instead of (?:\p{P}\p{N}+)*. In other words, the compulsory number after punctuation becomes optional. However, this modification would lead to over-recognition in other examples from chapter 3, e.g. “4-Port”, “2,-/Tag” and “13.Monatslohn”, and was not applied.

There is some overlapping between the regular expression that recognizes symbols and the regular expression that recognizes alphanumeric strings. The latter expression, for example, matches fragments of the strings that are completely matched by the former. To solve this conflict, one possibility would be to implement them in this sequence so that the larger string that has already been recognized will not be checked again unnecessarily.

Chapter 6

URLs

6.1 Introduction

A Uniform Resource Locator (URL) is a “a compact string representation for a resource available via the Internet”, IETF (1994). A URL consists at least of two compulsory parts:

- Scheme
- Hostname

The *scheme* is a set of instructions for data transfer, e.g.:[1]

- http (HyperText Transfer Protocol)
- https (HyperText Transfer Protocol Secure)
- ftp (File Transfer Protocol)

The *hostname* conforms to the domain schema, which defines two domain levels:

1. Top Level Domain (TLD): country-specific or generic.[2]
2. Subdomains: follow (from right to left) the TLD. Theoretically, a host can contain up to 127 subdomain levels.

The domain and subdomain levels are separated by a dot.

A URL has also optional constituents, see IETF (1994):

- Path: “data specific to the scheme [...]. It supplies the details of how the specific resource can be accessed”, IETF (1994).

[1] The examples are limited to the tested schemes.

[2] Country-specific TLDs consist of two letters and are reserved to countries and dependent territories, e.g. .de and .it. Generic TLDs consist of three or more letters and are reserved to particular classes of companies and organizations, e.g. .aero and .gov.

- User name and password (typical, e.g. of the ftp scheme, but not allowed for http).
- Port number, introduced by a colon. The number identifies the port that serves as a communication gateway for one or more specific processes.
- Query, introduced by a question mark. The query string contains data to be transmitted to a server program that processes it.
- Anchor or fragment, introduced by a number sign. The anchor defines a jump label within the HTML page and is, strictly speaking, "not considered part of the URL", IETF (1995).

In URLs, only some characters can be used freely and do not have a special reserved meaning. These unreserved ASCII characters "include uppercase and lowercase letters, decimal digits, hyphen, period, low line, and tilde", IETF (2005). The implementation and support of the Unicode repertoire in domain names (Internationalized Domain Names, IDN) is a work in progress, see (Korpela, 2006, 531) and (Yunker, 2011, 62), but has already been defined in the following requests for comments (RFCs):[3] IETF (2003a) and IETF (2003b), both with the status "Proposed Standard".

6.2 Tests

The tests are aimed at checking the following issues:

- Are URLs in hyperlinks recognized?
- Are URLs in plain text recognized?
- Are there specific URL patterns that are not recognized?

Recognition assessment does not require that target text be entered, see 2.3.2.1. In order to avoid privacy and copyright issues arising from the URLs, they were anonymized. Except for the TLD, file extension, scheme and few other constituents, all letters were substituted by an "x" (for lowercase) or an "X" (for uppercase). Digits were altered as well.

As was the case with numbers, see 3.2, URLs were embedded in larger sentences. These are not listed since they are not the focus of the test. However, the position of the URL in the sentence affects the way it is processed by some TM systems, as described in the results.

In MS Word 2003, Internet addresses that have been entered as plain text are replaced by default by a hyperlink: { HYPERLINK "(URL)"}. This option can be deactivated under Tools > AutoCorrect Options > AutoFormat As You Type > Internet And Network Paths With Hyperlinks. Both scenarios (hyperlink and plain text) are covered in the tests.

[3] RFCs are documents published by the Internet Engineering Task Force (IETF) on technical issues concerning the Internet. Some of them are standards.

6.2.1 TM system settings

6.2.1.1 Versions

TM system	**Version**
Across Standalone Personal Edition	4.00
Déjà Vu X Professional	7.5.303
Heartsome Translation Studio Ultimate	7.0.6 2008-09-12S
memoQ Corporate	3.2.17
MultiTrans	4.3.0.84
SDL Trados 2007 Freelance	8.2.0.835
STAR Transit XV Professional	3.1 SP22 631
Wordfast	5.53

Table 6.1: TM systems used in the tests

6.2.1.2 Customizability

None of the TM systems tested offers any specific setting for the recognition of URLs.

6.2.2 Test suite

6.2.2.1 URL as a hyperlink

This test comprises one URL that was converted into a hyperlink by MS Word: `http://xx-xxxxxxxx.xxxxxxxx.ch/Xxxx/xxxxxx/xxxxxxxxxxxxx/Xxxxxxxxxx_Xxxxxxxxxxxxx_X_2006.pdf`

If the hyperlink is recognized as a placeable element, it is displayed in the same way as shown under 10.2.2.

Results

See table 6.2 for a summary of the results.

	Recognition
Across	yes
Déjà Vu	no
Heartsome	no
memoQ	no
MultiTrans	yes
SDL Trados	yes
Transit	no
Wordfast	yes

Table 6.2: URL recognition in hyperlinks

Across, SDL Trados, Wordfast: the hyperlink is recognized as a placeable element.

Déjà Vu: the hyperlink is displayed as plain text. No tag precedes or follows the URL. After translating and exporting the document, the hyperlink is preserved.

Heartsome, memoQ, Transit: the hyperlink is displayed as plain text between tags.

MultiTrans: the URL is recognized as a placeable element. If the hyperlink is the sole element in the paragraph, the complete paragraph is skipped. If the hyperlink is in the middle of a segment, it is skipped but the text before and after it is highlighted for translation.

6.2.2.2 URL as plain text

This section aims to find out why URLs were successfully recognized as placeable elements: either URLs are recognized thanks to the conversion into hyperlink, or true pattern recognition is available.

In order to verify which hypothesis is correct for which TM system, an MS Word document was used with the same URL as in 6.2.2.1 formatted as plain text and not as a hyperlink. This test was limited to the TM systems that recognized the URL as a placeable element.

Results

See table 6.3 for a summary of the results.

	Recognition
Across	no
MultiTrans	no
SDL Trados	no
Wordfast	yes

Table 6.3: URL recognition in plain text

Across, MultiTrans, SDL Trados: the URL is displayed as plain text.

Wordfast: the URL is recognized as a placeable element.

6.2.2.3 Different URL patterns

A further issue arises from the test under 6.2.2.2: it is necessary to verify whether these results apply to all URL patterns. The tested URLs are listed in table 6.4.

Results

Across, MultiTrans, SDL Trados: URLs as plain text are never recognized as placeable elements.

Wordfast: 28 examples out of 30 are recognized correctly, which provides a recall value of 0.93, see 2.3.3.1. Only the ftp scheme prevents Wordfast from recognizing the URL (examples 26 and 27). If an anchor is available (example 30), the URL is recognized up to the number sign. As defined by IETF (1995), the fragment after the number sign is not part of the URL, therefore, this behavior is correct.

	URL
1	http://xx-xxxxxxxx.xxxxxxxx.ch/Xxxx/xxxxxx/xxxxxxxxxxxxx/Xxxxxxxxxx_Xxxxxxxxxxxxx_X_2006.pdf
2	http://xxx-xxxxxxxx.x-xxxxx.xxxxxxxx.ch/xxxx/xxxxxxx/xxxxxxxxxxxxxxxxxx
3	http://www.xxxxxxxxxxxx-xxxx.ch/de/5555/55555.html
4	https://xxx.xxx.xxxxxxx.ch/xx/xxxx/XxxxxxxxXxxx?lang=de%20xxxxxx=
5	http://xxxxxx-xx.xxxxxxxx.net/XxXxx/XX/xxxxxx/XXX1X11X-222X-3XXX-44X4-55X5555X5555.htm?XXXXXX=Xxxxxxxxxxx &xxx_xxxxxxx=Xxxxx&XXXXXXX=XxxxxxxxxxxxXxxxxxxxxxx
6	http://xxx.xxx.xxxxxxx.ch/de/xxxxx.php/xxxxxxxxxx/
7	http://xxxxxx-xx.xxxxxxxx.net
8	http://www.xxxxxxxx-xxxxxx.ch/xxxxx/xxxxx/Xxxxxxxxx/Xxxxxxxxxxxxxxxxxx/Xxxxx_XX.exe
9	http://www.xxxxxxxx-xxxxxx.ch/xxxxx/xxxxx/Xxxxxxxxx/Xxxxxxxxxxxxxxxxxx/xxx-xxxxx_xxxxx_xxxxxxxxxxx.wmv
10	http://www.x-xxxxxx-xxxxxx.ch/xxx
11	http://www.xxxxxxxxxx.ch/xxxxx.php?xx=xxxxxxxxxxxx
12	http://www.xxxxxxxx.de/xxxx/2005/xx66/x44444.html
13	http://www.xxx-xxxx.ch/xxx_xxx_111/XXXXXXXXXXXXxXX.html
14	http://www.xxxxxxxxxxx.com
15	http://www.xxxxxxxxxx.ch/xxxxxxxxxxxxxxx/xxxxxxx/XX_XxxxxxxXxxxx.jpg
16	https://www.xxxxxxxx-xxxxxx.ch/xxxxxxx/XxxxXxxxxxxxxxxx?xxx=/xxxxx/xxxxx_xxxx_xxxx-xx.aspx
17	http://www.xxxxxxxxx.xx.tv
18	http://www.xxxxxxxx-xxxxxx.ch/XxXxx/Xxxxx/XxxxxxxXXXX/Xxxxxxxxxx_XxxxxxxXXXX_xx.pdf
19	http://www.XxX24.net
20	www.xxx.de/
21	www.xxx.de/xxxxxxx
22	http://www.xxxxxxxxxxxxxxxx.com/xxxxxxxxxx_xxx/X/X/X/1/XXX11-1.shtml
23	http://xxx1.xxxxxxxxxxxx.com/xxxxxxxxx-xxx/xxxx/222222/XXXXXXX/X3333.html
24	http://177.77.77.77
25	http://www.Xxxxx.xxxxxx.org
26	ftp://ftp.xxxxxxxxx.com
27	ftp://ftp.xxxxxx-xxxxxxxxxx.de/xxxx/
29	http://www.xxx.de:80/xxxxxxx
30	www.xxx.de/xxxxxxx#abschnitt1

Table 6.4: URLs: test examples

6.3 Conclusions

The way in which TM systems handle URLs depends on how URLs are coded in the source text. When a URL is converted into a hyperlink, the URL structure is irrelevant because recognition is based on the hyperlink encapsulation. Still, not all TM systems identify such hyperlinks as placeable or localizable elements, see table 6.5.

	Plain text	**Plain text between tags**	**Placeable**
Across			x
Déjà Vu	x		
Heartsome		x	
memoQ		x	
MultiTrans			x
SDL Trados			x
Transit		x	
Wordfast			x

Table 6.5: Overview: handling of URLs in hyperlinks

Wordfast is the only TM system that is able to recognize most URLs if they occur as plain text.

6.4 Possible improvements

Specific URL recognition by means of a regular expression is required. A viable approach is given in (Goyvaerts and Levithan, 2009, 350-352).

```
1  m/
2  \b
3  (
4      (
5          (https?|ftp|file):\/\/
6          |
7          (www|ftp)\.
8      )
9      [-A-Z0-9+&\@#/%?=~_|\$!:,.;]*
10     [A-Z0-9+&\@#/%=~_|\$]
11 )
12 /gix
```

There are some minor adaptations with respect to the original version. The free spacing and greedy modifiers (line 12) as well as the outer parentheses (lines 3 and 11) were added. The dollar sign and the commercial at were escaped (lines 9 and 10), see (Friedl, 2006, 77).

The regular expression can be divided into two parts.

1. The first (lines 4 to 8) covers the transfer protocols, but – if URLs are written starting with www or ftp – the transfer protocol may be omitted.
2. The second (lines 9 and 10) includes all the standard characters allowed in a URL, but the last character cannot be a punctuation mark.

The aim of the regular expression is the recognition of URLs within a larger text, not their validation.[4] For example, if either http, https or ftp[5] are present, incorrect URLs are also recognized, e.g. `http://www.xxx.cam`.

The regular expression makes some trade-offs in order to avoid frequent misrecognitions, even though the "regex will work incorrectly with certain URLs that use odd punctuation, matching those URLs only partially", (Goyvaerts and Levithan, 2009, 352). For example, recognition will be incomplete with URLs including:

- Literal (i.e. unescaped) spaces.
- Punctuation marks at the end of an URL.

Both literal spaces and punctuation marks at the end of a URL are valid, but seldom.

The regular expression recognizes all examples. A major restriction is that it does not allow non-ASCII characters. If this is required, `A–Z0–9` and `_` in lines 9 and 10 should be replaced by \w, which – in Perl flavor – fully supports Unicode.

Another approach to URL recognition is provided by (Friedl, 2006, 74-75 and 206-208). It is not analyzed here, but it is worth stressing that the author points out the necessity of using heuristics to achieve better recognition, see (Friedl, 2006, 75 and 207).

[4] Depending on the regular expression purposes, (Goyvaerts and Levithan, 2009, 347-355 and 358-364) propose significantly different strategies, particularly if partial validation in accordance with RFC 3986 is necessary, see also IETF (2005).

[5] The regular expression also includes "file", but the file scheme is not relevant as directories are not covered in this thesis. A regular expression for directory recognition is presented in (Goyvaerts and Levithan, 2009, 395-397).

Chapter 7

E-mail addresses

7.1 Introduction

An e-mail address consists of two parts separated by the commercial at (@).[1]

1. Domain part, right of the commercial at

2. Local part, left of the commercial at

The *domain part* must be a fully qualified domain name and follows the restrictions of the Domain Name System (DNS), see also 6.1.

> A domain name [...] consists of one or more components, separated by dots if more than one appears. [...] These components [...] are restricted for SMTP purposes to consist of letters, digits, and hyphens drawn from the ASCII character set. (IETF, 2008b, 13)

The *local part* has to be unique. It does not have to conform to a specific pattern in the same way as the domain part, but some limitations apply to character choice. Non-ASCII letter characters can be used,[2] but might be problematic due to poor support.

Some non-alphanumeric characters are not allowed (see IETF (2008b) for further details), e.g. less-than sign, greater-than sign, left square bracket, right square bracket, colon, semicolon, reverse solidus, etc. Furthermore, several non-alphanumeric characters are theoretically permitted, but not always interpreted as valid (e.g. left curly bracket, right curly bracket, exclamation mark, plus sign, equals sign and tilde), see (IETF, 2008c, 13). As a result, the characters used are usually limited to letters, digits, dot, hyphen-minus sign and low line in the ASCII range.

[1] This description of the structure of an e-mail address follows the specifications of the Standard Mail Transfer Protocol (SMTP).

[2] Earlier RFCs did not allow for non-ASCII characters, see (IETF, 1982, 31) and, more recently, IETF (2008b) as well as IETF (2008a). Internationalization efforts on UTF-8 basis are being made, see IETF (2008e) and IETF (2008d). However, both RFCs did not have standard status at the time of writing.

7.2 Tests

The tests are aimed at checking the following issues:

- Are e-mail addresses in hyperlinks recognized?
- Are e-mail addresses in plain text recognized?
- Are there specific e-mail address patterns that are not recognized?

Recognition assessment does not require that target text be entered, see 2.3.2.1. In order to avoid privacy issues arising from real e-mail addresses, they were anonymized. Except for the top level domain, all ASCII letters were substituted by an x (for lowercase) or an X (for uppercase).[3] Characters outside the ASCII range were retained.

As was the case with URLs, see 6.2, e-mail addresses were embedded in larger sentences. These are not listed since they are not the focus of the test. However, the position of the e-mail address in the sentence affects the way it is processed by some TM systems, as described in the results.

As was the case with URLs, see 6.2.2.1, e-mail addresses that have been entered in MS Word 2003 as plain text are replaced by default by a hyperlink.

7.2.1 TM system settings

7.2.1.1 Versions

TM system	Version
Across Standalone Personal Edition	4.00
Déjà Vu X Professional	7.5.303
Heartsome Translation Studio Ultimate	7.0.6 2008-09-12S
memoQ Corporate	3.2.17
MultiTrans	4.3.0.84
SDL Trados 2007 Freelance	8.2.0.835
STAR Transit XV Professional	3.1 SP22 631
Wordfast	5.53

Table 7.1: TM systems used in the tests

7.2.1.2 Customizability

None of the TM systems tested offers any specific setting for the recognition of e-mail addresses.

[3] The local part “MAY be case-sensitive”, (IETF, 2008b, 41).

7.2.2 Test suite

The following e-mail addresses were tested first as hyperlinks, see 7.2.2.1, and then as plain text, see 7.2.2.2:

	E-mail address
1	`xxxxxxxxx@xxxxxxxx.ch`
2	`xxxxx.xxxxxx@xxxxxxxx.com`
3	`xxx-xxxx@xxxxxxxx.com`
4	`xxxxxxx.xxxxx1@xxxxxxxx.com`
5	`xxxxx.xxxxxxx-xxxxxxxxxxx@xxxxxxxx.com`
6	`Xxxxxxxx.X-Xxxxxxx@xxxxxxxx.com`
7	`xxxxx-xxxxx.xxxxxxxxxx@xxxxxxxx.com`
8	`xxxxxx.xxxxüxx@xxxxxxxx.ch`
9	`xxxxxxxx_xxxxxxx@xxxx.tv`
10	`xéxôxx.xxxxxxxx@xxxxxxxx.com`
11	`2xxxxxxxxxxxxxx.xxx@xxxxxxxx.com`
12	`xxxxxx@xxx-xxxxx.de`
13	`xxxxxxx.xxxxxxxxx@xxx.biz`
14	`xxxxx-xxxxxx-xxxxxx@de.xxx.com`

Table 7.2: E-mail addresses: test examples

7.2.2.1 E-mail addresses as a hyperlink

E-mail addresses were converted into hyperlinks before testing. If the hyperlink is recognized as a placeable element, it is displayed in the same way as shown under 10.2.2.

Results

See table 7.3 for a summary of the results.

	Recognition
Across	yes
Déjà Vu	no
Heartsome	no
memoQ	no
MultiTrans	yes
SDL Trados	yes
Transit	no
Wordfast	yes

Table 7.3: E-mail recognition in hyperlinks

Across, SDL Trados: e-mail addresses are recognized as placeable elements and are never skipped.

Déjà Vu, Heartsome, memoQ: e-mail addresses are presented as plain text.

MultiTrans: e-mail addresses are recognized as placeable elements. However, if the e-mail address is the sole element in the paragraph, the complete paragraph is skipped. If the e-mail address is at the beginning or in the middle of a segment, it is skipped but the text before and after it is highlighted for translation.

Transit: e-mail addresses are presented as plain text between tags.

Wordfast: e-mail addresses are recognized as placeable elements. If the e-mail address is the sole element in the paragraph, the complete paragraph is skipped.

7.2.2.2 E-mail addresses as plain text

The e-mail addresses were tested as plain text. This test was limited to the TM systems that recognized e-mail addresses as placeable elements (Across, MultiTrans, SDL Trados and Wordfast).

Results

No TM system recognizes e-mail addresses as plain text.

7.3 Conclusions

The way in which TM systems handle e-mail addresses depends on how e-mail addresses are coded in the source text. When an e-mail address is converted into a hyperlink, the e-mail address structure is irrelevant because recognition is based on the hyperlink encapsulation. Still, not all TM systems identify hyperlinks as placeable or localizable elements. If e-mail addresses occur as plain text, none of the TM systems is able to recognize them.

7.4 Possible improvements

E-mail addresses can be recognized by means of a regular expression. Since recognition of e-mail addresses is often required, many regular expressions have been presented. The following solution is proposed and discussed in detail by (Friedl, 2006, 70-73):[4]

```
m/
\b
(
    \w[-.\w]*
    \@
    [-a-z0-9]+(\.[-a-z0-9]+)*
    \.
    (com|edu|gov|int|mil|net|org|biz|info|name|museum|coop|aero|[a-z][a-z])
)
\b
```

[4] An explanation as to why the commercial at is escaped is given by (Friedl, 2006, 77).

```
11 /gix
```

The list of the possible top level domains (line 8) is no longer up-to-date as new ones have been introduced in the meantime (e.g. asia, mobi, etc.), but it can be easily expanded. Because of the /i modifier (line 11), the regular expression is case-insensitive.

The author specifies that matching the official e-mail address specification exactly "is difficult, but we can use something less complex that works for most email addresses [...]", (Friedl, 2006, 70). This point is made clear also by (Goyvaerts, 2007, 75): "there's no 'official' fool-proof regex to match email addresses". In fact, this regular expression allows for patterns that are not correct, e.g. name.@mail.com.

(Goyvaerts, 2007, 73-75) as well as (Goyvaerts and Levithan, 2009, 213-219) propose several regular expressions for recognizing e-mail addresses, describing the trade-offs made in each case. Basically, the variants differ depending on their adherence to the standard defined by the RFC 2822, see IETF (2001).[5]

One solution, provided in (Goyvaerts and Levithan, 2009, 215), is:

```
1 m/
2 \b
3 (
4     [\w!#$%&'*+/=?`{|}~^-]+(?:\.[\w!#$%&'*+/=?`{|}~^-]+)*
5     \@
6     (?:[A-Z0-9-]+\.)+[A-Z]{2,6}
7 )
8 \b
9 /gix
```

The regular expression has been slightly adapted: \b (lines 2 and 8) was used instead of ^ and $, respectively, because the regular expression is supposed to "search for email addresses in larger bodies of text", (Goyvaerts and Levithan, 2009, 218); capturing parentheses (lines 3 and 7) and the greedy modifier (line 9) were added; \w was added to the second character class (line 4, a typo in the original).

This variant is more precise as regards the characters and the structure that are allowed in the local part (line 4) – for example, the invalid e-mail address name.@mail.com would not be recognized – but less stringent on the domain name side, simply limiting the number of letters (line 6). In fact, some listed characters of the local part are admissible, but their use is either deprecated or not encouraged (e.g. = { } + and some others, see 7.1). As was the case with URLs in 6.4, the boundaries between recognition and validation are fuzzy and different approaches are possible.

Both regular expressions presented recognize all tested e-mail addresses. For the purpose of a TM system, the second one – which also includes less common characters and is less stringent for the domain part – may be more appropriate and less prone to obsolescence because new top level domains are regularly added.

[5] At the time of writing, RFC 5322 superseded RFC 2822. (Goyvaerts, 2007, 75) presents a regular expression that adheres to RFC 2822, but – by design – does not fulfill all requirements of a valid e-mail address.

Chapter 8

Tags

8.1 Introduction

Tags are defined as markup code[1] (Musciano, 2006, 6) providing information on the structure and format of specific content.

> [A] tag consists of a tag *name*, sometimes followed by an optional list of tag *attributes*, all placed between opening and closing brackets (< and >). [...] A tag's attribute value, if any, follows an equals sign (=) after the attribute name. (Musciano, 2006, 38)

Tags can be classified according to different criteria. Initial classification is based on their structure (Musciano, 2006, 18):

- Tag pairs: a start tag and an end tag define a discrete region of the document. In the end tag, the tag name is preceded by a solidus, e.g. <i> </i>.
- Single tags: the tag does not have any end tag, e.g. <img>.[2]

Start tags and single tags may include required or optional attributes that further specify or complete the tag instructions; the values of these attributes may have to be translated, see 8.2.3.3. Sometimes start or end tags (e.g. </p>) can be omitted in HTML, but not in XML.

Another classification is based on the type of direction provided by the tag.

- Content-based style tags: "attach meaning to various text passages", (Musciano, 2006, 21), e.g. <abbr>, which encloses an abbreviation.

[1] Since the examples used in this chapter are in HTML, this introduction will focus on HTML (4.01) and only on topics relevant to the examples provided. Please refer to Musciano (2006) for extensive information on HTML.

[2] In addition to <img>, further single tags include <area>, <base>, <basefont>, <br>, <col>, < frame>, <hr>, <input>, <isindex>, <link>, <meta> and <param>, (Musciano, 2006, 501).

- Physical style tags: “tell the browser to display (if it can) a character, word, or phrase in a particular physical style”, (Musciano, 2006, 21). In other words, they prescribe what the text should look like, e.g. <b> for bold.

The content-based style will be “rendered in a manner different from the regular text in the document”, (Musciano, 2006, 70-71), when the browser displays the document, but the manner is not hard-coded in the document (it can be controlled by applying a cascading style sheet or JavaScript to the page).

A third classification is more closely related to document structure and translation issues.

- Block tags: start on a new line and enclose paragraph text, e.g. <div>.
- Inline tags: can be used (nested) inside block tags, e.g. <em>.

This classification, see (Musciano, 2006, 291) for further information, is extremely important for the correct segmentation of the document before translation.[3]

Character entities

In HTML, some characters are semantically ambiguous because they can be part of tags or used in plain text. In order to disambiguate their meaning, if they are to be displayed as plain text, they are entered as character entities starting with an ampersand (&) and ending with a semicolon (;). The entity has a special name, e.g. &gt; for the greater-than sign. The decimal or hexadecimal Unicode codepoint of the character, preceded by a number sign (#), can be used to specify the entity instead of the name. For example, &#62; (decimal) or &#x3e; (hexadecimal).

Character entities are not to be confused with tags; they are aliases for special characters. However, it is useful to check whether they are correctly supported by TM systems and how they are displayed.[4] Problems affecting the display of special characters, also when they occur as plain text, is a general issue in the translation process. Central and Eastern European languages, for example, include several characters that may become corrupted, see Nedoma and Nedoma (2004).

8.2 Tests

The tests are aimed at checking the following issues, see also 2.2.2 and 2.4.1.6:

- How are tags displayed?
- How are character entities displayed?

[3] For a concise overview of segmentation issues in TM systems, see Mercier (2003).

[4] If character entities or other elements of HTML code occur within an XML file, their handling is even more complex, see Zetzsche (2011b).

- Are inline tags in the segment or are they interpreted as segment end markers?
- Are attributes translatable if their content has to be localized?
- How do transpositions, replacements, additions and deletions affect the similarity value?
- Does the segment length affect the similarity value?
- Does the number of modifications affect the similarity value?
- Are automatic adaptations applied?

The issues of segmentation and translatability have already been identified for different formats (MS Word, XML, HTML), see e.g. Reinke (2008) and Dockhorn and Reinke (2008). In particular, segmentation difficulties are reported by (Reinke and Höflich, 2002, 284) and Joy (2002); skipped translatable text is reported by Lagoudaki (2009).

The question of the translatability of some attributes relates to accessibility, defined as:

> Making your web content easy for a wide range of users to access. This may include people with vision impairments [...], people with mobility problems [...], and those with cognitive issues. (Lloyd, 2008, 24)

Some translatable attributes (e.g. alt for <img> and summary for <table>) are indeed intended to enhance the accessibility of web pages.

From the point of view of user effort, deletions and transpositions are less problematic than replacements and additions because no new elements are inserted. The calculation of the penalty value should not take the segment length into consideration, but reflect the number of modifications. Replacements and deletions can be managed using automatic adaptations so it is to be expected that these would be applied. All these considerations assume that the modification does not enclose plain text as with translatable attributes. Consequently, for different tags, diverging results as regards penalties and in terms of the effectiveness of automatic adaptations are to be expected. See 8.4 for more information on the recommended handling of tags.

HTML was used for the tests performed here. The examples were extracted from a dump of the English version of Wikipedia. In some cases, the original wording was shortened and minor adaptations were made in order to keep the examples as uncluttered as possible. The HTML code was adapted using the HTML editor phase 5.

8.2.1 TM system settings

8.2.1.1 Versions

TM system	Version
Across Standalone Personal Edition	4.00
Déjà Vu X Professional	7.5.303
Heartsome Translation Studio Ultimate	7.0.6 2008-09-12S
memoQ Corporate	3.2.17
MultiTrans	4.3.0.84
SDL Trados Studio	9.1.0.0
STAR Transit NXT	4.0.0.672.3
Wordfast 6	2.2.0.4

Table 8.1: TM systems used in the tests

8.2.1.2 Customizability

In this section, the availability of the following features is discussed.

- Is the penalty value visible and customizable?
- Can automatic adaptations be activated/deactivated/customized?
- Do the format filter settings influence the treatment of tags and is the filter customizable?

Any adaptations made to the display options are also reported.

Across: under Tools > Profile Settings > CrossTank > Advanced Settings, penalties for specific differences can be set; the default value for Different inline elements, is 1%. Inline elements include placeables, editable fields and tags, see Across (2009a).

Under Tools > System Settings > General > crossTank, the option Use autoadjustments is activated by default. Placeables, formatting and tags are adjusted automatically if they are "[...] the only difference between a segment to be translated and a crossTank entry", Across (2009a). The prerequisite is a Rich TM, see 2.4.3.2.

Under Tool > System Settings > Document Settings > Tagged HTML, it is possible to customize the conversion settings, for example, if elements are internal or external. Translatable tag attributes can also be modified. This option is useful if the defaults are not correct, see 8.3.3.

Déjà Vu: it is not possible to view or customize the penalty applied to differences concerning tags, referred to as "embedded codes", ATRIL (2008).

Tags are substituted automatically, see 8.2.3.6, thanks to an automatic conversion process, see (ATRIL, 2003, 196). This conversion cannot be deactivated.

The file format filter does not provide any option relevant to the treatment of tags.

Heartsome: under ADVANCED > MATCH ENGINE CONFIGURATION, it is possible to define a WRONG TAG PENALTY value. The default value of 0.5 was left unchanged.

There is no option for automatic adaptation of tags. The file format filter does not provide any option relevant to the treatment of tags.

memoQ: it is not possible to view or customize the penalty applied to differences concerning tags.

Under TOOLS > OPTIONS > TM DEFAULTS, USER INFO > ADJUSTMENTS, the option ADJUST FUZZY HITS AND INLINE TAGS is activated by default and enables "on-the-fly adjustment of [...] inline tags within translation memory hits with less than 100% match rate", Kilgray (2008).

In the document import settings, the option IMPORT MARKUP AS INLINE TAGS is activated by default in order to show the tag content, see 8.2.2, and not only placeholders in curly brackets, see 10.2.2. The option IMPORT HTML ENTITIES AS CHARACTERS is activated by default as well, see 8.2.3.1.

MultiTrans: it is not possible to view or customize the penalty applied to differences concerning tags. There is no option for automatic adaptation of tags either.

The file format filter (in the XLIFF Editor under TOOLS > MAPPING EDITOR > XHTML) provides different customization possibilities. It is possible to set elements as internal/external or translatable/not translatable.

SDL Trados: under TRANSLATION MEMORY AND AUTOMATED TRANSLATION > PENALTIES, it is possible to set penalties for MISSING FORMATTING and DIFFERENT FORMATTING; both are by default 1%. MISSING FORMATTING means that "one of the source segments (either the translation memory or the document source) has formatting that is not in the other source segment", SDL (2009). It is not clear whether inline tags are included in formatting and thus affected by this setting. There is no other setting for tag differences or for placeable differences in general.[5] The tests (see e.g. 8.2.3.4) provide an answer to this question.

For automatic adaptations, SDL Trados applies auto-substitutions, but, according to SDL (2009), these are limited to dates, times, numbers, measurements and variables. Again, there is no specific option for tags and the tests (see e.g. 8.2.3.6) show whether automatic adaptation is available.

The file format filter (under TOOLS > OPTIONS > FILE TYPES > HTML) provides different customization possibilities. Under ELEMENTS AND ATTRIBUTES, it is possible to set elements as internal or external, translatable or protected as well as to define whether the content of the attributes is translatable or not. The defaults were retained. Under ENTITY CONVERSION, the option CONVERT ENTITIES is activated by default.

The FULL TAG TEXT display mode was used in conjunction with the display of formatting tags.

[5] "Markup tags, placeholder tags, numbers, variables and dates are all examples of placeables", SDL (2009).

Transit: it is not possible to view or customize the penalty applied to differences concerning tags, referred to as “markups”, see STAR (2009).

For automatic adaptations, the following behavior applies to pretranslation (and interactive translation):

> If a source-language sentence from the reference material and the current source-language sentence only differ in terms of their numbers and/or formatting, Transit NXT only accepts the text itself from the translation in the reference material. The numbers and markups are carried over from the current source file [...]. (STAR, 2009, 121)

The relevant options can be found under Transit button > User preferences > Dual Fuzzy > Update Transit matches. These options specify how fuzzy matches should be updated. The option Markups in particular ensures that “any changes to markups will be updated and the modified segment accepted into the translation”, (STAR, 2009, 228).

The standard file format definition for HMTL files can be adapted under Project > Settings > File type > Define. However, its adaptation requires knowledge of the Transit regular expression flavor.

The NXT_5 (User) view was chosen, with several customizations.

- Full view for Text/Markups.
- Full view for Segment Markers with the options Word wrap and Each in new line.
- Deactivation of the option Show indent level for Language pair

Wordfast: under Edit > Preferences > Translations > Translation Memory > Penalties it is possible to define the penalty for Tag. The default value of 0.5 was left unchanged.

There is no option for automatic adaptation of tags. The file format filter does not provide any option relevant to the treatment of tags.

8.2.2 General remarks on TM system behavior

Display

The display of tags in the editor is essential for understanding the tag function. Therefore, the display is discussed even though it does not affect the retrieval performance of TM systems. An overview is provided in table 8.2.

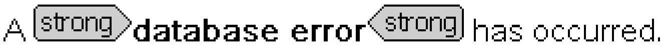
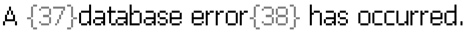
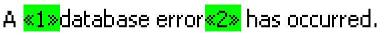
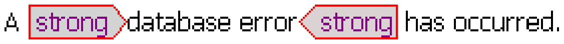
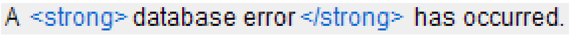
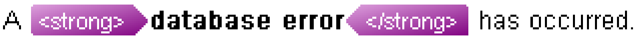
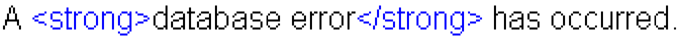
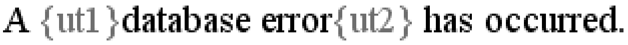
	Display
Across	 A strong database error strong has occurred.
Déjà Vu	 A {37}database error{38} has occurred.
Heartsome	 A «1»database error«2» has occurred.
memoQ	 A strong database error strong has occurred.
MultiTrans	 A <strong> database error </strong> has occurred.
SDL Trados	 A <strong> database error </strong> has occurred.
Transit	 A <strong>database error</strong> has occurred.
Wordfast	 A {ut1}database error{ut2} has occurred.

Table 8.2: Display of tags

Déjà Vu, Heartsome and Wordfast only display a placeholder tag,[6] see also 9.2.2 and 10.2.2. However, it is possible to show the content:

- Déjà Vu: by means of the Display Code function.
- Heartsome: by means of a tooltip displayed with the mouse pointer positioned over the tag.
- Wordfast: by means of a tooltip displayed with the mouse pointer positioned over the tag. In addition, the Txml Context view can be used to display the text and tags of the processed file in full.

The fact that inline code is presented in different ways is pointed out by (Savourel, 2007, 37). In fact, the different displays reflect the different methods of storing information, see 1.4.5. As regards tags with translatable attribute values, most TM systems deploy the value text in an extra segment, while others make the relevant portion of the tags editable, see 8.2.3.3.

[6] A discussion concerning the visibility of formatting tags and codes is provided by (Lagoudaki, 2009, 141).

8.2.3 Test suite

8.2.3.1 Character entities

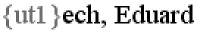
In order to evaluate the support of character entities, the following examples were processed:

1. The S&amp;W Cafeteria Building is also a fine example of Art Deco architecture in Asheville.
   displays
   The S&W Cafeteria Building is also a fine example of Art Deco architecture in Asheville.

2. Z&uuml;rich, Switzerland
   displays
   Zürich, Switzerland

3. Differences with HTML&#160;4
   displays
   Differences with HTML 4[7]

4. &#268;ech, Eduard
   displays
   Čech, Eduard

Table 8.3 summarizes the results. Most TM systems convert character entities into the respective character, but there are some exceptions. Heartsome does not convert the entity &#268; into a Č, but displays a tag, while all other tested character entities are converted. Wordfast converts the entities into tags (e.g.  {ut1}ech, Eduard). A tooltip shows the corresponding letter.

	Display
Across	wysiwyg character
Déjà Vu	wysiwyg character
Heartsome	tag / wysiwyg character
memoQ	wysiwyg character
MultiTrans	wysiwyg character
SDL Trados	wysiwyg character
Transit	wysiwyg character
Wordfast	tag

Table 8.3: Display of character entities

[7] No-break space between HTML and 4.

8.2.3.2 Inline tags

Inline tags and in particular their segmentation deserve special consideration. The following examples were used:

1. A <strong>database error</strong> has occurred.

2. Modern consolidation, created from the 70
 double-page spreads of the original atlas.

3. Programs that perform similar operations as the Unix <code>touch</code> utility are available for other operating systems, including Microsoft Windows and MAC OS.

Table 8.4 summarizes the results. If the tag is inline, "yes" is inserted. Generally, inline tags are dealt with correctly, but Déjà Vu and Heartsome segment in two cases. It would be interesting to check the remaining inline tags, see 8.1. This check, however, would constitute specific assessment of the HTML format filter, which was not the scope of these tests.

	strong	**br**	**code**
Across	yes	yes	yes
Déjà Vu	yes	no	no
Heartsome	yes	no	no
memoQ	yes	yes	yes
MultiTrans	yes	yes	yes
SDL Trados	yes	yes	yes
Transit	yes	yes	yes
Wordfast	yes	yes	yes

Table 8.4: Handling of inline tags

8.2.3.3 Translatable attributes

In order to determine whether attributes with translatable content are presented for translation, the following examples were used:

1. <img alt="A blond woman with rosy cheeks holds a white rose. She wears a gilded black shawl over her head, and a red robe trimmed in white spotted fur." src="http://upload.wikimedia.org/wikipedia/commons/thumb/c/c6/Elizabeth_of_York%2C_right_facing_portrait.jpg/140px-Elizabeth_of_York%2C_right_facing_portrait.jpg" width="140" height="173" class="thumbimage" />

2. <table class="wikitable" summary="This table contains some specifications for common MIPS microprocessors. Each microprocessor is given the frequency in megahertz, the release year, the fabrication process in micrometers, the number of transistors (in millions), the size of the die in square millimeters, the pin count, the power dissipation in watts, its voltage, and the sizes of the data, instruction, L2 and L3 caches.">

3. <th abbr="Frequency">Frequency (MHz)</th>

4. Economists have been trying to analyze the overall net benefit of Kyoto Protocol through <a href="/wiki/Cost-benefit_analysis" title="Cost-benefit analysis">cost-benefit analysis</a>.

Table 8.5 summarizes the results. Only Déjà Vu and memoQ correctly support all tested translatable attributes. Other TM systems only support some of them.

	img alt	**table summary**	**th abbr**	**a title**
Across	yes	no	no	yes
Déjà Vu	yes	yes	yes	yes
Heartsome	yes	no	no	yes
memoQ	yes	yes	yes	yes
MultiTrans	no	yes	yes	yes
SDL Trados	yes	no	no	yes
Transit	yes	no	yes	yes
Wordfast	yes	no	no	yes

Table 8.5: Handling of translatable attributes

The title attribute of <a> is always translatable. The alt attribute of <img> is not translatable in one instance. The abbr attribute of <th> is not correctly supported by several TM systems and the summary attribute of <table> is not translatable in the majority of cases. The fact that some attributes are supported more effectively than others is due to their role in the HTML standard. The alt attribute, for example, is required, while summary is only optional and far less frequent.

Whether an attribute value is presented for translation depends on the format filter used while importing the HTML file into the TM system. Although some TM systems allow these settings to be customized, these shortcomings are unexpected considering that HTML is a standardized format.

As table 8.6 illustrates, the TM systems present the content of translatable attributes (provided that they are recognized as such) in different ways. Across and Transit opt for displaying it as editable text within the segment, while most TM systems prefer to move it to a separate, text-only segment.

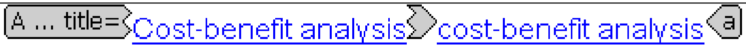
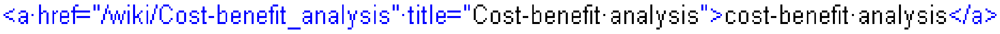
	Display
Across	 A ... title=Cost-benefit analysis cost-benefit analysis a
Déjà Vu	independent segment
Heartsome	independent segment
memoQ	independent segment
MultiTrans	independent segment
SDL Trados	independent segment
Transit	 <a·href="/wiki/Cost-benefit_analysis"·title="Cost-benefit·analysis">cost-benefit·analysis</a>
Wordfast	independent segment

Table 8.6: Display of translatable attributes

8.2.3.4 Addition

Tag pair

1. Amstrong stepped off Eagle's footpad and into history as the first human to set foot on another world
2. Amstrong stepped off <i>Eagle</i>'s footpad and into history as the first human to set foot on another world

Difference: "Eagle" was put in italics (<i>).

Results

Across: a 98% match is proposed. There is no automatic adaptation.

Déjà Vu: a 98% match is proposed. If the proposed fuzzy match is accepted, a partial adaptation is made; added tags are copied into the fuzzy match and have to be manually moved to the right position.

Heartsome, Wordfast: a 99% match is proposed. There is no automatic adaptation.

memoQ: an 85% match is proposed. There is no automatic adaptation.

MultiTrans: a 46% match is proposed. There is no automatic adaptation.

SDL Trados: a 91% match is proposed. There is no automatic adaptation. This result proves that inline tags are not treated as formatting elements, otherwise a 1% penalty would apply, see 8.2.1.2.

Transit: a 93% match is proposed. There is no automatic adaptation.

Single tag

1. Secularisation of the Monastic state of the Teutonic Knights

2. Secularisation of the Monastic
state of the Teutonic Knights

Difference: a line break (<br>) was added before “state”.

Results

Across, memoQ: a 98% match is proposed. There is no automatic adaptation.

Déjà Vu: a 98% match is proposed. If the proposed fuzzy match is accepted, a partial adaptation is made; added tags are copied into the fuzzy match and have to be manually moved to the right position.

Heartsome, SDL Trados, Transit, Wordfast: a 99% match is proposed. There is no automatic adaptation.

MultiTrans: a 45% match is proposed. There is no automatic adaptation.

8.2.3.5 Deletion

Tag pair

1. Also note that an agency can still be in <strong>legal</strong> compliance by meeting one of the &sect; 1194.3 General exceptions (e.g., the NSA)[8]

2. Also note that an agency can still be in legal compliance by meeting one of the &sect; 1194.3 General exceptions (e.g., the NSA)

Difference: the stressing (<strong>) was removed.

Results

Across, memoQ: a 98% match is proposed. The superfluous tags are deleted automatically.

Déjà Vu, Transit, Wordfast: a 99% match is proposed. The superfluous tags are deleted automatically.

Heartsome: a 100% match is proposed. The superfluous tags are deleted automatically.

MultiTrans: a 50% match is proposed. There is no automatic adaptation.

SDL Trados: a 97% match is proposed. The superfluous tags are deleted automatically. Automatic adaptations then also apply to tags, see 8.2.1.2.

Single tag

1. Modern consolidation, created from the 70
 double-page spreads of the original atlas.

2. Modern consolidation, created from the 70 double-page spreads of the original atlas.

Difference: the line break (<br>) was deleted.

[8] The character entity &sect; displays §.

Results

Across, memoQ: a 98% match is proposed. The superfluous tag is deleted automatically.

Déjà Vu, SDL Trados, Transit, Wordfast: a 99% match is proposed. The superfluous tag is deleted automatically.

Heartsome: a 100% match is proposed. The superfluous tag is deleted automatically.

MultiTrans: a 54% match is proposed. There is no automatic adaptation.

8.2.3.6 Replacement

Tag pair

1. Air Dolomiti <small>(a subsidiary company of Lufthansa)</small>

2. Air Dolomiti <strong>(a subsidiary company of Lufthansa)</strong>

Difference: the <small> tag was replaced by <strong>.

Results

Across, Déjà Vu, Heartsome, SDL Trados, Wordfast: a 100% match is proposed. The modified tags are replaced automatically.

memoQ: a 97% match is proposed.[9] There is no automatic adaptation.

MultiTrans: an 86% match is proposed. There is no automatic adaptation.

Transit: a 99% match is proposed. The modified tags are replaced automatically.

Tag attribute value

1. Munich has a <a href="/wiki/Continental_climate" title="Continental climate"> continental climate</a>.

2. Munich has a <a href="/wiki/Continental_climate" title="Climate">continental climate</a>.

Difference: the value of the title attribute of <a> was changed from "Continental climate" to "Climate".

Results

Across: a 79% match is proposed. There is no automatic adaptation.

Déjà Vu: an 86% match is proposed. There is no automatic adaptation.

Heartsome: a 73% match is proposed. There is no automatic adaptation.

[9] This value is caused by incorrect segmentation that leaves the closing tag </small> outside the first segment, whereas </strong> is correctly included in the second segment.

memoQ, SDL Trados, Wordfast: a 100% match is proposed. The modified tag is replaced automatically, but the content of the modified attribute is presented in a subsequent segment.

MultiTrans: a 97% match is proposed. There is no automatic adaptation.

Transit: an 81% match is proposed. There is no automatic adaptation.

8.2.3.7 Transposition

Style tag pair

1. This statement is true <em>only</em> when the subject distance is small in comparison with the hyperfocal distance, however.

2. This statement is <em>true</em> only when the subject distance is small in comparison with the hyperfocal distance, however.

Difference: the emphasis (<em>) was shifted from “only” to “true”.

Results

Across, Heartsome, memoQ: a 100% match is proposed. However, there is no automatic adaptation; the match should be in fact a fuzzy match.

Déjà Vu: a 98% match is proposed. There is no automatic adaptation.

MultiTrans: a 93% match is proposed. There is no automatic adaptation.

SDL Trados, Wordfast: a 99% match is proposed. There is no automatic adaptation.

Transit: a 99% match is proposed. The tags that have been moved are deleted from the fuzzy match when it is accepted.

Anchor tag pair

1. The two types of fission bomb <a href="/wiki/Nuclear_weapon_design" title="Nuclear weapon design">assembly methods</a> investigated during the <a href="/wiki/Manhattan_Project" title="Manhattan Project">Manhattan Project</a>.

2. The two types of <a href="/wiki/Nuclear_weapon_design" title="Nuclear weapon design">fission bomb</a> assembly methods investigated during the <a href="/wiki/Manhattan_Project" title="Manhattan Project">Manhattan Project</a>.

Difference: the hyperlink (<a>) was shifted from “assembly methods” to “fission bomb”, but the link destination remained the same.

Results

Across: a 93% match is proposed. There is no automatic adaptation.

Déjà Vu: a 77% match is proposed. There is no automatic adaptation.

Heartsome: a 92% match is proposed. There is no automatic adaptation.

memoQ: a 100% match is proposed. However, there is no automatic adaptation; the match should be in fact a fuzzy match.

MultiTrans: a 90% match is proposed. There is no automatic adaptation.

SDL Trados, Wordfast: a 99% match is proposed. There is no automatic adaptation.

Transit: a 99% match is proposed. The tags that have been moved are deleted from the fuzzy match when it is accepted.

8.2.3.8 Variable segment length

In this section, testing investigates whether the similarity value is calculated based on the segment length. The type of modification is always the same to ensure the comparability of the results that are presented in table 8.11 under 8.3.4.

Short segment: style tag pair

1. A <strong>database error</strong> has occurred.

2. A <strong>database <em>error</em></strong> has occurred.

Difference: emphasis (<em>) was added to “error”.

Long segment: style tag pair

1. A <strong>database error</strong> has occurred when attempting to delete a record from the specified file.

2. A <strong>database <em>error</em></strong> has occurred when attempting to delete a record from the specified file.

Difference: emphasis (<em>) was added to “error”.

Short segment: anchor tag pair

1. Munich has a <a href="/wiki/Continental_climate" title="Continental climate"> continental climate</a>.

2. <b>Munich</b> has a <a href="/wiki/Continental_climate" title="Continental climate">continental climate</a>.

Difference: bold face (<b>) was added to “Munich”.

Long segment: anchor tag pair

1. Munich has a <a href="/wiki/Continental_climate" title="Continental climate"> continental climate</a>, strongly modified by the proximity of the Alps.

2. <b>Munich</b> has a <a href="/wiki/Continental_climate" title="Continental climate">continental climate</a>, strongly modified by the proximity of the Alps.

Difference: bold face (<b>) was added to "Munich".

8.2.3.9 Variable number of modifications

In this section, testing investigates whether the similarity value depends on the number of modifications applied to one segment. The type of modification is always the same to ensure the comparability of the results that are presented in table 8.12 under 8.3.4.

Short segment

1. Last transmission February 6, 1966, 22:55 UTC.

2. Last transmission
February 6, 1966, 22:55 UTC.

3. Last transmission
February 6, 1966,
22:55 UTC.

Difference: in the second segment, a line break (<br>) was added before "February". In the third segment, a further line break was added before "22".

Long segment

1. Inverse spinel structures however are slightly different in that you must take into account the Crystal Field Stabilisation Energies (CFSE) of the Transition metals present.

2. <b>Inverse spinel structures</b> however are slightly different in that you must take into account the Crystal Field Stabilisation Energies (CFSE) of the Transition metals present.

3. <b>Inverse spinel structures</b> however are slightly different in that you must take into account the <b>Crystal Field Stabilisation Energies</b> (CFSE) of the Transition metals present.

Difference: in the second segment, bold face (<b>) was added to "Inverse spinel structures". In the third segment, bold face was added to "Crystal Field Stabilisation Energies" too.

8.3 Conclusions

8.3.1 Display and character entities

Many TM systems allow for customization of the tag display, but not all to the same extent. Possible displays range from hidden tags to full tags, where they are displayed with all their attributes and values. Some TM systems enable quick switching between views, e.g. by means of icons. However, for some TM systems it is not possible to show – by default – the content of the tag. This is a disadvantage for several reasons:

- It may slow down the translation process if users have to check the content with additional effort.

- It can increase errors due to misplaced and skipped tags.

Moreover, it remains true that "unless the translator knows what the tags signify, (s)he cannot use them [...]", Joy (2002). Criticism of cryptic displays was also expressed by (Pym, 2004, 163).

Most TM systems convert character entities into their corresponding characters. This conversion is not made by Wordfast and is not always made by Heartsome, which present tags instead.

8.3.2 Inline tags

Although the tests did not cover all inline tags, some general conclusions are possible. Most TM systems handle inline tags correctly. However, in Déjà Vu, Heartsome and memoQ, some inline tags do segment the text. This is a bug because a standardized (also as regards segmentation) and widespread format such as HTML is not reliably supported.

Moreover, the segmentation is not always correct in conjunction with block tags. For example, Heartsome sometimes treats <p> as an internal tag, e.g. in the fourth example in 8.2.3.1, in the first example in 8.2.3.6, in the examples in 8.2.3.7 and in the first example in 8.2.3.9.

8.3.3 Translatable attributes

Although the tests did not cover all translatable attributes, some general conclusions are possible. The support of translatable attributes is poor. Most TM systems do not present all relevant attribute values for translation. Only Déjà Vu and memoQ correctly support all tested translatable attributes.

Since attributes are standardized constituents of the HTML format, these bugs are severe, particularly if they affect compulsory attributes (such as alt for <img>).

8.3.4 Penalties

For additions, see table 8.7, the similarity value ranges between 85% and 99%.[10] However, if only one tag is changed, the penalty value is lower and ranges between 1% and 2%. Penalties between 0% and 3% are applied to deletions, see table 8.8. Sometimes no penalty is applied because automatic adaptations convert the fuzzy match into a 100% match.

	1	**2**
Across	98	98
Déjà Vu	98	98
Heartsome	99	99
memoQ	85	98
MultiTrans	46	45
SDL Trados	91	99
Transit	93	99
Wordfast	99	99

Table 8.7: Match values: addition

	1	**2**
Across	98	98
Déjà Vu	99	99
Heartsome	100	100
memoQ	98	98
MultiTrans	50	54
SDL Trados	97	99
Transit	99	99
Wordfast	99	99

Table 8.8: Match values: deletion

For transpositions, see table 8.9, the similarity value ranges between 90% and 100%.[11] 100% matches are proposed in conjunction with automatic adaptations. However, these adaptations are never correct and supposed 100% matches have to be edited manually. The fact that the position of certain elements is not taken into account when calculating the similarity value is one of the most interesting findings, see also (Nübel and Seewald-Heeg, 1999b, 28).[12]

For replacements, see table 8.10, the similarity value ranges between 73% and 100%. Unlike transpositions, automatic adaptations producing 100% matches are successful and common.

[10] MultiTrans produces erratic results for additions as well as for deletions by applying extremely high penalties.

[11] One erratic result produced by Déjà Vu is excluded.

[12] In fact, according to Nübel and Seewald-Heeg (1999b), 100% matches are proposed for textual transpositions, which is even more surprising.

	1	2
Across	100	93
Déjà Vu	98	77
Heartsome	100	92
memoQ	100	100
MultiTrans	93	90
SDL Trados	99	99
Transit	99	99
Wordfast	99	99

Table 8.9: Match values: transposition

	1	2
Across	100	79
Déjà Vu	100	86
Heartsome	100	73
memoQ	97	100
MultiTrans	86	97
SDL Trados	100	100
Transit	99	81
Wordfast	100	100

Table 8.10: Match values: replacement

In general, significant variations between the similarity values proposed by TM systems can be observed. The penalty values are not always transparent and adequate. TM systems sometimes do not apply any penalty although the user has to perform some manual adaptations (e.g. because automatic adaptations are not correct). In other case, the high penalty seems to overestimate the necessary adaptation effort (e.g. for replacements).

Segment length

In some TM systems, the penalty value for the same modification is calculated based the length of the segment where the modification occurs. On the one hand, Across, Heartsome, memoQ, SDL Trados, Transit and Wordfast apply a fixed penalty value irrespective of the segment length.[13] On the other hand, for Déjà Vu and MultiTrans the penalty value applied to the same modification decreases as the segment length increases (and vice-versa). The difference in the penalty value is slight for Déjà Vu (2%), and higher for MultiTrans (5% – 8%). This inverse correlation of penalty and length is not justified because the adaptation effort for the user remains virtually the same.

	Short 1	Long 1	Short 2	Long 2
Across	98	98	98	98
Déjà Vu	97	99	95	97
Heartsome	99	99	99	99
memoQ	98	98	98	98
MultiTrans	55	63	61	66
SDL Trados	97	97	98	98
Transit	93	93	93	93
Wordfast	99	99	99	99

Table 8.11: Match values: variable segment length

[13] For SDL Trados, it is directly confirmed by SDL: "Tags trigger penalties, while words are counted relative to the segment length", ProZ (2010a).

Number of modifications

Across, Heartsome, memoQ and Wordfast apply the same penalty value irrespective of the number of modifications. Déjà Vu, MultiTrans, SDL Trados and Transit increase the penalty values.

	Short 1	**Short 2**	**Long 1**	**Long 2**
Across	98	98	98	98
Déjà Vu	98	97	98	97
Heartsome	99	99	99	99
memoQ	98	98	98	98
MultiTrans	42	-	53	36
SDL Trados	99	98	98	96
Transit	99	99	93	87
Wordfast	99	99	99	99

Table 8.12: Match values: variable number of modifications

8.3.5 Automatic adaptations

Automatic adaptations are common for deletions, with the exception of MultiTrans. For replacements, a more differentiated situation can be observed. Only SDL Trados and Wordfast always apply an automatic adaptation, the success of which depends on the tag involved.

Automatic adaptations are generally not applied to additions and transpositions, with two exceptions. Déjà Vu inserts added tags, which have to be manually repositioned. When accepting a fuzzy match, Transit deletes repositioned tags, which have to be reinserted.

	Additions		**Deletions**		**Transpositions**		**Replacements**	
	1	2	1	2	1	2	1	2
Across	no	no	yes	yes	no	no	yes	no
Déjà Vu	p	p	yes	yes	no	no	yes	no
Heartsome	no	no	yes	yes	no	no	yes	no
memoQ	no	no	yes	yes	no	no	no	yes
MultiTrans	no	no	no	no	no	no	no	no
SDL Trados	no	no	yes	yes	no	no	yes	yes
Transit	no	no	yes	yes	p	p	yes	no
Wordfast	no	no	yes	yes	no	no	yes	yes

Table 8.13: Overview: automatic adaptations

8.4 Possible improvements

8.4.1 Display, inline tags and translatable attributes

As regards the display of tags, maximum flexibility should be possible. During translation, tag content should be visible without additional effort[14] and entities should be displayed as wysiwyg characters.

The segmentation of inline tags and the attribute translatability for standardized formats such as HTML should adhere to the standard. The problems encountered during testing were due to file format filters that need to be fixed.[15]

8.4.2 Penalties and automatic adaptations

Many observations can be made regarding the similarity value. The penalty should be weighted depending on the action necessary to adapt the target segment, see also Piperidis et al. (1999). Firstly, mere tag modifications generally require less adaptation effort than textual ones. Therefore, the special penalization adopted by the TM systems is justified. As pointed out by Joy (2002), the penalization applied to tag differences should not prevent the TM system from retrieving previously translated segments without tags or with different tags. On the other hand, if modified tags contain textual elements (e.g. translatable attributes), this should be given appropriate consideration. Secondly, the number of modifications should be taken into account and the penalty should be proportional because the more modifications, the more effort is needed for adaptation. Effectively, the segment length is irrelevant when modifications only apply to tags. Thirdly, generally speaking, the penalty depends on the automatic adaptations applied.

As regards automatic adaptations, it should be possible for the user to decide if automatic adaptations lead to a 100% match or if a penalty is applied in any case. For deletions, automatic adaptations are mostly unproblematic. However, it is not possible to exclude ambiguous situations, e.g. where the same tag appears twice (or more) in a segment and only one occurence is deleted. Without linguistic knowledge, it is difficult to ascertain which tag has to be deleted in the target segment because syntax reordering may have been performed. The following German source text and its Italian translation provide an example of an ambiguous situation:

1. Bei <b>Steigungen</b> erleichtert die <b>Berganfahrhilfe</b> (oder <b>Berganfahrassistent</b>) das Anfahren.

   L'<b>assistente di partenza in salita</b> (o <b>hill holder</b>) facilita la partenza in <b>salita</b>.

[14] The visibility of formatting tags and underlying code is not uncontroversial among users, see (Lagoudaki, 2008, 141) and Lagoudaki (2009), depending on experience, computer skills and the formats edited. Hiding tags completely can be helpful in the review process.

[15] Problems affecting file format filters are not limited to HTML, but are generally an issue, see Champollion (2003).

2. Bei Steigungen erleichtert die <b>Berganfahrhilfe</b> (oder <b>Berganfahrassistent</b>) das Anfahren.

   L'<b>assistente di partenza in salita</b> (o <b>hill holder</b>) facilita la partenza in salita.

Some replacements can be suitable for automatic adaptation, for example, if tags of the same type are replaced (tag pair with another tag pair, single tag with another single tag) and if the tag modification does not imply textual modifications. In other cases, no automatic adaptation is generally possible.

For transpositions, automatic adaptation is generally impossible because it usually requires linguistic knowledge. Still, there may be exceptions, e.g. if a tag pair is moved to an unambiguous position. However, these cases are likely to be very rare. For the above-mentioned reasons, 100% matches proposed for transpositions tend to be incorrect. It is questionable whether partial adaptation of the tags is useful, and no clear answer can be given without specific assessment.

For additions, no automatic adaptation is generally possible because it usually requires linguistic knowledge. Supposed 100% matches often need further correction. Again, there may be exceptions, e.g. when the position of the added tags can be determined unequivocally.

To summarize, automatic adaptations are not always possible and sometimes only partial. Nevertheless, successful automatic adaptations do speed up the translation.

Chapter 9

Inline graphics

9.1 Introduction

Inline graphics are small images that appear in the text flow. They are frequent in technical and particularly software documentation and often reproduce elements of the graphical user interface (GUI).

9.2 Tests

The tests are aimed at checking the following issues, see also 2.2.2 and 2.4.1.6:

- How are inline graphics displayed?
- How do transpositions, replacements, additions and deletions affect the similarity value?
- Does the segment length affect the similarity value?
- Are automatic adaptations applied?

From the point of view of user effort, deletions and transpositions are less problematic than replacements and additions because no new elements are inserted. The calculation of the penalty value should be independent of the segment length. Replacements and deletions are manageable by automatic adaptations and it can be expected that some are applied. See 9.4 for more information on the recommended handling of inline graphics.

The input file used for this test set was the English user manual of Wordfast (Champollion (2008b)), available in MS Word format. The reasons for selecting this document are explained in 2.4.1. The inline graphics are directly inserted in the document (not as an object, see chapter 10.1.2).

Several examples were adapted in order to better suit the aims of the test. Consequently, they do not contain the same wording in the original document. The presented modifications are constructed, see 2.4.1. They are not intended to be meaningful from a semantic point of view, therefore, they are to be considered exclusively from a formal point of view.

9.2.1 TM system settings

9.2.1.1 Versions

TM system	**Version**
Across Standalone Personal Edition	4.00
Déjà Vu X Professional	7.5.303
Heartsome Translation Studio Ultimate	7.0.6 2008-09-12S
memoQ Corporate	3.2.17
MultiTrans	4.3.0.84
SDL Trados 2007 Freelance	8.2.0.835
STAR Transit XV Professional	3.1 SP22 631
Wordfast	5.53

Table 9.1: TM systems used in the tests

9.2.1.2 Customizability

In this section, the availability of the following features will be discussed.

- Is the penalty value visible and customizable?
- Can automatic adaptations be activated/deactivated/customized?
- Do the format filter settings influence the treatment of inline graphics and is the filter customizable?

Across: under Tools > Profile Settings > CrossTank > Advanced Settings, penalties for specific differences can be set. The default value for Different inline elements is 1%. Inline elements include placeables (such as inline graphics), editable fields and tags, see Across (2009a).

Under Tools > System Settings > General > crossTank, the option Use autoadjustments is activated by default. Placeables, formatting and tags are adjusted automatically if they are "[...] the only difference between a segment to be translated and a crossTank entry", Across (2009a). The prerequisite is a Rich TM, see 2.4.3.2.

The file format filter does not present any option relevant to inline graphics.

Déjà Vu: inline graphics are always treated as a segment separator. If they occur at the beginning of the segment, they are skipped. Two drawbacks arise: firstly, the inline graphic is not displayed. Secondly, syntax reordering may be impossible in the target language. This problem can be solved by joining the segments.[1] Joining was necessary to ensure comparability of the results with other TM systems.

It is not possible to view or customize the penalty applied to differences concerning inline graphics. When inline graphics are converted into tags ("embedded codes"), automatic conversion is applied, see (ATRIL, 2003, 196). Inline graphics are then substituted automatically, see 9.2.3.3 for an example. This conversion cannot be deactivated.

The file format filter does not provide any option relevant to inline graphics.

Heartsome, memoQ, MultiTrans: inline graphics are treated as tags, see 8.2.1.2 for more information on penalties and automatic adaptation settings.

The file format filter does not provide any option relevant to inline graphics.

SDL Trados: TagEditor (8.2.835) was used as an editor instead of MS Word.

In Translator's Workbench, under OPTIONS > TRANSLATION MEMORY OPTIONS > PLACEABLE DIFFERENCE PENALTY[2] the penalty value can be set. In these tests, it was left at 2% (default value). However, this penalty only applies when the tags in the target and source segment, both coming from the translation memory, are different. This penalty is *not* applied when the difference concerns the source segment in the translation memory and the source segment in the document. For this to happen, the option APPLY PLACEABLE PENALTY ALSO WHEN SOURCE TAG DIFFER has to be activated. However, it was left deactivated (default setting).

There is no option concerning the automatic adaptation of inline graphics (e.g. under FILE > SETUP > SUBSTITUTIONS), but the following description applies to inline graphics because they are treated as tags: "If the current source segment is the same as a source segment from translation memory apart from its variable elements, Translator's Workbench produces a 100% match", SDL (2007). This automatic adaptation is a built-in feature.

Moreover, under TOOLS > OPTIONS > GENERAL, the option STRIP TAGS FROM FUZZY MATCHES IF NO TAGS APPEAR IN THE SOURCE SEGMENT is activated in order to "remove unwanted tags from fuzzy matches when no tag appears in the source segment [...]. Tags are removed when you insert the fuzzy match target segment into the document [...]", SDL (2007).

Finally, the file format filter does not provide any option relevant to inline graphics.

Transit: inline graphics are treated as tags, but there is no setting concerning the penalty for differences in tags. As regards automatic adaptations, see the considerations in 10.2.1.2.

The file format filter does not provide any option relevant to inline graphics.[3]

Wordfast: inline graphics are not considered as placeables and there is no setting concerning penalties or automatic adaptations.

Since MS Word is used as the editor, no file format conversion is necessary.

[1] The option PREVENT SEGMENTATION under TOOLS > OPTIONS > FILTERS > DOC > DEFAULT FILTER OPTIONS (deactivated by default) is not related to this problem.

[2] In SDL Trados terminology, "placeables" include also inline graphics: "[...] non-translatable elements occurring within segments, such as tags, graphics, date or name fields, formulas and so on. These elements are referred to as placeables", SDL (2007).

[3] The option "Process objects" only applies to embedded OLE objects, see STAR (2005).

9.2.2 General remarks on TM system behavior

Display

Inline graphics are displayed in different ways. Some customizations are possible, but the examples presented in table 9.2 are based on the standard display settings.

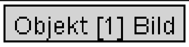
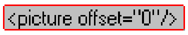
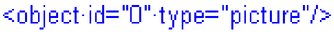
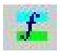
	Display as	**Example**
Across	object	 Objekt [1] Bild
Déjà Vu	placeholder	{31}
Heartsome	placeholder	«2»
memoQ	placeholder	{1}
MultiTrans	empty box	□
SDL Trados	tag	 <picture offset="0"/>
Transit	tag	 <object·id="0"·type="picture"/>
Wordfast	normal picture	

Table 9.2: Display of inline graphics

Across, SDL Trados and Transit show a tag whose name indicates that it stands for an inline graphic. Déjà Vu and Heartsome only display placeholders, but they can show the tag content if desired, see 8.2.2. However, Heartsome does not clearly display all inline graphics, see 9.2.3. memoQ shows a placeholder, but – unlike Déjà Vu and Heartsome – it is not possible to show more information on the tag content. MultiTrans displays an empty box as a placeholder in the Translation Agent. Finally, Wordfast uses MS Word as an editor and inline graphics are presented as normal pictures.

Automatic adaptations

Except for Heartsome and MultiTrans, most TM systems make automatic adaptations, see table 9.3. The test description in 9.2.3 provides further details.

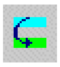
	Automatic adaptations?
Across	sometimes
Déjà Vu	often
Heartsome	no
memoQ	sometimes
MultiTrans	no
SDL Trados	sometimes
Transit	often
Wordfast	sometimes

Table 9.3: Support of automatic adaptations

9.2.3 Test suite

9.2.3.1 Addition

The following examples were used:

1. Another way is to use the CopySource icon or shortcut.
2. Another way is to use the CopySource icon  or shortcut.

Difference: the inline graphic was added.

Results

Across: a 93% match is proposed. There is no automatic adaptation. Of the 7% penalty, 5% is due to the difference in space, 2% to the inline element (inline graphic).

Déjà Vu: a 98% match is proposed. If the proposed fuzzy match is accepted, the inline graphic is inserted automatically, but has to be repositioned manually.

Heartsome: a 99% match is proposed. The graphic is displayed as a tag.

memoQ, SDL Trados: a 98% match is proposed. There is no automatic adaptation.

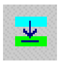
MultiTrans: a 100% match is proposed. The difference has not been recognised.

Transit: a 96% match is proposed. If the proposed fuzzy match is accepted, the inline graphic is inserted automatically, but has to be repositioned manually.

Wordfast: a 92% match is proposed. There is no automatic adaptation.

9.2.3.2 Deletion

The following examples were used:

1. Ctrl+Alt+Down  copies the selected placeable at the position of the cursor (in the target segment).

2. Ctrl+Alt+Down copies the selected placeable at the position of the cursor (in the target segment).

Difference: the inline graphic (and the space after it) were deleted.

Results

Across: a 93% match is proposed. If the proposed fuzzy match is accepted, the superfluous inline graphic is automatically deleted. However, the double space has to be corrected manually.

Déjà Vu: a 100% match is proposed. The inline graphic as well as the superfluous space are deleted automatically.

Heartsome: a 99% match is proposed. The graphic is displayed as a tag.

memoQ: a 99% match is proposed. There is no automatic adaptation.

MultiTrans: a 100% match is proposed. The difference has not been recognized.

SDL Trados: a 98% match is proposed. If the proposed fuzzy match is accepted, the superfluous inline graphic is automatically deleted. However, the double space has to be corrected manually.

Transit: a 99% match is proposed. If the proposed fuzzy match is accepted, the superfluous inline graphic is automatically deleted. However, the double space has to be corrected manually.

Wordfast: no match is proposed even though the fuzzy threshold was set to 50%.

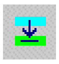
9.2.3.3 Replacement

In the source segment, an inline graphic can be replaced by another one or by a different placeable element (e.g. a number referring to a legend).

Replacement with another inline graphic

The following examples were used:

1. Click the icon to launch Wordfast

2. Click the  icon to launch Wordfast

Difference: a different inline graphic was used.

Results

Across, Déjà Vu, memoQ, SDL Trados, Transit: a 100% match is proposed. The inline graphic is replaced automatically.

Heartsome: a 99% match is proposed. The inline graphic is displayed as a tag.

MultiTrans: a 100% match is proposed. The difference has not been recognized.

Wordfast: a 96% match is proposed. The new icon is automatically inserted in the target segment; however, the placeholder for the old one (&'1;) has to be deleted manually.

Replacement with a different element

The following examples were used:

1. Ctrl+Alt+Down ↓ copies the selected placeable at the position of the cursor (in the target segment).
2. Ctrl+Alt+Down { AUTONUM } copies the selected placeable at the position of the cursor (in the target segment).

Difference: the inline graphic was replaced by an auto-numbering field.

Results

Across, Déjà Vu, SDL Trados: a 100% match is proposed. The inline graphic is replaced automatically by the field.

Heartsome: a 99% match is proposed. The auto-numbering field is not clearly identifiable, see also 9.2.3.4.

memoQ: an 85% match is proposed. There is no automatic adaptation.

MultiTrans: the auto-numbering field is treated as a segment separator. The first segment contains "Ctrl+Alt+Down": no match is proposed. The second contains "copies the selected placeable at the position of the cursor (in the target segment)": an 82% match is proposed. The field itself is not included in either of the segments.

Transit: a 94% match is proposed. If the proposed fuzzy match is accepted, the inline graphic is replaced automatically by the field.

Wordfast: a 97% match is proposed. There is no automatic adaptation.

9.2.3.4 Transposition

Single element modification

The following examples were used:

1. Click the f icon to launch Wordfast
2. f Click on the icon to launch Wordfast

Difference: the inline graphic was moved to the beginning of the segment.

Results

Across: a 93% match is proposed. The inline graphic is part of the second segment. If the proposed fuzzy match is accepted, there is no automatic adaptation.

Déjà Vu: a 98% match is proposed. The inline graphic is excluded from the second segment. If the proposed fuzzy match is accepted, the tag is automatically deleted and a space is added at the beginning of the segment.

Heartsome: a 99% match is proposed. The inline graphic is not clearly displayed. Several tags occur, but none stands for the inline graphic, which seems to have been excluded.

memoQ: a 98% match is proposed. The inline graphic is excluded from the second segment. If the proposed fuzzy match is accepted, there is no automatic adaptation.

MultiTrans: a 100% match is proposed. The inline graphic cannot be transferred to the target segment.

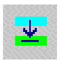
SDL Trados: a 91% match is proposed. The inline graphic is part of the second segment. If the proposed fuzzy match is accepted, there is no automatic adaptation.

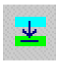
Transit: a 99% match is proposed. The inline graphic is part of the second segment. If the proposed fuzzy match is accepted, the tag is moved to the beginning of the segment. However, the adaptation is only partial because the inline graphic is not followed by a space. In addition, a double space is left where the inline graphic was originally positioned.

Wordfast: a 93% match is proposed. The inline graphic is excluded from the second segment. The proposed fuzzy match does not include the icon, but a placeholder (&'1;) that has to be deleted manually. If the proposed fuzzy match is accepted, there is no automatic adaptation.

Segment length modification

The following examples were used:

1. Ctrl+Alt+Down  copies the selected placeable at the position of the cursor (in the target segment).

2.  Ctrl+Alt+Down copies the selected placeable at the position of the cursor (in the target segment).

Difference: the inline graphic was moved to the beginning of the segment.

Results

Across: a 93% match is proposed. The inline graphic is part of the second segment. If the proposed fuzzy match is accepted, there is no automatic adaptation.

Déjà Vu: a 100% match is proposed. The inline graphic is excluded from the second segment. If the proposed fuzzy match is accepted, the superfluous tag is deleted automatically and a space is added at the beginning of the segment.

Heartsome: a 99% match is proposed. The inline graphic is not clearly displayed and seems to have been excluded.

memoQ: a 98% match is proposed. The inline graphic is excluded from the second segment. If the proposed fuzzy match is accepted, there is no automatic adaptation.

MultiTrans: a 100% match is proposed. The inline graphic cannot be transferred to the target segment.

SDL Trados: a 97% match is proposed. The inline graphic is part of the second segment. If the proposed fuzzy match is accepted, there is no automatic adaptation.

Transit: a 99% match is proposed. The inline graphic is part of the second segment. If the proposed fuzzy match is accepted, the tag is moved to the beginning of the segment. However, the adaptation is only partial because the inline graphic is not followed by a space. In addition, a double space is left where the inline graphic was originally positioned.

Wordfast: no match is proposed even though the minimum fuzzy match was lowered to 50%. The inline graphic is excluded from the second segment.

9.3 Conclusions

9.3.1 Display

Inline graphics are displayed in different ways, see 9.2.2. In Across, SDL Trados and Transit, it is possible to understand that the tag stands for an inline graphic. In other cases, it is possible only with additional effort (Déjà Vu and Heartsome) or not possible at all (memoQ[4]). This might pose a problem for users, in particular if the inline graphic is not the only tag in the segment. Incidentally, it has been noted that – even though the source file is exactly the same – some TM systems (Déjà Vu, Heartsome and memoQ) tend to present more tags than others (Across, SDL Trados and Transit). A specific investigation in this regard is beyond the scope of this thesis.

9.3.2 Segmentation

Table 9.4 shows whether inline graphics are identified as segment delimiters by the TM systems.

	Segmentation?
Across	no
Déjà Vu	yes
Heartsome	no
memoQ	yes
MultiTrans	yes
SDL Trados	no
Transit	no
Wordfast	yes

Table 9.4: Overview: segmentation of inline graphics

MultiTrans, for example, segments the sentence when a graphic of this kind occurs; moreover, the inline graphic is not displayed correctly and cannot be transferred from

[4] The result can nevertheless be seen in the translation preview window.

the source segment to the target segment. Déjà Vu also interprets an inline graphic as a segment end delimiter, however, the sentence fragments can be merged. In some TM systems, if inline graphics occur at the beginning of a segment, they are excluded from the editable segment. This is questionable because the syntax of the target segment may require reordering.

9.3.3 Penalties

The similarity values calculated by the TM systems do not differ strikingly, see table 9.5.

	Addition	**Deletion**	**Replacement**		**Transposition**	
Across	93	93	100	100	93	93
Déjà Vu	98	100	100	100	98	100
Heartsome	99	99	99	99	99	99
memoQ	98	99	100	85	98	98
MultiTrans	100	100	100	-	100	100
SDL Trados	98	98	100	100	91	97
Transit	96	99	100	94	99	99
Wordfast	92	0	96	97	93	0

Table 9.5: Overview: match values

Generally, all TM systems apply penalties up to 15%; only Wordfast does not propose any match in two cases. However, this is presumably due to some sort of bug, as such a high penalty is not justified by the modifications.

100% matches are relatively frequent. However, in the case of MultiTrans, this is the result of poor support of the inline graphics; in fact, the proposed matches must be adapted. On the other hand, other TM systems propose true 100% matches, which are usually the result of automatic adaptations. This is most frequently the case when inline graphics are replaced, but Déjà Vu sometimes also offers such matches when deletions or transpositions have been made. Only Heartsome applies a constant penalty (1%) to all proposed examples. All other TM systems have floating penalties and it is difficult to recognize a clear pattern.

The transposition test set aimed at checking whether the similarity value for the same modification is calculated based on the segment length. SDL Trados and Déjà Vu show a smaller penalty for the longer segment, but the latter case can be explained by the automatic adaptation. Transit, memoQ and Across apply the same penalty. Wordfast and MultiTrans cannot be considered.

9.3.4 Automatic adaptations

As mentioned in 9.2.2, TM systems clearly differ as regards automatic adaptations.[5] Table 9.6 summarizes the results: "partial" indicates that the TM system supports the translator to some extent, who then needs to fine-tune the suggestion.

	Addition	Deletion	Replacement		Transposition	
Across	no	partial	yes	yes	no	no
Déjà Vu	partial	yes	yes	yes	yes	yes
memoQ	no	no	yes	no	no	no
SDL Trados	no	partial	yes	yes	no	no
Transit	partial	partial	yes	yes	partial	partial
Wordfast	no	no	partial	no	no	no

Table 9.6: Overview: automatic adaptations

Déjà Vu is the most accurate TM system. Transit is less accurate so that minor manual modifications are often needed. Both TM systems offer at least a partial adaptation for all examples. Across and SDL Trados are not always able to propose adaptations. Finally, memoQ and Wordfast offer virtually no support.

9.4 Possible improvements

9.4.1 Display and segmentation

The tests show that there are two problems affecting the support of inline graphics: display and segmentation. Firstly, there are many possible displays for inline graphics, but the inline graphic should be immediately identifiable as such: therefore, solutions with placeholders (e.g. memoQ and MultiTrans) may not be clear enough, particularly if not accompanied by a real-time preview. The solutions presented by e.g. Across and Transit seem more straightforward. This problem does not exist if MS Word is used as the editor and DOC files are edited. Secondly, since inline graphics are intended for use in running text, they should not be taken as segment delimiters nor be excluded from the segment if they occur at the beginning. Because of the syntactic differences between languages, the initial position in the source language does not imply the initial position in the target language.

9.4.2 Penalties and automatic adaptations

A fixed penalty for all types of modification concerning inline graphics is a viable solution, provided that it does not prevent retrieval. The penalty values applied should be indepen-

[5] Heartsome and MultiTrans do not offer any automatic adaptation, therefore, they are not discussed.

dent of the length of the segment where the modification occurs because the effort involved in the modification is not length-dependent.

Although not implemented by all TM systems, automatic adaptations can modify fuzzy matches in order to obtain virtually 100% matches.[6] In the following cases, they achieve fairly precise results:

- Inline graphic was deleted.
- Inline graphic was replaced by another inline graphic or similar element.

A deletion may involve minor collateral modifications, e.g. the deletion of spaces. This was not considered by Across, SDL Trados and Transit, so that they only perform partial automatic adaptations. However, as Déjà Vu shows, the implementation of intelligent space management is able to deliver 100% matches.

Replacements are mainly performed correctly. The automatic adaptation should not be limited to replacements by another inline graphic, but could encompass also fields, tags or numbers. This would make the replacement more flexible. Nevertheless, the user should be prompted to check the result.

Automatic adaptations are often unsuccessful in the following two situations:

- The position of the inline graphic was modified.
- The inline graphic was added.

In both cases, the new position of the inline graphic is language-dependent, as explained in 9.2. If the position of an existing inline graphic was modified, automatic adaptation can be successful only if the new position is clearly identifiable. However, the result might be not suitable for all languages, so that a check by the user remains necessary. The usage of additional anchor points, e.g. numbers, would apply only to limited situations and is not likely to deliver sensible improvements.

When additions are made, the TM system can automatically insert the new inline graphic into the fuzzy match, but its correct positioning remains the responsibility of the user. The usefulness of this partial adaptation is a moot point and would require further investigation.

[6] Even if a segment does not need modifications because of automatic adaptations, it shoud not be presented as an ordinary 100% match. Manual revision is still advisable in most cases, as described later on.

Chapter 10

Fields

10.1 Introduction

Although fields can be found in different word processing and DTP programs, the following description focuses on MS Word 2003 and the examples are taken from MS Word documents.

A field is defined as a "set of instructions that you place in a document", (Camarda, 2003, 772). Fields mainly consist of a field name, possibly followed by switch(es)[1] and instructions that control the output of the field. The output of a field function is updatable, e.g. the current date is printed by the DATE field: `{ DATE \@ "d MMMM yyyy" \* MERGEFORMAT }`.

All fields can be displayed in two ways: either as the underlying function between curly brackets or as the output of the function. In MS Word, the output is displayed by default. The function can be displayed by pressing ALT+F9.

Fields can be categorized as follows:[2]

- Result fields: specify instructions that MS Word can use to determine which text to insert in the document.

- Action fields: perform a specific action that does not place new visible text in the document.

- Marker fields: mark text so that MS Word can find it later – for example, to compile an index or table of contents.

Examples:

- Result fields: `{ DATE }`, which inserts the current date into the document.

[1] Switches are field attributes that add special information to a field. Although they appear in several examples, they are not discussed. Refer to Camarda (2003) or Brodmüller-Schmitz (2003) for a detailed description.

[2] Categorization and examples taken from (Camarda, 2003, 772-773) with slight adaptations, see also Brodmüller-Schmitz (2003).

- Action fields: `{ HYPERLINK }`; when clicked on, MS Word jumps to a location indicated in the field.

- Marker fields: `{ XE }`, which marks index entries that can be compiled into indexes.

Another classification of fields is based on their content:[3]

- User information, e.g. USERNAME

- Date and time, e.g. DATE, TIME

- Document information, e.g. AUTHOR, FILENAME

- Formulae, e.g. =SUM

- Indexes, e.g. INDEX, TOC, XE

- Auto-numbering, e.g. LISTNUM, PAGE, SEQ

- Links, references and objects, e.g. EMBED, HYPERLINK, INCLUDEPICTURE, LINK, REF

Because of their relevance for the subsequent tests, references and objects are described in more detail.

10.1.1 References

Reference is a general term that includes cross-references and hyperlinks.

Cross-references, represented by the `{ REF bookmark}`[4] field, insert the content of a bookmark (heading, table, graphic etc.) defined in the active document.

Hyperlinks, represented by the `{ HYPERLINK "(...)" }` field, are references to another element in the document, to another (local or network) file or to URLs. E-mail addresses can also be included: `{ HYPERLINK "mailto:(...)" }`. Hyperlinks have a particular style (usually underlined and changing color once clicked).

In the case of hyperlinks, it is important to distinguish between:

- the hyperlink destination, e.g. `C:\\Documents and Settings\\student\\Desktop \\Install.log`.

- the display text, e.g. “install file”.

[3] Classification adapted from (Brodmüller-Schmitz, 2003, 557-642).

[4] “REF” can be omitted if the name of the bookmark cannot be confused with the name of another field.

10.1.2 Objects

Objects are data – e.g. tables, diagrams, graphics and formulae – created in one application and linked or embedded in another, e.g in a text document. The prerequisite for embedding/linking is that the application used to create the object supports OLE technology (Object Linking and Embedding).

There is an important difference concerning the storage location between embedded and linked objects. Embedded objects are a copy of the original file inserted in MS Word. Any modification is saved in the MS Word file and does not change the original file. Embedded objects are EMBED fields, e.g. `{ EMBED PBrush  }`.

If an object is linked, the MS Word file contains only a link to the location of the original file, which is stored separately. Any modification of the original file affects also the object in the MS Word file. Linked objects are LINK fields, e.g. `{ LINK Paint.Picture "C:\\Documents and Settings\\student\\Desktop\\Test.bmp""" \a \f 0 \p }`.

Double-clicking on an object (either linked or embedded) in MS Word opens an external application for editing it.

10.2 Tests

The tests are aimed at checking the following issues, see also 2.2.2 and 2.4.1.6:

- How are fields displayed?[5]
- Are they modifiable?
- How do transpositions, replacements, additions and deletions affect the similarity value?
- Are automatic adaptations applied?

From the point of view of user effort, deletions and transpositions are less problematic than replacements and additions because no new elements are inserted. Replacements and deletions are manageable by automatic adaptations and it can be expected that some are applied. All these considerations assume that the modification does not involve plain text as is the case with e.g. hyperlinks. Consequently, for the different categories of fields, diverging results can be expected as regards penalties and the effectiveness of automatic adaptations. See 10.4 for more information on the recommended handling of fields.

It is virtually impossible to test all fields, therefore, a representative selection was made. The input file used for this test set is the English user manual of Wordfast (Champollion (2008b)), available in MS Word format. The reasons for selecting this document are explained in 2.4.1.

Several examples were adapted in order to better suit the aims of the test. Consequently, they do not contain exactly the same wording in the original document. The presented

[5] Display and editing difficulties are known issues, see e.g. (Reinke and Höflich, 2002, 284).

modifications are constructed. They are not intended to be meaningful from a semantic point of view, therefore, they are to be considered exclusively from a formal point of view.

10.2.1 TM system settings

10.2.1.1 Versions

TM system	**Version**
Across Standalone Personal Edition	4.00
Déjà Vu X Professional	7.5.303
Heartsome Translation Studio Ultimate	7.0.6 2008-09-12S
memoQ Corporate	3.2.17
MultiTrans	4.3.0.84
SDL Trados 2007 Freelance	8.2.0.835
STAR Transit XV Professional	3.1 SP22 631
Wordfast	5.53

Table 10.1: TM systems used in the tests

10.2.1.2 Customizability

In this section, the availability of the following features will be discussed.

- Is the penalty value visible and customizable?
- Can automatic adaptations be activated/deactivated/customized?
- Do the format filter settings influence the treatment of fields and is the filter customizable?

Across: the remarks in 8.2.1.2 concerning penalties and automatic adaptations apply.

The default file format filter settings were left unchanged. Under Tools > System Settings > Document Settings > Word > Advanced it is possible to define how editable fields are treated; as described in 10.2.2 or with the editable content extracted in a separate segment.

Déjà Vu: it is not possible to view or customize the penalty applied to differences concerning fields.

For fields treated as tags ("embedded codes"), see 10.2.2, automatic conversion is applied, see (ATRIL, 2003, 196). This conversion cannot be deactivated.

Under Tools > Options > Filters > doc > Default Filter Options, it is possible to activate the option Import Field Result, which is deactivated by default. For these tests, it was left deactivated because it would otherwise extract the text from result fields such as a date or title.

Heartsome, memoQ: for fields treated as tags, the remarks in 8.2.1.2 concerning penalties and automatic adaptations apply.

The file format filter does not present any option relevant to fields.

MultiTrans: as fields are excluded from the segment, there is no corresponding option concerning automatic adaptation or the penalties applied.

Since MS Word is used as the editor, no file format conversion is necessary.

SDL Trados: TagEditor (8.2.835) was used as the editor instead of MS Word.

In Translator's Workbench, under OPTIONS > TRANSLATION MEMORY OPTIONS > PLACEABLE DIFFERENCE PENALTY[6], the penalty value can be set. In these tests, it was left at 2% (default value). However, the same remarks presented in 9.2.1.2 apply.

There is no option concerning the automatic adaptation of fields (e.g. under FILE > SETUP > SUBSTITUTIONS), but the following description seems to also apply to fields, as long as they are treated as tags: "If the current source segment is the same as a source segment from translation memory apart from its variable elements, Translator's Workbench produces a 100% match", SDL (2007). The automatic adaptation is a built-in feature.

Moreover, under TOOLS > OPTIONS > GENERAL, the option STRIP TAGS FROM FUZZY MATCHES IF NO TAGS APPEAR IN THE SOURCE SEGMENT is activated in order to "remove unwanted tags from fuzzy matches when no tag appears in the source segment [...]. Tags are removed when you insert the fuzzy match target segment into the document [...]", SDL (2007).

Finally, the file format filter does not present any option relevant to fields.

Transit: it is not possible to view or customize the penalty applied to differences concerning fields.

As regards automatic adaptations, since several fields are treated as tags, the following applies:

> If a source-language sentence in the reference material and a source-language sentence in the current project differ only in terms of numbers and/or formatting, Transit only uses the text of the target-language reference translation. The numbers and tags are taken from the current source-language file [...]. STAR (2005)

In addition, "if there is a different number of tags and digits, Transit places the 'excess' tags and digits at the end of the sentence", STAR (2005).[7]

The file format filter does not present any option relevant to fields.[8]

[6] In SDL Trados terminology, "placeables" include some fields: "[...] non-translatable elements occurring within segments, such as tags, graphics, date or name fields, formulas and so on. These elements are referred to as placeables", SDL (2007).

[7] This partial automatic adaptation can be prevented if the option SIMPLIFIED EXCEPTION HANDLING is activated in the project settings. This option is activated by default, i.e. no partial automatic adaptation is carried out. However, it seems to apply exclusively to the pretranslation, as interactive fuzzy matches are partially adapted.

[8] The option INDEX ENTRIES POSITION refers exclusively to index entries and determines their display position.

Wordfast: fields are considered as placeables by default, see (Champollion, 2008b, 25). It is not possible to view or customize the penalty applied to differences concerning fields.

For automatic adaptations, "[...] Wordfast uses a substitution algorithm to update the proposed segment and bring it closer to an exact match. The elements that are updated or substituted are [...] placeables", (Champollion, 2008b, 25) so that "Wordfast proposes [the] segment as 100% [...]", (Champollion, 2008b, 17). This default behavior cannot be changed.

Since MS Word is used as the editor, no file format conversion is necessary.

10.2.2 General remarks on TM system behavior

The display and the editability of fields are discussed in this section. As the findings vary depending on the type of field involved, three categories were isolated from among the fields tested.

1. URLs/references
2. Auto-numbering/dates
3. Objects

It is unlikely that other fields are treated in a completely different way so that this grouping essentially provides a complete overview.

Display

Fields are displayed in different ways, depending on the field type. Some customizations are possible, but the examples in table 10.2 are based on standard display settings.[9] MultiTrans is not included because fields segment the text and are never treated as inline elements in its Translation Agent.

Across: the display of fields depends on their editability. Editable fields such as hyperlinks are displayed as a rectangle with light green background color. Non-editable fields are displayed as a rectangle with gray background color. Within the rectangle it is usually possible to see the output of the field (e.g. a number), but not the underlying function. Although objects are non-editable fields, neither their output nor their underlying function can be viewed.

Déjà Vu: the output of fields containing URLs and references is displayed as plain text between tags. Fields containing auto-numbering and dates as well as embedded objects are displayed as tags.

Heartsome, memoQ: the output of fields containing URLs and references is displayed as plain text between tags. Auto-numbering fields are displayed as one tag followed by the

[9] For SDL Trados, the Complete Tag Text display was used throughout the tests. For Transit, view no. 5 ("As 2.7") was chosen.

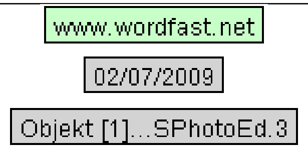
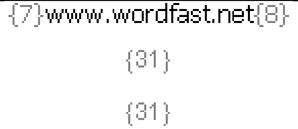
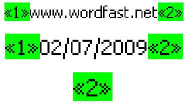
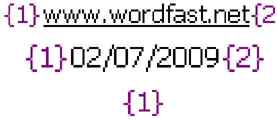
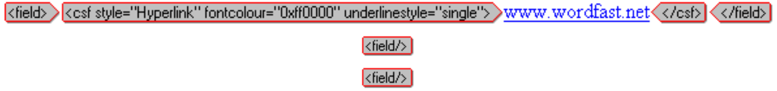
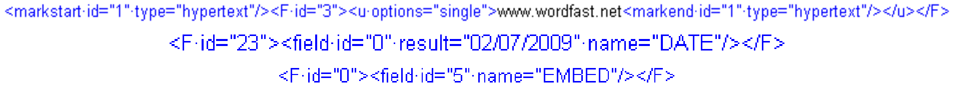
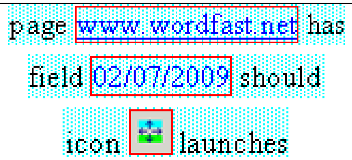
	Field type	Display
Across	hyperlinks/references	 www.wordfast.net
	dates/numberings	02/07/2009
	objects	Objekt [1]...SPhotoEd.3
Déjà Vu	hyperlinks/references	 {7}www.wordfast.net{8}
	dates/numberings	{31}
	objects	{31}
Heartsome	hyperlinks/references	 «1»www.wordfast.net«2»
	dates/numberings	«1»02/07/2009«2»
	objects	«2»
memoQ	hyperlinks/references	 {1}www.wordfast.net{2}
	dates/numberings	{1}02/07/2009{2}
	objects	{1}
SDL Trados	hyperlinks/references	 <field> <csf style="Hyperlink" fontcolour="0xff0000" underlinestyle="single"> www.wordfast.net </csf> </field>
	dates/numberings	<field/>
	objects	<field/>
Transit	hyperlinks/references	 <markstart·id="1"·type="hypertext"/><F·id="3"><u·options="single">www.wordfast.net<markend·id="1"·type="hypertext"/></u></F>
	dates/numberings	<F·id="23"><field·id="0"·result="02/07/2009"·name="DATE"/></F>
	objects	<F·id="0"><field·id="5"·name="EMBED"/></F>
Wordfast	hyperlinks/references	 page www.wordfast.net has
	dates/numberings	field 02/07/2009 should
	objects	icon launches

Table 10.2: Display of fields

output in plain text. Dates are displayed as plain text between tags, in the same way as hyperlinks and references. Objects are displayed as one tag only.

SDL Trados: the output of fields containing URLs and references is shown as plain text between tags. In addition, the address to which the hyperlink or reference points is presented in the segment that follows, as plain text for hyperlinks, or as a tag for references. For auto-numbering, dates and embedded objects, a tag is presented in the segment. The corresponding function is presented as tag in the segment that follows.

Transit: the output of fields containing URLs and references is displayed between tags as plain text. Auto-numbering, dates and objects are displayed as tags only.

Wordfast: fields are recognized as placeables in MS Word and are highlighted by means of a red box.

Editability

The editability of fields during the translation process allows for correct localization of the source text. Editability can apply to the field function itself or to the field output. TM systems apply different strategies depending on the field type involved. Table 10.3 presents an overview of the results. MultiTrans is not included because fields are never shown in its Translation Agent; the content of fields can be edited directly in MS Word. Wordfast is not included because MS Word is used as the editor and any modification can be performed directly.

	Field type	**Output**	**Function/destination**
Across	hyperlinks/references	yes	yes
	dates/numberings	no	no
	objects	-	no
Déjà Vu	hyperlinks/references	yes	yes
	dates/numberings	no	yes
	objects	-	yes
Heartsome	hyperlinks/references	yes	no
	dates/numberings	yes	no
	objects	-	no
memoQ	hyperlinks/references	yes	no
	dates/numberings	yes	no
	objects	-	no
SDL Trados	hyperlinks/references	yes	yes/no
	dates/numberings	no	no
	objects	-	no
Transit	hyperlinks/references	yes	no
	dates/numberings	no	no
	objects	-	no

Table 10.3: Editability of fields

Across: fields containing URLs and references are editable. By right-clicking on the rectangle, it is possible to modify the displayed text and the destination to which it points.

Fields containing auto-numbering and dates as well as embedded objects cannot be modified.

Déjà Vu: fields containing URLs and references are editable. The output can be modified directly. The field instructions are contained in the tags and can be modified as well by right-clicking the number and selecting Display Code or pressing Shift+F6.

Field instructions contained in the tags of auto-numbering, dates and embedded objects can be displayed and modified in the same manner.

Heartsome, memoQ: independently of the type of field, it is never possible to edit its underlying function. On the other hand, the field output is editable, not only for URLs and references, but also for dates and auto-numbering. For embedded objects the output is not editable.

SDL Trados: the output of fields containing URLs or references is directly editable. In addition, an extra segment presents the address to which the hyperlink points and is editable. For references, this extra segment only contains a tag and is skipped.

For auto-numbering, dates and embedded objects, neither the output not the underlying functions are editable.[10]

Transit: irrespective of the type of field, it is never possible to edit its underlying function, even if the tag protection is deactivated.

The output of fields containing URLs and references is directly editable.

10.2.3 Test suite

10.2.3.1 Addition

Addition of result field

The following examples were used:

1. The following "Today's date" field should toggle between the two views.

2. The following "Today's date" field { DATE * MERGEFORMAT } should toggle between the two views.
   displays
   The following "Today's date" field 14/07/2009 should toggle between the two views.

Difference: a date field was added.

Results

Across: a 93% match is proposed. The 7% penalty is the sum of 5% due to the difference in space and 2% due to the inline element (field). There is no automatic adaptation.

[10] This is not influenced by the settings under Tools > Options > Protection > Tag protection.

Déjà Vu: a 99% match is proposed. If the fuzzy match is accepted, the tag containing the field is automatically added at the end of the segment.

Heartsome, memoQ: an 83% match is proposed. There is no automatic adaptation.

MultiTrans: the sentence is split into two segments by the field. For each fragment segment a 50% match is proposed. There is no automatic adaptation.

SDL Trados: a 98% match is proposed. There is no automatic adaptation.

Transit: a 90% match is proposed. If the fuzzy match is accepted, the field is added automatically at the end of the segment.

Wordfast: a 96% match is proposed. There is no automatic adaptation.

Addition of hyperlink

The following examples were used:

1. Every line is actually a Find and Replace command, exactly as described in Pandora's Box.

2. Every line is actually a Find and Replace command, exactly as described in Pandora's Box { HYPERLINK \l "FR_PB" }.
   displays
   Every line is actually a Find and Replace command, exactly as described in Pandora's Box FR command section.

Difference: a hyperlink was added at the end of the segment.

Results

Across: a 93% match is proposed. There is no automatic adaptation.

Déjà Vu: an 82% match is proposed. If the fuzzy match is accepted, the tags are inserted at the end of the fuzzy match. However, the hyperlink text has to be entered manually and the tags have to be repositioned.

Heartsome: an 80% match is proposed. There is no automatic adaptation.

memoQ: an 86% match is proposed. There is no automatic adaptation.

MultiTrans: a 100% match is proposed. The hyperlink that has been added splits the sentence into two segments. A perfect match is proposed because the first subsegment is distinguished from the first segment only by the full stop. As explained in 11.2, the option STRICT PUNCTUATION MATCHING is not applied to matches coming from the propagation memory so that an incorrect perfect match is proposed.

SDL Trados: an 85% match is proposed. There is no automatic adaptation.

Transit: a 63% match is proposed. If the fuzzy match is accepted, the tags are added automatically at the end of the segment.

Wordfast: a 97% match is proposed. There is no automatic adaptation.

10.2.3.2 Deletion

Deletion of hyperlink

The following examples were used:

1. The Wordfast website download page { HYPERLINK "http://www.wordfast.net" } has training guides that are illustrated, step-by-step methods for beginners.
   displays
   The Wordfast website download page www.wordfast.net has training guides that are illustrated, step-by-step methods for beginners.

2. The Wordfast website download page has training guides that are illustrated, step-by-step methods for beginners.

Difference: the hyperlink containing the URL and the trailing space were deleted.

Results

Across: a 93% match is proposed. If the proposed fuzzy match is accepted, the superfluous object containing the URL is automatically deleted. However, a double space remains.

Déjà Vu: an 83% match is proposed. If the proposed fuzzy match is accepted, the tags embracing the URL are deleted automatically. However, the URL content represents a textual difference and has to be deleted manually.

Heartsome: an 84% match is applied. If the proposed fuzzy match is accepted, the tags embracing the URL are deleted automatically. However, the URL content represents a textual difference and has to be deleted manually.

memoQ: an 85% match is proposed. There is no automatic adaptation.

MultiTrans: since the first segment is split into two chunks by the hyperlink, no match is proposed.

SDL Trados: a 91% match is proposed. If the proposed fuzzy match is accepted, the tags embracing the URL are deleted automatically. However, the URL content represents a textual difference and has to be deleted manually.

Transit: a 74% match is applied. If the proposed fuzzy match is accepted, the tags embracing the URL are deleted automatically. However, the URL content represents a textual difference and has to be deleted manually.

Wordfast: a 97% match is applied. If the proposed fuzzy match is accepted, the superfluous placeable containing the URL and one space are deleted so that the target segment does not need further modification.

Deletion of object

The following examples were used:

1. The Ctrl+Alt+N shortcut or the icon { EMBED MSPhotoEd.3 } launches the Reference search from a document

   displays

   The Ctrl+Alt+N shortcut or the icon ✥ launches the Reference search from a document

2. The Ctrl+Alt+N shortcut or the icon launches the Reference search from a document

Difference: the embedded object (icon) was deleted.

Results

Across: a 93% match is proposed. If the fuzzy match is accepted, the object is automatically deleted, but a double space remains.

Déjà Vu, Transit: a 99% match is proposed. The object is automatically deleted, but a double space remains.

Heartsome: a 100% match is proposed. The superfluous tag and space are deleted automatically.

memoQ: a 98% match is proposed. There is no automatic adaptation.

MultiTrans: a 53% match is proposed. Since the first segment is split into two chunks by the object, the similarity value is very low.

SDL Trados: a 98% match is proposed. The object is automatically deleted, but a double space remains.

Wordfast: a 97% match is proposed. The superfluous tag and space are deleted automatically.

10.2.3.3 Replacement

Replacement of field

The following examples were used:

1. Figure { SEQ Abbildung * ARABIC }
   displays
   Figure 1

2. Figure { SEQ Tabelle * ARABIC }
   displays
   Figure 1

The test document was created with a German version of MS Word. Therefore, the field names are in German: *Abbildung* is figure, *Tabelle* is table.

Difference: the auto-numbering field for figures was replaced by the auto-numbering field for tables.

Results

Across, memoQ, SDL Trados, Transit, Wordfast: a 100% match is proposed. The field is replaced automatically.

Déjà Vu, MultiTrans: a 100% match is proposed. The field containing the autonumbering is excluded from the segment. Therefore, the two segments are considered identical.

Heartsome: a 100% match is proposed. The exported document contains only empty curly brackets in both segments. This type of field seems to be poorly supported.

Replacement of hyperlink

The following examples were used:

1. Of course, you can check { HYPERLINK `"http://www.wordfast.net"` } from time to time or join a mailing list to see if an upgrade has been released.
   displays
   Of course, you can check www.wordfast.net from time to time or join a mailing list to see if an upgrade has been released.

2. Of course, you can check { HYPERLINK `"http://www.wordfast.net/index.php?whichpage=faqbuying"` } from time to time or join a mailing list to see if an upgrade has been released.
   displays
   Of course, you can check www.wordfast.net/index.php?whichpage=faqbuying from time to time or join a mailing list to see if an upgrade has been released.

Difference: the hyperlink in the second segment points to another URL.

Results

Across, Wordfast: a 100% match is proposed. The hyperlink is replaced automatically.

Déjà Vu, memoQ: an 86% match is proposed. Since the hyperlink is presented as plain text between tags, after accepting the fuzzy match it is necessary to adapt the hyperlink text.

Heartsome: a 78% match is proposed. Since the hyperlink is presented as plain text between tags, after accepting the fuzzy match it is necessary to adapt the hyperlink text.

MultiTrans: since the first segment is split into two chunks by the hyperlink, and the hyperlink is excluded from the editable text, no match is proposed.

SDL Trados: a 95% match is proposed. Since the hyperlink is presented as plain text between tags, after accepting the fuzzy match it is necessary to adapt the hyperlink text. The new URL is presented in a subsequent segment where it is possible to adapt the address to which the hyperlink points.

Transit: a 67% match is proposed. Since the hyperlink is presented as plain text between tags, after accepting the fuzzy match it is necessary to adapt the hyperlink text.

10.2.3.4 Transposition

Transposition of hyperlinks

The following examples were used:

1. A more complete list of TMX-compliant language codes can be found on the { HYPERLINK `"http://www.lisa.org"` } web site or at { HYPERLINK `"http://www.wordfast.net/html/lang_frame.html"` }.
   displays
   A more complete list of TMX-compliant language codes can be found on the www.lisa.org web site or at http://www.wordfast.net/html/lang_frame.html.

2. A more complete list of TMX-compliant language codes can be found on the { HYPERLINK `"http://www.wordfast.net/html/lang_frame.html"` } web site or at { HYPERLINK `"http://www.lisa.org"` }.
   displays
   A more complete list of TMX-compliant language codes can be found on the http://www.wordfast.net/html/lang_frame.html web site or at www.lisa.org.

Difference: the two hyperlinks were swapped.

Results

Across: a 100% match is proposed. The two objects containing the hyperlinks are automatically swapped.

Déjà Vu: no match is proposed.

Heartsome: a 94% match is proposed. There is no automatic adaptation.

memoQ: an 81% match is proposed. There is no automatic adaptation.

MultiTrans: since the first segment is split at the point of the hyperlinks, and the hyperlinks are always excluded from the editable text, no match is proposed.

SDL Trados: a 94% match is proposed. There is no automatic adaptation. The URL addresses are proposed for translation in subsequent segments.

Transit: a 99% match is proposed. If the fuzzy match is accepted, the hyperlink tags are positioned at the beginning and at the end of the segment respectively.

Wordfast: a 95% match is proposed. There is no automatic adaptation.

Transposition of object

The following examples were used:

1. The Ctrl+Alt+N shortcut or the icon { EMBED MSPhotoEd.3 } launches the Reference search from a document
   displays
   The Ctrl+Alt+N shortcut or the icon ✥ launches the Reference search from a document

2. The Ctrl+Alt+N shortcut or the { EMBED MSPhotoEd.3 } icon launches the Reference search from a document
   displays

   The Ctrl+Alt+N shortcut or the ✥ icon launches the Reference search from a document

Difference: the object was repositioned.

Results

Across: a 93% match is proposed. There is no automatic adaptation.

Déjà Vu, Wordfast: a 98% match is proposed. There is no automatic adaptation.

Heartsome: a 100% match is proposed. There is no automatic adaptation. Therefore, the 100% match is in fact wrong. The difference is marked in the editing environment, but no penalty is applied.

memoQ: a 99% match is proposed. There is no automatic adaptation.

MultiTrans: since the sentence is split by the object into two subsegments, it is not possible to compare the two segments as a whole, but only the subsegments. For each subsegment of the second segment, an 88% match is proposed.

SDL Trados: a 96% match is proposed. There is no automatic adaptation.

Transit: a 99% match is proposed. If the fuzzy match is accepted, the tag is placed automatically at the end of the segment.

10.3 Conclusions

Field display and editability were described in 10.1 and will not be discussed further. Some general remarks on the support of fields can be made. In Déjà Vu, fields often cause a sentence to be split into two segments, as already observed in 9.2, and this might hinder correct translation. This problem can be solved by manually joining the segments. In some cases this solution is not viable, e.g. when the field occurs at the end of a sentence. In the case of MultiTrans, fields always segment the sentence and the fragments cannot be joined together. Additionally, the field is always excluded and has to be edited directly in MS Word.

10.3.1 Penalties

Table 10.4 summarizes the similarity values calculated by the TM systems. MultiTrans is not listed because fields were always excluded.

	Addition		Deletion		Replacement		Transposition	
	1	2	3	4	5	6	7	8
Across	93	93	93	93	100	100	100	93
Déjà Vu	99	82	83	93	(100)§	86	0	98
Heartsome	83	80	84	100	(100)§	78	94	100
memoQ	83	86	85	98	100	86	81	99
SDL Trados	98	85	91	98	100	95	94	96
Transit	90	63	74	99	100	67	99	99
Wordfast	96	97	97	97	100	100	95	98

Table 10.4: Overview: match values

§: the field is excluded from the segment. The match value refers only to the textual part.

These results show that there is generally no fixed penalty for differences concerning fields. The only exception is Across which applies a penalty of 7% irrespective of the type of modification or field modified. There are also a number of erratic match values that prove that field support is not always flawless. In example 7, Déjà Vu does not propose any fuzzy match. In example 8, Heartsome proposes a 100% match although the segments differ and there is no automatic adaptation.

Across, SDL Trados and Wordfast tend to apply lower penalties than Déjà Vu, Heartsome, memoQ and Transit. Transit applies the highest penalties in three out of eight examples. Automatic adaptations increase the similarity value.

A major factor influencing the similarity value is that most TM systems (Déjà Vu, Heartsome, memoQ, SDL Trados and Transit) present the output of hyperlinks as plain text. This textual difference is considered in the calculation of the similarity value. This explains the higher penalties applied to examples 2 and 3, which deal with the addition and the deletion of a hyperlink, respectively.

Heartsome and memoQ also present the output of dates and auto-numbering as plain text. This results in higher penalties for example 1 in comparison with all other TM systems. However, it does not make sense to change the output of result fields (see 10.1) that automatically generate text because any output modification is discarded after the first field update.

10.3.2 Automatic adaptations

Table 10.5 summarizes the automatic adaptations made by the TM systems. MultiTrans is not listed because fields were always excluded.

	Addition		**Deletion**		**Replacement**		**Transposition**	
	1	2	3	4	5	6	7	8
Across	no	no	p	p	yes	yes	yes	no
Déjà Vu	p	p	p	p	-	no	no	no
Heartsome	no	no	p	yes	-	no	no	no
memoQ	no	no	no	no	yes	no	no	no
SDL Trados	no	no	p	p	yes	no	no	no
Transit	p	p	p	p	yes	no	no	no
Wordfast	no	no	yes	yes	yes	yes	no	no

Table 10.5: Overview: automatic adaptations

All TM systems propose automatic adaptations at least once. Some TM systems (Transit and Wordfast) apply automatic adaptations more often than others (Heartsome and memoQ). The type of modification influences the capacity to adapt automatically. Additions and transpositions usually cannot be adapted automatically, whereas this is more likely to be the case with deletions and replacements.

When additions are made, only Déjà Vu and Transit can make partial adaptations. In the case of transpositions, only Across manages in one example to adapt fields, most likely because two fields are swapped so that the transposition resembles a replacement. There is a straightforward explanation for this: additions and transpositions involve changing the field position. Since this position is language dependent, any adaptation without linguistic knowledge may be incorrect. At most, the new field (for additions) can be inserted into the segment, but the user has to take care to position it correctly.

Deletions are easier to manage. Most TM systems automatically delete the superfluous field, but fail to make additional modifications. In most cases, double spaces remain after deletion and have to be removed manually. Moreover, depending on the TM system, some fields consist of plain text between tags; these tags are automatically deleted, but the text is not. Therefore, these adaptations seldom produce a match that does not need further modifications.

Replacements are most likely to be automatically adapted because no repositioning is necessary. However, automatic adaptation is usually successful only if the new field belongs to the same category as the old one.

10.4 Possible improvements

10.4.1 Display and editability

A problem affecting some TM systems (mainly Déjà Vu and MultiTrans) is that fields are considered segment delimiters and the resulting segmentation has to be corrected manually. Correct segmentation is therefore desirable.

Editability is also a cause for concern. As explained in 10.1, it is crucial to distinguish between the field function and its output. The output should be localizable in most cases, for example in a hyperlink that has to display a different website in each case, adapted for the country in question (e.g. `www.google.de` versus `www.google.it`). However, editability is not suitable for output fields where the output depends on the function (e.g. DATE and TIME). Any modification of the output is lost as soon as the field is updated. Consequently, presenting the output of date and auto-numbering fields as plain text for editing (as is the case with memoQ and Heartsome) can be considered misleading.

Editability of the function itself would be necessary for localization purposes. An example is given by the website hyperlink above: changing the output is only partial localization if the field still points to the old URL. In this case, it is essential to also change the field function itself. Currently, Heartsome, memoQ and Transit do not offer this possibility. The result is confusing because the translated document displays the localized website, but, if the hyperlink is followed, another website is shown.

For result fields, the localization of the field function should not be excluded altogether. For example, the output of the date field should conform to locale-dependent conventions, e.g. 10.12.2003 (dd.mm.yyyy) in German should be rewritten as 12-10-2003 (mm-dd-yyyy) in US English. On the other hand, adaptation of the field function requires format-specific knowledge.

Editability is closely related to the display: different approaches are possible, see 10.2.2. Fields with translatable output can be displayed with their output text between a tag pair. If also the function of the field has to be adapted, as for URLs, SDL Trados and Across propose two different approaches. SDL Trados presents the URL after the output text. However, the relationship with the output text presented in the previous segment might not be clear. On the other hand, Across presents a dedicated mask, opened by right-clicking the hyperlink field. In the mask, output text and URL can be edited. This approach has the disadvantage that the user has to open it actively. Apart from personal preferences, both approaches have the key advantage that the full editability is ensured.

Where the output text is generated automatically, a single tag is sufficient as is the case with auto-numbering fields e.g. in SDL Trados and Transit. Function editability can be implemented as described for fields with translatable output.

10.4.2 Penalties

The penalties applied by TM systems differ and depend, among other factors, on automatic adaptations, see 10.4.3. On the one hand, a fixed penalty for all types of modification concerning fields without translatable output is a viable solution. On the other hand, modifications of fields with translatable output, e.g. hyperlinks and references, should be applied additional penalties because of the textual differences involved.

Penalty values should be independent of the length of the segment where the modification occurs – at least for fields without translatable output. The number of modified fields should be taken into account in order to reflect the adaptation effort for the user.

10.4.3 Automatic adaptations

As discussed in 10.3.2, in the case of additions and transpositions automatic adaptations are problematic because the position of the fields cannot be set without linguistic knowledge. Users need to check the result of this kind of adaptation so that a 100% match does not seem generally justified.

In the case of additions, the new field can be inserted automatically so that it has to be repositioned (partial adaptation), but the usefulness of this is not clear and would require further investigation. For transpositions, any automatic modification does not seem to facilitate the translation. Field replacements can be adapted automatically, but only where the field is replaced with a new one of the same category.

Deletions are to some extent the type of modification that is most suited to automatic adaptation. A prerequisite for successful automatic adaptation is intelligent field management. This means that not only the field is deleted, but also spaces are adapted automatically so that double spaces do not need to be manually corrected as is currently the case in several TM systems. Furthermore, if e.g. a hyperlink field is deleted, and the output proposed for translation is plain text, the text should also be deleted.

Chapter 11

Common punctuation marks

11.1 Introduction

In this chapter, the following punctuation marks will be dealt with:

- Full stop
- Question mark
- Exclamation mark
- Semicolon
- Colon
- Comma

The full stop is the most frequent punctuation mark in the corpus (even if its occurrences in abbreviations are excluded). Full stops, question marks and exclamation marks usually indicate the end of a sentence. Semicolons and colons are often sentence delimiters too. In the corpus, the comma is the most common punctuation mark that occurs within sentences and is therefore included in this chapter.

Other punctuation marks and word delimiters such as spaces, quotes and dashes are described in chapter 12. Further, less common punctuation marks such as brackets, ellipses etc. are not tested.

The usage of punctuation marks is locale-dependent. For example, an insight into the differences between English and Spanish is given by Montserrat and Balonés (2009), who stress that punctuation also determines the quality and appropriateness of a translation for the locale in question. The importance of correct punctuation for translation quality is also acknowledged by Schulz (1999) and (Way, 2010b, 21). In addition, punctuation errors can be particularly problematic when texts are revised by a proofreader, see Groves (2008).

11.2 Tests

The tests are aimed at checking the following issues, see also 2.2.2 and 2.4.1.6:

1. Is the difference between the TM segment and source text segment recognized?
2. Does the type of the modified punctuation mark affect the similarity value?
3. Does the position of the modified punctuation mark affect the similarity value?
4. Does the segment length affect the similarity value?
5. Are automatic adaptations applied?

Tests concerning the handling of punctuation marks by TM systems can be found in (Nübel and Seewald-Heeg, 1999b, 21–22) and (Guillardeau, 2009, 76); in this thesis, they have been considerably expanded. The presented modifications are constructed. They are not intended to be meaningful from a semantic point of view, therefore, they are to be considered exclusively from a formal point of view.

The penalty for differences in punctuation marks should be low and reflect only a minor modification. The calculation of the penalty value should be independent of the segment length and of the type of punctuation mark. However, it is more difficult to define whether the position of the modified punctuation mark should be taken into account too. Nevertheless, the modification of punctuation marks can be easily managed using automatic adaptations when the punctuation marks occur at the end of the segment, and it can be expected that some automatic adaptations are applied. See 11.4 for more information on the recommended handling of differences in punctuation marks.

11.2.1 TM system settings

11.2.1.1 Versions

TM system	**Version**
Across Standalone Personal Edition	4.00
Déjà Vu X Professional	7.5.303
Heartsome Translation Studio Ultimate	7.0.6 2008-09-12S
memoQ Corporate	3.2.17
MultiTrans	4.3.0.84
SDL Trados 2007 Freelance	8.2.0.835
STAR Transit XV Professional	3.1 SP22 631
Wordfast	5.53

Table 11.1: TM systems used in the tests

11.2.1.2 Customizability

In this section, the availability of the following features will be discussed.

- Is the penalty value visible and customizable?
- Can automatic adaptations be activated/deactivated/customized?

Across: under Tools > Profile Settings > CrossTank > Advanced settings, the default penalty for Different punctuation is 5%. However, it is not possible to check which punctuation marks are included; the online help does not provide more information either.

For automatic adaptations, no option is available.

Déjà Vu: the penalty for differences in punctuation is neither visible nor customizable.

For automatic adaptations, the option Control leading and trailing symbols under Tools > Options > General > Conversions is active and it automatically inserts "symbols such as punctuation marks or spaces at the beginning or end of a sentence", ATRIL (2008), according to the source segment.

Heartsome, SDL Trados, Transit, Wordfast: there are no settings concerning either the penalty for differences in punctuation marks or corresponding automatic adaptation.

memoQ: the penalty for differences in punctuation is neither visible nor customizable.

For automatic adaptations, the option Adjust fuzzy hits and inline tags under Tools > Options > TM defaults, user info > Adjustments is activated by default and enables "on-the-fly adjustment of [...] ending punctuation marks [...] within translation memory hits with less than 100% match rate", Kilgray (2008).

MultiTrans: differences concerning punctuation marks are considered if the option Strict Punctuation Matching is activated under MultiTrans > Options > TextBase. If the option is active and the source segments in the TextBase and in the text differ only by a punctuation mark, a 1% penalty is applied. However, this option is ignored for matches coming from the propagation memory: in this case, 100% matches are proposed.

To avoid this problem, the test procedure for MultiTrans had to be adapted: only the first source segment was translated and added to the TextBase. The propagation memory was disabled. The translation of the remaining segments was performed in a separate session so that the matches came from the TextBase.

For automatic adaptations, no option is available.

11.2.2 General remarks on TM system behavior

Automatic adaptations

Table 11.2 presents general findings on automatic adaptations. Déjà Vu and memoQ will be discussed in detail in 11.2.3.

	Automatic adaptations?
Across	No
Déjà Vu	Generally yes
Heartsome	No
memoQ	Partially
MultiTrans	No
SDL Trados	No
Transit	No
Wordfast	No

Table 11.2: Support of automatic adaptations

Across, Heartsome, MultiTrans, SDL Trados, Transit, Wordfast: no automatic adaptation is ever made.

Déjà Vu: automatic adaptations are regularly made, with a few exceptions.

memoQ: automatic adaptations are made only to some extent.

11.2.3 Test suite

11.2.3.1 Different punctuation marks

1. Taste 5 Sekunden drücken, um diese Maske aufzurufen.
2. Taste 5 Sekunden drücken, um diese Maske aufzurufen:
3. Taste 5 Sekunden drücken, um diese Maske aufzurufen!
4. Taste 5 Sekunden drücken, um diese Maske aufzurufen;
5. Taste 5 Sekunden drücken, um diese Maske aufzurufen,
6. Taste 5 Sekunden drücken, um diese Maske aufzurufen?

Difference: each segment ends with a different punctuation mark.

Results

Across: a 95% match is proposed for all segments.

Déjà Vu, memoQ: a 99% match is proposed for all segments. The punctuation mark is always adapted automatically.

Heartsome, MultiTrans: a 98% match is proposed for all segments.

SDL Trados, Wordfast: a 97% match is proposed for all segments.

Transit: a 90% match is proposed for all segments.

11.2.3.2 Variable positions

1. Ermittelte Istwerte der Sensoren bei 60Hz
2. Ermittelte Istwerte der Sensoren, bei 60Hz
3. Ermittelte Istwerte der Sensoren bei 60Hz.

Difference: in segment 2, a comma was added after “Sensoren”. In segment 3, a full stop was added at the end.

Results

Across: a 90% match is proposed for segment 2. The penalty is higher than expected as the default penalty for different punctuation is 5%. The tooltip over the match rate indicates an additional 5% penalty because of other character differences, probably the space following the comma. This space is also highlighted in the editor window. A 95% match is proposed for segment 3.

Déjà Vu: a 99% match is proposed for segment 2 and 3. However, these two segments are not handled in the same manner. The source segment 2 is marked with a split bar (light green – grey) that, by default, identifies fuzzy matches. On the other hand, the source segment 3 is marked with a deep blue bar that, by default, identifies assembled matches. The full stop is indeed added automatically at the end.

Heartsome: a 95% is proposed for segment 2. A 97% is proposed for segment 3.

memoQ: a 99% match is proposed for segment 2. There is no automatic replacement of the punctuation mark. A 99% match is proposed for segment 3, the full stop is added automatically.

MultiTrans: a 99% match is proposed for segment 2 and 3.

SDL Trados: a 98% match is proposed for segment 2 and 3.

Transit: a 91% match is proposed for segment 2 and 3.

Wordfast: a 97% match is proposed for segment 2 and 3.

11.2.3.3 Variable segment length

1. Nicht verstellen!
   Nicht verstellen
2. Explosionsgefahr durch Vertauschen!
   Explosionsgefahr durch Vertauschen
3. A) Die zulässigen Betriebsbedingungen gemäß Datenblatt (Spannung, Strom, Lufttemperatur) sind einzuhalten.
   B) Die zulässigen Betriebsbedingungen gemäß Datenblatt (Spannung, Strom, Lufttemperatur) sind einzuhalten
   C) Die zulässige Betriebsbedingungen gemäß Datenblatt (Spannung, Strom, Lufttemperatur) sind einzuhalten.

Difference: in all three segment sets, the end punctuation mark was deleted. In addition, in segment C[1] of segment set 3, a textual modification was made: “zulässigen” is (wrongly) replaced by “zulässige”.

Results

Across: a 95% match is proposed for all segment sets. For segment C, an 89% match is proposed.

Déjà Vu: a 99% match is proposed for all segment sets. The punctuation mark is always deleted automatically. For segment C, a 90% match is proposed.

Heartsome: a 94% match is proposed for segment set 1. A 97% match is proposed for segment set 2. A 99% match is proposed for segment set 3. For segment C, a 97% match is proposed.

memoQ: a 99% match is proposed for all segment sets. There is no automatic adaptation. For segment C, a 93% match is proposed.

MultiTrans: a 99% match is proposed for all segment sets. For segment C, a 90% match is proposed.

SDL Trados: a 94% match is proposed for segment set 1. A 96% match is proposed for segment set 2. A 99% match is proposed for segment set 3. For segment C, a 98% match is proposed.

Transit: a 75% match is proposed for segment set 1. An 83% match is proposed for segment set 2. A 95% match is proposed for segment set 3. For segment C, a 93% match is proposed.

Wordfast: a 92% match is proposed for segment set 1. A 93% match is proposed for segment set 2. A 94% match is proposed for segment set 3. For segment C, a 93% match is proposed.

11.3 Conclusions

11.3.1 Penalties

The difference in punctuation marks is always recognized by all TM systems (with the exception of MultiTrans if matches come from its propagation memory). Column 1 of table 11.3 summarizes the results of 11.2.3.1. Each TM system proposes the same similarity value for all five examples, therefore only one value is presented. It is clear that the penalty does not depend on the change in punctuation mark.

Column 2 of table 11.3 lists the results of 11.2.3.2, where modified punctuation marks occur at different positions. In most cases, the position of the modified punctuation mark does not affect the similarity value. The exceptions are Across and Heartsome; with these systems, the penalty is lower if the modification is at the end of the segment. Moreover, the position affects the automatic adaptation function, as discussed in 11.3.2.

[1] The letters A, B and C are not part of the segments, but are used to distinguish them.

	1	**2**	
Across	95	90	95
Déjà Vu	99	99	99
Heartsome	98	95	97
memoQ	99	99	99
MultiTrans	99	99	99
SDL Trados	97	98	98
Transit	90	91	91
Wordfast	97	97	97

Table 11.3: Match values: different punctuation marks

Table 11.4 sheds some light on the question as to whether the penalty value is calculated based on the length of the segment. The column number identifies the segment set from 11.2.3.3. The double line in column 3 separates the results concerning the textual modification (segment C).

	1	**2**	**3**	
Across	95	95	95	89
Déjà Vu	99	99	99	90
Heartsome	94	97	99	97
memoQ	99	99	99	93
MultiTrans	99	99	99	90
SDL Trados	94	96	99	98
Transit	75	83	95	93
Wordfast	92	93	94	93

Table 11.4: Match values: variable segment length

Across, Déjà Vu, memoQ and MultiTrans apply a length-independent penalty to differences in punctuation marks. This fixed penalty is applied to all examples, irrespective of the type of modification (deletions, replacements, additions). Heartsome, SDL Trados, Transit and Wordfast apply a length-dependent penalty. The longer the segment, the lower the penalty if the same modification is applied. Nevertheless, similarity values are always above 90%, with the sole exception of Transit.

All TM systems apply a higher penalty to textual modifications than to non-textual modifications, even though in both cases only one character is modified. The higher penalty is justifiable because textual modifications generally entail more effort for the translator. The difference is sometimes bigger (e.g. 9% for Déjà Vu and MultiTrans), sometimes smaller (e.g. 1% for SDL Trados and Wordfast). Punctuation marks are presumably tokenized independently, see also (Nübel and Seewald-Heeg, 1999b, 23).

11.3.2 Automatic adaptations

Automatic adaptations are made only by Déjà Vu and memoQ. memoQ can automatically add missing end punctuation marks (as in 11.2.3.1 and 11.2.3.2), but cannot delete superfluous ones. Déjà Vu, on the other hand, is more flexible.

Adaptations are possible only if the punctuation mark can be positioned unequivocally. If the modification affecting the punctuation mark occurs within the segment, both TM systems do not propose any automatic adaptation. Interestingly, this does not affect the penalty value, as clearly visible in table 11.3.

11.4 Possible improvements

11.4.1 Penalties

TM systems almost always correctly handle the tested punctuation marks so that only minor improvements are necessary. It is difficult to answer to the question as to whether the position of the changed punctuation mark should affect the similarity value. The answer would require a test assessing whether translators need more time to manually adapt a punctuation mark at the end or in the middle of a segment. However, this would entail considerable effort and possibly produce language-dependent results. Therefore, applying the same penalty is a viable solution.

The penalty applied to differences in punctuation marks should be length-independent. Otherwise, particularly for relatively short segments, significant penalty values would be applied (e.g. 25%, but higher values are also possible) although they may not be justified by the effort required for the adaptation.[2] Finally, it is not appropriate to propose 100% matches when differences are not corrected by automatic adaptations.

11.4.2 Automatic adaptations

As far as end punctuation marks are concerned, automatic adaptations should be possible and are useful because these differences can be easily overlooked. Quality assurance/quality control routines can detect divergences; however, correcting possible errors requires additional time. Automatic adaptations are thus beneficial for translation speed and quality.

In fact, end punctuation marks may require adaptation (excluding proper localization issues due to different characters). For example, if list items in the target language end with a semicolon instead of a full stop as in the source language. These rules need not be linguistically motivated, but can be part of company style guides. Consequently, automatic adaptation should also be deactivatable.

[2] (Nübel and Seewald-Heeg, 1999b, 28) come to the same conclusion.

Chapter 12

Spaces, quotes, dashes

12.1 Introduction

Quotes and dashes are punctuation marks, whereas spaces are word delimiters. Spaces, quotes and dashes are discussed separately from common punctuation marks (see chapter 11) for various reasons. Firstly, quotes and dashes usually do not belong to sentence delimiters covered in that chapter. In addition, although quotes and dashes occur within sentences in the same way as commas, they are not as frequent in the corpus and are therefore not included with common punctuation marks. Some quotes and dashes (but also spaces) are not always as well-supported as common punctuation marks in several respects, consequently more complex tests were necessary, see 12.2.4. Finally, spaces need specific coverage.

Only spaces, quotes and dashes that are common in Western European languages are discussed. This thesis does not aim at defining proper usage.[1]

12.1.1 Spaces

In typography, spaces are distinguished according to different criteria.

With respect to their line breaking behavior:

- space
- no-break space

A no-break space is different from a space because it prevents a line break after it. For example, this can be useful between numbers and measuring units.

With respect to their width:

- em space

[1] For more information on this subject, see e.g. Ritter (2005) (for British English) or University of Chicago (2003)(for American English).

- en space
- thin space

The widths are not standardized. An *em* corresponds to the width of the lowercase m or to 1000 units of the point[2] size, see (Strizver, 2006, 176). An *en* is half as wide as an em and corresponds approximately to the width of the lowercase n or to 500 units of the point size, see (Strizver, 2006, 176). A thin space is one fifth or one sixth of an em, see (Korpela, 2006, 414). For a complete list of spaces, see (Korpela, 2006, 412-417).

12.1.2 Quotes

There are numerous typographical quotes, the usage of which depends on the locale. Several types of quotes may coexist in the same locale, sometimes with different meanings and functions (quotations, verbatim citations and so on).

- " quotation mark
- ' apostrophe
- „ double low-9 quotation mark
- “ left double quotation mark
- ‚ single low-9 quotation mark
- ‘ left single quotation mark
- ” right double quotation mark
- ’ right single quotation mark
- « left-pointing double angle quotation mark
- » right-pointing double angle quotation mark
- ‹ single left-pointing angle quotation mark
- › single right-pointing angle quotation mark

[2] A *point* is a typographic unit defined in various ways, see (Haralambous, 2007, 12) for more information.

12.1.3 Dashes

Dashes are distinguished according to their width (as spaces) or to their semantic meaning. Some of them are listed below; for a complete overview, see (Korpela, 2006, 418-421).

- ‐ hyphen
- − minus sign
- – en dash
- — em dash
- - hyphen-minus sign

The hyphen-minus sign is semantically ambiguous because it can be used as a hyphen and as a minus sign. It is generally used instead of the unambiguous characters, hyphen and minus sign, which are less known and are not supported by the standard keyboard keys.

12.2 Tests

The tests are aimed at checking the following issues, see also 2.2.2 and 2.4.1.6:

1. Is the difference between the TM segment and source text segment recognized?
2. Is the difference between the TM segment and source text segment highlighted?
3. How do transpositions, replacements, additions and deletions affect the similarity value?
4. Does the segment length affect the similarity value?
5. Does the changed element affect the similarity value?
6. Does the number of modifications affect the similarity value?
7. Are automatic adaptations applied?

In order to ensure that all characters are correctly displayed in the relevant font, all examples were written in MS Word using Arial Unicode MS, as suggested by (Korpela, 2006, 415).

Differences in spaces, quotes and dashes should be recognized and highlighted, but the penalty should be low and reflect a minor modification. The calculation of the penalty value should be independent of the segment length and of the modified element, but reflect the number of modifications. The modification of spaces, quotes and dashes might be manageable by automatic adaptations in some situations where no linguistic knowledge is required. See 12.4 for more information on the recommended handling of differences in spaces, quotes and dashes.

Because of the numerous tests conducted, the test section includes two test suites and each of them will be followed by a subsection “Specific conclusions” that presents the findings in detail. Section 12.3 summarizes the main findings of the whole chapter.

12.2.1 TM system settings

12.2.1.1 Versions

TM system	Version
Across Standalone Personal Edition	4.00
Déjà Vu X Professional	7.5.303
Heartsome Translation Studio Ultimate	7.0.6 2008-09-12S
memoQ Corporate	3.2.17
MultiTrans	4.3.0.84
SDL Trados 2007 Freelance	8.2.0.835
STAR Transit XV Professional	3.1 SP22 631
Wordfast	5.53

Table 12.1: TM systems used in the tests

12.2.1.2 Customizability

In this section, the availability of the following features will be discussed.

- Is the penalty value visible and customizable?
- Can automatic adaptations be activated/deactivated/customized?

Across: under Tools > Profile Settings > CrossTank > Advanced settings, the default penalty for different punctuation and different special characters (including control characters and spaces) is 5%. However, it is not possible to check which characters are included; the online help does not provide more information either.

For automatic adaptations, no option is available.

Déjà Vu: Déjà Vu tends to insert more tags than other TM systems when it converts the source text into its proprietary format. An analysis of this aspect was beyond the scope of this thesis. Some tags were deleted if they did not affect the tested elements to ensure the comparability of the results. The standard font of Déjà Vu was changed from Tahoma to Arial Unicode MS.

The penalty for differences in spaces, quotes or dashes is neither visible nor customizable.

For automatic adaptations, the option Control leading and trailing symbols under Tools > Options > General > Conversions is active and automatically inserts “symbols such as punctuation marks or spaces at the beginning or end of a sentence”, ATRIL (2008), according to the source segment.

Heartsome: the penalty for differences in spaces, quotes or dashes is neither visible nor customizable.

For automatic adaptations, the option CORRECT SPACES WHEN ACCEPTING TRANSLATIONS under OPTIONS is activated by default and “will ensure that spaces are accurately retained after the translated file is converted to the source format”, Heartsome (2008).

memoQ: as is the case with Déjà Vu, some tags were deleted if they did not affect the tested elements to ensure the comparability of the results.

The penalty for differences in spaces, quotes or dashes is neither visible nor customizable.

For automatic adaptations, the option ADJUST FUZZY HITS AND INLINE TAGS under TOOLS > OPTIONS > TM DEFAULTS, USER INFO > ADJUSTMENTS is activated by default and enables “on-the-fly adjustment of [...] ending punctuation marks [...] within translation memory hits with less than 100% match rate”, Kilgray (2008). The tests will show whether spaces, quotes and dashes are treated as common punctuation marks.[3]

MultiTrans: for the test procedure and the options concerning the penalty value, see 11.2.1.2.

For automatic adaptations, no option is available.

SDL Trados, Transit: there are no settings concerning either the penalty for differences in punctuation in general or corresponding automatic adaptation.

Wordfast: by default, the penalty value for differences concerning spaces as well as quotes, apostrophes and dashes is 0. The penalty value was set to 1 under WORDFAST > SETUP > TRANSLATION MEMORY > TM RULES for the following items:[4]

- PENALTY FOR WHITESPACE DIFFERENCE
- PENALTY FOR DIFFERENT QUOTES/APOSTROPHES/DASHES

For automatic adaptations, no option is available.

12.2.2 General remarks on TM system behavior

Difference display and automatic adaptations

General remarks concerning the following two aspects are presented here.

[3] The option “Smart Quotes” is available for quotes, but its purpose is to replace “straight quotation marks with others that match your target language’s typographical conventions”, Kilgray (2008). This feature does not adapt the fuzzy match on the basis of a modification in the source text.

[4] Under WORDFAST > SETUP > SETUP > PB, it is possible to activate several advanced options from the so called “Pandora’s Box”, among others, PROCESSQUOTES and PROCESSDASHES. Wordfast will always use a previously defined quote or dash, no matter which type is used in the source text. However, this is a substitution option that does not affect the similarity value and was left deactivated.

- Correct display of the difference between the source segment in the TM and the source segment in the text.

- Automatic adaptation of fuzzy matches.

In several cases, differences are neither recognized nor highlighted in the editor. Sometimes the difference is highlighted, but the segment is still presented as a 100% match.

	Difference highlighting?	**Automatic adaptations?**
Across	Yes, except for spaces	No
Déjà Vu	No	Partly
Heartsome	Partly	No
memoQ	Yes, except for spaces	No[1]
MultiTrans	No	No
SDL Trados	Yes	No
Transit	Yes[2]	No
Wordfast	Partly	No[1]

Table 12.2: Difference in highlighting and support of automatic adaptations

1 = with one exception, 2 = with special settings

Across: differences are displayed correctly unless they affect spaces, which are never highlighted. No automatic adaptation is performed.

Déjà Vu: differences concerning spaces, quotes and dashes are never highlighted. Only textual modifications are highlighted correctly. Automatic adaptations are made when spaces, quotes or dashes are at the beginning or at the end of the segment. If they are in the middle, adaptations are done only if their position is unequivocally identifiable, e.g. just before or just after numbers. The description of the results, see 12.2.3 and 12.2.4, shows where automatic adaptation took place.

Heartsome: differences are usually displayed correctly even if no penalty is applied. However, sometimes a word is marked as changed although only a space, quote or dash next to it was modified. Sometimes the difference is not highlighted. All these cases are specified in the result description. No automatic adaptation is performed.

memoQ: differences are displayed correctly unless they affect spaces, which are never highlighted. Only one fuzzy match is automatically adapted and the adaptation is not fully correct. This erratic result is described in the relevant test set, see 12.2.3.1.

MultiTrans: if differences concern spaces, dashes and quotes, the segment is marked as a fuzzy match but the changed element is never highlighted. Only textual modifications are highlighted correctly. No automatic adaptation is performed, but a kind of intelligent management of quotation marks and dashes was observed and is described in 12.2.4.

SDL Trados: differences are always displayed correctly. No automatic adaptation is performed.

Transit: in order to correctly display different spaces, special settings are necessary: under OPTIONS > PROFILE > SETTINGS > EDITOR > SPECIAL CHARACTERS, different symbols have to be assigned to different spaces. No automatic adaptation is performed.

Wordfast: differences affecting spaces are not displayed correctly. Only some differences affecting quotes and dashes are highlighted. Sometimes the whole word is marked as changed although only a character next to it was modified. These cases are described in the results. Only one fuzzy match is automatically adapted. This erratic result is described in the relevant test set, see 12.2.3.1.

12.2.3 Test suite: basic modifications

12.2.3.1 Deletion or addition

Quotes

1. "Letztes Jahr konnte ich leider nicht teilnehmen.
2. Letztes Jahr konnte ich leider nicht teilnehmen.
3. "Letztes Jahr konnte ich leider nicht teilnehmen."

Difference: in segment 2, the quotation mark at the beginning was deleted. In segment 3, a quotation mark was added at the end.

Results

Across: an 80% match is proposed for segment 2. An 85% match is proposed for segment 3. The tooltip over the fuzzy matches helps explain the penalties. A 5% penalty is due to the different character. A 10% penalty is due to differences in formatting/inline-objects, although there is no such difference. For segment 2 an additional 5% penalty is applied due to differences in non-alphanumeric characters.

Déjà Vu: a 99% match is proposed for segment 2. If the proposed fuzzy match is accepted, the superfluous quotation mark is automatically deleted. A 99% match is proposed for segment 3. If the proposed fuzzy match is accepted, the missing quotation mark is added.

Heartsome: a 97% match is proposed for segment 2. The word "Letztes" is also highlighted. A 98% match is proposed for segment 3.

memoQ: a 99% match is proposed for segment 2 and 3. For segment 3, if the proposed fuzzy match is accepted, the missing quotation mark is added, but the full stop is deleted.

MultiTrans: a 99% match is proposed for segment 2. The deleted quotation mark is automatically excluded from the proposed match. A 100% match is proposed for segment 3 because the closing quotation mark is excluded.

SDL Trados, Transit: a 99% match is proposed for segment 2 and 3.

Wordfast: a 94% match is proposed for segment 2. The difference is highlighted. A 97% match is proposed for segment 3. The difference is not highlighted.

Mixed elements

1. Durchmesser 5.

2. Durchmesser 5

3. Durchmesser 5".

4. Durchmesser 5.

Difference: in segment 2, the full stop was deleted. In segment 3, a quotation mark was added before the full stop. In segment 4, the space was replaced by a thin space.

Results

Across: an 85% match is proposed for segment 2 to 4.

Déjà Vu: a 99% match is proposed for segment 2. The superfluous full stop is deleted automatically. A 99% match is proposed for segment 3. The missing quotation mark is added at the right position. A 99% match is proposed for segment 4. The space is automatically replaced.

Heartsome: a 92% match is proposed for segment 2. A 91% match is proposed for segment 3. A 99% match is proposed for segment 4 and the words "Durchmesser" and "5" are highlighted even though only the space between them differs.

memoQ: a 99% match is proposed for segment 2 to 4. In segment 4, the thin space appears in the editor as a square.

MultiTrans: a 99% match is proposed for segment 2 to 4.

SDL Trados: a 94% match is proposed for segment 2. A 96% match is proposed for segment 3. A 77% match is proposed for segment 4.

Transit: a 73% match is proposed for segment 2. A 99% match is proposed for segment 3 and 4.

Wordfast: no match is proposed for segment 2 even though the minimum fuzzy match value was 50%. A 100% match is proposed for segment 3: the difference is highlighted and the quotation mark is added automatically. No match is proposed for segment 4.

12.2.3.2 Replacement

No-break space

1. Montage von 180 Anlagen

2. Montage von 180 Anlagen

Difference: the space between "180" and "Anlagen" was replaced by a no-break space.

Results

Across: an 85% match is proposed. The tooltip explains the penalty: 5% is due to the difference in non-alphanumeric character (no-break space); 10% is due to differences in formatting/inline-objects although there is no such difference.

Déjà Vu: a 99% match is proposed. The difference is not highlighted and there is no automatic adaptation.

Heartsome, SDL Trados: a 100% match is proposed. The difference is not recognized.

MultiTrans, memoQ, Transit: a 99% match is proposed.

Wordfast: a 99% match is proposed. The difference is not highlighted.

Thin space

1. Montage von 180 Anlagen

2. Montage von 180 Anlagen

Difference: the space between "180" and "Anlagen" was replaced by a thin space.

Results

Across: an 85% match is proposed and the same explanation for no-break spaces as above applies.

Déjà Vu, memoQ, MultiTrans, Transit: a 99% match is proposed.

Heartsome: a 92% match is proposed. The difference is not highlighted.

SDL Trados: an 82% match is proposed.

Wordfast: a 68% match is proposed. The difference is not highlighted.

Dash

1. Benutzer – Optionen

2. Benutzer — Optionen

Difference: the en dash between "Benutzer" and "Optionen" was replaced by an em dash.

Results

Across: a 95% match is proposed.

Déjà Vu, memoQ, Transit: a 99% match is proposed.

Heartsome: a 92% match is proposed.

MultiTrans: a 100% match is proposed.

SDL Trados: an 89% match is proposed.

Wordfast: a 99% match is proposed. The difference is not highlighted.

Quotes

1. "Letztes Jahr konnte ich leider nicht teilnehmen.

2. „Letzes Jahr konnte ich leider nicht teilnehmen.

Difference: the quotation mark was replaced by a double low-9 quotation mark.

Results

Across: an 85% match is proposed and the same explanation for no-break spaces as above applies.

Déjà Vu: a 99% match is proposed. If the proposed fuzzy match is accepted, the quotation mark is automatically replaced.

Heartsome: a 97% match is proposed. The word “Letztes” is also highlighted.

memoQ, MultiTrans, Transit, Wordfast: a 99% match is proposed.

SDL Trados: a 97% match is proposed.

12.2.3.3 Variable segment length

In this test section, the same modification patterns as in 12.2.3.2 are investigated. However, the tested segments are longer in order to check whether the penalty value is calculated based on the length of the segment. Section 12.2.3.4 will summarize and compare the results of 12.2.3.2 and 12.2.3.3 to answer this question.

Space

1. Die Kosten für eine Installation betragen 289 €

2. Die Kosten für eine Installation betragen 289 €

Difference: the no-break space between “289” and “€” was replaced by a thin space.

Results

Across: a 95% match is proposed.

Déjà Vu: a 99% match is proposed. If the proposed fuzzy match is accepted, the space is automatically replaced.

Heartsome: a 100% match is proposed. The difference is not highlighted.

memoQ: a 99% match is proposed. The thin space appears in the editor as a square.

MultiTrans, Transit: a 99% match is proposed.

SDL Trados: an 88% match is proposed.

Wordfast: a 96% match is proposed. The difference is not highlighted.

Dash

1. Sie telefonieren bequem über das Internet - wie gewohnt mit Ihrem Telefon oder von Ihrem Computer aus.

2. Sie telefonieren bequem über das Internet – wie gewohnt mit Ihrem Telefon oder von Ihrem Computer aus.

Difference: the hyphen-minus sign between “Internet” and “wie” was replaced by an en dash.

Results

Across: a 95% match is proposed.
Déjà Vu, memoQ, SDL Trados, Transit: a 99% match is proposed.
Heartsome: a 97% match is proposed.
MultiTrans: a 100% match is proposed. The difference is not recognized.
Wordfast: a 99% is proposed. The difference is not highlighted.

12.2.3.4 Specific conclusions

Tables 12.3 and 12.4 show that Across, Déjà Vu, memoQ, MultiTrans and Transit apply a fixed penalty to modifications concerning quotes, dashes or spaces irrespective of the type of modification (deletion, addition or replacement). Some results seem to contradict this conclusion, however, they are due to other reasons. Furthermore, for these TM systems, the segment length does not influence the similarity value, see table 12.5. Instead, the segment length affects the similarity value applied by Heartsome, SDL Trados and Wordfast.

False 100% matches are occasionally proposed by Heartsome, MultiTrans and SDL Trados. In several examples the match value proposed by Across does not conform to the penalty rule explained in 12.2.1.2. A similar problem can be observed for Wordfast. The most reliable performance is shown by Déjà Vu and memoQ, which always recognize the modification and propose a 99% match.

Déjà Vu makes several automatic adaptations that are always successful. memoQ and Wordfast make only once a suggestion: memoQ fails, Wordfast is successful. In the case of MultiTrans, a sort of intelligent management of quotation marks was observed: if the segment in the TextBase and the one in the text differ only by a quotation mark positioned at the end of the segment, the Translation Agent dynamically excludes it so that better matches can be proposed.

	Quotes		**Mixed elements**		
Across	80	85	85	85	85
Déjà Vu	99	99	99	99	99
Heartsome	97	98	92	91	99
memoQ	99	99	99	99	99
MultiTrans	99	100	99	99	99
SDL Trados	99	99	94	96	77
Transit	99	99	73	99	99
Wordfast	94	97	0	100	0

Table 12.3: Match values: deletion or addition (12.2.3.1)

	No-break space	**Thin space**	**Dash**	**Quotes**
Across	85	85	95	85
Déjà Vu	99	99	99	99
Heartsome	100	92	92	97
memoQ	99	99	99	99
MultiTrans	99	99	100	99
SDL Trados	100	82	89	97
Transit	99	99	99	99
Wordfast	99	68	99	99

Table 12.4: Match values: replacement (12.2.3.2)

	Space		**Dash**	
Across	85	95	95	95
Déjà Vu	99	99	99	99
Heartsome	100	100	92	97
memoQ	99	99	99	99
MultiTrans	99	99	100	100
SDL Trados	100	88	89	99
Transit	99	99	99	99
Wordfast	99	96	99	99

Table 12.5: Match values: variable segment length (12.2.3.3)

12.2.4 Test suite: complex modifications

12.2.4.1 Multiple modifications

In the previous test sets, the segments differed only by one character. The following tests are aimed at checking the following issues:

- How do similarity values develop when several modifications are made?
- If the similarity value is not constant, is its development linear?

For the sake of clarity and comparability, the modifications will affect the same type of element and the same type of modification will be made (e.g. only additions).

Replacements of spaces

1. 100 Ohm +/- 5%
2. 100 Ohm +/- 5%
3. 100 Ohm +/- 5%
4. 100 Ohm +/- 5%

Difference: segment 1 includes only spaces. In segment 2, a thin space is used between "100" and "Ohm". In segment 3, a thin space is used also between "-" and "5%". Segment 4 includes only thin spaces.

Results

Across: an 85% match is proposed for segment 2 to 4.

Déjà Vu: a 99% match is proposed for segment 2. The space is automatically replaced. A 99% match is proposed for segment 3. Both spaces are automatically replaced. A 99% match is proposed for segment 4. All three spaces are automatically replaced.

Heartsome: a 100% match is proposed for segment 2, although the words "100 Ohm" are highlighted. A 99% match is proposed for segment 3 and 4; in both cases the whole segment is highlighted.

memoQ: a 99% match is proposed for segment 2 to 4. The thin space appears in the editor as a square.

MultiTrans, Transit: a 99% match is proposed for segment 2 to 4.

SDL Trados: an 88% match is proposed for segment 2. A 72% match is proposed for segment 3. A 58% match is proposed for segment 4. The decrease of the similarity value is not linear: -12%, -28%, -42%.

Wordfast: no match is proposed for segment 2; the difference due to a thin space is not handled correctly, as already ascertained in 12.2.3.1 and 12.2.3.2. An 89% match is proposed for segment 3. The expression "100 Ohm" is highlighted. No match is proposed for segment 4.

Additions of quotes

1. Sind Sie mit Xyz zufrieden?
2. „Sind Sie mit Xyz zufrieden?
3. „Sind Sie mit Xyz zufrieden?“
4. „Sind Sie mit ‚Xyz zufrieden?“
5. „Sind Sie mit ‚Xyz‘ zufrieden?“

Note: the original product name was replaced by “Xyz”.

Difference: in segment 2, a double low-9 quotation mark was added at the beginning. In segment 3, a left double quotation mark was added at the end. In segment 4, a single low-9 quotation mark was added before “Xyz”. In segment 5, a left single quotation mark was added after “Xyz”.

Results

Across: an 80% match is proposed for segment 2 to 5.

Déjà Vu: a 99% match is proposed for segment 2. The double low-9 quotation mark is added automatically. A 99% match is proposed for segment 3: both double quotation marks are added automatically. A 99% match is proposed for segment 4: both double quotation marks are added automatically, but the single low-9 quotation mark before “Xyz” is not. A 99% match is proposed for segment 5: both double quotation marks are added automatically, but single quotation marks are not.

Heartsome: a 96% match is proposed for segment 2. A 93% match is proposed for segment 3. An 88% match is proposed for segment 4. An 83% match is proposed for segment 5. The penalty progression is almost linear: 4%, 7%, 12%, 17%. The difference is highlighted, but together with the word preceded or followed by the quotation mark.

memoQ: a 99% match is proposed for segment 2. A 92% match is proposed for segment 3, where the word “zufrieden?” is highlighted. A 92% match is proposed for segment 4. A 64% match is proposed for segment 5, where the word “Xyz” is highlighted.

MultiTrans, Transit: a 99% match is proposed for segment 2 to 5.

SDL Trados: a 98% match is proposed for segment 2. A 91% match is proposed for segment 3. An 85% match is proposed for segment 4. A 75% match is proposed for segment 5. As in 12.2.4.1, the penalty progression is not linear: 2%, 9%, 15%, 25%.

Wordfast: a 90% match is proposed for segment 2. The difference is not highlighted. A 91% match is proposed for segment 3. The difference is not highlighted. A 76% match is proposed for segment 4. The difference is not highlighted. A 74% match is proposed for segment 5. Only the left single quotation mark is highlighted.

Deletion of dashes

1. –Leistungsbeschreibung Paket Xyz −− 3m
2. –Leistungsbeschreibung Paket Xyz − 3m
3. Leistungsbeschreibung Paket Xyz − 3m
4. Leistungsbeschreibung Paket Xyz 3m

Note: the original product name was replaced by “Xyz”.

Difference: at the beginning of segment 1, there is an en dash; between “Xyz” and “3m”, there are two hyphen-minus signs. In segment 2, one hyphen-minus sign was deleted. In segment 3, the en dash was deleted. In segment 4, both hyphen-minus signs were deleted, two spaces are left.

Results

Across: an 80% match is proposed for segment 2 to 4.

Déjà Vu: a 99% match is proposed for segment 2. A 99% match is proposed for segment 3 and 4, in both cases the en dash at the beginning is deleted automatically, but the two hyphen-minus signs remain.

Heartsome: a 95% match is proposed for segment 2. A 92% match is proposed for segment 3. An 87% match is proposed for segment 4 and the words not directly affected by the modifications are highlighted too. The penalty progression is not linear: 5%, 8%, 12%.

memoQ, Transit: a 99% match is proposed for segment 2 to 4.

MultiTrans: a 99% match is proposed for segment 2 to 4. For segments 3 and 4, the leading en dash is automatically excluded from the proposed fuzzy match.

SDL Trados: a 98% match is proposed for segment 2. A 95% match is proposed for segment 3. An 85% match is proposed for segment 4. The penalty progression is not linear: 2%, 5%, 15%.

Wordfast: A 92% match is proposed for segment 2. An 84% match is proposed for segment 3. An 81% match is proposed for segment 4. The difference is always highlighted. The penalty progression is not linear: 8%, 16%, 19%.

12.2.4.2 Mixed modifications

Until this section, tests concentrated on modifications of the same type (replacement, addition or deletion) applied to the same element. Additionally, the role of the segment length was investigated.

This test set aims at comparing the results for different types of modifications and for different elements in order to test whether some modifications have a bigger impact on the similarity value.

To obtain comparable results, only one segment will be modified in each test set. Otherwise the segment length would affect the similarity value, as shown in section 12.2.3.3.

Replacements of different elements

1. PSS 21"-Schrank
2. PSS 21“-Schrank
3. PSS 21"-Schrank
4. PSS 21"–Schrank
5. PST 21"-Schrank

Difference: in segment 1, there is a space, a quotation mark and a hyphen-minus sign. In segment 2, the quotation mark was replaced by a left double quotation mark. In segment 3, the space was replaced by a thin space. In segment 4, the hyphen-minus sign was replaced by an en dash. In segment 5, “PSS” was replaced by “PST”.

Results

Across: an 85% match is proposed for segment 2 to 4. However, the difference is not highlighted for segment 3. A 70% match is proposed for segment 5.

Déjà Vu: a 99% match is proposed for segment 2 to 4. No match is proposed for segment 5.

Heartsome: a 91% match is proposed for segment 2; the word “21“-Schrank” is highlighted. A 100% match is proposed for segment 3, the difference is not highlighted. A 91% match is proposed for segment 4 and 5.

memoQ: a 64% match is proposed for segment 2 and 4; the word “21“-Schrank” is highlighted. A 99% match is proposed for segment 3, the thin space in the source segment is displayed with a square. A 64% match is proposed for segment 5.

MultiTrans: a 100% match is proposed for segment 2 and 4, the difference is not recognized. A 99% match is proposed for segment 3. A 67% match is proposed for segment 5.

SDL Trados: a 94% match is proposed for segment 2 and 4. An 85% match is proposed for segment 3. A 90% match is proposed for segment 5.

Transit: a 99% match is proposed for segment 2 to 4. A 66% match is proposed for segment 5.

Wordfast: a 99% match is proposed for segment 2. The word “21“-Schrank” is highlighted. No match is proposed for segment 3. A 72% match is proposed for segment 4. The word “21“-Schrank” is highlighted. An 87% match is proposed for segment 5.

Different types of modifications

1. –Touchscreen-Farbbildschirm
2. Touchscreen-Farbbildschirm

3. – Touchscreen-Farbbildschirm
4. —Touchscreen-Farbbilschirm
5. –Touchscreen-Farbbildschirme
6. –Touchscreen-Farbbildscherm

Difference: in segment 2, the en dash was deleted. In segment 3, a space was added between the en dash and “Touchscreen”. In segment 4, the en dash was replaced by an em dash. In segment 5, an “e” was added at the end. In segment 6, “schirm” was replaced by “scherm”.

Results

Across: an 85% match is proposed for segment 2. An 85% match is proposed for segment 3, but the difference is not highlighted. A 95% match is proposed for segment 4. No match is proposed for segments 5 and 6.

Déjà Vu: a 99% match is proposed for segment 2; the superfluous en dash is automatically deleted. A 99% match is proposed for segment 3; a space is automatically added. A 99% match is proposed for segment 4; the en dash is automatically replaced. No match is proposed for segments 5 and 6.

Heartsome: a 96% match is proposed for segment 2, the whole word is highlighted. A 94% match is proposed for segment 3, the whole word is highlighted. A 96% match is proposed for segments 4 and 5. A 94% match is proposed for segment 6.

memoQ: a 99% match is proposed for segment 2 to 4. A 92% match is proposed for segment 5. An 84% match is proposed for segment 6.

MultiTrans: a 99% match is proposed for segment 2 and 3. A 100% match is proposed for segment 4, the difference is not recognized. A 50% match is proposed for segment 5. No match is proposed for segment 6.

SDL Trados: a 96% match is proposed for segment 2. A 100% match is proposed for segment 3, the difference is not recognized. A 92% match is proposed for segment 4 to 6.

Transit: a 99% match is proposed for segment 2 and 4. A 100% match is proposed for segment 3, the difference is not recognized. A 50% match is proposed for segment 5 and 6.

Wordfast: no match is proposed for segment 2. Most likely, segment 1 was treated as a single word, including the en dash. Without the en dash, the word is not recognized. No match is proposed for segment 3. A 99% match is proposed for segment 4. The difference is highlighted. An 87% match is proposed for segment 5. An 85% match is proposed for segment 6.

12.2.4.3 Specific conclusions

Table 12.6 summarizes the results of 12.2.4.1 and shows that for Across, Déjà Vu, MultiTrans and Transit the number of modifications does not affect the similarity value. On the other hand, SDL Trados and Wordfast increase the penalty value according to the number of modifications. A mixed approach is followed by Heartsome and memoQ:

- Replacements: the penalty value is fixed no matter how many modifications are made.
- Additions: the penalty value increases with the number of modifications.
- Deletions: Heartsome shows a progression of the penalty value whereas memoQ applies a fixed penalty.

The TM systems with a variable penalty value mostly show a loose correlation (and not a linear progression) between the penalty value and the number of modifications.

	Replacements			**Additions**				**Deletions**		
Across	85	85	85	80	80	80	80	80	80	80
Déjà Vu	99	99	99	99	99	99	99	99	99	99
Heartsome	100	99	99	96	93	88	83	95	92	87
memoQ	99	99	99	99	92	92	64	99	99	99
MultiTrans	99	99	99	99	99	99	99	99	99	99
SDL Trados	88	72	58	98	91	85	75	98	95	85
Transit	99	99	99	99	99	99	99	99	99	99
Wordfast	0	89	0	90	91	76	74	92	84	81

Table 12.6: Match values: multiple modifications (12.2.4.1)

The comparison of the similarity values of the same TM system in table 12.7 (for 12.2.4.2) show that it is irrelevant which element is changed (except for poorly supported ones, e.g. thin spaces). For Across, Déjà Vu, memoQ, MultiTrans and Transit, it is not relevant which type of modification was applied. For Heartsome and SDL Trados, the penalty value varies, but no clear pattern is ascertainable.

	Replacements of different elements				**Different types of modifications**				
Across	85	85	85	70	85	85	95	0	0
Déjà Vu	99	99	99	0	99	99	99	0	0
Heartsome	91	100	91	91	96	94	96	96	94
memoQ	64	99	64	64	99	99	99	92	84
MultiTrans	100	99	100	67	99	99	100	50	0
SDL Trados	94	85	94	90	96	100	92	92	92
Transit	99	99	99	66	99	100	99	50	50
Wordfast	99	0	72	87	0	0	99	87	85

Table 12.7: Match values: mixed modifications (12.2.4.2)

Note: textual modifications are separated by a double line

The results for modifications affecting placeable and localizable elements provide some interesting insights. A few TM systems sometimes cannot recognize the modification of spaces, dashes or quotes. This problem, which already occurred in the tests under 12.2.3, is particularly evident for MultiTrans, but Heartsome, SDL Trados and Transit are affected too. However, if erratic results are excluded, the similarity values across the TM systems are relatively uniform; ranging from 85% to 99%, with the majority above 90%.

Table 12.7 also points out differences between textual modifications and modifications affecting spaces, quotes or dashes. Across, Déjà Vu, memoQ, MultiTrans and Transit apply higher penalties to textual modifications. This is particularly evident for Déjà Vu: in contrast to 99% matches for modifications of spaces, quotes or dashes, no match is proposed for textual modifications. In the case of SDL Trados and Wordfast, it is more difficult to recognize a clear trend, particularly for Wordfast where several results are biased by poor support of the investigated characters. Still, penalties applied to textual modifications are usually higher. On the other hand, Heartsome is the only TM system in which the similarity value remains more or less the same.

There are also differences within textual modifications. One word ("Farbbildschirm") was modified once at the end ("Farbbildschirme") and once in the middle ("Farbbildscherm"), see the last two columns of table 12.7, respectively. Heartsome, memoQ, MultiTrans and Wordfast show a decrease in the similarity value; their matching algorithm presumably takes into account the position of the modification. A modification at the end entails a smaller penalty than a modification within the word. For most Western European languages this heuristic approach is correct because it accounts for flections. SDL Trados and Transit, on the other hand, apply the same penalty; however, this test set is too limited to allow for general conclusions. For Across and Déjà Vu, no match is proposed in both cases.

Comparing the differences between the similarity values proposed for textual modifications indicates significant discrepancies between the TM systems: for the three segments where textual modifications occurred, the similarity values range from 0% to 91%, from 0% to 96% and from 0% to 94%, respectively.

An explanation of the no matches could be that some TM systems calculate the similarity value on a word basis. If one word character differs from the corresponding word in the TM, the whole word is considered as changed. Since the examples presented contain few words, the penalty becomes so high that the match is not recognized. This has a detrimental effect on retrieval, in particular for very flective languages. Poor support of some characters – resulting in high penalty values – was observed too, in particular for Wordfast, but also for memoQ. In addition, TM systems (for example, memoQ) sometimes cannot distinguish between true textual modifications and modifications of spaces, quotes and dashes. This problem, probably due to poor tokenization, results in high penalties for segments that are in fact very similar. While analyzing different systems, (Nübel and Seewald-Heeg, 1999b, 28) also come to these conclusions.

12.3 Conclusions

This section summarizes the results of all test sets by answering the questions formulated in 12.2.

12.3.1 Recognition and display

Differences affecting spaces, quotes or dashes are not always recognized by all TM systems; incorrect 100% matches were proposed at least once by Heartsome, MultiTrans, SDL Trados and Transit. The opposite case of no match despite marginal modifications is described in section 12.3.2.

When the difference is recognized and penalized, it is not always easy for the user to spot it. Sometimes no highlighting is available, e.g. differences in spaces are not highlighted by most TM systems. In other cases, presentation is unclear or only possible after customization of the editor interface. To summarize, as systematically described in 12.2.2, the display is not optimal.[5] Support for spaces, quotes and dashes is worse than for common punctuation marks, see 11.3.1.

12.3.2 Penalties and automatic adaptations

Most TM systems apply a fixed penalty value to modifications affecting spaces, quotes or dashes. By default, this is low across all TM systems (1%–5%). Only some (e.g. Across) allow for customization in this respect. In other TM systems (Heartsome, SDL Trados and Wordfast), the penalty value is calculated based on the segment length; the same modification is penalized more in a short segment than in a longer one.

Checks were performed to see whether some elements are penalized more than others. The overall result was that occasional higher penalties are due to poor support of the modified element and not to a systematic difference in penalty weighting. On the other hand, textual modifications are regularly subject to higher penalties and there are significant disparities between the similarity values proposed by the different TM systems. Generally speaking, TM systems treat spaces, quotes and dashes as special sentence constituents. However, sometimes they are considered as part of neighboring words because of poor tokenization. A modification affecting these elements causes the whole word to be identified as changed, which increases the penalty and can result in no matches although the TM contains a – from a human point of view – very similar segment.

The penalty, if fixed, does not change according to the type of modification; deletions, additions and replacements are in general weighted equally. Surprisingly, Across, Déjà Vu, MultiTrans and Transit do not apply higher penalties when the number of modifications increases. The remaining TM systems show a penalty progression, which is – however – not linear.

Finally, most TM systems do not apply automatic adaptations to fuzzy matches. Déjà Vu is the most notable exception, as already noted in 11.3.2: its automatic adaptation

[5] The importance of displaying non-printing characters such as spaces is stressed by Zetzsche (2007a).

delivers reliable results. Automatic adaptations are possible if the modification can be positioned unequivocally, e.g. at the beginning or at the end of the segment.

12.4 Possible improvements

12.4.1 Recognition and display

The most important improvement is a complete and correct recognition of the modifications concerning (sometimes less common) punctuation marks and word delimiters. Complete and clear highlighting is necessary too: *complete* because sometimes even recognized differences were not highlighted; *clear* because some modifications were difficult to spot despite highlighting.

12.4.2 Penalties and automatic adaptations

A special fixed penalty applied to spaces, quotes and dashes is a viable solution, particularly if its value is customizable. This approach seems more beneficial than a length-dependent penalty. Thus, modifications affecting these placeable and localizable elements are distinguished from textual modifications that justify higher penalties. Spaces, quotes and dashes (and other punctuation marks) can be penalized with the same value and it is unnecessary to apply different penalty values to different types of modifications.

Penalties should increase with the number of modifications because more modifications entail more work for the user. Furthermore, the increase of the penalty value should be linear. None of the TM systems tested fulfills both these requirements yet.

Automatic adaptations proved to be helpful. However, they cannot be applied in all situations because of their intrinsic limitations, see 12.3.2, and should be checked by the user as more sophisticated locale-dependent adaptation strategies might be necessary, see 12.1.

Part III

Conclusions

Chapter 13

General conclusions

This chapter summarizes the test findings and also serves as a starting point for further reading in the relevant chapters where complete descriptions are provided. Section 13.1 briefly recalls the main concepts of chapter 1 and summarizes the research introduced in chapter 2. The following sections are intended as a response to the research questions presented in 2.2, while section 13.6 tries to formulate in few words a global assessment of the TM systems tested. The final section 13.7 lists some possible directions for further research.

13.1 Background and outline of the conducted research

This thesis is the product of research focused on placeable and localizable elements in the context of the translation process with the help of TM systems. Placeable and localizable elements are portions of a document that remain unchanged or are adapted according to specific conventions in the target language, see also 2.1.1.1: mainly, numbers, dates, proper nouns and identifiers, URLs, e-mail addresses, tags, inline graphics, fields, punctuation marks and word delimiters.

The background of computer-assisted translation is discussed in chapter 1. The key concepts of translation environment tool and TM system (among others) are introduced and explained primarily for readers who are not acquainted with this field.

This research is motivated by the popularity of translation environment tools in the translation process as indicated by past surveys. The popularity of these applications is due to the substantial advantages that they provide in comparison with the traditional translation process, such as improved productivity, consistency and flexibility. However, they are not free of disadvantages, such as the financial investment required and the time it takes to learn how to use them. Further disadvantages are possible, such as context disjunction. As these products have now reached a certain maturity, different strategies have been elaborated to mitigate the drawbacks, although they have not completely disappeared.

TM systems are a component of translation environment tools and are essentially information retrieval systems. They have to be able – given a source language text unit – to retrieve an identical or the most similar text units as well as their target language counterparts so that the translation production effort is minimized. The similarity value can be calculated using different approaches and based on different text units. A segment usually sets the boundaries of the comparison, but unsegmented strings are used too, see 1.3.5.1. Within the segment or the string, the similarity can be calculated based on words, which however requires some enhancements of a linguistic nature. Alternatively, characters can be taken as comparison units; this has the advantage of also being suitable for languages without explicit word boundaries such as Chinese and Thai. Hybrid approaches are also conceivable. Regardless of how calculation is performed, the similarity is expressed with a match value, usually a percentage.

The main directions of future developments lie in the exploitation of the potential enabled by the web, by interaction with MT and by retrieval at a more granular level. Firstly, the field of translation environment tools has undergone major changes in particular with the spread of the Internet. Connectivity has enabled forms of remote collaboration that were nearly impossible 20 years ago. Secondly, the awareness of MT has increased as well as its usage, but the coexistence of MT and TM still offers potential for optimization. Thirdly, TM systems increasingly offer subsegment retrieval, thus permitting greater reuse of past translations and enhancing productivity.

In chapter 2, placeable and localizable elements are presented in detail with explanations as to why their correct handling is advantageous for the translation process. Higher translation throughput, better quality, more successful retrieval and more effective memorization are the advantages that have been identified.

The main aims of this thesis included the assessment of recognition, retrieval and automatic adaptations. Corollary objectives such as the assessment of display, support and customizability were also pursued. All aims can be summarized by the goal of correctly estimating and effectively reducing the translation effort.

Once the research scope and its objectives were clear, the methodology was defined. The evaluation comprised a set of black-box tests with a test procedure modeled on the task under examination; either recognition or retrieval of placeable and localizable elements. To evaluate recognition, many different placeable and localizable elements were tested. When evaluating retrieval, the variety of elements was more restricted, with a standard set of modifications applied in order to reproduce the issues encountered in translation. The strategy used to interpret the results varied accordingly. A standardized metric was applied to a limited test subset, where the recognition of numbers, dates and proper nouns was evaluated. A similar metric could have been applied to URLs and e-mail addresses too, but the relative effort was not justified because the interpretation of the results was straightforward. For the results of retrieval evaluation, a more general interpretation was preferable because a metric would not have been suitable for assessing less formalized aspects.

In order to build adequate test sets, several data sources – mainly consisting of technical documentation – were exploited. The principal data source was a collection of four translation memories. Thus, the examples were taken from real translation jobs and reflected their complexity and challenges. This source was supplemented with a software manual and Wikipedia dump where they allowed for the extraction of better examples. The source formats were MS Word and HTML.

The evaluation could encompass only some of the available commercial TM systems, see 2.4.3. The selection included the most popular translation environment tools and some additional systems. The tests were conducted over a long period of time so that several versions were considered. However, the research was not aimed at identifying the best translation environment tool, but at pointing out which aspects are relevant when dealing with translatable and localizable elements. The results apply exclusively to the tested versions and identified shortcomings might have been solved in the meantime. The settings and the customizability of the TM systems tested were given particular attention because they can significantly affect the results. Customizability can be a very efficient way of adapting a TM system to the requirements e.g. of different projects or different language combinations.

Possible improvements were formulated as follows: for better recognition, regular expressions were developed; for all other objectives, less formalized remarks were presented. The regular expressions conform to the Perl flavor and are partly personal ad hoc suggestions and partly taken from reference works that propose viable solutions for the recognition difficulties encountered by TM systems.

Each group of placeable or localizable elements was presented in a separate chapter with tests, conclusions and possible improvements. The conclusions compared the results among the TM systems and pointed out strengths and weaknesses. In general, the results show different grades of maturity of the TM systems. On the other hand, it is impossible to identify the best (or worst) TM system because the performance varies significantly depending on the placeable or localizable element under investigation, see 13.6. The differences ascertained within this limited research scope – only very particular features of the TM systems were assessed – have far reaching consequences for the translation itself and for further steps of the translation process such as quality control. The general opinion formed in the course of this research is that some TM features did not receive sufficient attention and testing during development. Furthermore, some assistance functionalities, namely automatic adaptations, are still underexploited by many TM systems.

13.2 Recognition

In chapters 3 to 7, assessing the recognition of the placeable and localizable elements is the main test objective. The results are summarized below.

13.2.1 Numbers and dates

Almost all TM systems are principally able to recognize numbers. However, only some of them make use of this feature during translation and mark numbers as placeable or localizable elements with the advantages described under 2.1.1.2. Others recognize numbers only in the quality checks they perform on the translated text. The results of a specific metric in chapter 3 show that only Wordfast exceeds the defined baseline. memoQ almost reaches it while the rest show poor recognition. Poor recognition might be a minor problem during translation, but it becomes serious if erroneous numbers are overlooked by the quality checks that most TM systems offer. The consequence is that errors are not spotted despite the quality check. In addition, poor recognition causes automatic conversions, see 2.1.1.2, to fail.

Generally speaking, the mere recognition of single digits or digit groupings can be achieved without major problems, as digits can be identified by means of their Unicode property, which encompasses not only the characters from 0 to 9 but also e.g. Arabic-Indic digits as well as fractions. However, the tests (see 3.2.3) have proven that this basic recognition is not always provided, in particular when digits occur in alphanumeric strings. Such strings cause digit recognition to fail repeatedly.

Many numbers include decimal and thousand separators. Failing to recognize them means that automatic conversions, e.g. of decimal separators between English and German, will not be successful. Difficulties arise because it is not easy to identify exactly which combinations should be allowed. Digit sequences such as telephone numbers, numeric dates and standards make up a single unit from a semantic point of view. However, TM systems often fail to recognize them because of separators between digits.

Number recognition can be extended to some characters that semantically form a unit with numbers. This is the case for the plus sign, the minus sign, etc. However, the tests have shown that few TM systems include these characters and that it is quite difficult to achieve complete recognition. An evident problem is that some characters (e.g. the hyphen-minus sign) are ambiguous so that the risk of over-recognition is high.

The recognition of measuring units is important if the automatic conversion of measuring units is supported (so far only by SDL Trados). Such support entails difficulties because of the variety of measuring units possible; as a result, several are not recognized.

Alphanumeric dates are recognized only by SDL Trados, which consequently scores the highest in date recognition as a whole. Most TM systems do not have the necessary linguistic knowledge. The tests also show that there are several variations (e.g. abbreviated month names) that make complete recognition difficult to achieve.

To summarize, digit-only recognition can work reliably, but is far from being full-fledged. If additional characters are included, error-proneness increases. Some TM systems do not achieve complete digit-only recognition with at least two consequences. Firstly, they are not able to quality-check all digits in the text although they claim to do so. Secondly, if they offer automatic conversion of some sort, shortcomings in recognition cause adaptations to fail. Therefore, such functions are not fully reliable. The proposed regular expressions are geared towards complete and comprehensive recognition.

13.2.2 Proper nouns and identifiers

Complete language-independent recognition of proper nouns is beyond the scope of TM systems. Therefore, proper nouns are not dealt with as a semantic category, but as a formal category that also includes other identifiers. These elements are given a formal definition specifying what they look like when they most likely do not need any translation. The definition concentrates on specialties regarding letter case as well as the presence of non-alphanumeric characters.

Only two TM systems tested try to identify proper nouns, see 5.2. SDL Trados provides very basic recognition of completely capitalized alphabetic strings. Wordfast goes beyond that, and can also recognize alphanumeric strings, mixed-case strings, and strings containing some non-alphanumeric characters. Extensive customization options are given so that advanced users can adapt this function to suit their needs. The advantages of this recognition are twofold. Firstly, transfer from the source text can speed up the translation and prevent typing errors, see 2.1.1.2. Additionally, the recognized elements can be included in a quality check routine, but this feature is not yet implemented in any of the TM systems tested.[1] This check should be optional: when solely formal criteria are chosen, misrecognitions are unavoidable. However, powerful quality checks are already modular and can be toggled on and off as suitable.

To conclude, proper nouns and identifiers are still given little attention in the TM systems tested, even though their recognition could be advantageous in the translation process and is called for by users, see e.g. ProZ (2011). A set of regular expressions has been crafted to cover several possibilities. However, customization may be required because it is virtually impossible to account for all scenarios.

13.2.3 URLs and e-mail addresses

When URLs and e-mail addresses occur, it is of crucial importance to distinguish between two possibilities. They can be included in fields and processed as fields so that their content is not relevant for recognition itself. Such fields consist of a displayed text and a function defining the target. Alternatively, they occur as plain text. Only in the latter case it is possible to assess the specific recognition offered by TM systems. The results are less than impressive: only Wordfast can recognize most URLs, and no TM system recognizes e-mail addresses. Regular expressions for recognizing those elements are readily available in reference textbooks, such as Friedl (2006) and Goyvaerts and Levithan (2009). As the validation of URLs and e-mail addresses is not pursued by TM systems, see 2.4.4.1, the proposed regular expressions are quite lax.

When URLs and e-mail addresses are fields, some TM systems only support the modification of the displayed text so that the target text seems correct, but the underlying function still points to the URL or e-mail address of the source text. These problems are related to the support of fields and the file format filter, see 13.5, not to the structure of the URL or e-mail address.

URLs and e-mail addresses are not as frequent as numbers or proper nouns, therefore they are of secondary importance, see 2.4.1.5. Nevertheless, their recognition does not pose particular problems, they are easily identifiable and could be included in quality

[1] The acronym check of the stand-alone quality control tool ErrorSpy, see 1.4.6, is an implementation example.

check routines, which can be very beneficial for particular projects where these elements occur with above-average frequency, see e.g. Zhechev and van Genabith (2010).

13.2.4 Recognition ranking

Table 13.1 ranks the recognition functionality provided by TM systems. If a TM system does not provide any recognition, it is omitted. E-mail addresses are not included in the ranking because they are not recognized by any TM system, see chapter 7. Wordfast is the most successful TM system in this regard because it recognizes the most placeable and localizable elements and because its level of recognition is always the best or the second-best. SDL Trados provides, on the one hand, comparatively accurate recognition of some placeable and localizable elements, however its number recognition is extremely poor. Other TM systems show on average more or less the same performance, with the exceptions of Heartsome and MultiTrans, which do not provide any recognition.

Numbers		**Dates**		**Proper nouns**		**URLs**	
1	memoQ	1	SDL Trados	1	Wordfast	1	Wordfast
2	Wordfast	2	Wordfast	2	SDL Trados		
3	Across	3	Déjà Vu				
4	Transit		Transit				
5	Déjà Vu	5	memoQ				
6	SDL Trados	6	Across				

Table 13.1: Overview: recognition ranking

13.3 Retrieval

Retrieval is the main research objective in chapters 8 to 12 and is primarily assessed based on segment matches. Are matches presented? Does the proposed match value reflect an intuitive human judgment of the difference between the segments?

13.3.1 Customization

Several TM systems offer the possibility of customizing the penalty value for differences concerning some placeable and localizable elements. Others, however, do not allow these settings to be viewed or modified.

Even if the penalty is customizable, it is often not clear to which elements the penalty applies because the user interface and other resources such as the online help and the documentation do not provide enough information.[2] This applies for example to differences

[2] Documentation is part of a general usability evaluation, see (White, 2003, 230).

in punctuation marks: are all punctuation marks considered? Is their position within the segment irrelevant? Although this problem is not crucial, it can be annoying if the desired result is not achieved after customization.

Customizable penalties can be lowered to 0%, but this is not advisable, even if automatic adaptations apply. As the tests have shown, errors in these adaptations are not seldom. Correction remains necessary and the required effort should be reflected in the penalty.

13.3.2 Factors influencing the penalties

Different TM systems apply different penalties to the same modification, see Fifer (2007). On average, some TM systems tend to apply higher penalties than others. Since the tests assume that TM systems operate like black boxes, see 2.3.1, it is often not possible to fully reconstruct how penalty values are calculated.

Generally, higher penalties are applied to textual modifications than to modifications of placeable and localizable elements. Differences in placeable and localizable elements only (without discursive text, see below) should not prevent match retrieval. However, if the changed element includes discursive text (for example, translatable attributes for tags), this textual modification has to be considered in the penalty calculation and, depending on the context, a match may be justified or not.

The penalty applied by some TM systems depends on the length of the segment. This behavior cannot be customized and is questionable because the necessary translation effort is largely independent of the segment length and may cause high penalties being applied to short segments, see e.g. 11.3.1.

In some TM systems, see 13.3.3 for more details, the number of modifications is not taken into account. This is wrong because the more modifications are made, the more adaptation is necessary and the penalty value should be proportional. Even if automatic adaptations have been applied, the user still has to spend time checking whether they are correct.

Furthermore, it is worth considering whether the type of modification should influence the penalty value. For textual modifications the effort for adapting a fuzzy match is different; for example, deletions involve less effort than additions because, apart from minor adaptations (if any), *usually* no target text production is required. However, for modifications exclusively involving placeable and localizable elements, differentiation seems of limited usefulness because the effort required for adapting a fuzzy match does not vary significantly depending on the type of modification. A fixed penalty can be successfully applied. This is already the case in several TM systems, even though the actual implementations differ. This approach eases customization, see 13.3.1.

13.3.3 Errors

Manifest retrieval errors can comprise incorrect 100% matches or incorrect no matches. On the one hand, differences are not recognized and incorrect 100% matches are proposed in two cases:

- For some transpositions, although there is no automatic adaptation.
- For some differences concerning spaces, quotes and dashes, sometimes because of inappropriate TM system settings.

On the other hand, high penalties sometimes prevent a match from being retrieved although the differences are minimal, for example when only a quote has been modified, but the whole word following or preceding it is considered as changed.

Table 13.2 on page 252 provides an overview of the number of errors found in the tests. The criteria for the calculation are:

- Only segments containing differences pertaining to placeable and localizable elements are relevant.[3]
- Only errors pertaining to retrieval are considered. Errors affecting segmentation, editability, display, etc. are ignored.
- Only manifest errors as defined above are considered. While incorrect 100% matches are clear, a threshold had to be defined for very low matches. The match value of 70% was set because it usually corresponds to the lower limit for minimum matches recommended for TM systems, see 2.4.3.2.

TM systems are ranked according to the number of errors. The fewer the errors, the better the TM system. The comparison highlights the fact that some TM system are more robust than others: Déjà Vu makes almost no errors, while MultiTrans makes significantly more errors than any other TM system. Conversely, some placeable and localizable elements tend to pose more difficulties than others. Common punctuation marks, for example, are very well supported by all TM systems, while tags are problematic for the majority of the TM systems.

Besides manifest retrieval errors, fuzzy matches may be inadequate, but not completely incorrect, e.g. if a high penalty is applied to a minimal modification, but does not prevent matches from being retrieved. Trying to determine an ideal similarity value is pointless because the underlying concept of similarity is elusive and idiosyncratic. Different users judge differently, see (Fifer, 2007, 97-106). Consensus can be achieved on values that are clearly inappropriate, e.g. a 20% penalty for a difference in punctuation marks. However, a suitable penalty can be defined at most in terms of a range of possible values. This is the reason why customizable penalty values, see 13.3.1, are a user-friendly feature.

13.3.4 Automatic adaptations

Automatic adaptations are productivity enhancements, see (Lagoudaki, 2009, 144). In order to assess the retrieval of TM systems, automatic adaptations play a role too, see 2.2.2. The support of automatic adaptations varies significantly between TM systems,

[3] Segments containing modifications affecting running text in chapter 11 and 12 are excluded.

see table 13.3 discussed later. Not all modifications regarding placeable and localizable elements can be dealt with automatic adaptations, but in some TM systems they are not provided even if they would be possible. Some TM systems are very powerful but do not allow their automatic adaptations to be customized. Conversely, some TM systems allow for customization but their functionality is limited.

As regards tags, inline graphics and some fields, automatic adaptations are possible and successful, particularly when deletions and replacements are made. However, the necessary modification does not always encompass only the changed element itself, but includes e.g. the adaptation of spaces after a deletion. This is only handled correctly by Déjà Vu. The replaced element is also relevant: if a result field is replaced by a reference field, automatic adaptation cannot produce perfect results. During the tests, there were some instances of incorrect automatic adaptations. Even for modifications such as deletions, which allow for more reliable automatic adaptations, ambiguous situations cannot be excluded altogether, particularly when the segment includes more than one placeable or localizable element of the same type. This observation does not contradict the case for more automatic adaptations formulated throughout this work. In many instances, the adapted segment does not require any further intervention. However, this cannot be taken for granted. This is why a review is advisable even if a 100% match is proposed.

A minority of TM systems also offers partial automatic adaptations. When additions and transpositions are made, correct placement of the modified element would require linguistic knowledge, which is not available. Still, some TM systems automatically insert the added element at the end of the segment, for example, so that the user has to reposition it correctly. Partial adaptations may lead users to accept incorrect matches if they do not pay sufficient attention. Without specific tests, it is not possible to state whether they are really helpful or not.

The automatic adaptations observed were beneficial, but it cannot be excluded that some of them, in particular combinations of source and target languages, may not be useful or may even be irritating. For this reason, it should be possible to deactivate these default features if necessary. This is not always the case in current TM systems.

Table 13.3 on the following page summarizes the automatic adaptations applied by TM systems. Automatic adaptations were counted only if they were correct and complete, in other words, incorrect and partial adaptations are not considered. Partial adaptations were excluded because their actual usefulness could not be proven.

The TM systems are ranked according to the number of automatic adaptations. The more automatic adaptations, the better the TM system. The comparison highlights clear differences between TM systems. Déjà Vu provides the most automatic adaptations, while MultiTrans cannot provide any. Automatic adaptations are not equally frequent for all placeable and localizable elements.

	Tags	**Inline graphics**	**Fields**	**Common punctuation marks**	**Spaces, quotes, dashes**	**Total**
Tested segments	16	6	8	10	27	67
Déjà Vu			1			1
SDL Trados					3	3
Transit			2		1	3
Across	5					5
memoQ	6				3	9
Heartsome	5		1		4	10
Wordfast	4	2			8	14
MultiTrans	11	2	6		5	24

Table 13.2: Overview: retrieval errors

	Tags	**Inline graphics**	**Fields**	**Common punctuation marks**	**Spaces, quotes, dashes**	**Total**
Tested segments	16	6	8	10	27	67
Déjà Vu	3	5		9	15	32
memoQ	3	1	1	6		11
Wordfast	4		4		1	9
Across	3	2	3			8
SDL Trados	4	2	1			7
Transit	3	2	1			6
Heartsome	3		1			4
MultiTrans						0

Table 13.3: Overview: automatic adaptations

13.3.5 Retrieval ranking

This section takes a closer look at the figures presented in table 13.2 and 13.3 in order to provide a complete overview of the retrieval performance of the TM systems. To start with, the performance of TM systems can be ranked for each type of placeable and localizable element. The following criteria are applied:

1. Number of errors: the fewer the errors, the higher the ranking.

2. Number of automatic adaptations: the more adaptations, the higher the ranking.

The criteria were strictly applied in this sequence, i.e. automatic adaptations can only determine the sequence of TM systems having the same number of errors. This is because errors are always more detrimental to retrieval than the failure to perform automatic adaptation. The results are presented in table 13.4.

The most interesting finding is probably that no TM system is always in the top 3 for each type of placeable and localizable element. Even a good TM system can score poorly in certain cases, e.g. Déjà Vu with fields.

In order to define a global ranking of retrieval performance for all the TM systems tested, the mean of their ranking positions was calculated. The results are presented in table 13.5 and show that Déjà Vu is the TM system with the best retrieval performance. MultiTrans ranks at the bottom of the list.

Tags		Inline graphics		Fields		Common punctuation marks		Spaces, quotes, dashes	
1	SDL Trados	1	Déjà Vu	1	Wordfast	1	Déjà Vu	1	Déjà Vu
2	Déjà Vu	2	Across	2	Across	2	memoQ	2	Across
	Transit		SDL Trados	3	memoQ	3	Across	3	Transit
4	Wordfast		Transit		SDL Trados		Heartsome	4	memoQ
5	Across	5	memoQ	5	Heartsome		MultiTrans		SDL Trados
	Heartsome	6	Heartsome	6	Déjà Vu		SDL Trados	6	Heartsome
7	memoQ	7	MultiTrans	7	Transit		Transit	7	MultiTrans
8	MultiTrans		Wordfast	8	MultiTrans		Wordfast	8	Wordfast

Table 13.4: Overview: retrieval performance ranking

	Tags	Inline graphics	Fields	Common punctuation marks	Spaces, quotes, dashes	Mean
Déjà Vu	2	1	6	1	1	**2.2**
SDL Trados	1	2	3	3	4	**2.6**
Across	5	2	2	3	2	**2.8**
Transit	2	2	7	3	3	**3.4**
memoQ	7	5	3	2	4	**4.2**
Wordfast	4	7	1	3	8	**4.6**
Heartsome	5	6	5	3	6	**5**
MultiTrans	8	7	8	3	7	**6.6**

Table 13.5: Overview: global retrieval performance ranking

13.4 Display

The general look and feel of the editing environment varies significantly between TM systems. Preferences are highly personal as pointed out by Lagoudaki (2008) and McBride (2009) and TM systems allow for extensive customization. However, in conjunction with placeable and localizable elements, there are some display issues that affect the proper functionality of TM systems. These issues can either affect the display of the element itself or the display of the modifications concerning the element.

13.4.1 Placeable and localizable elements

It is important that placeable and localizable elements are displayed during translation and convey enough information. Otherwise, the quality of the translation can be jeopardized. Shortcomings were ascertained for inline graphics, HTML character entities and fields.

When the source format is imported into the TM system, inline graphics are converted into tags. However, sometimes these tags do not indicate that they stand for an inline graphic. HTML character entities are not always displayed as the character they stand for either. For result fields and objects, in some TM systems only a numbered placeholder is shown. Checking the original file in a side-by-side view or the translated document in a real-time preview can solve these problems, but entails additional effort.

General text display problems have been recently reported for some language combinations, see (Lagoudaki, 2009, 134). They were not observed in this thesis, most likely because the languages involved in the tests were Western European languages.

13.4.2 Modifications

A further problem concerning the display is that for spaces, quotes and dashes, for example, some TM systems correctly recognize the difference between segments, but do not highlight it. The user has to search for the difference; this is a time-consuming task that a translation environment tool should eliminate.

The opposite problem occurs with, for example, modified quotes if the whole neighboring word is highlighted. This gives the impression that the modification affects the word itself. This problem is probably due to the fact that some characters are not tokenized as independent elements.

13.5 Format filters

Most TM systems use format filters to convert the file to be translated from the original format into a proprietary bilingual format supported by their editors, see 1.3.1. The tests made with placeable and localizable elements have shown that the format filters deployed by TM systems do not always work flawlessly.[4] The quality of format filters was not the

[4] This finding is confirmed by (Lagoudaki, 2009, 165).

main focus of the tests and only DOC and HMTL files were processed. Nevertheless, shortcomings were ascertained with both source formats.

The first shortcoming concerns the segmentation. In HTML, some tags should be internal because they occur within a sentence or a paragraph, see 8.3.2. However, they are not always interpreted correctly. In DOC, segmentation problems arise with inline graphics and fields that split segments in the middle. In addition, if inline graphics and fields occur at the beginning of a segment, they are excluded from the segment. Translation difficulties can arise because the target language might need syntax reordering that requires the element to be within the segment and not before it. This issue also affects numbers. Finally, in HTML some external tags are included in the segment. This is incorrect too, although less problematic. Similar shortcomings have also been reported for other file formats, e.g. MIF, see Mitschke (2010).

A further problem concerning the HTML format is that some elements have attributes whose values are language-dependent and should be translatable. Their translatability is defined in the format filters, but errors are relatively frequent, see 8.3.2. This problem can be solved adapting the format filters. However, about half of the TM systems do not allow for any customization in this respect. Even when customization is possible, there are significant differences between the TM systems. The user-friendliness of the adaptation is not homogeneous, with some TM systems helping the user with input forms and others relying on the knowledge of regular expressions. However, the main problem is that during translation it is not always possible to see that something is left untranslated. Subsequent customization and corrections can entail considerable additional effort.

Further shortcomings of the format filters concern the editability of fields in DOC. The most interesting examples are result fields, where the content is automatically generated by MS Word according to the instructions contained in the field itself, see 10.1. It does not make sense to present the content as plain text – as some TM systems do – because that text is discarded after a field update. In order to modify the content, it would be necessary to modify the instructions. A very similar problem affects reference fields that consist of text and of an underlying destination; some TM systems do not allow the destination of URLs and e-mail addresses, see 13.2.3, to be changed.

While not all TM systems are affected by these faults, it is still surprising that two of the most common formats are sometimes not correctly supported. These general issues concerning format filters have already been reported in recent years, see 1.3.1, and it is not possible to tell whether there has been any improvement, see also 13.7.

13.6 Global assessment

This section provides – as far as possible – a global assessment of the TM systems tested. This assessment concentrates on the main test objectives formulated under 2.2, i.e. recognition and retrieval performance for placeable and localizable elements.[5]

[5] Secondary objectives, see 2.2.3, are not explicitly discussed in this section. Some of them (e.g. correct segmentation) influence recognition and retrieval performance so that they impact indirectly on the global

If the results of recognition, see 13.2.4, are compared to those of retrieval performance, see 13.3.5, it can be noted that the most successful TM system in recognition tests, i.e. Wordfast, is positioned in the lower half of the retrieval ranking. Conversely, Déjà Vu shows the best retrieval performance, but does not stand out as regards recognition. SDL Trados scores well as regards ranking, provides accurate recognition for some placeable and localizable elements, but its poor performance in number recognition, a key feature for any TM system, represents a major setback. Despite shortcomings, these three TM systems offer above-average results in several cases. Heartsome and MultiTrans, on the contrary, tend to be consistently at the lower end of the ranking. They seem to be the least robust among the TM systems tested. All other TM systems offer a recognition and retrieval performance that more or less equates to the average.

The remarks above demonstrate that it is not possible to name one TM system that can be globally identified as superior. Furthermore, rankings in general reflect the situation when the tests were conducted and the performance of the tested version. As new versions of the TM systems are constantly released, it is likely that the ranking at the time of reading is different as software updates make evaluations of previous versions less useful, see (Quah, 2006, 132). In any case, the aim of this thesis was to highlight which problems arise in conjunction with placeable and localizable elements and which improvements can be implemented in order to solve these problems.

13.7 Further research

The research on placeable and localizable elements as well as on translation environment tools can be expanded in several directions that could not be investigated in this thesis. The most obvious further research is to repeat the described tests on up-to-date versions of the TM systems in order to ascertain whether improvements have been implemented.

The regular expressions suggested in chapters 3, 5, 6 and 7 were tailored to recognize the examples presented. The risk of overfitting is obvious and to some extent unavoidable because of the difficulty of developing regular expressions that fit all situations, see 2.4.4.1. It would be interesting to extract further examples from corpora covering legal or financial texts, for example, and to test the proposed regular expressions. In addition, as the standards for URLs, for example, continue to develop, the relative regular expressions may need to be updated.

A limiting factor in some tests, particularly in chapters 3, 4 and 11, were the tested languages. Carrying out the tests with non-European languages (Arabic, Chinese, Hindi, etc.) may provide more insight into the support of language peculiarities and enable verification of whether any TM system was (unwittingly) tailored to European languages, e.g. as regards the automatic adaptation functionality.

To my knowledge, no specific comprehensive investigation on the file format filters of TM systems is available. The shortcomings described in this thesis suggest that this field

assessment.

should be further investigated with a test set that systematically covers the peculiarities of the desired format (HTML, IDML, MIF, etc.).

There certainly appears to be room for improvement in the usability of TM systems and in particular their operability, defined as “effort in using and controlling the system”, (Trujillo, 1999, 262). As the principles of good usability become increasingly extensive (Lagoudaki, 2008, 174-177), a further investigation in this regard would be worthwhile.

Part IV

Appendix

Appendix A

Unicode codepoints

Official names come first, even if less common. Alternative names are specified in square brackets. Spaces are shown between left and right double quotation marks.

Character	Codepoint	Representative glyph
ampersand	0026	&
apostrophe [single quotation mark]	0027	'
asterisk	002A	*
colon	003A	:
comma	002C	,
commercial at [at sign]	0040	@
degree sign	00B0	°
division slash	2215	∕
dollar sign	0024	$
dot operator	22C5	⋅
double low-9 quotation mark	201E	„
em dash	2014	—
em space	2003	“ ”
en dash	2013	–
en space	2002	“ ”
equals sign	003D	=
exclamation mark	0021	!
full stop [dot, period, point]	002E	.
greater-than sign	003E	>
hyphen	2010	‐
hyphen-minus sign [hyphen]	002D	-
left curly bracket	007B	{
left double quotation mark	201C	“
left parenthesis	0028	(
left-pointing double angle quotation mark [guillemet]	00AB	«
left single quotation mark	2018	‘

left square bracket	005B	[
less-than sign	003C	<
low line [underscore]	005F	_
minus sign	2212	−
multiplication sign	00D7	×
no-break space [non-breaking space]	00A0	“ ”
number sign [hash, sharp sign]	0023	#
percent sign	0025	%
plus sign	002B	+
question mark	003F	?
quotation mark	0022	"
ratio	2236	∶
reverse solidus [backslash]	005C	\
right curly bracket	007D	}
right double quotation mark	201D	”
right parenthesis	0029	)
right-pointing double angle quotation mark [guillemet]	00BB	»
right single quotation mark	2019	’
right square bracket	005D	]
semicolon	003B	;
single left-pointing angle quotation mark [guillemet]	2039	‹
single low-9 quotation mark	201A	‚
single right-pointing angle quotation mark [guillemet]	203A	›
solidus [slash]	002F	/
space	0020	“ ”
thin space	2009	“ ”
tilde	007E	~

Table A.1: Unicode codepoints

Bibliography

Abekawa, T. and Kageura, K. (2008). Constructing a Corpus that Indicates Patterns of Modification between Draft and Final Translations by Human Translators. In Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odjik, J., Piperidis, S., and Tapias, D., editors, *Proceedings of the Sixth International Conference on Language Resources and Evaluation*, pages 2000–2005, Marrakech, Morocco. European Language Resources Association. `http://www.lrec-conf.org/proceedings/lrec2008/pdf/512_paper.pdf`. Last visited on 13th February 2011.

Across (2009a). Across 4 Online Help. Across Systems, Karlsbad.

Across (2009b). *User Manual crossMining.* Across Systems, Karlsbad.

Alcina, A. (2008). Translation technologies: Scope, tools and resources. *Target*, 20(1):79–102.

Anastasiou, D. (2010). Survey on the Use of XLIFF in Localisation Industry and Academia. In *Proceedings of the 7th International Conference on Language Resources and Evaluation*, Valletta, Malta. European Language Resources Association. `http://ai.cs.uni-sb.de/~stahl/d-anastasiou/Publications/XLIFF_survey.pdf`. Last visited on 13th February 2011.

Arenas, A. G. (2010). Project management and machine translation. *MultiLingual*, 21(3):34–38.

Arntz, R., Picht, H., and Mayer, F. (2002). *Einführung in die Terminologiearbeit.* Georg Olms, Hildesheim.

ATRIL (2003). *Déjà Vu X Professional Users' Guide.* ATRIL Language Engineering, Madrid.

ATRIL (2008). Déjà Vu Online Help. ATRIL Language Engineering, Madrid.

Austermühl, F. (2001). *Electronic tools for translators.* St. Jerome, Manchester.

Bahagi, B. (2009). QA-Maßnahmen für Translation Memorys: Weg mit dem Datenballast! In *Jahrestagung 2009. Zusammenfassung der Referate*, pages 343–345, Stuttgart. Gesellschaft für technische Kommunikation.

Baldwin, T. (2001). Low-cost, high-performance translation retrieval: dumber is better. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 18–25, Morristown, USA. Association for Computational Linguistics. `http://www.aclweb.org/anthology-new/P/P01/P01-1004.pdf`. Last visited on 23rd June 2010.

Baldwin, T. (2010). The hare and the tortoise: speed and accuracy in translation retrieval. *Machine Translation*, 23(4):195–240.

Baldwin, T. and Tanaka, H. (2000). The effects of word order and segmentation on translation retrieval performance. In *Proceedings of the 18th conference on Computational linguistics*, pages 35–41, Morristown, USA. Association for Computational Linguistics. `http://www.aclweb.org/anthology/C/C00/C00-1006.pdf`. Last visited on 23rd June 2010.

Baun, C. and Kunze, M. (2010). Wolkige Zeiten. In *Programmieren heute*, volume 1 of *iX Special*, pages 111–116. Heise, Hannover.

Benito, D. (2009). Future Trends in Translation Memory. *Tradumàtica: traducció i tecnologies de la informació i la comunicació*, (7). `http://www.raco.cat/index.php/Tradumatica/article/viewFile/154834/206728`. Last visited on 16th February 2011.

Bernardini, S. (2001). Think-aloud protocols in translation research. *Target*, 13(2):241–263.

Biçici, E. and Dymetman, M. (2008). Dynamic Translation Memory: Using Statistical Machine Translation to Improve Translation Memory Fuzzy Matches. In Gelbukh, A., editor, *Proceedings of the 9th international conference on Computational linguistics and intelligent text processing*, pages 454–465, Haifa, Israel. Springer.

Born, A. (2010). Get off of my Cloud. In *Cloud, Grid, Virtualisierung*, volume 2 of *iX Special*, pages 16–19. Heise, Hannover.

Bourdaillet, J., Huet, S., Langlais, P., and Lapalme, G. (2010). TransSearch: from a bilingual concordancer to a translation finder. *Machine Translation*, 24(3-4):241–271.

Bowker, L. (2002). *Computer-aided translation technology: a practical introduction.* University of Ottawa Press, Ottawa.

Bowker, L. and Barlow, M. (2004). Bilingual concordancers and translation memories: A comparative evaluation. In Yuste, E., editor, *COLING 2004 Second International Workshop on Language Resources for Translation Work, Research and Training*, pages 52–61, Geneva, Switzerland. International Committee on Computational Linguistics. `http://www.mt-archive.info/Coling-2004-Bowker.pdf`. Last visited on 23rd June 2010.

Brockmann, D. (2009). Übersetzen in die Zukunft mit SDL Trados Studio 2009. In Baur, W., Kalina, S., Mayer, F., and Witzel, J., editors, *Übersetzen in die Zukunft*, pages 466–472, Berlin. Bundesverband der Dolmetscher und Übersetzer.

Brodmüller-Schmitz, A. (2003). *Microsoft Office Word 2003 – das Handbuch.* Microsoft Press, Unterschleißheim.

Camarda, B. (2003). *Special edition using Microsoft Office Word 2003.* Que, Indianapolis.

Cancedda, N., Dymetman, M., Foster, G., and Goutte, C. (2009). A Statistical Machine Translation Primer. In Goutte, C., Cancedda, N., Dymetman, M., and Foster, G., editors, *Learning Machine Translation*, pages 1–37. MIT Press, Cambridge (Massachusetts).

Carbonell, J. and Brown, R. (1999). Generalized Example-Based Machine Translation: Generalizing EBMT. `http://www.cs.cmu.edu/~ralf/ebmt/general.html`. Last visited on 8th December 2009.

Carl, M. (2009). Grounding Translation Tools in Translator's Activity Data. In *The twelfth Machine Translation Summit*, Ottawa, Canada. International Association for Machine Translation. `http://www.mt-archive.info/MTS-2009-Carl.pdf`. Last visited on 23rd June 2010.

Carl, M. and Jakobsen, A. (2009). Towards statistical modelling of translators' activity data. *International Journal of Speech Technology*, 12(4):125–138.

Carl, M., Jakobsen, A. L., and Jensen, K. T. (2008). Modelling Human Translator Behaviour with User-Activity Data. In *12th EAMT conference*, pages 21–26, Hamburg, Germany. European Association for Machine Translation. `http://www.mt-archive.info/EAMT-2008-Carl.pdf`. Last visited on 23rd June 2010.

Carl, M. and Way, A. (2003). Introduction. In Carl, M. and Way, A., editors, *Recent Advances in Example-Based Machine Translation*, pages xvii–xxxi. Kluwer Academic Publishers, Dordrecht.

Carson-Berndsen, J., Somers, H., Vogel, C., and Way, A. (2009). Integrated Language Technology as part of the Next Generation Localisation. *The International Journal of Localisation*, 8(1):53–66.

Carstensen, K.-U., Jekat, S., Ebert, C., Klabunde, R., Ebert, C., and Langer, H., editors (2010). *Computerlinguistik und Sprachtechnologie.* Spektrum, Heidelberg.

Casacuberta, F., Civera, J., Cubel, E., Lagarda, A., Lapalme, G., Macklovitch, E., and Vidal, E. (2009). Human Interaction For High-Quality Machine Translation. *Communications of the ACM*, 52(10):135–138.

Chama, Z. (2009). Arbeitserleichterung für Übersetzer. *MDÜ: Mitteilungen für Dolmetscher und Übersetzer*, 5:37–40.

Chama, Z. (2010a). Die Tücken der Tags. *MDÜ: Mitteilungen für Dolmetscher und Übersetzer*, 4:14–17.

Chama, Z. (2010b). Vom Segment zum Kontext. *technische kommunikation*, 32(2):21–25.

Champollion, Y. (2003). Convergence in CAT: Blending MT, TM, OCR & SR to boost productivity. In *Translating and the Computer 25: Proceedings of the Twenty-fifth International Conference on Translating and the Computer*, London. Aslib.

Champollion, Y. (2008a). The Road toward Collaborative Translation Memories. *The ATA Chronicle*, 37(6):30–32.

Champollion, Y. (2008b). *Wordfast User Manual.* Wordfast, Wilmington.

Circé, K. (2005). Traduction automatique, mémoire de traduction ou traduction humaine? Proposition d'une approche pour déterminer la meilleure méthode à adopter, selon le texte. Master's thesis, University of Ottawa.

Colominas, C. (2008). Towards chunk-based translation memories. *Babel*, 54(4):343–354.

Commission of the EU (1996). Benchmarking Translation Memories. `http://www.issco.unige.ch/en/research/projects/ewg96/node157.html`. Evaluation of Natural Processing Systems, Final Report (EAG-EWG-PR.2). Last visited on 22nd June 2010.

Common Sense Advisory (2011). Research. `http://www.commonsenseadvisory.com/Research.aspx`. Last visited on 28th April 2011.

de Almeida, G. and O'Brien, S. (2010). Analysing Post-Editing Performance: Correlations with Years of Translation Experience. In *Proceedings of the 14th Conference of the European Association for Machine Translation*, St. Raphael, France. European Association for Machine Translation. `http://www.mt-archive.info/EAMT-2010-Almeida.pdf`. Last visited on 12th February 2011.

Deschamps-Potter, C. (2010). Moving toward multimedia content. *MultiLingual*, 21(7):36–39.

Dietz, F. (2006). Issues in localizing computer games. In Dunne, K., editor, *Perspectives on Localization*, pages 121–134. John Benjamins, Amsterdam, Philadelphia.

Directorate-General for Translation (2010). The DGT Multilingual Translation Memory of the Acquis Communautaire: DGT-TM. `http://langtech.jrc.it/DGT-TM.html`. Last visited on 2nd June 2010.

Dockhorn, R. and Reinke, U. (2008). Wie gut spricht Ihr TM-System Dita, Xliff und Co.? In *Jahrestagung 2008. Zusammenfassung der Referate*, pages 319–320, Stuttgart. Gesellschaft für technische Kommunikation.

Doherty, S. and O'Brien, S. (2009). Can MT Output be Evaluated Through Eye Tracking? In *MT Summit XII: Proceedings of the Twelfth Machine Translation Summit*, pages 214–221, Ottawa, Canada. International Association for Machine Translation. `http://www.mt-archive.info/MTS-2009-Doherty.pdf`. Last visited on 23rd June 2010.

Doherty, S., O'Brien, S., and Carl, M. (2010). Eye tracking as an MT evaluation technique. *Machine Translation*, 24(1):1–13.

Dragsted, B. (2005). Segmentation in translation: Differences across levels of expertise and difficulty. *Target*, 17(1):49–70.

Dunne, K. (2006). A Copernican revolution. In Dunne, K., editor, *Perspectives on Localization*, pages 1–11. John Benjamins, Amsterdam, Philadelphia.

Esselink, B. (2000). *A practical guide to localization.* John Benjamins, Amsterdam, Philadelphia.

Esteban, J., Lorenzo, J., Valderrábanos, A. S., and Lapalme, G. (2004). TransType2: an innovative computer-assisted translation system. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, Morristown, USA. Association for Computational Linguistics. `http://www.mt-archive.info/ACL-2004-Esteban.pdf`. Last visited on 23rd June 2010.

Esteves-Ferreira, J., Cebulla, M., and Bauer, S. C. (2009). Urheberrechte an Translation Memories. In Baur, W., Kalina, S., Mayer, F., and Witzel, J., editors, *Übersetzen in die Zukunft*, pages 491–494, Berlin. Bundesverband der Dolmetscher und Übersetzer.

Exton, C., Wasala, A., Buckley, J., and Schäler, R. (2009). Micro Crowdsourcing: A new Model for Software Localisation. *The International Journal of Localisation*, 8(1):81–89.

Fairman, G. (2010). Checking 100% matches. *MultiLingual*, 21(4):62.

Fiederer, R. and O'Brien, S. (2009). Quality and Machine Translation: A realistic objective? *The Journal of Specialised Translation*, (11):52–74.

Fifer, M. T. (2007). The Fuzzy Factor: An Empirical Investigation of Fuzzy Matching in the Context of Translation Memory Systems. Master's thesis, University of Ottawa.

Flournoy, R. and Rueppel, J. (2010). One Technology: Many Solutions. In *Proceedings of the Ninth Conference of the Association for Machine Translation in the Americas*, Denver, USA. Association for Machine Translation in the Americas. `http://www.mt-archive.info/AMTA-2010-Flournoy.pdf`. Last visited on 10th February 2011.

Foster, G., Isabelle, P., and Plamondon, P. (1996). Word Completion: A First Step Toward Target-Text Mediated IMT. In *Proceedings of the 16th International Conference on Computational Linguistics*, pages 394–399, Copenhagen, Denmark. International Committee on Computational Linguistics. `http://acl.ldc.upenn.edu/C/C96/C96-1067.pdf`. Last visited on 26th July 2010.

Foster, G., Langlais, P., and Lapalme, G. (2002). User-friendly text prediction for translators. In *Proceedings of the Conference on Empirical Methods in Natural Language*

Processing, pages 148–155, Morristown, USA. Association for Computational Linguistics. http://www.aclweb.org/anthology/W/W02/W02-1020.pdf. Last visited on 23rd June 2010.

Foster, G. F. (2002). *Text prediction for translators*. PhD thesis, Université de Montréal, Canada.

Friedl, J. E. F. (2006). *Mastering regular expressions*. O'Reilly, Sebastopol.

Fulford, H. and Granell-Zafra, J. (2005). Translation and Technology: a Study of UK Freelance Translators. *The Journal of Specialised Translation*, (4):2–17.

García, I. (2005). Long term memories: Trados and TM turn 20. *The Journal of Specialised Translation*, (4):18–31.

García, I. (2006a). The Bottom Line: Does Text Reuse Translate into Gains in Productivity? *International Journal of technology, knowledge and society*, 2(1):103–110.

García, I. (2006b). Translators on translation memories: a blessing or a curse? In Pym, A., Perekrestenko, A., and Starink, B., editors, *Translation Technology and its Teaching*, pages 97–105. Intercultural Studies Group, Terragona.

García, I. (2007). Power shifts in web-based translation memory. *Machine Translation*, 21(1):55–68.

García, I. (2008). Translating and Revising for Localisation: What do We Know? What do We Need to Know? *Perspectives*, 16(1):49–60.

García, I. (2009a). Beyond Translation Memory: Computers and the Professional Translator. *Journal of Specialised Translation*, 12:199–214.

García, I. (2009b). Research on translation tools. In Pym, A. and Perekrestenko, A., editors, *Translation Research Projects 2*, pages 34–41. Intercultural Studies Group, Terragona.

García, I. (2010a). Is machine translation ready yet? *Target*, 22(1):7–21.

García, I. (2010b). The proper place of professionals (and non-professionals and machines) in web translation. *Tradumàtica: traducció i tecnologies de la informació i la comunicació*, (8). http://www.fti.uab.cat/tradumatica/revista/num8/articles/02/02art.htm. Last visited on 28th April 2011.

García, I. and Stevenson, V. (2006). Lingotek: Introducing a new web-based 'language search engine'. *MultiLingual*, 17(8):22–25.

García, I. and Stevenson, V. (2007). Logoport: An important move in technology application and management. *MultiLingual*, 18(6):22–25.

García, I. and Stevenson, V. (2009a). Google Translator Toolkit. *MultiLingual*, 20(6):16–18.

García, I. and Stevenson, V. (2009b). Translation trends and the social web. *MultiLingual*, 20(3):28–31.

García, I. and Stevenson, V. (2010). Déjà Vu X. *MultiLingual*, 21(1):16–19.

García, I. and Stevenson, V. (2011). MT and translating ideas. *MultiLingual*, 22(1):28–31.

Gaussier, E., Langé, J.-M., and Meunier, F. (1992). Towards Bilingual Terminology. In *Proceedings of the ALLC/ACH Conference*, pages 121–124, Oxford. Oxford University Press.

Geldbach, S. (2009). Neue Werkzeuge zur Autorenunterstützung. *MDÜ: Mitteilungen für Dolmetscher und Übersetzer*, 4:10–19.

Geldbach, S. (2010a). Ein Twitter-Stern für eine gute Übersetzung. *MDÜ: Mitteilungen für Dolmetscher und Übersetzer*, 4:39–42.

Geldbach, S. (2010b). Neues Leben für Standards. *MDÜ: Mitteilungen für Dolmetscher und Übersetzer*, 3:54.

Geldbach, S. (2010c). Schöne Open-Source-Welt. *MDÜ: Mitteilungen für Dolmetscher und Übersetzer*, 2:51–56.

Geldbach, S. (2011). Der Kunde ist König. *MDÜ: Mitteilungen für Dolmetscher und Übersetzer*, 5:38–41.

Giammarresi, S. (2007). The development of new translation technologies. A case of adaptation or adoption? Presentation given at the conference Transadaptation, Technology, Nomadism, 8-10th March 2007, Montréal, Canada.

Giménez, J. and Màrquez, L. (2010). Linguistic measures for automatic machine translation evaluation. *Machine Translation*, 24(3-4):209–240.

Gintrowicz, J. and Jassem, K. (2007). Using Regular Expressions in Translation Memories. In *Proceedings of the International Multiconference on Computer Science and Information Technology*, pages 87–92, Wisla, Poland. Polish Information Processing Society. `http://www.mt-archive.info/IMCSIT-2007-Gintrowicz.pdf`. Last visited on 24th June 2010.

Gonzalez-Barahona, J. and Peña, J. F.-S. (2008). How Debian GNU/Linux is translated into Spanish. *Tradumàtica: traducció i tecnologies de la informació i la comunicació*, (6). `http://www.fti.uab.cat/tradumatica/revista/num6/articles/07/07art.htm`. Last visited on 28th April 2011.

Gordon, I. (1997). The TM Revolution – What does it really mean? In *Translating and the Computer 19: Papers from the Aslib conference held on 13 & 14 November 1997*, London. Aslib. `http://www.mt-archive.info/Aslib-1997-Gordon.pdf`. Last visited on 24th June 2010.

Gotti, F., Langlais, P., Macklovitch, E., Bourigault, D., Robichaud, B., and Coulombe, C. (2005). 3GTM: A Third-Generation Translation Memory. In *3rd Computational Linguistics in the North-East (CLiNE) Workshop*, pages 8–15, Gatineau, Canada. Université du Québec. `http://www.iro.umontreal.ca/~felipe/Papers/paper-cline-3gtm-2005.pdf`. Last visited on 24th June 2010.

Gough, J. (2010). A troubled relationship: the compatibility of CAT tools. `http://www.translationautomation.com/technology/a-troubled-relationship-the-compatibility-of-cat-tools.html`. Last visited on 12th February 2011.

Gow, F. (2003). Metrics for Evaluating Translation Memory Software. Master's thesis, University of Ottawa, Ottawa.

Goyvaerts, J. (2007). *Regular Expressions*. `http://www.regular-expressions.info/print.html`. Last visited on 16th September 2010.

Goyvaerts, J. (2010a). RegexBuddy. `http://www.regexbuddy.com/`. Last visited on 18th September 2010.

Goyvaerts, J. (2010b). *RegexBuddy*. Just Great Software, Rawai Phuket.

Goyvaerts, J. and Levithan, S. (2009). *Regular expressions cookbook*. O'Reilly, Sebastopol.

Graliński, F., Jasem, K., and Marcińczuk, M. (2009). An Environment for Named Entity Recognition and Translation. In *Proceedings of the 13th Annual Conference of the EAMT*, pages 88–95, Barcelona, Spain. European Association for Machine Translation. `http://www.mt-archive.info/EAMT-2009-Gralinski.pdf`. Last visited on 12th December 2010.

Grefenstette, G. and Tapanainen, P. (1994). What is a word, What is a sentence? Problems of Tokenization. In *Proceedings of COMPLEX-94: 3rd Conference on Computational Lexicography and Text Research*, pages 79–87, Budapest, Hungary. Research Institute for Linguistics, Hungarian Academy of Sciences.

Gross, M. (1997). The Construction of Local Grammars. In Roche, E. and Schabes, Y., editors, *Finite-State Language Processing*, pages 329–354. MIT Press, Cambridge (Massachusetts).

Groves, D. (2008). Bringing Humans into the Loop: Localization with Machine Translation at Traslán. In *Proceedings of the Eighth Conference of the Association for Machine*

Translation in the Americas, pages 11–22, Waikiki, USA. Association for Machine Translation in the Americas. `http://www.traslan.ie/traslan-amta08.pdf`. Last visited on 24th June 2010.

Grünwied, G. (2006). Gelungene Benutzerführung in Online-Hilfen dank Eye-Tracking. In *Jahrestagung 2006. Zusammenfassung der Referate*, pages 122–126, Stuttgart. Gesellschaft für technische Kommunikation.

Guerberof, A. (2008). Post-editing MT and TM: a Spanish case. *MultiLingual*, 19(6):45–50.

Guillardeau, S. (2009). Freie Translation Memory Systeme für die Übersetzungspraxis: Ein kritischer Vergleich. Master's thesis, Universität Wien.

Haralambous, Y. (2007). *Fonts & encodings*. O'Reilly, Sebastopol.

Härtinger, H. (2009). Textsortenbezogene linguistische Untersuchungen zum Einsatz von Translation-Memory-Systemen an einem Korpus deutscher und spanischer Patentschriften. *Journal for Language Technology and Computational Linguistics*, 24(3):87–112.

Härtinger, H. (2010). Formulierungen verwalten. *MDÜ: Mitteilungen für Dolmetscher und Übersetzer*, 4:34–38.

He, Y., Ma, Y., Roturier, J., Way, A., and van Genabith, J. (2010). Improving the Post-Editing Experience using Translation Recommendation: A User Study. In *Proceedings of the Ninth Conference of the Association for Machine Translation in the Americas*, Denver, USA. Association for Machine Translation in the Americas. `http://amta2010.amtaweb.org/AMTA/papers/2-27-HeMaEtal.pdf`. Last visited on 10th February 2011.

Hearne, M. and Way, A. (2011). Statistical Machine Translation: A Guide for Linguists and Translators. *Language and Linguistics Compass*, 5(5):205–226.

Heartsome (2008). Heartsome Translation Studio Online Help. Heartsome, Hong Kong.

Herrmann, A. (2011). Geeignete Werkzeuge. *MDÜ: Mitteilungen für Dolmetscher und Übersetzer*, 2:20–24.

Hewavitharana, S., Vogel, S., and Waibel, A. (2005). Augmenting a Statistical Translation System with a Translation Memory. In *Proceedings of the 10th Conference of the European Association for Machine Translation*, pages 126–132, Budapest, Hungary. European Association for Machine Translation. `http://www.mt-archive.info/EAMT-2005-Hewavitharana.pdf`. Last visited on 12th December 2010.

Hodász, G. (2006a). Evaluation methods of a linguistically enriched translation memory system. In *Proceedings of LREC-2006: Fifth International Conference on Language Resources and Evaluation*, pages 2044–2047, Genoa, Italy. European Language Resources Association. `http://www.mt-archive.info/LREC-2006-Hodasz.pdf`. Last visited on 24th June 2010.

Hodász, G. (2006b). Towards a comprehensive evaluation method of Memory-Based Translation Systems. In *Proceedings of the 11th Annual Conference of the European Association for Machine Translation*, pages 213–217, Oslo, Norway. European Association for Machine Translation. `http://www.mt-archive.info/EAMT-2006-Hodasz.pdf`. Last visited on 24th June 2010.

Hodász, G., Grőbler, T., and Kis, B. (2004). Translation memory as a robust example-based translation system. In Hutchins, J., editor, *Proceedings of the 9th EAMT Workshop "Broadening horizons of machine translation and its applications"*, pages 82–89, La Valletta, Malta. EAMT, Foundation for International Studies.

Höge, M. (2002). *Towards a Framework for the Evaluation of Translators' Aids' Systems.* PhD thesis, University of Helsinki.

Huddleston, R. D. and Pullum, G. K., editors (2002). *The Cambridge grammar of the English language.* Cambridge University Press, Cambridge.

Hudík, T. and Ruopp, A. (2011). The Integration of Moses into Localization Industry. In Forcada, M. L., Depraetere, H., and Vandeghinste, V., editors, *Proceedings of the 15th Conference of the European Association for Machine Translation*, pages 47–53, Leuven, Belgium. European Association for Machine Translation. `www.mt-archive.info/EAMT-2011-Hudik.pdf`. Last visited on 11th October 2011.

Hunt, T. (2003). Translation Technology Failures and Future. *Globalization Insider*, 12(1.4).

Hutchins, J. (1998). The origins of the translator's workstation. *Machine Translation*, 13(4):287–307.

Hutchins, J. (2005). Current commercial machine translation systems and computer-based translation tools: system types and their uses. *International Journal of Translation*, 17(1-2):5–38.

Hutchins, J. and Somers, H. (1992). *An Introduction to Machine Translation.* Academic Press, London.

IETF (1982). RFC 821: Simple Mail Transfer Protocol. `http://www.ietf.org/rfc/rfc0821.txt`. Last visited on 22nd June 2010.

IETF (1987). RFC 1034: Domain Concepts and Facilities. `http://www.ietf.org/rfc/rfc1034.txt`. Last visited on 22nd June 2010.

IETF (1994). RFC 1738: Uniform Resource Locators (URL). `http://www.ietf.org/rfc/rfc1738.txt`. Last visited on 22nd June 2010.

IETF (1995). RFC 1808: Relative Uniform Resource Locators. `http://www.ietf.org/rfc/rfc1808.txt`. Last visited on 22nd June 2010.

IETF (2001). RFC 2822: Internet Message Format. `http://www.ietf.org/rfc/rfc2822.txt`. Last visited on 20th September 2010.

IETF (2003a). RFC 3490: Internationalizing Domain Names in Applications (IDNA). `http://www.ietf.org/rfc/rfc3490.txt`. Last visited on 22nd June 2010.

IETF (2003b). RFC 3492: Punycode: A Bootstring encoding of Unicode for Internationalized Domain Names in Applications (IDNA). `http://www.ietf.org/rfc/rfc3492.txt`. Last visited on 22nd June 2010.

IETF (2005). RFC 3986: Uniform Resource Identifier (URI): Generic Syntax. `http://www.ietf.org/rfc/rfc3986.txt`. Last visited on 22nd June 2010.

IETF (2008a). RFC 5234: Augmented BNF for Syntax Specifications: ABNF. `http://www.ietf.org/rfc/rfc5234.txt`. Last visited on 22nd June 2010.

IETF (2008b). RFC 5321: Simple Mail Transfer Protocol. `http://www.ietf.org/rfc/rfc5321.txt`. Last visited on 22nd June 2010.

IETF (2008c). RFC 5322: Internet Message Format. `http://www.ietf.org/rfc/rfc5322.txt`. Last visited on 22nd June 2010.

IETF (2008d). RFC 5335: Internationalized Email Headers. `http://www.ietf.org/rfc/rfc5335.txt`. Last visited on 22nd June 2010.

IETF (2008e). RFC 5336: SMTP Extension for Internationalized Email Addresses. `http://www.rfc-editor.org/rfc/rfc5336.txt`. Last visited on 1st March 2011.

Ikonomidis, A. (2011). Mit Strichen eine saubere Linie fahren. *technische kommunikation*, 33(2):38–42.

ISO/IEC (1991). ISO/IEC 9126 Information Technology – Software Product Evaluation, Quality Characteristics and Guidelines for their Use. Geneva.

Jääskeläinen, R. (2002). Think-aloud protocol studies into translation. *Target*, 14(1):107–136.

Jakobsen, A. L. (2003). Effects of think aloud on translation speed, revision, and segmentation. In Alves, F., editor, *Triangulating translation*, pages 69–95. John Benjamins, Amsterdam, Philadelphia.

Jakobsen, A. L. (2006). Research Methods in Translation – Translog. In Sullivan, K. P. H., editor, *Computer Keystroke Logging and Writing*, pages 95–105. Elsevier, Oxford.

Jelinek, R. (2004). Modern MT Systems and the Myth of Human Translation: Real World Status Quo. In *Translating and the Computer 26. Proceedings of the Twenty-sixth International Conference on Translating and the Computer*, London. Aslib.

Joscelyne, A. (2009a). Putting language data sharing to work. http://www.translationautomation.com/user-cases/putting-language-data-sharing-to-work.html. Last visited on 15th August 2010.

Joscelyne, A. (2009b). Taking the MT decision. http://www.translationautomation.com/best-practices/taking-the-mt-decision-selection-build-out-and-hosting.html. Last visited on 15th August 2010.

Joscelyne, A. and van der Meer, J. (2007a). Translation 2.0: market forces. *MultiLingual*, 18(1):26–27.

Joscelyne, A. and van der Meer, J. (2007b). Translation 2.0: technology. *MultiLingual*, 18(2):36–37.

Joscelyne, A. and van der Meer, J. (2007c). Translation 2.0: transmutation. *MultiLingual*, 18(3):30–31.

Joy, L. (2002). Translating tagged text – imperfect matches and a good finished job. In *Translating and the Computer 24: Proceedings of the Twenty-fourth International Conference on Translating and the Computer*, London. Aslib.

Junginger, M. (2011). Androiden auf dem Vormarsch. In *Programmieren heute*, volume 1 of *iX Special*, pages 18–24. Heise, Hannover.

Jüngst, H. E. (2010). *Audiovisuelles Übersetzen*. Narr Francke Attempto, Tübingen.

Kanavos, P. and Kartsaklis, D. (2010). Integrating Machine Translation with Translation Memory: A Practical Approach. In Zhechev, V., editor, *Proceedings of the Second Joint EM+/CNGL Workshop Bringing MT to the User: Research on Integrating MT in the Translation Industry*, pages 11–20, Denver, USA. EuroMatrixPlus/Centre for Next Generation Localisation. http://web.me.com/emcnglworkshop/JEC2010_Kanavos_and_Kartsaklis.pdf. Last visited on 12th February 2011.

Karamanis, N., Luz, S., and Doherty, G. (2010). Translation practice in the workplace and Machine Translation. In *Proceedings of the 14th Conference of the European Association for Machine Translation*, St. Raphael, France. European Association for Machine Translation. http://www.mt-archive.info/EAMT-2010-Karamanis.pdf. Last visited on 13th February 2011.

Karamanis, N., Luz, S., and Doherty, G. (2011). Translation practice in the workplace: contextual analysis and implications for machine translation. *Machine Translation*, 25(1):35–52.

Karttunen, L., Chanod, J.-P., Grefenstette, G., and Schiller, A. (1996). Regular expressions for language engineering. *Natural Language Engineering*, 2(4):305–328.

Keller, N. (2011). Neun auf einen Blick. *MDÜ: Mitteilungen für Dolmetscher und Übersetzer*, 5:16–31.

Kenning, M.-M. (2010). What are parallel and comparable corpora and how can we use them? In O'Keeffe, A. and McCarthy, M., editors, *The Routledge Handbook of Corpus Linguistics*, pages 487–500. Routledge, New York.

Khadivi, S. (2008). *Statistical Computer-Assisted Translation.* PhD thesis, Rheinisch-Westfälische Technische Hochschule Aachen.

Kilgray (2008). memoQ Online Help. Kilgray Translation Technologies, Budapest.

King, M. and Maegaard, B. (1998). Issues in Natural Language Systems Evaluation. In *Proceedings of the First International Conference on Language Resources and Evaluation (LREC)*, volume 1, pages 225–230, Granada, Spain. European Language Resources Association.

Kleinophorst, I. (2010). Untersuchung zum Einsatz von Translation-Memory-Systemen in der Technischen Redaktion. Master's thesis, Hochschule Merseburg.

Koehn, P. (2009a). A process study of computer-aided translation. *Machine Translation*, 23(4):241–263.

Koehn, P. (2009b). A Web-Based Interactive Computer Aided Translation Tool. In *Proceedings of the ACL-IJCNLP 2009 Software Demonstrations*, pages 17–20, Suntec, Singapore. Association for Computational Linguistics. `http://aclweb.org/anthology/P/P09/P09-4005.pdf`. Last visited on 24th June 2010.

Koehn, P. (2010). *Statistical Machine Translation.* Cambridge University Press, Cambridge.

Koehn, P. and Haddow, B. (2009). Interactive Assistance to Human Translators using Statistical Machine Translation Methods. In *Machine Translation Summit XII*, Ottawa, Canada. Association for Machine Translation in the Americas. `http://www.mt-archive.info/MTS-2009-Koehn-2.pdf`. Last visited on 24th June 2010.

Koehn, P. and Senellart, J. (2010). Convergence of Translation Memory and Statistical Machine Translation. In Zhechev, V., editor, *Proceedings of the Second Joint EM+/CNGL Workshop Bringing MT to the User: Research on Integrating MT in the Translation Industry*, pages 21–31, Denver, USA. EuroMatrixPlus/Centre for Next Generation Localisation. `http://www.mt-archive.info/JEC-2010-Koehn.pdf`. Last visited on 12th February 2011.

Koester, A. (2010). Building small specialised corpora. In O'Keeffe, A. and McCarthy, M., editors, *The Routledge Handbook of Corpus Linguistics*, pages 66–79. Routledge, New York.

Korpela, J. (2006). *Unicode explained.* O'Reilly, Sebastopol.

Kranias, L. and Samiotou, A. (2004). Automatic Translation Memory Fuzzy Match Post-Editing: A Step beyond Traditional TM/MT Integration. In *LREC-2004: Proceedings of the Fourth International Conference on Language Resources and Evaluation*, pages 331–334, Lisbon, Portugal. European Language Resources Association. `http://www.mt-archive.info/LREC-2004-Kranias.pdf`. Last visited on 25th June 2010.

Kreckwitz, S. (2007). Future Web-Based Translation Environments. Presentation held at the Localisation Research Forum (Dublin), available at `http://www.across.net/presentations/LRC_070928_sk.pdf`. Last visited on 27th January 2011.

Kreimeier, U. (2010). Schöne neue Datenwelt. *MDÜ: Mitteilungen für Dolmetscher und Übersetzer*, 4:44–47.

Krenz, M. (2008). XML im Übersetzungsprozess. In Seewald-Heeg, U., editor, *Maschinelle Übersetzung und XML im Übersetzungsprozess*, pages 151–360. Frank & Timme, Berlin.

Krings, H. P. (1994). *Texte reparieren. Empirische Untersuchungen zum Prozess der Nachredaktion von Maschinenübersetzungen.* Universität Hildesheim, Hildesheim.

Kübler, N. and Aston, G. (2010). Using corpora in translation. In O'Keeffe, A. and McCarthy, M., editors, *The Routledge Handbook of Corpus Linguistics*, pages 501–515. Routledge, New York.

Kuhn, R., Isabelle, P., Goutte, C., Senellart, J., Simard, M., and Ueffing, N. (2010). Automatic post-editing. *MultiLingual*, 21(2):43–46.

Lagoudaki, E. (2006). Translation Memory systems: Enlightening users' perspective. `http://www3.imperial.ac.uk/portal/pls/portallive/docs/1/7307707.PDF`. Last visited on 22nd June 2010.

Lagoudaki, E. (2009). Translation Editing Environments. In *Machine Translation Summit XII*, Ottawa, Canada. International Association for Machine Translation. `http://www.mt-archive.info/MTS-2009-Lagoudaki.pdf`. Last visited on 25th June 2010.

Lagoudaki, P. M. (2008). *Expanding the Possibilities of Translation Memory Systems.* PhD thesis, Imperial College of London.

Langé, J.-M., Gaussier, E., and Daille, B. (1997). Bricks and Skeletons: Some Ideas for the Near Future of MAHT. *Machine Translation*, 12(1/2):39–51.

Langlais, P., Loranger, M., and Lapalme, G. (2002). Translators at Work with TransType: Resource and Evaluation. In *Proceedings of the International Conference on Language Resources and Evaluation*, pages 2128–2134, Las Palmas, Spain. European Language Resources Association. `http://www.iro.umontreal.ca/~lapalme/Publications/Confs/LanglaisTranslatorsLREC02.pdf`. Last visited on 25th June 2010.

Lee, D. (2010). What corpora are available? In O'Keeffe, A. and McCarthy, M., editors, *The Routledge Handbook of Corpus Linguistics*, pages 107–121. Routledge, New York.

Lemnitzer, L. and Zinsmeister, H. (2010). *Korpuslinguistik*. Narr, Tübingen.

Levitt, G. (2003). Internet-based sharing of translation memory. New solutions that are changing translation and localization workflows. *MultiLingual*, 14(5):38–41.

Levitt, G. (2007). Toward a cleaner information environment. *tcworld*, (3):18–22.

Li, D. (2004). Trustworthiness of think-aloud protocols in the study of translation processes. *International Journal of Applied Linguistics*, 4(3):301–313.

Lieske, C. (2011). Insights into the future of XLIFF. *MultiLingual*, 22(5):51–52.

Lieske, C. and Sasaki, F. (2009). Standards-based Translation with W3C ITS and OASIS XLIFF. In *Jahrestagung 2009. Zusammenfassung der Referate*, pages 160–162, Stuttgart. Gesellschaft für technische Kommunikation.

Linguee (2011). About Linguee. `http://www.linguee.com/english-german/page/about.php`. Last visited on 2nd May 2011.

LISA (2005). TMX 1.4b Specification. `http://www.lisa.org/fileadmin/standards/tmx1.4/tmx.htm`. Last visited on 29th July 2010.

Lloyd, I. (2008). *The ultimate HTML reference*. Sitepoint, Collingwood.

Lommel, A. (2006). Localization standards, knowledge- and information-centric business models, and the commodization of linguistic information. In Dunne, K., editor, *Perspectives on Localization*, pages 223–239. John Benjamins, Amsterdam, Philadelphia.

Lopez, A. (2008). Statistical Machine Translation. *ACM Computing Surveys*, 40(3):8:1–8:49.

Loshin, P. (2000). *Essential email standards: RFCs and protocols made practical*. Wiley, New York.

Lutz, H.-D. (2001). Software-ergonomische Entwicklung – eine Herausforderung für die Computerlinguistik. *Sprache und Datenverarbeitung*, 1:5–20.

Macken, L. (2009). In search of the recurrent units of translation. In Daelemans, W. and Hoste, V., editors, *Evaluation of Translation Technology*, number 8 in Linguistica Antverpiensia New Series, pages 195–212. Academic and Scientific Publishers, Brussels.

Macken, L. (2010). *Sub-sentential alignment of translational correspondences*. PhD thesis, Universiteit Antwerpen. `http://lt3.hogent.be/media/uploads/publications/2010/Macken2010b.pdf`. Last visited on 23rd March 2011.

Macklovitch, E. (1995). TransCheck – or the Automatic Validation of Human Translations. In *Proceedings of the MT Summit V*, Luxembourg. International Association for Machine Translation. http://www.mt-archive.info/MTS-1995-Macklovitch.pdf. Last visited on 25th June 2010.

Macklovitch, E. (2000). Two Types of Translation Memory. In *Proceedings of the Twenty-second International Conference on Translating and the Computer*, London. Aslib. http://www.mt-archive.info/Aslib-2000-Macklovitch.pdf. Last visited on 25th June 2010.

Macklovitch, E. (2006). TransType2: The Last Word. In *Proceedings of the 5th International Conference on Languages Resources and Evaluation*, pages 167–172, Genoa, Italy. European Language Resources Association. http://www.mt-archive.info/LREC-2006-Macklovitch.pdf. Last visited on 25th June 2010.

Macklovitch, E., Lapalme, G., and Chan, N. (2009). Term-spotting with TransCheck: A Capital Idea. In Alcina, A. and L'Homme, M.-C., editors, *First International Workshop on Terminology and Lexical Semantics*, pages 3–12, Montréal, Canada. Université de Montréal. http://olst.ling.umontreal.ca/pdf/ProceedingsTLS09.pdf. Last visited on 25th June 2010.

Macklovitch, E. and Russell, G. (2000). What's Been Forgotten in Translation Memory. In *Proceedings of the 4th Conference of the Association for Machine Translation in the Americas on Envisioning Machine Translation in the Information Future*, volume 1934 of *Lecture Notes In Computer Science*, pages 137–146, London. Springer.

Makoushina, J. (2008). A comparison of eight quality assurance tools. *MultiLingual*, 19(4):52–56.

Manning, C. D., Raghavan, P., and Schütze, H. (2009). *Introduction to Information Retrieval*. Cambridge University Press, Cambridge. Online edition available at http://nlp.stanford.edu/IR-book/information-retrieval-book.html. Last visited on 10th October 2010.

Manning, C. D. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge (Massachusetts).

Massion, F. (2005). *Translation-Memory-Systeme im Vergleich*. Doculine, Reutlingen.

Massion, F. (2009). Aufsteigender Stern aus dem Osten. *MDÜ: Mitteilungen für Dolmetscher und Übersetzer*, 4:24–28.

Massion, F. (2010). Automating translation quality management. *MultiLingual*, 21(1):41–43.

Massion, F. (2011). Ist die Zeit reif für maschinelle Übersetzung? *technische kommunikation*, 33(5):19–26.

Mateos, J. (2011). Is XLIFF positioned correctly? *MultiLingual*, 22(2):48–52.

Matusov, E., Leusch, G., and Ney, H. (2009). Learning to Combine Machine Translation Systems. In Goutte, C., Cancedda, N., Dymetman, M., and Foster, G., editors, *Learning Machine Translation*, pages 257–276. MIT Press, Cambridge (Massachusetts).

Maynard, D., Tablan, V., Ursu, C., Cunningham, H., and Wilks, Y. (2001). Named Entity Recognition from Diverse Text Types. In *Recent Advances in Natural Language Processing 2001 Conference*, pages 257–274, Tzigov Chark, Bulgaria. Bulgarian Academy of Sciences. `http://gate.ac.uk/sale/ranlp2001/maynard-etal.pdf`. Last visited on 25th June 2010.

McBride, C. (2009). Translation Memory Systems: An Analysis of Translation Attitudes and Opinions. Master's thesis, University of Ottawa.

McTait, K., Olohan, M., and Trujillo, A. (1999). A Building Blocks Approach to Translation Memory. In *Translating and the Computer 21. Proceedings of the 21st ASLIB International Conference on Translating and the Computer*, London. Aslib. `http://www.mt-archive.info/Aslib-1999-McTait.pdf`. Last visited on 25th June 2010.

Melby, A. (2006). MT+TM+QA: The Future is Ours. *Tradumàtica: traducció i tecnologies de la informació i la comunicació*, (4). `http://www.fti.uab.cat/tradumatica/revista/num4/articles/04/04.pdf`. Last visited on 18th February 2011.

Mercier, P. (2003). Do TM Systems Compare Apples and Pears? *Localisation Focus*, 2(2):14–17.

Mikheev, A. (1999). Periods, Capitalized Words, etc. *Computational Linguistics*, 28(3):289–318.

Mikheev, A. (2003). Text segmentation. In Mitkov, R., editor, *The Oxford Handbook of Computational Linguistics*, pages 201–218. Oxford University Press, Oxford, New York.

Miller, K. S. and Sullivan, K. P. H. (2006). Keystroke Logging: An Introduction. In Sullivan, K. P. H., editor, *Computer Keystroke Logging and Writing*, pages 1–9. Elsevier, Oxford.

Mitschke, U. (2010). Praxistipps FrameMaker: Störungsfrei übersetzen. *MDÜ: Mitteilungen für Dolmetscher und Übersetzer*, 2:29–32.

Montserrat, S. F. and Balonés, L. (2009). What's the Deal with Punctuation? In *50th ATA Annual Conference*, New York, USA. American Translators Association.

Mozilla Foundation (2010). Download a Firefox version that speaks your language. `http://www.mozilla.com/en-US/firefox/all.html`. Last visited on 23rd May 2010.

Muegge, U. (2010). Ten good reasons for using a translation memory. *tcworld*. http://www.tcworld.info/tcworld/translation-and-localization/article/ten-good-reasons-for-using-a-translation-memory/. Last visited on 11th October 2011.

Müller, H., Amerl, V., and Natalis, G. (1980). Worterkennungsverfahren als Grundlage einer Universalmethode zur automatischen Segmentierung von Texten in Sätze. Ein Verfahren zur maschinellen Satzgrenzenbestimmung im Englischen. *Sprache und Datenverarbeitung*, 1:46–63.

MultiCorpora (2008). *MultiTrans Translator Training Guide*. MultiCorpora R&D, Gatineau.

Münz, S. and Nefzger, W. (2005). *HTML-Handbuch*. Franzis, Poing.

Musciano, C. (2006). *HTML & XHTML: the definitive guide*. O'Reilly, Sebastopol.

Nagel, S., Hezel, S., Hinderer, K., and Pieper, K. (2009). *Audiovisuelle Übersetzung*. Peter Lang, Frankfurt am Main.

Nedoma, A. and Nedoma, J. (2004). Problems with CAT tools related to translations into Central and Eastern European Languages. In *Translating and the Computer 26. Proceedings of the Twenty-sixth International Conference on Translating and the Computer*, London. Aslib.

Nelson, M. (2010). Building a written corpus. In O'Keeffe, A. and McCarthy, M., editors, *The Routledge Handbook of Corpus Linguistics*, pages 53–65. Routledge, New York.

Nirenburg, S., Somers, H., and Wilks, Y., editors (2002). *Readings in Machine Translation*. The MIT Press, Cambridge (Massachusetts).

Nogueira, D. and Semolini, K. (2010). Will We Be Here Tomorrow? *Translation Journal*, 14(3). http://translationjournal.net/journal/53mt.htm. Last visited on 17th August 2010.

Nübel, R. and Seewald-Heeg, U. (1999a). Ausblick. *LDV-Forum*, 16(1/2):118–121.

Nübel, R. and Seewald-Heeg, U. (1999b). Translation-Memory-Module automatischer Übersetzungssysteme. *LDV-Forum*, 16(1/2):16–35.

O'Brien, S. (2004). Machine Translatability and Post-Editing Effort: How do they relate? In *Translating and the Computer 26. Proceedings of the Twenty-sixth International Conference on Translating and the Computer*, London. Aslib.

O'Brien, S. (2007). Eye-Tracking and Translation Memory Matches. *Perspectives*, 14(3):185–205.

Och, F. J. (2002). *Statistical Machine Translation: From Single-Word Models to Alignment Templates*. PhD thesis, Rheinisch-Westfälische Technische Hochschule Aachen. http://sylvester.bth.rwth-aachen.de/dissertationen/2003/059/03_059.pdf. Last visited on 26th July 2010.

Odcházel, O. and Bojar, O. (2009). Computer-Aided Translation Backed by Machine Translation. http://ufal.mff.cuni.cz/~bojar/publications/2009-FILE-odchazel_bojar_2009-2009-trcomp-abstract.pdf. Last visited on 22nd June 2010.

Oehmig, P. (2006). Übersetzung – einwandfrei und reibungslos. *technische kommunikation*, 29(5):50–52.

Ortiz-Martínez, D., Leiva, L. A., Alabau, V., and Casacuberta, F. (2010). Interactive machine translation using a web-based architecture. In *Proceeding of the 14th international conference on Intelligent user interfaces*, pages 423–425, New York. Association for Computing Machinery. http://smt.speedzinemedia.com/smt/docs/IUI2010.pdf. Last visited on 25th June 2010.

Ottmann, A. (2004). *Translation-Memory-Systeme: Nutzen, Risiken, erfolgreiche Anwendung*. GFT, Schenkenzell.

Oxford University Press (2009). Oxford English Dictionary Online. http://www.oed.com/. Last visited on 22nd June 2010.

Palmer, M. and Xue, N. (2010). Linguistic Annotation. In Clark, A., Fox, C., and Lappin, S., editors, *The Handbook of Computational Linguistics and Natural Language Processing*, pages 238–270. Wiley-Blackwell, Chichester.

Panzer, M., Lehmann, S., and Hausmann, S. (2010). Termine und Prozesse im Griff. *MDÜ: Mitteilungen für Dolmetscher und Übersetzer*, 3:38–42.

Pelster, U. (2011). XML für den passenden Zweck. *technische kommunikation*, 33(1):54–57.

Perrino, S. (2009). User-generated Translation: The future of translation in a Web 2.0 environment. *The Journal of Specialised Translation*, (12):55–78.

Phillips, L. A. (2000). *Special edition using XML*. Que, Indianapolis.

Piperidis, S., Malavazos, C., and Triantafyllou, I. (1999). A Multi-level Framework for Memory-Based Translation Aid Tools. In *Translating and the Computer 21. Proceedings of the 21st ASLIB International Conference on Translating and the Computer*, London. Aslib. http://www.mt-archive.info/Aslib-1999-Piperidis.pdf. Last visited on 25th June 2010.

Planas, E. (1998). *Structures et algorithmes pour la traduction fondée sur la mémoire*. PhD thesis, Université de Grenoble 1.

Planas, E. (2000). Extending Translation Memories. In *Proceedings of the Fifth EAMT Workshop, Harvesting Existing Resources*, Ljubljana, Slovenia. European Association for Machine Translation. `http://nl.ijs.si/eamt00/proc/Planas.pdf`. Last visited on 25th June 2010.

Plitt, M. and Masselot, F. (2010). A Productivity Test of Statistical Machine Translation Post-Editing in a Typical Localisation Context. *The Prague Bulletin of Mathematical Linguistics*, (93):7–16. `http://www.mtmarathon2010.info/web/Program_files/art-plitt-masselot.pdf`. Last visited on 16th February 2011.

Popović, M. (2009). *Machine translation: statistical approach with additional linguistic knowledge*. PhD thesis, Rheinisch-Westfälische Technische Hochschule Aachen.

Porsiel, J. (2009). Übersetzungsprozesse und maschinengestütztes Übersetzen. In Baur, W., Kalina, S., Mayer, F., and Witzel, J., editors, *Übersetzen in die Zukunft*, pages 418–425, Berlin. Bundesverband der Dolmetscher und Übersetzer.

Porsiel, J. (2011). MT data security. *MultiLingual*, 22(1):35–36.

Pouliquen, B. and Steinberger, R. (2009). Automatic Construction of Multilingual Name Dictionaries. In Goutte, C., Cancedda, N., Dymetman, M., and Foster, G., editors, *Learning Machine Translation*, pages 59–78. MIT Press, Cambridge (Massachusetts).

Prasad, R. and Sarkar, A. (2000). Comparing test-suite based evaluation and corpus-based evaluation of a wide-coverage grammar for English. In *Proceedings of LREC 2000 Satellite Workshop: Using Evaluation with HLT Programs: Results and Trends*, pages 7–12, Athens, Greece. European Language Resources Association. `http://www.cis.upenn.edu/~rjprasad/papers/hlt-eval.pdf`. Last visited on 25th June 2010.

Prestifilippo, C. (2002). Die Problematik der Lokalisierung Technischer Dokumentationen: typografische und landesspezifische Anpassungen. In *Jahrestagung 2002. Zusammenfassung der Referate*, pages 93–95, Stuttgart. Gesellschaft für technische Kommunikation.

ProZ (2010a). How to have Trados concentrate on relevant text rather than tag material for finding matches. `http://deu.proz.com/forum/sdl_trados_support/183991-how_to_have_trados_concentrate_on_relevant_text_rather_than_tag_material_for_finding_matches.html`. Last visited on 1st November 2010.

ProZ (2010b). Studio SP2: fuzzy matches make no sense. `http://fra.proz.com/forum/sdl_trados_support/163876-studio_sp2:_fuzzy_matches_make_no_sense.html`. Last visited on 1st November 2010.

ProZ (2011). Create generic placeables. `http://deu.proz.com/forum/sdl_trados_support/190271-create_generic_placeables.html`. Last visited on 26th January 2011.

Puscher, F. (2011). Reicher surfen. In *Software-Qualität: Bessere Programme sind machbar*, volume 1 of *iX Kompakt*, pages 24–29. Heise, Hannover.

Pym, A. (2004). *The moving text: localization, translation, and distribution.* John Benjamins, Amsterdam, Philadelphia.

Quah, C. K. (2006). *Translation and Technology.* Palgrave Macmillan, Basingstoke.

Raybaud, S., Langlois, D., and Smaïli, K. (2011). "This sentence is wrong." Detecting errors in machine-translated sentences. *Machine Translation*, 25(1):1–34.

Reddy, A. and Rose, R. (2010). Integration of Statistical Models for Dictation of Document Translations in a Machine-Aided Human Translation Task. *IEEE transactions on audio, speech and language processing*, 18(8):2015–2027.

Reinke, U. (1999). Evaluierung der linguistischen Leistungsfähigkeit von Translation-Memory-Systemen – ein Erfahrungsbericht. *LDV Forum*, 16(1/2):100–117.

Reinke, U. (2004). *Translation memories: Systeme – Konzepte – linguistische Optimierung.* Saarbrücker Beiträge zur Sprach- und Translationswissenschaft. Peter Lang, Frankfurt am Main.

Reinke, U. (2006). Translation Memories. In Brown, K., editor, *Encyclopedia of Language & Linguistics*, pages 61–65. Elsevier, Oxford.

Reinke, U. (2008). XML-Unterstützung in Translation-Memory-Systemen. In *Jahrestagung 2008. Zusammenfassung der Referate*, pages 389–391, Stuttgart. Gesellschaft für technische Kommunikation.

Reinke, U. (2009). Computergestützte Werkzeuge zur Qualitätssicherung, -kontrolle und -messung. In Baur, W., Kalina, S., Mayer, F., and Witzel, J., editors, *Übersetzen in die Zukunft*, pages 170–179, Berlin. Bundesverband der Dolmetscher und Übersetzer.

Reinke, U. and Höflich, I. (2002). Translation-Memory-Systeme – Vergleich der unterschiedlichen Konzepte. In *Jahrestagung 2002. Zusammenfassung der Referate*, pages 283–288, Stuttgart. Gesellschaft für technische Kommunikation.

Resnik, P. and Lin, J. (2010). Evaluation of NLP Systems. In Clark, A., Fox, C., and Lappin, S., editors, *The Handbook of Computational Linguistics and Natural Language Processing*, pages 271–295. Wiley-Blackwell, Chichester.

Rico, C. (2000). Evaluation Metrics for Translation Memories. *Language International*, 12(6):36–37.

Rico, C. (2001). Reproducible models for CAT tools evaluation: A user-oriented perspective. In *Translating & the Computer 23: Proceedings of the Twenty-third International Conference on Translating and the Computer*, London. Aslib. `http://www.mt-archive.info/Aslib-2001-Rico.pdf`. Last visited on 25th June 2010.

Riggio, F. (2010). Dubbing vs. subtitling. *MultiLingual*, 21(7):31–35.

Rinsche, A. and Zanotti, N. P. (2009). Study on the size of the language industry in the EU. `http://ec.europa.eu/dgs/translation/publications/studies/size_of_language_industry_en.pdf`. Last visited on 19th June 2010.

Ritter, R. M., editor (2005). *New Hart's rules.* Oxford University Press, Oxford, New York.

Rösener, C. (2010). Computational Linguistics in the Translator's Workflow – Combining Authoring Tools and Translation Memory Systems. In *Proceedings of the NAACL Human Language Technologies 2010 Workshop on Computational Linguistics and Writing*, pages 1–6, Los Angeles, USA. Association for Computational Linguistics. `http://www.aclweb.org/anthology/W/W10/W10-0401.pdf`. Last visited on 16th February 2011.

Russell, G. (2005). Automatic Detection of Translation Errors: The TransCheck System. In *Translating and the Computer 27: Proceedings of the Twenty-seventh International Conference on Translating and the Computer*, London. Aslib.

Sager, J. C. (1994). *Language engineering and translation: Consequences of automation.* John Benjamins, Amsterdam, Philadelphia.

Savourel, Y. (2007). CAT tools and standards: a brief summary. *MultiLingual*, 18(6):37.

Schubert, M. (2009). STAR Transit – The NXT Generation. *Translorial*, 31(4):12–13.

Schulz, J. (1999). Deploying the SAE J2450 Translation Quality Metric in Language Technology Evaluation Projects. In *Translating and the computer 21: Proceedings of the Twenty-first International Conference on Translating and the Computer*, London. Aslib. `http://www.mt-archive.info/Aslib-1999-Schutz.pdf`. Last visited on 25th June 2010.

SDL (2007). SDL Trados 2007 Online Help. SDL, Maidenhead.

SDL (2009). SDL Trados Studio Online Help. SDL, Maidenhead.

Seewald-Heeg, U. (2005). Der Einsatz von Translation-Memory-Systemen am Übersetzerarbeitsplatz: Aufbau, Funktionsweise und allgemeine Kaufkriterien. *MDÜ: Mitteilungen für Dolmetscher und Übersetzer*, 4-5:8–38.

Seewald-Heeg, U. (2007). Evaluation der Übersetzungsleistung maschineller Werkzeuge und Möglichkeiten der Qualitätssicherung. In Schmitt, P. A. and Jüngst, H. E., editors, *Translationsqualität*, pages 561–571. Peter Lang, Frankfurt am Main.

Seewald-Heeg, U. (2009). Werkzeuge für die Softwarelokalisierung. In Baur, W., Kalina, S., Mayer, F., and Witzel, J., editors, *Übersetzen in die Zukunft*, pages 484–490, Berlin. Bundesverband der Dolmetscher und Übersetzer.

Shuttleworth, M. and Lagoudaki, E. (2006). Translation Memory Systems: Technology in the service of the translation professional. In *Proceedings of the 1st Athens International Conference of Translation and Interpretation*, Athens, Greece. Hellenic American Union and Hellenic American University. `http://translation.hau.gr/telamon/files/MarkShuttleworth_ElinaLagoudaki_PaperAICTI.pdf`. Last visited on 25th June 2010.

Sikes, R. (2007). Fuzzy matching in theory and practice. *MultiLingual*, 18(6):39–43.

Sikes, R. (2009). Across Language Server v5. *MultiLingual*, 20(7):16–21.

Sikes, R. (2010). Plunet BusinessManager. *MultiLingual*, 21(3):13–17.

Simard, M. (2003). Translation spotting for translation memories. In *Proceedings of the HLT-NAACL 2003 Workshop on Building and using parallel texts*, pages 65–72, Morristown, USA. Association for Computational Linguistics.

Simard, M. and Isabelle, P. (2009). Phrase-based Machine Translation in a Computer-assisted Translation Environment. In *Proceedings of The twelfth Machine Translation Summit*, pages 120–127, Ottawa, Canada. International Association for Machine Translation. `http://www.mt-archive.info/MTS-2009-Simard.pdf`. Last visited on 25th June 2010.

Simard, M. and Langlais, P. (2001). Sub-sentential Exploitation of Translation memories. In *Proceedings of MT Summit VIII: Machine Translation in the Information Age*, pages 335–339, Santiago de Compostela, Spain. International Association for Machine Translation. `http://www.mt-archive.info/MTS-2001-Simard.pdf`. Last visited on 25th June 2010.

Software and Information Industry Association (2001). Software as a Service: Strategic Backgrounder. `http://www.siia.net/estore/pubs/SSB-01.pdf`. Last visited on 22nd June 2010.

Somers, H., editor (2003). *Computers and translation.* John Benjamins, Amsterdam, Philadelphia.

Somers, H. and Diaz, G. F. (2004). Translation Memory vs. Example-based MT – What's the difference? *International Journal of Translation*, 16(2):5–33.

Specia, L. and Farzindar, A. (2010). Estimating Machine Translation Post-Editing Effort with HTER. In Zhechev, V., editor, *Proceedings of the Second Joint EM+/CNGL Workshop Bringing MT to the User: Research on Integrating MT in the Translation Industry*, pages 33–41, Denver, USA. EuroMatrixPlus/Centre for Next Generation Localisation. `http://www.mt-archive.info/JEC-2010-Specia.pdf`. Last visited on 12th February 2011.

Speer, F. (2010). Das Maß der Dinge. *technische kommunikation*, 32(3):53–54.

STAR (2005). Transit XV Online Help. STAR Group, Ramsen.

STAR (2009). *Transit NXT User's Guide.* STAR Group, Ramsen.

Strizver, I. (2006). *Type rules!* Wiley, Hoboken.

TAUS Data Association (2010). TAUS Data Association. `http://www.translationautomation.com/taus-data-association.html`. Last visited on 2nd June 2010.

Thicke, L. (2009). Optimized MT for Higher Translation Quality. *MultiLingual*, 20(7):9–11.

Thicke, L. (2011). Improving MT results: a study. *MultiLingual*, 22(1):37–40.

Thor, A. J. (1994). Terminology for quantities and units in International Standards. *Terminology*, 1(1):137–146.

Tomás, J. and Casacuberta, F. (2006). Statistical phrase-based models for interactive computer-assisted translation. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 835–841, Morristown, USA. Association for Computational Linguistics.

Translation Automation User Society (2009). TAUS Annual Plan 2010. `http://www.translationautomation.com/taus/annual-plan.html`. Last visited on 22nd June 2010.

Translation Automation User Society (2010). Mission. `http://www.translationautomation.com/about-taus/mission.html`. Last visited on 2nd June 2010.

Trujillo, A. (1999). *Translation Engines: Techniques for Machine Translation.* Springer, London.

University of Chicago, editor (2003). *The Chicago manual of style.* University of Chicago Press, Chicago, London.

Vallianatou, F. (2005). CAT Tools and Productivity: Tracking Words and Hours. *Translation Journal*, 9(4).

Vashee, K. (2010). Best Practices for Using MT & Post-Editing to make Critical and Dynamic Technical Content Multilingual. In *Jahrestagung 2010. Zusammenfassung der Referate*, pages 170–172, Stuttgart. Gesellschaft für technische Kommunikation.

Vidal, E., Casacuberta, F., Rodríguez, L., Civera, J., and Martínez-Hinarejos, C. D. (2006). Computer-assisted translation using speech recognition. *IEEE Transactions on Audio, Speech & Language Processing*, 14(3):941–951.

Wallis, J. (2006). Interactive Translation vs Pre-translation in the Context of Translation Memory Systems. Master's thesis, University of Ottawa.

Wang, W., Knight, K., and Marcu, D. (2006). Capitalizing machine translation. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 1–8, Morristown, USA. Association for Computational Linguistics. http://www.isi.edu/natural-language/mt/hlt-naacl-06-wang.pdf. Last visited on 25th June 2010.

Way, A. (2009). A critique of Statistical Machine Translation. In Daelemans, W. and Hoste, V., editors, *Evaluation of Translation Technology*, number 8 in Linguistica Antverpiensia New Series, pages 17–41. Academic and Scientific Publishers, Brussels.

Way, A. (2010a). Machine Translation. In Clark, A., Fox, C., and Lappin, S., editors, *The Handbook of Computational Linguistics and Natural Language Processing*, pages 531–573. Wiley-Blackwell, Chichester.

Way, A. (2010b). Panning for EBMT gold, or "Remembering not to forget". *Machine Translation*, 24(3-4):177–208.

Way, A. and Hearne, M. (2011). On the Role of Translations in State-of-the-Art Statistical Machine Translation. *Language and Linguistics Compass*, 5(5):227–248.

White, J. (2003). How to evaluate machine translation. In Somers, H., editor, *Computers and translation*, pages 211–244. John Benjamins, Amsterdam, Philadelphia.

Whyman, E. K. and Somers, H. L. (1999). Evaluation metrics for a translation memory system. *Software – Practice and Experience*, 29(14):1265–1284.

Wiedl, W. (2006). *Reguläre Ausdrücke.* Galileo Press, Bonn.

Wills, L. (1996). Introduction to the Special Double Issue on Reverse Engineering. In Wills, L. and Newcomb, P., editors, *Reverse Engineering*, pages 7–8. Kluwer Academic Publishers, Norwell, Dordrecht.

Wittner, J. (2011). Improving translation of variables in interactive games. *MultiLingual*, 22(6):26–29.

Wordfast (2010a). Very Large Translation Memory. http://www.wordfast.com/products_vltm.html. Last visited on 2nd June 2010.

Wordfast (2010b). Wordfast Anywhere: Free Web-based TM for All Translators. *MultiLingual*, 21(5):31.

Wordfast (2010c). Wordfast Online Help. Wordfast LLC, Wilmington.

Wright, S. E. (2006). The creation and application of language industry standards. In Dunne, K., editor, *Perspectives on Localization*, pages 241–278. John Benjamins, Amsterdam, Philadelphia.

Wu, D. (2005). MT model space: statistical versus compositional versus example-based machine translation. *Machine Translation*, 19(3-4):213–227.

Wyld, D. (2010). Cloud computing around the world. *MultiLingual*, 21(1):44–48.

Yamagata Europe (2011). QA Distiller features. `http://www.qa-distiller.com/en/features`. Last visited on 4th January 2011.

Yanishevsky, A. (2009). The Emerging Role of Machine Translation. *MultiLingual*, 20(3):12–13.

Yunker, J. (2011). Making the internet accessible to the world. *MultiLingual*, 22(3):62.

Zajontz, Y., Kuhn, M., and Kollmann, V. (2010). Markteffizienz durch Translation Memory Systeme? Intelligente Übersetzungstechnologien zur Reduktion von Transaktionskosten international agierender Unternehmen. *Journal for Language Technology and Computational Linguistics*, 25(1):41–56.

Zerfaß, A. (2002a). Comparing Basic Features of TM Tools. *Language Technology*, 51:11–14. Supplement of MultiLingual Computing & Technology.

Zerfaß, A. (2002b). Evaluating Translation Memory Systems. In *LREC-2002: Third International Conference on Language Resources and Evaluation*, pages 49–52, Las Palmas, Spain. European Language Resources Association. `http://www.mt-archive.info/LREC-2002-Zerfass.pdf`. Last visited on 25th June 2010.

Zerfaß, A. (2004). Testing TMX Import/Export in Several Translation Tools. *MultiLingual*, 15(8):51–54.

Zerfaß, A. (2007). Exchange standards in localization. *tcworld*, (3):14–16.

Zerfaß, A. (2010). memoQ 4. *MultiLingual*, 21(4):14–17.

Zerfaß, A. (2011). MultiTrans Version 4.4, R2 SP1. *MultiLingual*, 22(3):16–20.

Zetzsche, J. (2003). TMX Implementation in Major Translation Tools. *MultiLingual*, 14(2):23–27.

Zetzsche, J. (2005). De-Hyping Translation Memory: True Benefits, Real Differences, and an Educated Guess about its Future. *The ATA Chronicle*, 34(6):29–33.

Zetzsche, J. (2007a). Creating the Ideal Word Processing Environment in Translation Environment Tools. *Translation Journal*, 11(4). `http://accurapid.com/journal/42toolbox.htm`. Last visited on 22nd June 2010.

Zetzsche, J. (2007b). Online Translation Memories. *The ATA Chronicle*, 36(10):47–49.

Zetzsche, J. (2007c). The World According to Gap. *The ATA Chronicle*, 36(4):27–30.

Zetzsche, J. (2009a). Let's Talk: Trados and the Google Translator Toolkit. *The ATA Chronicle*, 38(10):18–21.

Zetzsche, J. (2009b). The Google Translation Center That Was to Be. *Translation Journal*, 13(3). `http://translationjournal.net/journal/49google.htm`. Last visited on 22nd June 2010.

Zetzsche, J. (2010a). Better a Bottle In Front of Me Than a Frontal Lobotomy. *The Tool Kit: A computer newsletter for translation professionals*, (10-8-173).

Zetzsche, J. (2010b). Hostile Takeover? Welcome Addition? Machine Translation Enters the World of the Translator. *Translation Journal*, 14(3). `http://translationjournal.net/journal/53mt1.htm`. Last visited on 17th August 2010.

Zetzsche, J. (2010c). Transcreation. *The ATA Chronicle*, 39(8):32–33.

Zetzsche, J. (2010d). What's Up? *The ATA Chronicle*, 39(2):32–33.

Zetzsche, J. (2011a). Building Blocks. *Translation Journal*, 15(4). `http://translationjournal.net/journal/58blocks.htm`. Last visited on 6th October 2011.

Zetzsche, J. (2011b). Dealing with Embedded HTML Content in XML Files. `http://www.internationalwriters.com/XML_HTMLFiles.pdf`. Last visited on 12th February 2011.

Zetzsche, J. (2011c). Individual Translators and Data Exchange Standards. `http://www.translationautomation.com/perspectives/individualtranslatorsanddataexchangestandards.html`. Last visited on 3rd October 2011.

Zetzsche, J. (2011d). The New Five-Year-Rule. *Translation Journal*, 15(2). `http://translationjournal.net/journal/56mt.htm`. Last visited on 24th March 2011.

Zetzsche, J. (2011e). Things Can Only Get Better! *The ATA Chronicle*, 40(3):24–26.

Zhechev, V. and van Genabith, J. (2010). Maximising TM Performance through Sub-Tree Alignment and SMT. In *Proceedings of the Ninth Conference of the Association for Machine Translation in the Americas*, Denver, USA. Association for Machine Translation in the Americas. `http://www.mt-archive.info/AMTA-2010-Zhechev.pdf`. Last visited on 10th February 2011.

Index