Multilingual Representations and Models for Improved Low-Resource Language Processing

Dissertation
an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig–Maximilians–Universität München

vorgelegt von
Masoud Jalili Sabet
aus Tehran

München, den 15. December 2021

Erstgutachter: *Prof. Dr. Hinrich Schütze*
Zweitgutachter: *Prof. Dr. Andy Way*
Drittgutachter: *Prof. Dr. Ehsan Shareghi*

Tag der Einreichung: 15. December 2021
Tag der mündlichen Prüfung: 18. July 2022

Eidesstattliche Versicherung

(siehe Promotionsordnung vom 12.07.11, § 8, Abs. 2 Pkt. .5.)

Hiermit erkläre ich an Eides statt, dass die Dissertation von mir selbstständig ohne unerlaubte Beihilfe angefertigt ist.

München, den 15. December 2021

Masoud Jalili Sabet

Abstract

Word representations are the cornerstone of modern NLP. Representing words or characters using real-valued vectors as static representations that can capture the Semantics and encode the meaning has been popular among researchers. In more recent years, Pretrained Language Models using large amounts of data and creating contextualized representations achieved great performance in various tasks such as Semantic Role Labeling. These large pretrained language models are capable of storing and generalizing information and can be used as knowledge bases.

Language models can produce multilingual representations while only using monolingual data during training. These multilingual representations can be beneficial in many tasks such as Machine Translation. Further, knowledge extraction models that only relied on information extracted from English resources, can now benefit from extra resources in other languages.

Although these results were achieved for high-resource languages, there are thousands of languages that do not have large corpora. Moreover, for other tasks such as machine translation, if large monolingual data is not available, the models need parallel data, which is scarce for most languages. Further, many languages lack tokenization models, and splitting the text into meaningful segments such as words is not trivial. Although using subwords helps the models to have better coverage over unseen data and new words in the vocabulary, generalizing over low-resource languages with different alphabets and grammars is still a challenge.

This thesis investigates methods to overcome these issues for low-resource languages. In the first publication, we explore the degree of multilinguality in multilingual pretrained language models. We demonstrate that these language models can produce high-quality word alignments without using parallel training data, which is not available for many languages. In the second paper, we extract word alignments for all available language pairs in the public bible corpus (PBC). Further, we created a tool for exploring these alignments which are especially helpful in studying low-resource languages. The third paper investigates word alignment in multiparallel corpora and exploits graph algorithms for extracting new alignment edges. In the fourth publication, we propose a new model to iteratively generate cross-lingual word embeddings and extract word alignments when only small parallel corpora are available. Lastly, the fifth paper finds that aggregation of different granularities of text can improve word alignment quality. We propose using subword sampling to produce such granularities.

Zusammenfassung

Wortdarstellungen sind der Eckpfeiler der modernen Maschinellen Sprachverarbeitung. Die Darstellung von Wörtern oder Zeichen mit Hilfe von Vektoren als statistische Repräsentationen, die die Semantik erfassen und die Bedeutung kodieren können, ist in der Forschung weitverbreitet. In den letzten Jahren haben vortrainierte Sprachmodelle, die große Datenmengen verwenden und kontextualisierte Repräsentationen erstellen, bei verschiedenen Aufgaben, wie z.B. Semantic Role Labeling, große Erfolge erzielt. Diese großen vortrainierten Sprachmodelle sind in der Lage, Informationen zu speichern und zu verallgemeinern und können als Wissensbasis verwendet werden.

Sprachmodelle können mehrsprachige Repräsentationen erzeugen, obwohl sie beim Training nur einsprachige Daten verwenden. Diese mehrsprachigen Repräsentationen können bei vielen Aufgaben wie der maschinellen Übersetzung von Vorteil sein. Außerdem können Modelle zur Wissensextraktion, die nur auf Informationen aus englischen Ressourcen basieren, nun von zusätzlichen Ressourcen in anderen Sprachen profitieren.

Obwohl diese Ergebnisse für Sprachen mit vielen Ressourcen erzielt wurden, gibt es Tausende von Sprachen, die nicht über große Korpora verfügen. Außerdem benötigen die Modelle für andere Aufgaben wie die maschinelle Übersetzung, wenn keine großen einsprachigen Daten verfügbar sind, parallele Daten, die für die meisten Sprachen nur in geringem Umfang vorhanden sind. Außerdem gibt es in vielen Sprachen keine Tokenisierungsmodelle, und die Aufteilung des Textes in sinnvolle Segmente wie Wörter ist nicht trivial. Obwohl die Verwendung von Subwords den Modellen hilft, eine bessere Abdeckung von ungesehenen Daten und neuen Wörtern im Vokabular zu erreichen, ist die Verallgemeinerung auf ressourcenarme Sprachen mit unterschiedlichen Alphabeten und Grammatiken immer noch eine Herausforderung.

In dieser Arbeit werden Methoden zur Überwindung dieser Probleme für Sprachen mit geringen Ressourcen untersucht. In der ersten Veröffentlichung untersuchen wir den Grad der Mehrsprachigkeit in mehrsprachigen vortrainierten Sprachmodellen. Wir zeigen, dass diese Sprachmodelle hochwertige Wortalignierungen erzeugen können, ohne parallele Trainingsdaten zu verwenden, welche für viele Sprachen nicht verfügbar sind. In der zweiten Arbeit extrahieren wir Wortalignierungen für alle verfügbaren Sprachpaare im Public Bible Corpus (PBC). Darüber hinaus haben wir ein Tool zur Untersuchung dieser Alignierungen entwickelt, das insbesondere bei der Untersuchung von Sprachen mit geringen Ressourcen hilfreich ist. Die dritte Arbeit untersucht die Wortalignierungen in multiparallelen Korpora und nutzt Graphalgorithmen zur Extraktion neuer Ausrichtungskanten. In der vierten Veröffentlichung schlagen wir ein neues Modell

zur iterativen Erzeugung von crosslingual word embeddings und zur Extraktion von Wortalignierungen vor, wenn nur kleine parallele Korpora verfügbar sind. Im fünften Beitrag schließlich wird festgestellt, dass die Aggregation verschiedener Textgranularitäten die Qualität der Wortalignierungen verbessern kann. Wir schlagen die Verwendung von sogenanntem subword sampling vor, um solche Granularitäten zu erzeugen.

Publications and Declaration of Co-Authorship

Chapter 2 corresponds to the following publication:

> **Masoud Jalili Sabet***, Philipp Dufter*, François Yvon, and Hinrich Schütze. *SimAlign: High Quality Word Alignments Without Parallel Training Data Using Static and Contextualized Embeddings.* In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP): Findings, pp. 1627–1643. 2020. *equal contribution.

The basis of this publication was my earlier (unpublished) work on using embeddings for word alignments. Except for the initial proof of concept, I performed the implementation and most evaluation for this work. Along with my coauthors, I co-wrote the published paper.

Chapter 3 corresponds to the following publication:

> Ayyoob Imani, **Masoud Jalili Sabet**, Philipp Dufter, Michael Cysouw, and Hinrich Schütze. *ParCourE: A Parallel Corpus Explorer for a Massively Multilingual Corpus.* In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL): System Demonstrations, pp. 63-72. 2021.

Philipp Dufter and I conceived of the original research contributions. I built a proof-of-concept of ParCourE that tested and developed a subset of the ParCourE functionality. Ayyoob Imani then built the final ParCourE system based on the proof-of-concept. I implemented the alignment reader in the backend for the final system. All authors wrote the initial draft of the article and did the subsequent corrections. All authors regularly discussed this work with each other and improved the draft.

Chapter 4 corresponds to the following publication:

> Ayyoob Imani*, **Masoud Jalili Sabet***, Lutfi Kerem Senel, Philipp Dufter, François Yvon, and Hinrich Schütze. *Graph Algorithms for Multiparallel Word Alignment.* In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2021. *equal contribution.

Ayyoob Imani and I conceived of the original research contributions. My original contribution consisted of classical graph algorithms, Ayyoob Imani's original contribution consisted of recommendation algorithms. I also contributed the idea of using translated datasets. I performed implementation and evaluation for the classical graph algorithms (Section 3.3). I performed the experiments for the analysis of the model (Section 5.3). All authors wrote the initial draft of the article and did the subsequent corrections. All authors regularly discussed this work with each other and improved the draft.

Chapter 5 corresponds to the following publication:

> Nina Poerner, **Masoud Jalili Sabet**, Benjamin Roth, and Hinrich Schütze. *Aligning Very Small Parallel Corpora Using Cross-Lingual Word Embeddings and a Monogamy Objective.* In Computing Research Repository 1811.00066.

I conceived of the original idea of using and fine-tuning bilingual word embeddings for low-resource word alignment. I implemented the initial proof-of-concept using word2vec representations trained with S-IDs. Based on the proof-of-concept, Nina Poerner implemented our final experimental system. She also contributed a new loss function ("monogamy"). I performed the experiments with the baseline methods and evaluated the model (Figure 2). Nina Poerner and I wrote related work, evaluation and discussion sections together. She wrote the initial draft of the rest of the paper. I regularly discussed this work with my coauthors and assisted in improving the draft.

Chapter 6 corresponds to the following publication:

> Ehsaneddin Asgari*, **Masoud Jalili Sabet***, Philipp Dufter, Christopher Ringlstetter, and Hinrich Schütze. *Subword Sampling for Low Resource Word Alignment.* In Computing Research Repository 2012.11657. *equal contribution.

I conceived of the original idea of using subword tokenization for word alignment and implemented a proof-of-concept using a new subword tokenization model

trained with an entropy-based loss function. Ehsaneddin Asgari implemented the iterative subword sampling algorithm. I implemented subword alignment and evaluation. I performed most of the experiments and evaluations (Tables 2 and 3). All authors wrote the initial draft of the article. I regularly discussed this work with my coauthors and assisted in improving the draft.

Contents

Chapter 1

Introduction

1.1 Motivation

Natural language is one of the most important features of human civilization. As a result, the understanding, processing, and generation of natural language have been a major research topic in computer science from the start. The beginning of natural language processing dates back to the 1950s when Alan Turing proposed the task of automated natural language understanding and generation as a criterion of intelligence for machines, which is now called the Turing test (Turing, 1950).

During the early years, most works on language processing were rule-based. After the introduction of machine learning methods for language processing, statistical models such as statistical machine translation (Brown et al., 1988a,b) became popular. Since in natural languages an infinite number of different sentences are possible, which all contain a different number of words, models have to divide the text into smaller meaningful units, such as sentences and words, in order to process it. Statistical models then calculate the frequency of words and their co-occurrences in order to estimate conditional probability distributions – for example, based on n-grams – and use them to represent words and the relationships between them. However, these simple techniques were at their limits in many tasks, and scaling up the size of the datasets did not result in any significant progress (Mikolov et al., 2013a).

Word or token representation forms the basis of most modern NLP approaches. Following this approach, the text is segmented into smaller units, which can be words, tokens, or characters. Each segment is represented by a real-valued vector that can encode the semantics of the segment and its context in the text (Schütze, 1992). It should be mentioned that these distributed representations can be used as a similarity measure between different segments. Word representations were used as the first layer of neural networks for different tasks such as part-of-speech

tagging, named entity recognition, and semantic role labeling (Collobert et al., 2011). Although these models achieve good performance, there are several open questions about the best way to create the underlying representations. There are multiple challenges regarding this process:

- Handling new tokens and out-of-vocabulary (OOV) words: creating embeddings for new tokens that fit the representations of known tokens is a challenge. Schick and Schütze (2019) propose a method to leverage both the surface form and the context of the word to create embeddings, but only investigate static representations.

- Choosing the smallest meaningful unit: while using subwords and characters is a promising potential solution to represent the semantics of OOV words, it is hard to capture their meaning through representation learning (Park et al., 2021). Pretrained language models mostly use subword tokenizers that are generated by compression algorithms such as byte pair encoding (Sennrich et al., 2016) and WordPiece (Schuster and Nakajima, 2012), which are not morphologically informed (Hofmann et al., 2021). As an example, in the BERT language model (Devlin et al., 2019), the word "Superbizarre" is represented by the subwords { "superb", "iza", "rre" }, while "Superb" can have a contrastive sentiment with "Superbizarre"; this means that segmenting words into meaningful subunits is even more challenging in multilingual settings.

- Hierarchical representations: producing representations for a group of units (e.g., an embedding for a sentence or phrase, which is produced by individual word embeddings) or for subunits (e.g., generating character or subword embeddings that are comparable with word embeddings) is desirable, but non-trivial. As an example, the word "war" can appear as a subunit in other words, such as "**war**zone", "**war**ehouse", "**war**ranty", and "**war**drobe". Since the language models should be able to produce the word representations from the corresponding subunits, assigning a single representation to "war", which is compatible with the meanings of all the mentioned words, can be difficult. Surprisingly, averaging word embeddings of all words in a sentence has proven to be a strong baseline (Faruqui et al., 2015), but for long sentences, paragraphs, and documents, averaging the representations may lose important information.

In more recent years, pretrained language models (LMs) that create contextualized word representations have achieved good performance in a wide range of tasks (Peters et al., 2018; Devlin et al., 2019). Various factors contributed to their exceptional performance. First, the leveraging of contextualized, rather than

static representations, provides more information for downstream tasks. Second, using the transformer architecture allows for more parallelization and therefore for the training of larger models on larger amounts of data. As a consequence, these large pretrained models are capable of storing and generalizing information and can be used as knowledge bases (Petroni et al., 2019). Third, the usage of subwords (Sennrich et al., 2016) has increased the coverage of the models and provided a solution for OOV words with high performance. Previous works use morphological information (in the form of a table of possible affixes) to increase the coverage over OOV words in morphologically rich languages (Passban et al., 2018), but building a large list of all possible affixes for many languages is a costly and time-consuming task, and might not be a good solution for all phenomena for all languages (e.g., compound words in German).

Although these contributions have paved the way for more innovations in natural language processing, they have not had the same impact on many low-resource languages (Blasi et al., 2021). There are more than 7000 languages in the world (Eberhard et al., 2020), only a few of which have adequate resources for training such models. Due to memory constraints and to overcome sparsity, pretrained multilingual language models use subword tokenization models with limited vocabulary sizes, which means that most languages (e.g., Khmer, Lao, Kurdish, and Oromo) are not included in the vocabulary, and the tokens are mostly chosen from dominant languages (the languages with the most training data, such as English, German, and French). Additionally, for some low-resource languages (e.g., Arabic, Finnish, Korean, Russian, and Turkish) that are included in the vocabulary, the tokens are over-segmented, which results in poor quality token representations (Rust et al., 2021).

Language models can produce multilingual representations even though they only use monolingual data during training (Devlin et al., 2019; Conneau et al., 2020a). Training multilingual models can be beneficial for many reasons. First, the model requires less data to train and generalize on multiple languages. In addition, the multilingual representations can be beneficial for many tasks that directly make use of them, such as machine translation. Furthermore, knowledge extraction models, which formerly only had access to information extracted from resources available in one language (e.g., English), can now benefit from resources in all languages (Roy et al., 2020).

This extends further to annotated data from high-resource languages, which can be used to train a model that can be applied to other languages. The last advantage of multilingual language models is that they are easier to maintain compared to individual models for each language.

1.1.1 Approach

In this work, we approach the mentioned issues of multilingual NLP models for low-resource languages using different methods:

- Tokenization: we examine different tokenization models for languages, to improve the quality of their representations and enable models to reach better performance when large training datasets are not available.
- Multilingual embeddings: we experiment with different multilingual models in order to study the strengths and weaknesses of such models for various languages. This will provide an insight into the effectiveness of different features of these models and may help develop stronger models.
- Multi-parallel datasets: we investigate multi-parallel datasets and different methods of employing them to improve the quality of representations for all languages.

It should be noted that the above approaches are language agnostic. Despite the fact that languages have different characteristics, which means that applying the same procedures to all of them may lead to poor results for some, we pursue pipelines and methods that can be applied to models effortlessly. We argue that using language-specific rules hinders the growth and scalability of multilingual models.

1.1.2 Research Questions

Considering the necessity of multilingual representations and models, and their aforementioned benefits, we can pursue a range of interesting questions:

(i) *Data:* What kinds of datasets can help improve multilingual models? How much data is needed for the model during training time to show desirable performance on a language (or a language pair)?

(ii) *Models:* Do models require supervision in order to obtain better generalization? Are there any additional signals from the training data that the models can use to perform better for more languages?

(iii) *Analysis:* It is not clear how similar concepts are connected in different languages. How can we study this similarity? Is multilinguality based on the similarity of the languages? Does the model performance improve when the information from the other languages is added?

In this work, we aim to address all of these questions.

1.1.3 Outline

In this chapter, we start by defining the notation used throughout this thesis and clarifying the terminology. We then introduce the basic methods of monolingual and multilingual representation learning, both in static and contextualized forms. We finally describe applications that can benefit from these representations and show common evaluation methods for them.

In Chapter 2, we analyze existing multilingual representations and their performance in different languages. We show that we can obtain high-quality word alignments from them. In the second publication, presented in Chapter 3, we generate word alignments for all language pairs in the Parallel Bible Corpus (PBC) and present a tool to study the connections between languages. In Chapter 4, we propose a graph model that uses multi-parallel datasets to improve the quality of word alignments. This method especially aims for improvements in low-resource languages. Chapter 5 presents a model to iteratively improve the multilingual representations and word alignments. The last paper, presented in Chapter 6, measures the word alignment performance based on the quality of the subword tokenization. We propose a new model which samples from several subword tokenizations and improves the word alignment quality for multiple language pairs.

1.2 Foundations

1.2.1 Notation

With positive integers $d, t \in \mathbb{N}^+$, we denote vectors as boldface lowercase $\mathbf{x} \in \mathbb{R}^d$, and matrices as boldface uppercase letters $\mathbf{X} \in \mathbb{R}^{t \times d}$. The i-th element of $\mathbf{x}$ can be referred to as $\mathbf{x}_i$ and matrices can be indexed $(X_{ij})_{i=1,2\ldots,t,j=1,2,\ldots d} = \mathbf{X}$. The i-th row and j-th column of $\mathbf{X}$ are denoted as $\mathbf{X}_i \in \mathbb{R}^d, \mathbf{X}_{\star,j} \in \mathbb{R}^t$. When a textual unit is used to index a matrix or vector, instead of an index, it refers to the vector or scalar corresponding to that unit; i.e., $\mathbf{X}_{\text{book}}$ refers to the vector $\mathbf{X}_i$ where $index(\text{book}) = i$ and $index()$ is a bijective function that assigns each textual unit a unique integer. The cardinality of a set S is denoted by $|S|$, and the Euclidean norm of a vector is $\|\mathbf{x}\|$. The transposed vector and matrix are denoted as $\mathbf{x}^\intercal$ and $\mathbf{X}^\intercal$. We denote the cosine similarity of two vectors $\mathbf{x}, \mathbf{y}$ as $\text{cos-sim}(\mathbf{x}, \mathbf{y}) := \mathbf{x}^\intercal\mathbf{y}/(\|\mathbf{x}\|\,\|\mathbf{y}\|)$.

1.2.2 Local and Distributed Representations

Natural language text has an order and can be interpreted as sequential data. We can therefore denote it as $(u_1, u_2, \ldots, u_t)$, where u_i is some unit of text. The set

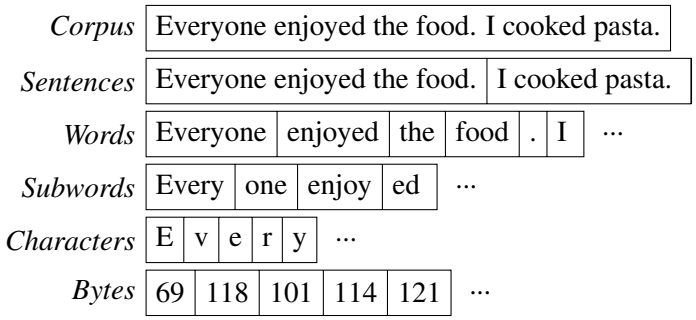

Figure 1.1 – *Common ways to split text data into units. The byte representation depends on the encoding for unicode points (e.g., utf-8).*

of all distinct text units is called the vocabulary $V = \{v_1, v_2, \ldots, v_n\}$. Common choices for units are shown in Figure 1.1. In order to process units of text, these units need to be assigned numerical representations, called embeddings. We denote an embedding function as a map $e : V \to E$, which assigns each unit in the vocabulary an embedding. A simple choice of embedding function is the one-hot encoding, which chooses $E = \{0, 1\}^{|V|}$ and assigns each element $v_i \in V$ the i-th unit in E. This requires one computing element for each unit v_i and is called a *local representation* (Hinton et al., 1990). In contrast, *distributed representations* use multiple computing elements to represent each v_i. This implies that E is a d-dimensional space with $d \ll |V|$.

For example, assume a vocabulary of n elements $V = \{$“cook”, “cookies”, “begin”, “began”, $\ldots\}$. We can use one-hot encodings as local representations by assigning $e($“cook”$) = (1, 0, 0 \ldots)$, $e($“cookies”$) = (0, 1, 0 \ldots)$, etc. A possible distributed representation could consist of d dimensional vectors, where d is the number of all characters across all elements in V. Afterward, we assign $e(v_i)_j = \mathbb{1}\{c_j \in v_i\}$ where $\{c_1, c_2, \ldots, c_d\}$ is the set of all characters. This means $e(v_i)$ assings 1 to all the characters in v_i.

At first glance, it can be noticed that unlike distributed representations, in the local representations, all vectors are orthogonal to each other, and developing a useful similarity metric is therefore challenging. On the other hand, on distributed representations, a metric such as cosine similarity can be used: as “cook” and “cookies” share more common characters, their respective vectors are more similar than “cook” and “begin”. Another advantage of this is that adding new elements to the vocabulary does not require the embedding function to change the number of its dimensions.

Finding a meaningful mapping for a distributed representation model is not trivial. Furthermore, it is not clear how to assess a representation model. One common way of evaluating the resulting representations is to measure their performance on

downstream tasks. Another is to check whether the similarity measures correlate with the semantic similarity of the text units. When looking at multilingual representations specifically, more challenges regarding definition and evaluation arise. In the literature, the term *multilingual* is sometimes used for models that can process multiple languages with no cross-language connections, whereas *crosslingual* models can leverage the connections between languages and provide a meaningful similarity measure between text units in different languages. These two groups naturally require different methods of evaluation.

1.3 Static Representations

A static embedding function over a vocabulary V is defined as

$$e : V \rightarrow \mathbb{R}^d, \tag{1.1}$$

with dimensionality d. Most static representation methods use the context of the text unit to create the embedding for that unit (Mikolov et al., 2013a). The supporting argument is that two words that occur in similar contexts are likely to have similar meanings. For example, the words "bank" and "money" are more likely to occur in the same sentences, therefore their embeddings are expected to have high similarity.

Given a sequence of text units $S = (s_1, s_2, \ldots, s_t) \in V^t$, the co-occurrence matrix of the vocabulary V with size n is denoted as $\mathbf{C} \in \mathbb{R}^{n \times n}$, where $\mathbf{C}_{ij}$ is the co-occurrence of s_i and s_j in S. Formally, a text unit s_i is in the context of s_j if $|i - j| \leq c$, where c is the size of the context window. The co-occurrence can then be calculated as the number of times that two units occur in the same context window. A more relaxed version of co-occurrence is calculated by the number of times two units co-occur in the same batch of units in the corpus S, such as sentences or paragraphs (Levy et al., 2017). Note that these co-occurrence counts can be weighted based on the distance between words or normalized over units, i.e., rows of $\mathbf{C}$.

1.3.1 Monolingual Representations

A simple method for creating monolingual representations is to use the co-occurrence matrix $\mathbf{C}$ as embeddings; i.e., the embedding function is defined as $e(v_i) := \mathbf{C}_i$. Consequently, if v_i and v_j occur in similar contexts, the embeddings $e(v_i)$ and $e(v_j)$ will have higher cosine similarity. Although this method is preferable to local representations, there are certain disadvantages. The main issue is that since the size of the vectors depends on the vocabulary size n, a large vocabulary may result in large input size and therefore higher computational costs for the model that uses

them. Furthermore, the addition of new units to the vocabulary will increase the dimensionality of overall representations, resulting in a need to update the entire pipeline. Lastly, $\mathbf{C}$ is sparse since most units never co-occur which may result in problems in downstream models.

One way of overcoming this issue is the use of matrix factorization as proposed by Schütze (1992). More specifically, using the singular value decomposition (SVD) of $\mathbf{C}$ for the embeddings enables the representations to be of any chosen dimensionality. During recent years, as neural networks became more popular in natural language processing, the need for low-dimensional representations became more prominent. For instance, Levy and Goldberg (2014) and Pennington et al. (2014) introduced the use of matrix factorization for word representations.

As neural networks became more popular in natural language processing, new methods were introduced that exploited them for the generation of representations. Mikolov et al. (2013a) employed shallow neural networks to encode the text and used the hidden layer representation as word embeddings, which is a widely popular method nowadays. We move forward by discussing two such methods in more details.

Continuous Bag-of-Words and Skip-gram

Mikolov et al. (2013a) proposed two new methods for generating word representations using neural networks. In the first method, continuous bag of words (CBOW), the model predicts a word u_i given the representations of words u_j in the context window. The context window includes words both from the left and the right of the target word.

The second proposed architecture is skip-gram with negative sampling. Similar to CBOW, the main idea of the model is to predict, given a word u_i, whether other units u_j are likely to appear in the context window of u_i. To summarize, CBOW predicts a word given all the words in the context while skip-gram predicts all the context words given a word.

More formally, assume two matrices $\mathbf{E}, \mathbf{W} \in \mathbb{R}^{|V| \times d}$ where $\mathbf{E}_{u_i}$ is the word embedding of u_i and $\mathbf{W}_{u_i}$ is the context embedding of the word. Further, consider $c_p(u_i) \subset \mathbf{V}$ as the set of words in the context window of u_i and let $c_n(u_i) \subset \mathbf{V}$ be a set of random negative samples for the training. The skip-gram model with negative sampling tries to minimize the following objective:

$$\begin{aligned}\mathcal{L}(\mathbf{E}, \mathbf{W}) = -\sum_{i=1}^{|V|} \sum_{w \in c_p(u_i)} \log\left(\sigma(\mathbf{E}_{u_i}^{\intercal} \mathbf{W}_w)\right) \\ - \sum_{w \in c_n(u_i)} \log\left(\sigma(-\mathbf{E}_{u_i}^{\intercal} \mathbf{W}_w)\right),\end{aligned} \tag{1.2}$$

where $\sigma : \mathbb{R} \to \mathbb{R}$ is the sigmoid function. $\sigma(x) = 1/(1 + e^{-x})$. Mikolov et al. (2013a) used a context window size of $C = 10$.

Incorporating Subword Information

Both skip-gram and CBOW use only the context of the words for prediction, while ignoring the internal structure of the words. For example, *bank* and *banks* while spelled similarly, are assined independent vector representations.

Bojanowski et al. (2017) proposed a method that makes use of the additional information source of subword structure. They published the implementation of this model under the name *fastText*. We denote $\mathcal{G}_k : V \to C_k$ as a function that maps a word to the set of *k*-grams contained in the word. To allow the model to distinguish prefixes and suffixes from other sequences, special boundary symbols $<$ and $>$ are added at the beginning and end of words. Considering the word "book" and $k = 3$ as an example, $\mathcal{G}_3(book) = \{$*<bo*, *boo*, *ook*, *ok>*$\}$. Let C_k be the set of all possible n-grams in V. Each n-gram g in C_k will be represented with a vector $\mathbf{z}_g$ and finally, for each word u_i, $\mathbf{E}_{u_i}$ in Eq. 1.2 will be replaced with

$$\mathbf{E}_{u_i} + \sum_{g \in \mathcal{G}_k(u_i)} \mathbf{z}_g. \tag{1.3}$$

This allows the model to integrate the subword information into the word representations.

1.3.2 Multilingual Representations

So far, we have described two static monolingual representation learning models. In this section, we revisit the methods for creating multilingual embedding spaces in more detail. For languages e and f with vocabularies V_e and V_f, let us consider $\mathbf{E}^{(e)}$ and $\mathbf{E}^{(f)}$ as the static embeddings with dimensions d_e and d_f. For simplicity, we assume $d_e = d_f =: d$.

It is not clear what properties should be expected from a multilingual embedding space. It is desirable that the multilingual representations preserve the monolingual features and can be used in monolingual tasks. However, high quality multilingual representations should also exhibit transferability between different language spaces, i.e., if a model is trained on a task with embeddings $\mathbf{E}^{(e)}$, it should be able to process $\mathbf{E}^{(f)}$ for the same task without a significant decrease in performance. One way to achieve this is to require semantically similar units to have similar representations. For example, if monolingual embedding spaces for English and German are learned separately, there is no correlation between the similar words in the two languages. For instance, $\mathbf{E}^{(e)}_{drive}$ and $\mathbf{E}^{(f)}_{fahren}$ are in different

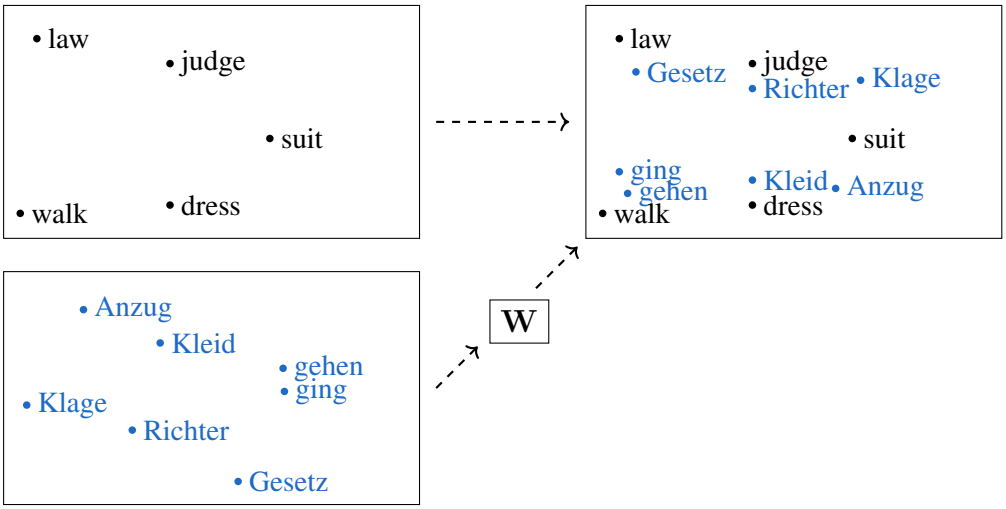

Figure 1.2 – *Mapping a German monolingual embedding space into English embeddings using a rotation matrix* $\mathbf{W}$. *Figure taken from Dufter (2021).*

spaces and therefore have random cosine similarity. However, in a multilingual space with embeddings $\mathbf{E}^{e+f}$, cos-sim$(\mathbf{E}^{e+f}_{drive}, \mathbf{E}^{e+f}_{fahren})$ should be close to 1.

Mapping Approach

One popular approach to the creation of multilingual embedding spaces is to individually learn monolingual representations $\mathbf{E}^{(e)}, \mathbf{E}^{(f)}$ for both languages, and then learn a mapping $w : \mathbb{R}^d \to \mathbb{R}^d$, which will be applied to one or both embedding spaces such that $\bar{E}^{(f)}_i := w(\mathbf{E}^{(f)}_i)$, and $\mathbf{E}^{(e)}$ and $\bar{E}^{(f)}$ are in a multilingual space. This is illustrated in Figure 1.2. This mapping approach relies on the assumption that the monolingual spaces have similar structure and a mapping can therefore properly transfer one embedding space to the other (Vulić et al., 2020). As the early research on this approach was done on European languages, which have similar structure and training data (Mikolov et al., 2013b), the facts that many languages have different structures and that their training data comes from various domains were neglected.

Assume that a bilingual dictionary is available for the target pair of languages

$$\mathcal{D} := \{(u^{(e)}_1, u^{(f)}_1), (u^{(e)}_2, u^{(f)}_2), \ldots, (u^{(e)}_m, u^{(f)}_m)\}, \tag{1.4}$$

where each tuple has a unit from both languages and m is the number of entries in the dictionary. Further, consider creating modified embedding matrices $\tilde{\mathbf{E}}^{(e)}, \tilde{\mathbf{E}}^{(f)}$ that only contain embeddings from the units in $\mathcal{D}$. A function $w(\mathbf{x}) = \mathbf{W}^\intercal\mathbf{x}$ with $\mathbf{W} \in \mathbb{R}^{d\times d}$, can be used for the mapping by minimizing the loss function

$$\mathcal{L}(\mathbf{W}) = \frac{1}{2}\sum_{i=1}^{m}\left\|\tilde{\mathbf{E}}_i^{(e)}\mathbf{W} - \tilde{\mathbf{E}}_i^{(f)}\right\|^2. \tag{1.5}$$

This optimization problem is an instance of the Procrustes Problem. Mikolov et al. (2013b) used a gradient descent method for this optimization and showed that the multilingual spaces created by this method can perform well for word translation.

An extension to this approach is to constrain the matrix $\mathbf{W}$ to be orthonormal, i.e., $\mathbf{W}^\intercal\mathbf{W} = \mathbf{I}$ (Xing et al., 2015). This transformation does not modify the structure of the embedding space which is desirable. This constraint will change Eq. 1.5 to the *Orthogonal Procrustes Problem* (Schönemann, 1966), which can be solved with singular value decomposition (SVD). By computing the SVD of matrix $(\tilde{\mathbf{E}}^{(f)})^\intercal\tilde{\mathbf{E}}^{(e)}$ that is $(\tilde{\mathbf{E}}^{(f)})^\intercal\tilde{\mathbf{E}}^{(e)} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\intercal$, with $\mathbf{E}, \mathbf{V}$ as complex unitary matrices, and $\mathbf{\Sigma}$ as a rectangular diagonal matrix with non-negative real numbers on the diagonal, the transformation is given by $\mathbf{W}^* = \mathbf{V}\mathbf{U}^\intercal$.

The aforementioned approaches are effective when a bilingual dictionary is available. In the unsupervised setting, the underlying challenge is that $\mathbf{E}^{(e)}$ and $\mathbf{E}^{(f)}$ are unaligned across both axes: neither the ith vocabulary item $\mathbf{E}_{i*}^{(e)}$ and $\mathbf{E}_{i*}^{(f)}$ nor the jth dimension of the embeddings $\mathbf{E}_{*j}^{(e)}$ and $\mathbf{E}_{*j}^{(f)}$ are aligned.

Since creating these dictionaries is challenging for many language pairs, unsupervised embedding alignment approaches gained popularity, in particular the approach introduced by Lample et al. (2018). They proposed a generative adversarial learning method following the training method of Goodfellow et al. (2014). Consider a discriminator model with parameters θ_D, where $f_{\theta_D}(z)$ shows the probability of vector z being an embedding from language e. Values output by the discriminator that are closer to zero show that z is an embedding from language f. The discriminator and the mapping losses are written as

$$\mathcal{L}_D(\theta_D|\mathbf{W}) = -\frac{1}{n}\sum_{i=1}^{n}\log\left(f_{\theta_D}(\mathbf{W}\mathbf{E}_i^{(e)})\right) - \frac{1}{m}\sum_{i=1}^{m}\log\left(1 - f_{\theta_D}(\mathbf{E}_i^{(f)})\right) \tag{1.6}$$

$$\mathcal{L}_G(\mathbf{W}|\theta_D) = -\frac{1}{n}\sum_{i=1}^{n}\log\left(1 - f_{\theta_D}(\mathbf{W}\mathbf{E}_i^{(e)})\right) - \frac{1}{m}\sum_{i=1}^{m}\log\left(f_{\theta_D}(\mathbf{E}_i^{(f)})\right) \tag{1.7}$$

The transformation $\mathbf{W}$ is trained with the mapping loss so that the discriminator is unable to predict the original language of an embedding. During the training, both the discriminator and the transformation matrix are trained alternatingly with gradient descent. The authors also propose a refinement procedure that uses well-aligned units as a noisy dictionary which is used in a Procrustes solution to find a final transformation.

Artetxe et al. (2018) shows that the generator-discriminator model is rather unstable and frequently fails to find a transformation. Thus they introduced a model called Vecmap that exploits the structural similarity of the embeddings and a robust self-learning algorithm that iteratively improves this solution.

Joint Learning

The main assumption of the mapping approaches is that the embedding spaces $\mathbf{E}^{(e)}, \mathbf{E}^{(f)}$ are monolingual representations that have been already learned individually for both languages. Joint learning approaches introduce other embedding learning algorithms that aim to learn $\mathbf{E}^{(e)}, \mathbf{E}^{(f)}$ simultaneously in the same multilingual space.

Many joint learning methods employ a parallel corpus with two or more languages. Assume that this corpus consists of n parallel sentences in two languages $S^{(e)} = (s_1^{(e)}, s_2^{(e)}, \ldots, s_n^{(e)})$, $S^{(f)} = (s_1^{(f)}, s_2^{(f)}, \ldots, s_n^{(f)})$ where each sentence pair $s_i^{(e)} = (u_1^{(e)}, \ldots, u_{l_i^{(e)}}^{(e)}), s_i^{(f)} = (u_1^{(f)}, \ldots, u_{l_i^{(f)}}^{(f)})$ consists of two sentences that are translations of each other, and $l_i^{(e)}, l_i^{(f)}$ are the number of units in each sentence. The Parallel Bible Corpus (PBC) (Mayer and Cysouw, 2014) and the Proceedings of the European Parliament (Koehn, 2005) are two examples of a parallel corpus. Both of these corpora are also multi-parallel, meaning that the translations of each sentence is provided in multiple languages, more than a thousand in the case of PBC.

As a simple approach of joint learning, Vulić and Moens (2015) proposed to create *pseudo-bilingual* sentences, where the sentences $s_i^{(e)}$ and $s_i^{(f)}$ are concatenated and randomly shuffled. For instance, with two sentences "*Ich habe eine Katze*" and "*I have a cat*", the pseudo-bilingual sentence could be "*eine I Katze habe Ich cat a have*". Afterwards, a monolingual learning algorithm, such as skip-gram, is trained on the resulting corpus. The intuition behind this approach is that the monolingual learning algorithms treat the sentences as context for words. Creating pseudo-bilingual sentences will allow the model to treat the words from all languages as the context. In the example sentence, the tokens *have* and *habe* will more likely occur in similar contexts and therefore will have similar vector representations. While there are certainly better ways of constructing pseudo-bilingual sentences, this work only considered random shuffling. Other works learn distinct embedding models for the source and the target languages that keep the sentence orders, while jointly learning a cross-lingual regularizer, enforcing word pairs aligned in the parallel text to have similar representations (Klementiev et al., 2012; Gouws et al., 2015). The word pairs are obtained from a dictionary with 1:1 mappings between source and target words.

Levy et al. (2017) proposed an algorithm based on sentence IDs, where the

intuition is similar to pseudo-bilingual sentences: units across languages that occur in similar sentences in a parallel corpus are likely to have a similar meaning. In this approach, instead of using the individual words as context, the parallel sentences will be represented with a special token and used as context. A new corpus will be constructed, where each sentence consists of a pair of tokens. The first token in each pair is the special token of the parallel sentence and the second token is one of the words in that sentence. After training the skip-gram learning method on the final corpus, the words that occur in similar sentences will have more matching sentence IDs as their context and thus similar vector representations.

Another approach for joint training uses matrix factorization to create the representations. In this approach, an inverted index of words and sentences is created, where the vector for each word has the same dimensionality as the number of sentences in the corpus. Subsequently, the matrix factorization is used to reduce the dimensionality of word vectors to smaller numbers (Søgaard et al., 2015; Jalili Sabet et al., 2016).

1.4 Contextualized Representations

Static representations are created as mappings $e : V \rightarrow \mathbb{R}^d$, which assign a particular vector to each unit in the vocabulary (Mikolov et al., 2013a). To give an example, in static models, the unit *bank* is always represented by the same embedding vector, whether it occurs in the phrase *river bank* (where its meaning is "the land alongside or sloping to a river or lake") or *the bank account* (where its meaning is "a financial establishment"). It is up to debate whether aggregating multiple meanings of a unit and representing them with a single embedding vector is problematic. Neelakantan et al. (2014) approaches this issue with learning separate vectors for each word sense.

Contextualized representation methods attempt to take into account the context in which a unit appears (McCann et al., 2017). We can define a contextualized embedding function as

$$e : V^{t_{max}} \rightarrow \mathbb{R}^{t_{max} \times d}, \tag{1.8}$$

where t_{max} is the maximum number of units that the function can process at once. This number can vary between the size of a phrase, a sentence, a paragraph or a document. Therefore, the contextualized embedding of the unit u_i in a sentence $(u_1, \ldots, u_i, \ldots u_t)$ depends on all other units in the sentence. In the case of the above example, the two different contexts will result in different contextualized representations for the unit *bank*.

The model proposed by Peters et al. (2017) uses bidirectional LSTMs (Hochreiter and Schmidhuber, 1997) to process the text in a forward and reverse fashion,

which means the model can potentially use all the previous and all the following words as context. Recent transformer-based language models, such as Devlin et al. (2019), use self-attention operations, and their computation costs scale quadratically with the sequence length. This means that these models are not optimized to process long sequences. A solution for this issue is to reduce their cost closer to linear complexity (Beltagy et al., 2020).

1.4.1 Monolingual Representations

This section presents various methods of learning contextualized monolingual representations.

Pretrained Language Models

Peters et al. (2017) is one of the first works that attempted to learn contextualized vector representations for words. The key idea is to train a neural language model without the need for a labeled training dataset. The hidden states of the model can then be used as contextualized embeddings.

A language model is a probability distribution over strings of text. It takes unstructured text as input for training and estimates the probability of a sequence of units occurring in that text. Formally, it models the probability $P(u_1, u_2, \ldots, u_t)$. As text is sequential in nature and in order to simplify the estimation of the probability distribution, the chain rule of probabilities is commonly applied in various approaches. The resulting forward language model is given by

$$P(u_0, u_1, u_2, \ldots, u_t, u_{t+1}) = P(u_0) \prod_{i=0}^{t} P(u_{i+1}|u_{\leq i}), \tag{1.9}$$

where $u_{\leq i}$ are all tokens with index $j \leq i$, and u_0, u_{t+1} are extra tokens that indicate the start and end of a sequence. Analogously, the backward language model can be formulated as

$$P(u_0, u_1, u_2, \ldots, u_t, u_{t+1}) = P(u_{t+1}) \prod_{i=0}^{t} P(u_i|u_{\geq i+1}). \tag{1.10}$$

The probability model $P_\theta(u_i|u_{<i})$ can be parameterized with different model architectures such as recurrent neural networks (RNNs). A RNN is a model that is recursively computed as

$$\mathbf{h}^{(i)} = \sigma_h(\mathbf{W}^{(u)}\mathbf{e}^{(i)} + \mathbf{W}^{(h)}\mathbf{h}^{(i-1)} + \mathbf{b}^{(h)}) \tag{1.11}$$

$$\mathbf{y}^{(i)} = \sigma_y(\mathbf{W}^{(y)}\mathbf{h}^{(i)} + \mathbf{b}^{(y)}) \tag{1.12}$$

for $i = 1, \ldots, n$, where $P_\theta(u_i|u_{<i}) = \hat{\mathbf{y}}^{(i)}_{u_i}$ is the probability of the unit u_i occurring at position i,

$$\theta = (\mathbf{W}^{(u)}, \mathbf{W}^{(h)} \in \mathbb{R}^{d\times d}; \mathbf{h}^{(0)}, \mathbf{b}^{(h)} \in \mathbb{R}^d; \mathbf{E}, \mathbf{W}^{(y)} \in \mathbb{R}^{n\times d}; \mathbf{b}^{(y)} \in \mathbb{R}^n) \quad (1.13)$$

are the parameters of the model. The $\mathbf{W}$'s are weight matrices and the $\mathbf{b}$'s are biases. $\mathbf{e}^{(i)}$ is an embedding vector for the unit u_i, i.e., $\mathbf{e}^{(i)} = \mathbf{E}_{u_i}$, $\mathbf{h}^{(i)}$ is the hidden layer representation of the unit u_i, and $\sigma : \mathbb{R}^m \to \mathbb{R}^m$ are activation functions. A common choice is to apply tangens hyperbolicus to the hidden layer vectors $\sigma_h(\mathbf{x})_k = \tanh(\mathbf{x}_k)$ component-wise, and the softmax function $\sigma_y(\mathbf{x})_k = e^{\mathbf{x}_k} / \sum_{i=1}^n e^{\mathbf{x}_i}$ to generate the probability over tokens in the output vector. Negative cross entropy between the $\hat{\mathbf{y}}^{(i)}$ and the observed units is commonly used as the objective function. The latter is formulated as:

$$\mathcal{L}(\theta, U) = -\frac{1}{m}\sum_{k=1}^{m}\sum_{i=1}^{t_k} CE(y^{(i)}, \hat{y}^{(i)}) = -\frac{1}{m}\sum_{k=1}^{m}\sum_{i=1}^{t_k} \log(\hat{\mathbf{y}}^{(i)}_{u_i}), \quad (1.14)$$

where $U = (s_1, \ldots, s_m)$ is a corpus with m sentences with $s_k = (u_1, \ldots, u_{t_k})$, and $y^{(i)}$ is a one-hot vector. This objective function can then be minimized using stochastic gradient descent. One of the popular variants of RNN models is Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) which has shown good performance for language modeling.

Using either a forward or a backward language model neglects one side of the sequence while creating the contextualized representations. Peters et al. (2017) proposed to train a language model that consists of both forward and backward LSTMs. They considered the concatenation of the hidden states of the LSTMs as embeddings and used them together with static embeddings as input to models for solving downstream tasks, such as named entity recognition. This approach yielded state of the art performance and was further developed in their next paper: Peters et al. (2018) introduced deep contextualized embeddings called *Embeddings from Language Models* (ELMo), which outperformed the state of the art performance across many tasks. In ELMo, bidirectional LSTMs are trained with the task of language modeling.

The advantage of these models is that language models do not need any manually labeled data and can be (pre-)trained on large amounts of text data, which could for example be sourced from the Internet. These models can then be used for downstream tasks. This motivates the name *Pretrained Language Models* (PLMs).

Transformer Models

Recurrent neural networks take advantage of the inherent sequential nature of text. However, studies show that recurrent neural networks cannot properly propagate

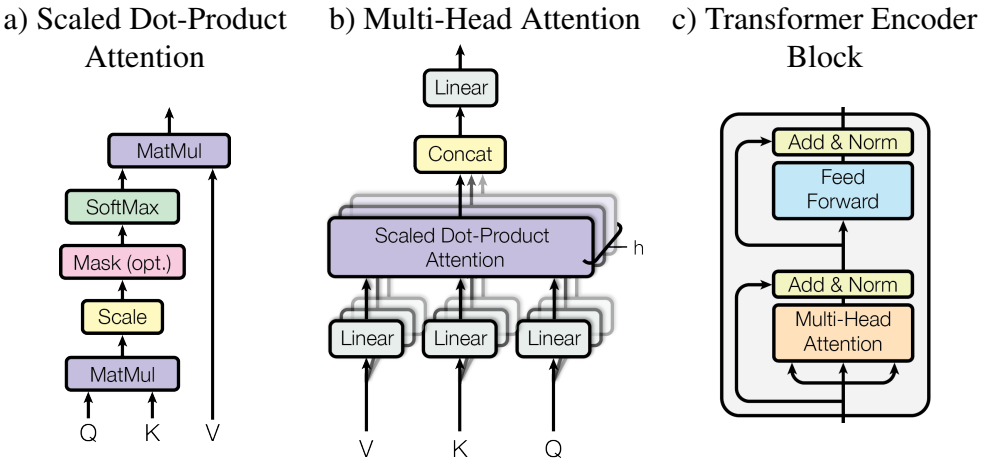

Figure 1.3 – *Transformer model.* ***a)*** *Scaled Dot-Product Attention.* ***b)*** *Multi-Head Attention consists of several attention layers running in parallel.* ***c)*** *Schema of a transformer encoder block that can be repeated for l layers. Figure taken from Vaswani et al. (2017)*

information across long spans of text, as analyzed for example by Cho et al. (2014) for machine translation. Bahdanau et al. (2015) proposed using other extensions such as *attention* to overcome this issue.

A key milestone in this area is Vaswani et al. (2017), where the authors introduced a machine translation system that only uses the attention mechanism, and this model exhibits superior performance compared to recurrent neural networks. They proposed to stack *Transformer Encoder Blocks* as shown in Figure 1.3. These blocks are functions $t_\theta : \mathbb{R}^{n\times d} \to \mathbb{R}^{n\times d}$ with $t_\theta(\mathbf{X}) =: \mathbf{Z}$, which is computed as follows:

$$
\begin{aligned}
\mathbf{Q}, \mathbf{K}, \mathbf{V} &= \mathbf{X}\mathbf{W}^{(q)}, \mathbf{X}\mathbf{W}^{(k)}, \mathbf{X}\mathbf{W}^{(v)} \\
\mathbf{A} &= \text{Softmax}(\frac{\mathbf{Q}\mathbf{K}^\intercal}{\sqrt{d_h}}) \\
\mathbf{M} &= \mathbf{A}\mathbf{V} \\
\mathbf{O} &= \text{LayerNorm}_1(\mathbf{M} + \mathbf{X}) \\
\mathbf{F} &= \text{ReLU}(\mathbf{O}\mathbf{W}^{(f_1)} + \mathbf{b}^{(f_1)})\mathbf{W}^{(f_2)} + \mathbf{b}^{(f_2)} \\
\mathbf{Z} &= \text{LayerNorm}_2(\mathbf{F} + \mathbf{O}),
\end{aligned}
\tag{1.15}
$$

where $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ are respectively projections of the embeddings for queries, keys, and values, $\mathbf{A}$ is the self attention matrix, and the softmax function is applied

row-wise. $\text{LayerNorm}(\mathbf{X})_i = \mathbf{g} \odot (\mathbf{X}_i - \mu(\mathbf{X}_i))/\sigma(\mathbf{X}_i) + \mathbf{b}$ is layer normalization with $\mu(\mathbf{x}), \sigma(\mathbf{x})$ returning the mean and standard deviation of a vector and g, b as parameters (Ba et al., 2016), and $\text{ReLU}(\mathbf{X}) = \max(0, \mathbf{X})$. The parameters of such a block are

$$\begin{aligned}\theta = (&\mathbf{W}^{(q)}, \mathbf{W}^{(k)}, \mathbf{W}^{(v)} \in \mathbb{R}^{d\times d}; \mathbf{g}^{(1)}, \mathbf{g}^{(2)}, \mathbf{b}^{(1)}, \mathbf{b}^{(2)} \in \mathbb{R}^{d}; \\ &\mathbf{W}^{(f_1)} \in \mathbb{R}^{d\times d_f}; \mathbf{W}^{(f_2)} \in \mathbb{R}^{d_f\times d}; \mathbf{b}^{(f_1)} \in \mathbb{R}^{d_f}; \mathbf{b}^{(f_2)} \in \mathbb{R}^{d}),\end{aligned} \tag{1.16}$$

where d is called the hidden dimension, d_f the intermediate dimension, and n is the sequence length. Multi-head attention runs several attention layers in parallel and concatenates their results. In this setting, each of the h heads learns its own projection matrices for queries, keys, and values, that is $\mathbf{W}^{(q)}, \mathbf{W}^{(k)}, \mathbf{W}^{(v)} \in \mathbb{R}^{d\times d_h}$ where $d = hd_h$. The output matrices $\mathbf{M}^{(h)} \in \mathbb{R}^{t\times d_h}$ are then concatenated along the second dimension to obtain the final $\mathbf{M}$.

As an example, consider how a Transformer is applied to text data, for the sequence $U = (u_1, u_2, \ldots, u_t)$. The token embeddings $\mathbf{T} \in \mathbb{R}^{t\times d}$ are created by a projection of the token IDs to the embedding matrix $\mathbf{E} \in \mathbb{R}^{v\times d}$, where v is the vocabulary size. The Transformer block, specifically the multihead attention, does not take into consideration the position information in a sequence and is therefore invariant to reorderings of the input. The way the attention mechanism is designed makes it suitable to operate on sets (Lee et al., 2019). Therefore, the sequential information has to be injected into the model with a different way. The simplest method is to add positional encodings to the input embeddings by creating a matrix of absolute position embeddings $\mathbf{P} \in \mathbb{R}^{t\times d}$, and changing the final input to T_θ to $\mathbf{U} + \mathbf{P}$. In some models additional embeddings such as language embeddings or layer embeddings are also added to the input.

Transformer-Based Language Models

Devlin et al. (2019) proposed a new language model *Bidirectional Encoder Representations from Transformers* (BERT). BERT is pretrained on a new variant of language modeling. Most of the previously proposed language modeling methods use the sequential nature of text and learn a unidirectional model. The main innovation of BERT is that it is not a unidirectional language model, but rather bidirectional. Instead of only utilizing the right or left context of a word, BERT has access to context words on both sides simultaneously. This is realized by the use of *masked language modeling* (*MLM*).

Consider a corpus $U = (u_1, \ldots, u_t)$. The MLM objective is to receive a corrupted version of U, here U', as input, and attempt to reconstruct U. The corrupted input U' is created by randomly altering some tokens in U. For instance, in Devlin et al. (2019), each token u_i has the probability 0.15 to be corrupted, and the model

learns to reconstruct such tokens. The possible changes applied to corrupted tokens are: (i) replacing it with the special token [MASK], (ii) replacing it with another random token from the vocabulary, and (iii) keeping the token as is. This process can be defined by two sequences of independent and identically distributed (iid) random variables $(R_i^{(1)})_{i=1,\dots,t}, (R_i^{(2)})_{i=1,\dots,t}$ with uniform distributions on $[0, 1]$. Then U' is created by

$$u'_i = \begin{cases} \text{[MASK]} & \text{if } R_i^{(1)} \leq \rho_1 \wedge R_i^{(2)} \leq \rho_2 \\ u_r & \text{if } R_i^{(1)} \leq \rho_1 \wedge \rho_2 < R_i^{(2)} \leq \rho_3 \\ u_i & \text{if } R_i^{(1)} \leq \rho_1 \wedge \rho_3 < R_i^{(2)} \\ u_i & \text{otherwise,} \end{cases} \tag{1.17}$$

where u_r is a randomly sampled unit from the vocabulary and [MASK] is a special token. The values used in Devlin et al. (2019) are $\rho_1 = 0.15, \rho_2 = 0.8, \rho_3 = 0.9$. The MLM uses this modified corpus as input to a Transformer. The original input $\mathbf{X}$ can then be used as the targets for prediction. The final hidden vectors corresponding to the corrupted tokens are fed into an output softmax over the vocabulary. In contrast to previous language modeling methods, the model only predicts the masked tokens, rather than reconstructing the entire input. Although MLM is useful for learning a bidirectional pre-trained model, one downside is that this creates a mismatch between pre-training and fine-tuning. This is caused by the fact that the [MASK] tokens only appear during pre-training and do not appear during fine-tuning. To mitigate this issue, the corrupted tokens are not always replaced with [MASK] tokens, but instead are sometimes left unchanged or replaced by a random token. All of the corruped tokens are then used to predict the original token with cross entropy loss.

Other technical aspects were taken into consideration during training. Using words as the basic units requires large vocabularies, which leads to high memory consumption. To mitigate this, BERT uses the wordpiece tokenizer (Schuster and Nakajima, 2012) to split the text into subword units. With this, the vocabulary size can be adjusted before training the model, which reduces the memory requirements of the embedding matrix $\mathbf{E}$, while at the same time enabling the model to cover various text sequences. For pre-training, the Adam optimizer (Kingma and Ba, 2015) is used and regularization methods such as dropout (Srivastava et al., 2014) are applied. The model also includes a second loss term, *next sentence prediction*, which aims at generating better sentence representations. However, experiments later showed that it has no impact on the performance of the model (Liu et al., 2019). Hence, in the follow-up works, this term is mostly excluded from the overall loss.

1.4.2 Multilingual Representations

BERT presented a method for learning high-quality contextualized representations. This section discusses methods for training *multilingual* contextualized representations.

Joint Training

The authors of Devlin et al. (2019) have also published a multilingual version of BERT (*mBERT*). This model is trained on Wikipedia across multiple languages, selected for largest amount of data. In case of mBERT, the 104 languages with the largest amount of available text on Wikipedia were used for training. The articles from these languages are concatenated and shuffled and a shared vocabulary across all these corpora is learned. Afterwards, a standard BERT model is trained on this data.

In this method, the model does not use any explicit crosslingual supervision. This means no parallel data or dictionary is used, and the loss function does not include any term that contains special signals for multilinguality. The underlying idea behind the multilinguality in this approach is that the tokens in the shared vocabulary can appear in multiple languages. As an example, the word *find* appears both in the English word *finding* and the German word *finden*. Experiments show that this model yields promising multilingual representations when evaluated with the methods described in Section 1.5.

Other works attempted to further improve mBERT's multilinguality. Conneau and Lample (2019) proposed a new loss term, *Translation Language Modeling* (*TLM*) that uses parallel sentences. In this model, a pair of parallel sentences are used as the input of the Transformer with a similar task of predicting the masked tokens, so that the model can also use the sentence in the other language as context. The intuition is that this can help increase the multilinguality of the model. In follow-up work, Conneau et al. (2020a) propose (*XLM-R*), which does crosslingual learning at scale by using more data, and drops the next sequence prediction loss term. Similar to static representation mapping approaches, Conneau et al. (2020b) show that monolingual contextualized embeddings can also be mapped into a common embedding space using linear transformations.

1.5 Evaluation

Previous sections have discussed how to learn static and contextualized representation models for both monolingual and multilingual settings. In this section we study different methods to evaluate the quality of such representations.

There are *intrinsic* and *extrinsic* evaluations for assessing the quality of monolingual static embeddings. The intrinsic evaluations include tasks for analyzing different properties of the embeddings, and they range from word similarity, e.g., (Hill et al., 2015; Gerz et al., 2016), and word analogy (Mikolov et al., 2013c; Gladkova et al., 2016), to correlation with linguistic features (Tsvetkov et al., 2015). In extrinsic evaluations, the main objective is a downstream task and the embeddings are used as input to a model that solves the task. Examples of downstream tasks are part-of-speech tagging and named entity recognition.

Monolingual contextualized embeddings can also be evaluated using perplexity in language modeling. However, to better understand the quality of the contextualization, there is often a range of extrinsic tasks for which the model is finetuned and then evaluated, e.g., the GLUE or SuperGLUE benchmarks (Wang et al., 2018, 2019) for natural language understanding, SQUAD and SQUAD 2.0 benchmarks (Rajpurkar et al., 2016, 2018) with questions for machine comprehension of text, the natural language inference (NLI) task (Bowman et al., 2015), and common tasks like named entity recognition.

Multilingual representations can be applied to many use cases, which in turn can be used for evaluating the representation models.

1. **Translation.** The representations of text units such as words, phrases, or sentences, that are semantically similar, should be close to each other across languages. This means that these representations can be used for translation or other applications including word alignment and crosslingual sentence retrieval.

2. **Zero-shot transfer.** Consider a model that is only trained on a specific dataset or language, e.g., English, for a downstream task such as part-of-speech tagging. If this model, without additional training, can be applied to other languages and is able to correctly tag words in non-English sentences, the model is performing zero-shot transfer across languages. Using multilingual representations can enable the models to achieve this. This is a desirable quality of these representations since annotating training data, especially for more complex tasks, requires labor, which is costly and time-consuming. If we aim to annotate data for tens or hundreds of languages, the costs quickly become infeasible. For this reason, zero-shot transfer is a common trend in modern NLP.

3. **Low-resource coverage.** Training datasets of various tasks are mostly annotated for the English language. On the other hand, a large number of languages are *low-resource*: there are only a few or no datasets available for them. One of the main objectives of multilingual NLP is to use the representations to make a multilingual model that exhibits sensible improvements in

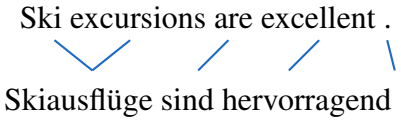

Figure 1.4 – *Example of a word alignment. Figure taken from Jalili Sabet et al. (2020)*

performance for low-resource languages. Another advantage of such models is that it is much easier to maintain a single model than to have multiple language-specific models.

In this thesis we intend to cover all these use cases with different experiments: the word alignment task covers the first use case as an *intrinsic evaluation*, and part-of-speech tagging with annotation projection as an *extrinsic evaluation* covers the other two use cases.

1.5.1 Word Alignment

Word alignment is a task where translations of words in two parallel sentences should be identified, as in the example shown in Figure 1.4. Consider a corpus with parallel sentences $\mathcal{U} = \{(s_1^{(e)}, s_1^{(f)}), (s_2^{(e)}, s_2^{(f)}), \ldots, (s_m^{(e)}, s_m^{(f)})\}$. For a parallel sentence pair $s_i^{(e)}, s_i^{(f)}$, the word alignment of the whole sentence can be considered as a bipartite graph, where the units in $s_i^{(e)}$ and $s_i^{(f)}$ are the nodes in the graph denoted by $V_i^{(e)}, V_i^{(f)}$. In some datasets, instead of one type of alignment, there are two sets of sure and possible edges, $S_i, P_i \subset V_i^{(e)} \times V_i^{(f)}$ where $S_i \subset P_i$. The task is to automatically predict the edges that were marked as correct in the manually created gold standard. The model generates a set of prediction edges A_i, while the edge sets without index denote the union of word alignments across sentences, i.e., $S = \bigcup_{i=1}^{m} S_i$. Standard evaluation metrics are then precision, recall, F_1 and alignment error rate (AER) (Och and Ney, 2000) computed by

$$\begin{aligned} \text{precision} &= \frac{|A \cap P|}{|A|}, \ \text{recall} = \frac{|A \cap S|}{|S|}, \\ \text{AER} &= 1 - \frac{|A \cap S| + |A \cap P|}{|A| + |S|}, \\ F_1 &= \frac{2 \ \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}. \end{aligned} \tag{1.18}$$

Since word alignment is also an important step in the statistical machine translation (SMT) pipeline, there are several works that proposed different methods

for generating high quality word alignments. Most statistical methods are either an implementation of the IBM alignment models or inspired by them (Brown et al., 1993). Namely, Giza++ (Och and Ney, 2003), fast-align (Dyer et al., 2013), as well as follow-up models such as eflomal (Östling and Tiedemann, 2016), are widely used for word alignment.

More recently, several works attempted to use multilingual representations for this task. In this set of approaches, a simple method to induce the alignment edges could be to match each unit with another unit in the parallel sentence, such that the corresponding embeddings have the highest similarity. Most such models require training or finetuning on the same task, or similar tasks such as translation. Garg et al. (2019) pursued a multitask approach for alignment and translation, while in other works only the word alignment task is used for end-to-end training or finetuning, and only new loss terms help the model to learn better representations (Zenkel et al., 2020; Dou and Neubig, 2021).

Both statistical and neural approaches have their advantages for different language pairs. However, the output alignments of these models can be aggregated to form an ensemble method for word alignment with better performance (Steingrímsson et al., 2021).

1.6 Conclusion

This introductory chapter has described the main concepts of multilingual representations that are relevant to this thesis. We presented mathematical notation and linguistic foundations, which were used to introduce the existing static and contextualized representation learning methods. We discussed the approaches for achieving multilinguality through joint training or mapping. Finally, we described the evaluation methods with possible use cases for multilingual models. The next chapters use the presented methods in a series of research papers and aim to improve the performance of different tasks for low-resource languages.

1.6.1 Contributions

In light of the three research questions posed at the beginning, we can categorize and summarize our contributions in this thesis as follows.

i) *Data:* In Chapter 2, we introduce SimAlign, a high-quality word alignment tool that uses static and contextualized embeddings and does not require parallel training data. We examine multiple datasets and show that, with only using monolingual training data, our unsupervised model performs better than popular supervised models. We train statistical word aligners, such as

fast-align, eflomal, and Giza++, with several training data sizes and show the effect of training data size on word alignment performance. Our experiments reveal that when we use SimAlgin with Vecmap embeddings, the alignment quality is on par with fast-align trained on ten thousand sentence pairs, while using SimAlign with mBERT performs better than all statistical aligners trained with more than 1 million sentence pairs. In Chapter 4, we show that training on multi-parallel corpora can enable the word aligner to improve its performance over bilingual corpora. Interestingly, the experiments reveal that multi-parallel corpora created by translations can still contribute to better alignment performance for the target language pair.

ii) *Models:* We incorporate various signals into existing word alignment models and introduce a new word aligner, SimAlign, that does not require supervision or parallel training data. We use contextualized representations trained on monolingual corpora as features for a word aligner (Chapter 2). We study the graph structures and use parallel sentences in a multi-parallel corpus for extra information in order to generate better bilingual word alignments (Chapter 4). In another work, subword structures and different granularities of text are studied (Chapter 6). We show that the aggregation of subword samplings of a language (subword models with different vocabulary sizes) can improve the quality of word-level alignments. We improve the state-of-the-art for the word alignment task in several evaluation datasets, while aiming to use little or no training data (Chapter 2, Chapter 5).

iii) *Analysis:* We build a tool named ParCourE to study languages and their word connections to better understand the quality of representations (Chapter 3). ParCourE can also help linguists as an interactive explorer for studying low-resource languages in PBC. In Chapter 4, we investigate the effect of anchor languages on bilingual word alignments of a target language pair and show that using similar languages as anchors is more effective for improving performance. Furthermore, we explore the effect of using similar granularities of text learned for a language pair and applying it to other languages for word alignment (Chapter 6). The experiments show that such features can be transferred to similar languages and improve the model performance.

1.6.2 Future Work

We now describe future work and related literature that address our research questions.

Training a pretrained multilingual language model that covers more than 1000 languages can be beneficial for the performance of NLP tasks in low-resource

languages. The popular pretrained language models mostly use some form of subword tokenization. Since large amounts of text are not available for low-resource languages, this puts such languages at a disadvantage when subword tokenization methods are learning a shared vocabulary (Maronikolakis et al., 2021). One way to solve this issue is to use character-level encoders. Other works have tried to replace subwords by using visual representations (Salesky et al., 2021), downsampling characters (Clark et al., 2021), and using deep character encoders (Xue et al., 2021). These models need large datasets for training. The next step towards character-level encoders can be to use multi-parallel corpora, and the bilingual signals in such corpora, to enable the models to converge faster and with less training data.

Our work in Chapter 4 shows that using multi-parallel corpora can boost the performance of word alignment for low-resource and distant languages by using other languages as anchors. We used graph algorithms that only rely on the information from the graph of words. This means that important considerations of word aligners, such as the distortion (introduced in IBM Model 2), and the context of the sentence, are neglected in the current model. Having a better encoding of the graph of languages might be beneficial for creating better word representations (nodes in the graph), and alignment prediction (edges in the graph). Using graph neural networks (GNN) (Scarselli et al., 2009) could be the key to solving this challenge. The GNN encoders can take the representations of words and their positions as input. It is also possible to include additional information as input to GNNs, such as part-of-speech tags of the words and the dependency trees of the sentences. Furthermore, by adding language identifiers as features for word representations, the GNN encoder can learn which anchor languages to use for each word. All of this suggests that by using proper features and models, the performance of word alignment can be further improved.

Using the graph of multiple languages can also be used for tasks other than word alignment. The word alignment edges can be used to project tags, such as part-of-speech tags and semantic role labels, from several languages to a target language (Agić et al., 2016). This method can create a training dataset for a target low-resource language while only parallel data and proper taggers for other languages (e.g., English and German) are available. It is worth investigating the effect of using GNN encoders for annotation projection and creating training datasets for all languages in a multi-parallel corpus such as PBC.

Chapter 2

SimAlign: High Quality Word Alignments Without Parallel Training Data Using Static and Contextualized Embeddings

SimAlign: High Quality Word Alignments Without Parallel Training Data Using Static and Contextualized Embeddings

Masoud Jalili Sabet[*1], **Philipp Dufter**[*1], **François Yvon**[2], **Hinrich Schütze**[1]
[1] Center for Information and Language Processing (CIS), LMU Munich, Germany
[2] Université Paris-Saclay, CNRS, LIMSI, France
{masoud,philipp}@cis.lmu.de, francois.yvon@limsi.fr

Abstract

Word alignments are useful for tasks like statistical and neural machine translation (NMT) and cross-lingual annotation projection. Statistical word aligners perform well, as do methods that extract alignments jointly with translations in NMT. However, most approaches require parallel training data, and quality decreases as less training data is available. We propose word alignment methods that require no parallel data. The key idea is to leverage multilingual word embeddings – both static and contextualized – for word alignment. Our multilingual embeddings are created from monolingual data only without relying on any parallel data or dictionaries. We find that alignments created from embeddings are superior for four and comparable for two language pairs compared to those produced by traditional statistical aligners – even with abundant parallel data; e.g., contextualized embeddings achieve a word alignment F_1 for English-German that is 5 percentage points higher than eflomal, a high-quality statistical aligner, trained on 100k parallel sentences.

1 Introduction

Word alignments are essential for statistical machine translation and useful in NMT, e.g., for imposing priors on attention matrices (Liu et al., 2016; Chen et al., 2016; Alkhouli and Ney, 2017; Alkhouli et al., 2018) or for decoding (Alkhouli et al., 2016; Press and Smith, 2018). Further, word alignments have been successfully used in a range of tasks such as typological analysis (Lewis and Xia, 2008; Östling, 2015b), annotation projection (Yarowsky et al., 2001; Padó and Lapata, 2009; Asgari and Schütze, 2017; Huck et al., 2019) and creating multilingual embeddings (Guo et al., 2016; Ammar et al., 2016; Dufter et al., 2018).

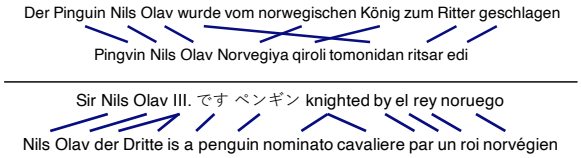

Figure 1: Our method does not rely on parallel training data and can align distant language pairs (German-Uzbek, top) and even mixed sentences (bottom). Example sentence is manually created. Algorithm: Itermax.

Statistical word aligners such as the IBM models (Brown et al., 1993) and their implementations Giza++ (Och and Ney, 2003), fast-align (Dyer et al., 2013), as well as newer models such as eflomal (Östling and Tiedemann, 2016) are widely used for alignment. With the rise of NMT (Bahdanau et al., 2014), attempts have been made to interpret attention matrices as soft word alignments (Cohn et al., 2016; Koehn and Knowles, 2017; Ghader and Monz, 2017). Several methods create alignments from attention matrices (Peter et al., 2017; Zenkel et al., 2019) or pursue a multitask approach for alignment and translation (Garg et al., 2019). However, most systems require parallel data (in sufficient amount to train high quality NMT systems) and their performance deteriorates when parallel text is scarce (Tables 1–2 in (Och and Ney, 2003)).

Recent unsupervised multilingual embedding algorithms that use only non-parallel data provide high quality static (Artetxe et al., 2018; Conneau et al., 2018) and contextualized embeddings (Devlin et al., 2019; Conneau et al., 2020). *Our key idea is to leverage these embeddings for word alignments – by extracting alignments from similarity matrices induced from embeddings – without relying on parallel data.* Requiring no or little parallel data is advantageous, e.g., in the low-resource case and in domain-specific settings without parallel data. A lack of parallel data cannot be easily

[*] Equal contribution - random order.

40 *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1627–1643
November 16 - 20, 2020.

remedied: mining parallel sentences is possible (Schwenk et al., 2019) but assumes that comparable, monolingual corpora contain parallel sentences. Further, we find that large amounts of mined parallel data do not necessarily improve alignment quality.

Our main **contribution** is that we show that *word alignments obtained from multilingual pretrained language models are superior for four and comparable for two language pairs, compared to strong statistical word aligners like eflomal even in high resource scenarios.* Additionally, **(1)** we introduce three new alignment methods based on the matrix of embedding similarities and two extensions that handle null words and integrate positional information. They permit a flexible tradeoff of recall and precision. **(2)** We provide evidence that subword processing is beneficial for aligning rare words. **(3)** We bundle the source code of our methods in a tool called *SimAlign*, which is available.[1] An interactive online demo is available.[2]

2 Methods

2.1 Alignments from Similarity Matrices

We propose three methods to obtain alignments from similarity matrices. Argmax is a simple baseline, IterMax a novel iterative algorithm, and Match a graph-theoretical method based on identifying matchings in a bipartite graph.

Consider parallel sentences $s^{(e)}, s^{(f)}$, with lengths l_e, l_f in languages e, f. Assume we have access to some embedding function $\mathcal{E}$ that maps each word in a sentence to a d-dimensional vector, i.e., $\mathcal{E}(s^{(k)}) \in \mathbb{R}^{l_k \times d}$ for $k \in \{e, f\}$. Let $\mathcal{E}(s^{(k)})_i$ denote the vector of the i-th word in sentence $s^{(k)}$. For static embeddings $\mathcal{E}(s^{(k)})_i$ depends only on the word i in language k whereas for contextualized embeddings the vector depends on the full context $s^{(k)}$. We define the *similarity matrix* as the matrix $S \in [0,1]^{l_e \times l_f}$ induced by the embeddings where $S_{ij} := \text{sim}\left(\mathcal{E}(s^{(e)})_i, \mathcal{E}(s^{(f)})_j\right)$ is some normalized measure of similarity, e.g., cosine-similarity normalized to be between 0 and 1. We now describe our methods for extracting alignments from S, i.e., obtaining a binary matrix $A \in \{0,1\}^{l_e \times l_f}$.

Argmax. A simple baseline is to align i and j when $s_i^{(e)}$ is the most similar word to $s_j^{(f)}$ and vice-versa. That is, we set $A_{ij} = 1$ if

$$(i = \arg\max_l S_{l,j}) \wedge (j = \arg\max_l S_{i,l})$$

and $A_{ij} = 0$ otherwise. In case of ties, which are unlikely in similarity matrices, we choose the smaller index. If all entries in a row i or column j of S are 0 we set $A_{ij} = 0$ (this case can appear in Itermax). Similar methods have been applied to co-occurrences (Melamed, 2000) ("competitive linking"), Dice coefficients (Och and Ney, 2003) and attention matrices (Garg et al., 2019).

Algorithm 1 Itermax.

procedure ITERMAX(S, $n_{\max}$, $\alpha \in [0,1]$)
  $A, M = \text{zeros\_like}(S)$
  for $n \in [1, \ldots, n_{\max}]$ **do**
    $\forall i, j:$
    $$M_{ij} = \begin{cases} 1 \text{ if } \max\left(\sum_{l=0}^{l_e} A_{lj}, \sum_{l=0}^{l_f} A_{il}\right) = 0 \\ 0 \text{ if } \min\left(\sum_{l=0}^{l_e} A_{lj}, \sum_{l=0}^{l_f} A_{il}\right) > 0 \\ \alpha \text{ otherwise} \end{cases}$$
    $A_{\text{to\_add}} = \text{get\_argmax\_alignments}(S \odot M)$
    $A = A + A_{\text{to\_add}}$
  end for
  return A
end procedure

Figure 2: Description of the Itermax algorithm. *zeros_like* yields a matrix with zeros and with same shape as the input, *get_argmax_alignments* returns alignments obtained using the Argmax Method, $\odot$ is elementwise multiplication.

Itermax. There are many sentences for which Argmax only identifies few alignment edges because mutual argmaxes can be rare. As a remedy, we apply Argmax iteratively. Specifically, we modify the similarity matrix conditioned on the alignment edges found in a previous iteration: if two words i and j have *both* been aligned, we zero out the similarity. Similarly, if *neither* is aligned we leave the similarity unchanged. In case only one of them is aligned, we multiply the similarity with a discount factor $\alpha \in [0,1]$. Intuitively, this encourages the model to focus on unaligned word pairs. However, if the similarity with an already aligned word is exceptionally high, the model can add an additional edge. Note that this explicitly allows one token to be aligned to multiple other tokens. For details on the algorithm see Figure 2.

Match. Argmax finds a local, not a global optimum and Itermax is a greedy algorithm. To find global optima, we frame alignment as an assign-

[1] https://github.com/cisnlp/simalign
[2] https://simalign.cis.lmu.de/

ment problem: we search for a maximum-weight maximal matching (e.g., (Kuhn, 1955)) in the bipartite weighted graph which is induced by the similarity matrix. This optimization problem is defined by

$$A^* = \text{argmax}_{A \in \{0,1\}^{l_e \times l_f}} \sum_{i=1}^{l_e} \sum_{j=1}^{l_f} A_{ij} S_{ij}$$

subject to A being a matching (i.e., each node has at most one edge) that is maximal (i.e., no additional edge can be added). There are known algorithms to solve the above problem in polynomial time (e.g., (Galil, 1986)).

Note that alignments generated with the match method are inherently bidirectional. None of our methods require additional symmetrization as post-processing.

2.2 Distortion and Null Extensions

Distortion Correction [Dist]. Distortion, as introduced in IBM Model 2, is essential for alignments based on non-contextualized embeddings since the similarity of two words is solely based on their surface form, independent of position. To penalize high distortions, we multiply the similarity matrix S componentwise with

$$P_{i,j} = 1 - \kappa \left(i/l_e - j/l_f\right)^2,$$

where κ is a hyperparameter to scale the distortion matrix P between $[(1-\kappa), 1]$. We use $\kappa = 0.5$. See supplementary for different values. We can interpret this as imposing a locality-preserving prior: given a choice, a word should be aligned to a word with a similar relative position ($\left(i/l_e - j/l_f\right)^2$ close to 0) rather than a more distant word (large $\left(i/l_e - j/l_f\right)^2$).

Null. Null words model untranslated words and are an important part of alignment models. We propose to model null words as follows: if a word is not particularly similar to any of the words in the target sentence, we do not align it. Specifically, given an alignment matrix A, we remove alignment edges when the normalized entropy of the similarity distribution is above a threshold τ, a hyperparameter. We use normalized entropy (i.e., entropy divided by the log of sentence length) to account for different sentence lengths; i.e., we set $A_{ij} = 0$ if

$$\min\left(-\frac{\sum_{k=1}^{l_f} S^h_{ik} \log S^h_{ik}}{\log l_f}, -\frac{\sum_{k=1}^{l_e} S^v_{kj} \log S^v_{kj}}{\log l_e}\right) > \tau,$$

where $S^h_{ik} := S_{ik} / \sum_{m=1}^{l_f} S_{im}$, and $S^v_{kj} := S_{kj} / \sum_{m=1}^{l_e} S_{mj}$. As the ideal value of τ depends on the actual similarity scores we set τ to a percentile of the entropy values of the similarity distribution across all aligned edges (we use the 95th percentile). Different percentiles are in the supplementary.

3 Experiments

3.1 Embedding Learning

Static. We train monolingual embeddings with fastText (Bojanowski et al., 2017) for each language on its Wikipedia. We then use VecMap (Artetxe et al., 2018) to map the embeddings into a common multilingual space. Note that this algorithm works without any crosslingual supervision (e.g., multilingual dictionaries). We use the same procedure for word and subword levels. We use the label **fastText** to refer to these embeddings as well as the alignments induced by them.

Contextualized. We use the multilingual BERT model (mBERT).[3] It is pretrained on the 104 largest Wikipedia languages. This model only provides embeddings at the subword level. To obtain a word embedding, we simply average the vectors of its subwords. We consider word representations from all 12 layers as well as the concatenation of all layers. Note that the model is not finetuned. We denote this method as mBERT[i] (when using embeddings from the i-th layer, where 0 means using the non-contextualized initial embedding layer) and mBERT[conc] (for concatenation).

In addition, we use XLM-RoBERTa base (Conneau et al., 2020), which is pretrained on 100 languages on cleaned CommonCrawl data (Wenzek et al., 2020). We denote alignments obtained using the embeddings from the i-th layer by XLM-R[i].

3.2 Word and Subword Alignments

We investigate both alignments between subwords such as wordpiece (Schuster and Nakajima, 2012) (which are widely used for contextualized language models) and words. We refer to computing alignment edges between words as *word level* and between subwords as *subword level*. Note that gold standards are all word-level. In order to evaluate alignments obtained at the subword level we convert subword to word alignments using the heuristic "two words are aligned if any of their subwords are

[3] https://github.com/google-research/bert/blob/master/multilingual.md

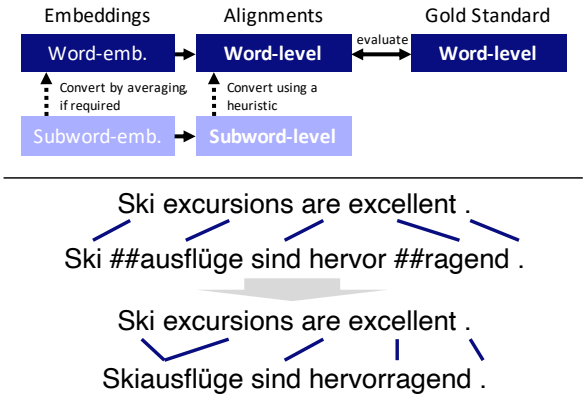

Figure 3: Subword alignments are always converted to word alignments for evaluation.

aligned" (see Figure 3). As a result a single word can be aligned with multiple other words.

For the *word* level, we use the NLTK tokenizer (Bird et al., 2009) (e.g., for tokenizing Wikipedia in order to train fastText). For the *subword* level, we generally use multilingual BERT's vocabulary[3] and BERT's wordpiece tokenizer. For XLM-R we use the XLM-R subword vocabulary. Since gold standards are already tokenized, they do not require additional tokenization.

3.3 Baselines

We compare to three popular statistical alignment models that all require parallel training data. **fast-align/IBM2** (Dyer et al., 2013) is an implementation of an alignment algorithm based on IBM Model 2. It is popular because of its speed and high quality. **eflomal**[4] (based on efmaral by Östling and Tiedemann (2016)), a Bayesian model with Markov Chain Monte Carlo inference, is claimed to outperform fast-align on speed and quality. Further we use the widely used software package **Giza++/IBM4** (Och and Ney, 2003), which implements IBM alignment models. We use its standard settings: 5 iterations each for the HMM model, IBM Models 1, 3 and 4 with $p_0 = 0.98$.

Symmetrization. Probabilistic word alignment models create forward and backward alignments and then symmetrize them (Och and Ney, 2003; Koehn et al., 2005). We compared the symmetrization methods grow-diag-final-and (GDFA) and intersection and found them to perform comparably; see supplementary. We use GDFA throughout the paper.

[4] github.com/robertostling/eflomal

3.4 Evaluation Measures

Given a set of predicted alignment edges A and a set of sure, possible gold standard edges S, P (where $S \subset P$), we use the following evaluation measures:

$$\text{prec} = \frac{|A \cap P|}{|A|}, \text{rec} = \frac{|A \cap S|}{|S|},$$
$$F_1 = \frac{2 \text{ prec rec}}{\text{prec} + \text{rec}},$$
$$\text{AER} = 1 - \frac{|A \cap S| + |A \cap P|}{|A| + |S|},$$

where $|\cdot|$ denotes the cardinality of a set. This is the standard evaluation (Och and Ney, 2003).

3.5 Data

Our **test data** are a diverse set of 6 language pairs: Czech, German, Persian, French, Hindi and Romanian, always paired with English. See Table 11 for corpora and supplementary for URLs.

For our baselines requiring parallel training data (i.e., eflomal, fast-align and Giza++) we select additional parallel **training data** that is consistent with the target domain where available. See Table 11 for the corpora. Unless indicated otherwise we use the whole parallel training data. Figure 5 shows the effect of using more or less training data.

Given the large amount of possible experiments when considering 6 language pairs we do not have space to present all numbers for all languages. If we show results for only one pair, we choose ENG-DEU as it is an established and well-known dataset (EuroParl). If we show results for more languages we fall back to DEU, CES and HIN, to show effects on a mid-resource morphologically rich language (CES) and a low-resource language written in a different script (HIN).

4 Results

4.1 Embedding Layer

Figure 4 shows a parabolic trend across layers of mBERT and XLM-R. We use layer 8 in this paper because it has best performance. This is consistent with other work (Hewitt and Manning, 2019; Tenney et al., 2019): in the first layers the contextualization is too weak for high-quality alignments while the last layers are too specialized on the pre-training task (masked language modeling).

Lang.	Gold Standard	Gold St. Size	$\|S\|$	$\|P \setminus S\|$	Parallel Data	Parallel Data Size	Wikipedia Size
ENG-CES	(Mareček, 2008)	2500	44292	23132	EuroParl (Koehn, 2005)	646k	8M
ENG-DEU	EuroParl-based[a]	508	9612	921	EuroParl (Koehn, 2005)	1920k	48M
ENG-FAS	(Tavakoli and Faili, 2014)	400	11606	0	TEP (Pilevar et al., 2011)	600k	5M
ENG-FRA	WPT2003, (Och and Ney, 2000),	447	4038	13400	Hansards (Germann, 2001)	1130k	32M
ENG-HIN	WPT2005[b]	90	1409	0	Emille (McEnery et al., 2000)	3k	1M
ENG-RON	WPT2005[b]	203	5033	0	Constitution, Newspaper[b]	50k	3M

[a] www-i6.informatik.rwth-aachen.de/goldAlignment/
[b] http://web.eecs.umich.edu/~mihalcea/wpt05/

Table 1: Overview of datasets. "Lang." uses ISO 639-3 language codes. "Size" refers to the number of sentences. "Parallel Data Size" refers to the number of parallel sentences in addition to the gold alignments that is used for training the baselines. Our sentence tokenized version of the English Wikipedia has 105M sentences.

	Method	ENG-CES F_1	ENG-CES AER	ENG-DEU F_1	ENG-DEU AER	ENG-FAS F_1	ENG-FAS AER	ENG-FRA F_1	ENG-FRA AER	ENG-HIN F_1	ENG-HIN AER	ENG-RON F_1	ENG-RON AER
Prior Work	(Östling, 2015a) Bayesian							**.94**	**.06**	.57	.43	**.73**	**.27**
	(Östling, 2015a) Giza++							.92	.07	.51	.49	.72	.28
	(Legrand et al., 2016) Ensemble Method	.81	.16					.71	.10				
	(Östling and Tiedemann, 2016) efmaral							.93	.08	.53	.47	.72	.28
	(Östling and Tiedemann, 2016) fast-align							.86	.15	.33	.67	.68	.33
	(Zenkel et al., 2019) Giza++				.21				**.06**				.28
	(Garg et al., 2019) Multitask				.20				.08				
Baselines Word	fast-align/IBM2	.76	.25	.71	.29	.57	.43	.86	.15	.34	.66	.68	.33
	Giza++/IBM4	.75	.26	.77	.23	.51	.49	.92	.09	.45	.55	.69	.31
	eflomal	.85	.15	.77	.23	.61	.39	.93	.08	.51	.49	.71	.29
Baselines Subword	fast-align/IBM2	.78	.23	.71	.30	.58	.42	.85	.16	.38	.62	.68	.32
	Giza++/IBM4	.82	.18	.78	.22	.57	.43	.92	.09	.48	.52	.69	.32
	eflomal	.84	.17	.76	.24	.63	.37	.91	.09	.52	.48	.72	.28
This Work Word	fastText - Argmax	.70	.30	.60	.40	.50	.50	.77	.22	.49	.52	.47	.53
	mBERT[8] - Argmax	**.87**	**.13**	.79	.21	.67	.33	**.94**	**.06**	.54	.47	.64	.36
	XLM-R[8] - Argmax	**.87**	**.13**	.79	.21	.70	.30	.93	**.06**	.59	.41	.70	.30
This Work Subword	fastText - Argmax	.58	.42	.56	.44	.09	.91	.73	.26	.04	.96	.43	.58
	mBERT[8] - Argmax	.86	.14	**.81**	**.19**	.67	.33	**.94**	**.06**	.55	.45	.65	.35
	XLM-R[8] - Argmax	**.87**	**.13**	**.81**	**.19**	**.71**	**.29**	.93	.07	**.61**	**.39**	.71	.29

Table 2: Comparison of our methods, baselines and prior work in unsupervised word alignment. Best result per column in bold. A detailed version of the table with precision/recall and Itermax/Match results is in supplementary.

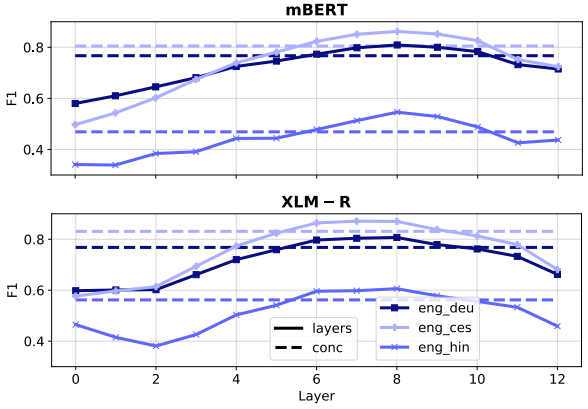

Figure 4: Word alignment performance across layers of mBERT (top) and XLM-R (bottom). Results are F_1 with Argmax at the subword level.

4.2 Comparison with Prior Work

Contextual Embeddings. Table 2 shows that mBERT and XLM-R consistently perform well with the Argmax method. XLM-R yields mostly higher values than mBERT. Our three baselines, eflomal, fast-align and Giza++, are always outperformed (except for RON). We outperform all prior work except for FRA where we match the performance and RON. This comparison is not entirely fair because methods relying on parallel data have access to the parallel sentences of the test data during training whereas our methods do not.

Romanian might be a special case as it exhibits a large amount of many to one links and further lacks determiners. How determiners are handled in the gold standard depends heavily on the annotation guidelines. Note that one of our settings, XLM-R[8] with Itermax at the subword level, has an F1 of .72 for ENG-RON, which comes very close to the performance by (Östling, 2015a) (see Table 3).

In summary, extracting alignments from similarity matrices is a very simple and efficient method that performs surprisingly strongly. It outperforms strong statistical baselines and most prior work in unsupervised word alignment for CES, DEU, FAS and HIN and is comparable for FRA and RON. We attribute this to the strong contextualization in mBERT and XLM-R.

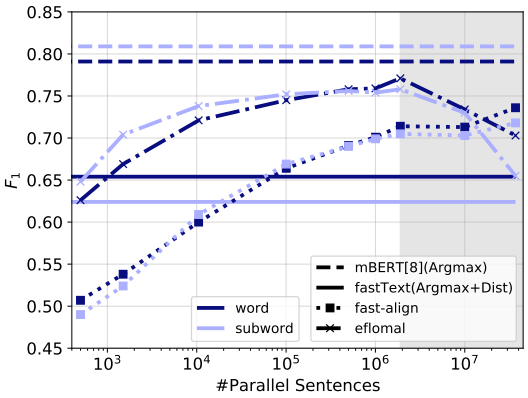

Figure 5: Learning curves of fast-align/eflomal vs. embedding-based alignments. Results shown are F_1 for ENG-DEU, contrasting subword and word representations. Up to 1.9M parallel sentences we use EuroParl. To demonstrate the effect with abundant parallel data we add up to 37M *additional* parallel sentences from ParaCrawl (Esplà et al., 2019) (see grey area).

Static Embeddings. fastText shows a solid performance on word level, which is worse but comes close to fast-align and outperforms it for HIN. We consider this surprising as fastText did not have access to parallel data or any multilingual signal. VecMap can also be used with crosslingual dictionaries. We expect this to boost performance and fastText could then become a viable alternative to fast-align.

Amount of Parallel Data. Figure 5 shows that fast-align and eflomal get better with more training data with eflomal outperforming fast-align, as expected. However, even with 1.9M parallel sentences mBERT outperforms both baselines. When adding up to 37M additional parallel sentences from ParaCrawl (Esplà et al., 2019) performance for fast-align increases slightly, however, eflomal decreases (grey area in plot). ParaCrawl contains mined parallel sentences whose lower quality probably harms eflomal. fastText (with distortion) is competitive with eflomal for fewer than 1000 parallel sentences and outperforms fast-align even with 10k sentences. Thus for very small parallel corpora (<10k sentences) using fastText embeddings is an alternative to fast-align.

The main takeaway from Figure 5 is that mBERT-based alignments, a method that does not need any parallel training data, outperforms state-of-the-art aligners like eflomal for ENG-DEU, even in the very high resource case.

Emb.	Method	ENG-CES	ENG-DEU	ENG-FAS	ENG-FRA	ENG-HIN	ENG-RON
mBERT[8]	Argmax	**.86**	**.81**	.67	**.94**	.55	.65
	Itermax	**.86**	**.81**	**.70**	.93	**.58**	**.69**
	Match	.82	.78	.67	.90	**.58**	.67
XLM-R[8]	Argmax	**.87**	**.81**	.71	**.93**	.61	.71
	Itermax	.86	.80	**.72**	.92	**.62**	**.72**
	Match	.81	.76	.68	.88	.60	.70

Table 3: Comparison of our three proposed methods across all languages for the best embeddings from Table 2: mBERT[8] and XLM-R[8]. We show F_1 at the subword level. Best result per embedding type in bold.

Emb.	$n_{\max}$	α	ENG-DEU Prec.	Rec.	F_1	AER	ENG-CES Prec.	Rec.	F_1	AER	ENG-HIN Prec.	Rec.	F_1	AER
mBERT[8]	1	-	**.92**	.69	.79	.21	**.95**	.80	**.87**	**.13**	**.84**	.39	.54	.47
	2	.90	.85	.77	**.81**	**.19**	.87	.87	**.87**	.14	.75	.47	.58	.42
		.95	.83	.80	**.81**	**.19**	.85	.89	**.87**	**.13**	.73	.48	.58	.42
		1	.77	.79	.78	.22	.80	.86	.83	.17	.63	.46	.53	.47
	3	.90	.81	.80	.80	.20	.83	.88	.85	.15	.70	.49	.57	.43
		.95	.78	**.83**	**.81**	.20	.81	**.91**	.86	.15	.68	**.52**	**.59**	**.41**
		1	.73	**.83**	.77	.23	.76	**.91**	.82	.18	.58	.51	.54	.46
fastText	1	-	**.81**	.48	.60	.40	**.86**	.59	.70	.30	**.75**	.36	.49	.52
	2	.90	.69	.56	**.62**	**.38**	.74	.69	**.72**	**.29**	.63	.42	**.51**	**.49**
		.95	.66	.56	.61	.39	.71	.69	.70	.30	.59	.41	.48	.52
		1	.59	.55	.57	.43	.62	.65	.63	.37	.53	.39	.45	.55
	3	.90	.63	**.59**	.61	.39	.67	.72	.70	.31	.57	.43	.49	.51
		.95	.59	**.59**	.59	.41	.63	**.73**	.68	.33	.53	**.44**	.48	.52
		1	.53	.58	.55	.45	.55	.70	.62	.39	.48	.43	.45	.55

Table 4: Itermax with different number of iterations ($n_{\max}$) and different α. Results are at the word level.

4.3 Additional Methods and Extensions

We already showed that Argmax yields alignments that are competitive with the state of the art. In this section we compare all our proposed methods and extensions more closely.

Itermax. Table 4 shows results for Argmax (i.e., 1 Iteration) as well as Itermax (i.e., 2 or more iterations of Argmax). As expected, with more iterations precision drops in favor of recall. Overall, Itermax achieves higher F_1 scores for the three language pairs (equal for ENG-CES) both for mBERT[8] and fastText embeddings. For Hindi the performance increase is the highest. We hypothesize that for more distant languages Itermax is more beneficial as similarity between wordpieces may be generally lower, thus exhibiting fewer mutual argmaxes. For the rest of the paper if we use Itermax we use 2 Iterations with $\alpha = 0.9$ as it exhibits best performance (5 out of 6 wins in Table 4).

Argmax/Itermax/Match. In Table 3 we compare our three proposed methods in terms of F_1 across all languages. We chose to show the two

Emb.	Method	ENG-DEU Prec.	Rec.	F_1	AER	ENG-CES Prec.	Rec.	F_1	AER	ENG-HIN Prec.	Rec.	F_1	AER
fastText	Argmax	.81	.48	.60	.40	.86	.59	.70	.30	**.75**	**.36**	**.49**	**.52**
	+Dist	**.84**	**.54**	**.65**	**.35**	**.89**	**.68**	**.77**	**.23**	.64	.30	.41	.59
	+Null	.81	.46	.59	.41	.86	.56	.68	.32	.74	.34	.46	.54
	Itermax	.69	.56	.62	.38	.74	.69	.72	.29	**.63**	**.42**	**.51**	**.49**
	+Dist	**.71**	**.62**	**.66**	**.34**	**.75**	**.76**	**.76**	**.25**	.54	.37	.44	.57
	+Null	.69	.53	.60	.40	.74	.66	.70	.30	**.63**	.40	.49	.51
	Match	.60	.58	.59	.41	.65	.71	.68	.32	.55	**.43**	**.48**	**.52**
	+Dist	**.67**	**.64**	**.65**	**.35**	**.72**	**.78**	**.75**	**.25**	.50	.39	.43	.57
	+Null	.61	.56	.58	.42	.66	.69	.67	.33	**.56**	.41	**.48**	**.52**
mBERT[8]	Argmax	.92	**.69**	**.79**	**.21**	**.95**	**.80**	**.87**	**.13**	.84	**.39**	**.54**	**.47**
	+Dist	.91	.67	.77	.23	.93	.79	.85	.15	.68	.29	.41	.59
	+Null	**.93**	.67	.78	.22	**.95**	.77	.85	.15	**.85**	.38	.53	**.47**
	Itermax	.85	**.77**	**.81**	**.19**	.87	**.87**	**.87**	**.14**	.75	**.47**	**.58**	**.43**
	+Dist	.82	.75	.79	.21	.84	.85	.85	.15	.56	.34	.43	.58
	+Null	**.86**	.75	.80	.20	**.88**	.84	.86	**.14**	**.76**	.45	.57	**.43**
	Match	.78	**.74**	**.76**	**.24**	.81	**.85**	**.83**	**.17**	.67	**.52**	**.59**	**.42**
	+Dist	.75	.71	.73	.27	.79	.83	.81	.20	.45	.35	.39	.61
	+Null	**.80**	.73	**.76**	**.24**	**.83**	.83	**.83**	**.17**	**.68**	.51	.58	**.42**

Table 5: Analysis of Null and Distortion Extensions. All alignments are obtained at word-level. Best result per embedding type and method in bold.

best performing settings from Table 2: mBERT[8] and XLM-R[8] at the subword level. Itermax performs slightly better than Argmax with 6 wins, 4 losses and 2 ties. Itermax seems to help more for more distant languages such as FAS, HIN and RON, but harms for FRA. Match has the lowest F_1, but generally exhibits a higher recall (see e.g., Table 5).

Null and Distortion Extensions. Table 5 shows that Argmax and Itermax generally have higher precision, whereas Match has higher recall. Adding Null almost always increases precision, but at the cost of recall, resulting mostly in a lower F_1 score. Adding a distortion prior boosts performance for static embeddings, e.g., from .70 to .77 for ENG-CES Argmax F_1 and similarly for ENG-DEU. For Hindi a distortion prior is harmful. Dist has little and sometimes harmful effects on mBERT indicating that mBERT's contextualized representations already match well across languages.

Summary. Argmax and Itermax exhibit the best and most stable performance. For most language pairs Itermax is recommended. If high recall alignments are required, Match is the recommended algorithm. Except for HIN, a distortion prior is beneficial for static embeddings. Null should be applied when one wants to push precision even higher (e.g., for annotation projection).

4.4 Words and Subwords

Table 2 shows that subword processing slightly outperforms word-level processing for most methods. Only fastText is harmed by subword processing.

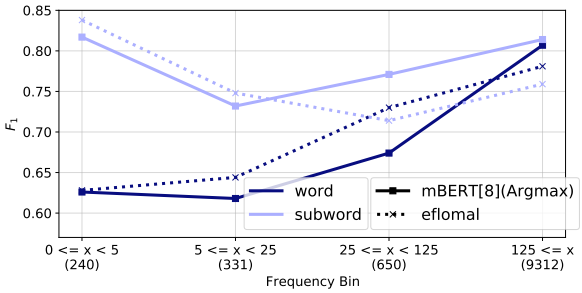

Figure 6: Results for different frequency bins on ENG-DEU. An edge in S, P, or A is attributed to exactly one bin based on the minimum frequency of the involved words (denoted by x). Number of gold edges in brackets. Eflomal is trained on all 1.9M parallel sentences. Frequencies are computed on the same corpus.

		ADJ	ADP	ADV	AUX	NOUN	PRON	VERB
eflomal	Word	**0.83**	0.69	**0.72**	0.63	0.85	0.79	0.63
	Subword	0.82	0.68	0.71	0.57	0.85	0.77	0.62
mBERT[8]	Word	0.79	0.74	0.71	0.71	0.81	**0.84**	**0.69**
	Subword	0.81	**0.75**	**0.72**	**0.72**	**0.87**	**0.84**	**0.69**

Table 6: Alignment performance (F_1) on ENG-DEU for POS. We use mBERT[8](Argmax) and Eflomal trained on 1.9M parallel sentences on the word level.

We use VecMap to match (sub)word distributions across languages. We hypothesize that it is harder to match subword than word distributions – this effect is strongest for Persian and Hindi, probably due to different scripts and thus different subword distributions. Initial experiments showed that adding supervision in form of a dictionary helps restore performance. We will investigate this in future work.

We hypothesize that subword processing is beneficial for aligning rare words. To show this, we compute our evaluation measures for different frequency bins. More specifically, we only consider gold standard alignment edges for the computation where at least one of the member words has a certain frequency in a reference corpus (in our case all 1.9M lines from the ENG-DEU EuroParl corpus). That is, we only consider the edge (i, j) in A, S or P if the minimum of the source and target word frequency is in $[\gamma_l, \gamma_u)$ where γ_l and γ_u are bin boundaries.

Figure 6 shows F_1 for different frequency bins. For rare words both eflomal and mBERT show a severely decreased performance at the word level, but not at the subword level. Thus, subword processing is indeed beneficial for rare words.

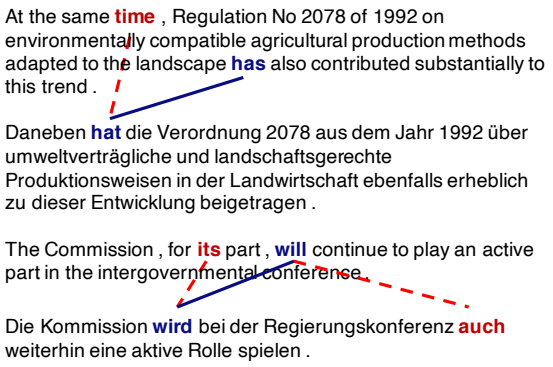

Figure 7: Example alignment of auxiliary verbs. Same setting as in Table 6. Solid lines: mBERT's alignment, identical to the gold standard. Dashed lines: eflomal's incorrect alignment.

4.5 Part-Of-Speech Analysis

To analyze the performance with respect to different part-of-speech (POS) tags, the ENG-DEU gold standard was tagged with the Stanza toolkit (Qi et al., 2020). We evaluate the alignment performance for each POS tag by only considering the alignment edges where at least one of their member words has this tag. Table 6 shows results for frequent POS tags. Compared to eflomal, mBERT aligns auxiliaries, pronouns and verbs better. The relative position of auxiliaries and verbs in German can diverge strongly from that in English because they occur at the end of the sentence (verb-end position) in many clause types. Positions of pronouns can also diverge due to a more flexible word order in German. It is difficult for an HMM-based aligner like eflomal to model such high-distortion alignments, a property that has been found by prior work as well (Ho and Yvon, 2019). In contrast, mBERT(Argmax) does not use distortion information, so high distortion is not a problem for it.

Figure 7 gives an example for auxiliaries. The gold alignment ("has" – "hat") is correctly identified by mBERT (solid line). Eflomal generates an incorrect alignment ("time" – "hat"): the two words have about the same relative position, indicating that distortion minimization is the main reason for this incorrect alignment. Analyzing all auxiliary alignment edges, the average absolute value of the distance between aligned words is 2.72 for eflomal and 3.22 for mBERT. This indicates that eflomal is more reluctant than mBERT to generate high-distortion alignments and thus loses accuracy.

5 Related Work

Brown et al. (1993) introduced the IBM models, the best known statistical word aligners. More recent aligners, often based on IBM models, include fast-align (Dyer et al., 2013), Giza++ (Och and Ney, 2003) and eflomal (Östling and Tiedemann, 2016). (Östling, 2015a) showed that Bayesian Alignment Models perform well. Neural network based extensions of these models have been considered (Ayan et al., 2005; Ho and Yvon, 2019). All of these models are trained on parallel text. Our method instead aligns based on embeddings that are induced from monolingual data only. We compare with prior methods and observe comparable performance.

Prior work on using learned representations for alignment includes (Smadja et al., 1996; Och and Ney, 2003) (Dice coefficient), (Jalili Sabet et al., 2016) (incorporation of embeddings into IBM models), (Legrand et al., 2016) (neural network alignment model) and (Pourdamghani et al., 2018) (embeddings are used to encourage words to align to similar words). Tamura et al. (2014) use recurrent neural networks to learn alignments. They use noise contrastive estimation to avoid supervision. Yang et al. (2013) train a neural network that uses pretrained word embeddings in the initial layer. All of this work requires parallel data. mBERT is used for word alignments in concurrent work: Libovický et al. (2019) use the high quality of mBERT alignments as evidence for the "language-neutrality" of mBERT. Nagata et al. (2020) phrase word alignment as crosslingual span prediction and finetune mBERT using gold alignments.

Attention in NMT (Bahdanau et al., 2014) is related to a notion of soft alignment, but often deviates from conventional word alignments (Ghader and Monz, 2017; Koehn and Knowles, 2017). One difference is that standard attention does not have access to the target word. To address this, Peter et al. (2017) tailor attention matrices to obtain higher quality alignments. Li et al. (2018)'s and Zenkel et al. (2019)'s models perform similarly to and Zenkel et al. (2020) outperform Giza++. Ding et al. (2019) propose better decoding algorithms to deduce word alignments from NMT predictions. Chen et al. (2016), Mi et al. (2016) and Garg et al. (2019) obtain alignments and translations in a multitask setup. Garg et al. (2019) find that operating at the subword level can be beneficial for alignment models. Li et al. (2019) propose two methods to extract alignments from NMT

models, however they do not outperform fast-align. Stengel-Eskin et al. (2019) compute similarity matrices of encoder-decoder representations that are leveraged for word alignments, together with supervised learning, which requires manually annotated alignment. We find our proposed methods to be competitive with these approaches. In contrast to our work, they all require parallel data.

6 Conclusion

We presented word aligners based on contextualized embeddings that outperform in four and match the performance of state-of-the-art aligners in two language pairs; e.g., for ENG-DEU contextualized embeddings achieve an alignment F_1 that is 5 percentage points higher than eflomal trained on 100k parallel sentences. Further, we showed that alignments from static embeddings can be a viable alternative to statistical aligner when few parallel training data is available. In contrast to all prior work our methods do not require parallel data for training at all. With our proposed methods and extensions such as Match, Itermax and Null it is easy to obtain higher precision or recall depending on the use case.

Future work includes modeling fertility explicitly and investigating how to incorporate parallel data into the proposed methods.

Acknowledgments

We gratefully acknowledge funding through a Zentrum Digitalisierung.Bayern fellowship awarded to the second author. This work was supported by the European Research Council (# 740516). We thank Matthias Huck, Jindřich Libovický, Alex Fraser and the anonymous reviewers for interesting discussions and valuable comments. Thanks to Jindřich for pointing out that mBERT can align mixed-language sentences as shown in Figure 1.

References

Tamer Alkhouli, Gabriel Bretschner, and Hermann Ney. 2018. On the alignment problem in multi-head attention-based neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, Belgium, Brussels. Association for Computational Linguistics.

Tamer Alkhouli, Gabriel Bretschner, Jan-Thorsten Peter, Mohammed Hethnawi, Andreas Guta, and Hermann Ney. 2016. Alignment-based neural machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, Berlin, Germany. Association for Computational Linguistics.

Tamer Alkhouli and Hermann Ney. 2017. Biasing attention-based recurrent neural networks using external alignment information. In *Proceedings of the Second Conference on Machine Translation*, Copenhagen, Denmark. Association for Computational Linguistics.

Waleed Ammar, George Mulcaire, Yulia Tsvetkov, Guillaume Lample, Chris Dyer, and Noah A Smith. 2016. Massively multilingual word embeddings. *arXiv preprint arXiv:1602.01925*.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Melbourne, Australia. Association for Computational Linguistics.

Ehsaneddin Asgari and Hinrich Schütze. 2017. Past, present, future: A computational investigation of the typology of tense in 1000 languages. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark. Association for Computational Linguistics.

Necip Fazil Ayan, Bonnie J. Dorr, and Christof Monz. 2005. NeurAlign: Combining word alignments using neural networks. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations*.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. O'Reilly Media, Inc.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2).

Wenhu Chen, Evgeny Matusov, Shahram Khadivi, and Jan-Thorsten Peter. 2016. Guided alignment training for topic-aware neural machine translation. *AMTA 2016*.

Trevor Cohn, Cong Duy Vu Hoang, Ekaterina Vymolova, Kaisheng Yao, Chris Dyer, and Gholamreza Haffari. 2016. Incorporating structural alignment biases into an attentional neural translation model. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 876–885, San Diego, California. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online. Association for Computational Linguistics.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *Proceedings of the Sixth International Conference on Learning Representations*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota. Association for Computational Linguistics.

Shuoyang Ding, Hainan Xu, and Philipp Koehn. 2019. Saliency-driven word alignment interpretation for neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, Florence, Italy. Association for Computational Linguistics.

Philipp Dufter, Mengjie Zhao, Martin Schmitt, Alexander Fraser, and Hinrich Schütze. 2018. Embedding learning through multilingual concept induction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Melbourne, Australia. Association for Computational Linguistics.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Atlanta, Georgia. Association for Computational Linguistics.

Miquel Esplà, Mikel Forcada, Gema Ramírez-Sánchez, and Hieu Hoang. 2019. ParaCrawl: Web-scale parallel corpora for the languages of the EU. In *Proceedings of Machine Translation Summit XVII Volume 2: Translator, Project and User Tracks*, Dublin, Ireland. European Association for Machine Translation.

Zvi Galil. 1986. Efficient algorithms for finding maximum matching in graphs. *ACM Computing Surveys (CSUR)*, 18(1).

Sarthak Garg, Stephan Peitz, Udhyakumar Nallasamy, and Matthias Paulik. 2019. Jointly learning to align and translate with transformer models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China. Association for Computational Linguistics.

Ulrich Germann. 2001. Aligned Hansards of the 36th parliament of Canada.

Hamidreza Ghader and Christof Monz. 2017. What does attention in neural machine translation pay attention to? In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Jiang Guo, Wanxiang Che, David Yarowsky, Haifeng Wang, and Ting Liu. 2016. A representation learning framework for multi-source transfer parsing. In *Thirtieth AAAI Conference on Artificial Intelligence*.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota. Association for Computational Linguistics.

Anh Khoa Ngo Ho and François Yvon. 2019. Neural baselines for word alignment. In *Proceedings of the 16th International Workshop on Spoken Language Translation*.

Matthias Huck, Diana Dutka, and Alexander Fraser. 2019. Cross-lingual annotation projection is effective for neural part-of-speech tagging. In *Proceedings of the Sixth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 223–233, Ann Arbor, Michigan. Association for Computational Linguistics.

Masoud Jalili Sabet, Heshaam Faili, and Gholamreza Haffari. 2016. Improving word alignment of rare words with word embeddings. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, Osaka, Japan. The COLING 2016 Organizing Committee.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Machine Translation Summit*, volume 5.

Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne, and

David Talbot. 2005. Edinburgh system description for the 2005 IWSLT speech translation evaluation. In *International Workshop on Spoken Language Translation (IWSLT) 2005*.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, Vancouver. Association for Computational Linguistics.

Harold W Kuhn. 1955. The Hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2).

Joël Legrand, Michael Auli, and Ronan Collobert. 2016. Neural network-based word alignment through score aggregation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, Berlin, Germany. Association for Computational Linguistics.

William D. Lewis and Fei Xia. 2008. Automatically identifying computationally relevant typological features. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II*.

Xintong Li, Guanlin Li, Lemao Liu, Max Meng, and Shuming Shi. 2019. On the word alignment from neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy. Association for Computational Linguistics.

Xintong Li, Lemao Liu, Zhaopeng Tu, Shuming Shi, and Max Meng. 2018. Target foresight based attention for neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, New Orleans, Louisiana. Association for Computational Linguistics.

Jindřich Libovický, Rudolf Rosa, and Alexander Fraser. 2019. How language-neutral is multilingual BERT? *arXiv preprint arXiv:1911.03310*.

Lemao Liu, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. 2016. Neural machine translation with supervised attention. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, Osaka, Japan. The COLING 2016 Organizing Committee.

David Mareček. 2008. Automatic alignment of tectogrammatical trees from Czech-English parallel corpus. Master's thesis, Charles University, MFF UK.

Anthony McEnery, Paul Baker, Rob Gaizauskas, and Hamish Cunningham. 2000. Emille: Building a corpus of South Asian languages. *VIVEK-BOMBAY-*, 13(3).

I. Dan Melamed. 2000. Models of translation equivalence among words. *Computational Linguistics*, 26(2).

Haitao Mi, Zhiguo Wang, and Abe Ittycheriah. 2016. Supervised attentions for neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas. Association for Computational Linguistics.

Rada Mihalcea and Ted Pedersen. 2003. An evaluation exercise for word alignment. In *Proceedings of the HLT-NAACL 2003 Workshop on Building and Using Parallel Texts: Data Driven Machine Translation and Beyond*.

Masaaki Nagata, Chousa Katsuki, and Masaaki Nishino. 2020. A supervised word alignment method based on cross-language span prediction using multilingual BERT. *arXiv preprint arXiv:2004.14516*.

Franz Josef Och and Hermann Ney. 2000. Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, Hong Kong. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1).

Robert Östling. 2015a. *Bayesian models for multilingual word alignment*. Ph.D. thesis, Department of Linguistics, Stockholm University.

Robert Östling. 2015b. Word order typology through multilingual word alignment. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, Beijing, China. Association for Computational Linguistics.

Robert Östling and Jörg Tiedemann. 2016. Efficient word alignment with Markov Chain Monte Carlo. *The Prague Bulletin of Mathematical Linguistics*, 106(1).

Sebastian Padó and Mirella Lapata. 2009. Cross-lingual annotation projection for semantic roles. *Journal of Artificial Intelligence Research*, 36.

Jan-Thorsten Peter, Arne Nix, and Hermann Ney. 2017. Generating alignments using target foresight in attention-based neural machine translation. *The Prague Bulletin of Mathematical Linguistics*, 108(1).

Mohammad Taher Pilevar, Heshaam Faili, and Abdol Hamid Pilevar. 2011. TEP: Tehran English-Persian parallel corpus. In *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer.

Nima Pourdamghani, Marjan Ghazvininejad, and Kevin Knight. 2018. Using word vectors to improve word alignments for low resource machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, New Orleans, Louisiana. Association for Computational Linguistics.

Ofir Press and Noah A Smith. 2018. You may not need attention. *arXiv preprint arXiv:1810.13409*.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, Online. Association for Computational Linguistics.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2019. Wikimatrix: Mining 135m parallel sentences in 1620 language pairs from wikipedia. *arXiv preprint arXiv:1907.05791*.

Frank Smadja, Kathleen R. McKeown, and Vasileios Hatzivassiloglou. 1996. Translating collocations for bilingual lexicons: A statistical approach. *Computational Linguistics*, 22(1).

Elias Stengel-Eskin, Tzu-ray Su, Matt Post, and Benjamin Van Durme. 2019. A discriminative neural model for cross-lingual word alignment. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China. Association for Computational Linguistics.

Akihiro Tamura, Taro Watanabe, and Eiichiro Sumita. 2014. Recurrent neural networks for word alignment model. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Baltimore, Maryland. Association for Computational Linguistics.

Leila Tavakoli and Heshaam Faili. 2014. Phrase alignments in parallel corpus using bootstrapping approach. *International Journal of Information & Communication Technology Research*, 6(3).

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy. Association for Computational Linguistics.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020. CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of The 12th Language Resources and Evaluation Conference*, Marseille, France. European Language Resources Association.

Nan Yang, Shujie Liu, Mu Li, Ming Zhou, and Nenghai Yu. 2013. Word alignment modeling with context dependent deep neural network. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Sofia, Bulgaria. Association for Computational Linguistics.

David Yarowsky, Grace Ngai, and Richard Wicentowski. 2001. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of the First International Conference on Human Language Technology Research*.

Thomas Zenkel, Joern Wuebker, and John DeNero. 2019. Adding interpretable attention to neural translation models improves word alignment. *arXiv preprint arXiv:1901.11359*.

Thomas Zenkel, Joern Wuebker, and John DeNero. 2020. End-to-end neural word alignment outperforms GIZA++. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1605–1617, Online. Association for Computational Linguistics.

A Additional Non-central Results

A.1 Comparison with Prior Work

A more detailed version of Table 2 from the main paper that includes precision and recall and results on Itermax can be found in Table 7.

A.2 Rare Words

Figure 8 shows the same as Figure 6 from the main paper but now with a reference corpus of 100k/1000k instead of 1920k parallel sentences. The main takeaways are similar.

A.3 Symmetrization

For asymmetric alignments different symmetrization methods exist. Dyer et al. (2013) provide an overview and implementation (fast-align) for these methods, which we use. We compare intersection and grow-diag-final-and (GDFA) in Table 9. In terms of F1, GDFA performs better (Intersection wins four times, GDFA eleven times, three ties). As expected, Intersection yields higher precision while GDFA yields higher recall. Thus intersection is preferable for tasks like annotation projection,

		Method	ENG-CES Prec.	Rec.	F_1	AER	ENG-DEU Prec.	Rec.	F_1	AER	ENG-FAS Prec.	Rec.	F_1	AER	ENG-FRA Prec.	Rec.	F_1	AER	ENG-HIN Prec.	Rec.	F_1	AER	ENG-RON Prec.	Rec.	F_1	AER
Prior Work		(Östling, 2015a) Bayesian													.96	.92	**.94**	**.06**	.85	.43	.57	.43	.91	.61	**.73**	**.27**
		(Östling, 2015a) Giza++													**.98**	.87	.92	.07	.63	.44	.51	.49	.85	.63	.72	.28
		(Legrand et al., 2016) Ensemble Method	.79	.83	.81	.16									.59	.90	.71	.10								
		(Östling and Tiedemann, 2016) efmaral															.93	.08			.53	.47			.72	.28
		(Östling and Tiedemann, 2016) fast-align															.86	.15			.33	.67			.68	.33
		(Zenkel et al., 2019) Giza++								.21								**.06**								.28
		(Garg et al., 2019) Multitask								.20								.08								
Baselines	Word	fast-align/IBM2	.71	.81	.76	.25	.70	.73	.71	.29	.60	.54	.57	.43	.81	.93	.86	.15	.34	.33	.34	.66	.69	**.67**	.68	.33
		Giza++/IBM4	.71	.79	.75	.26	.79	.75	.77	.23	.55	.48	.51	.49	.90	.95	.92	.09	.47	.43	.45	.55	.74	.64	.69	.31
		eflomal	.84	.86	.85	.15	.80	.75	.77	.23	.68	.55	.61	.39	.91	.94	.93	.08	.61	.44	.51	.49	.81	.63	.71	.29
	Subword	fast-align/IBM2	.72	.84	.78	.23	.67	.74	.71	.30	.60	.56	.58	.42	.80	.92	.85	.16	.39	.37	.38	.62	.69	**.67**	.68	.32
		Giza++/IBM4	.79	.86	.82	.18	.78	.78	.78	.22	.58	.56	.57	.43	.89	.95	.92	.09	.52	.44	.48	.52	.74	.64	.69	.32
		eflomal	.80	.88	.84	.17	.74	.78	.76	.24	.66	.60	.63	.37	.88	.95	.91	.09	.58	.47	.52	.48	.78	**.67**	.72	.28
This Work	Word	fastText - Itermax	.74	.69	.72	.29	.69	.56	.62	.38	.63	.45	.53	.48	.74	.78	.76	.24	.63	.42	.51	.49	.64	.40	.50	.51
		mBERT[8] - Itermax	.87	.87	**.87**	.14	.85	.77	**.81**	**.19**	.80	.63	.70	.30	.91	.95	.93	.08	.75	.47	.58	.43	.82	.58	.68	.32
		XLM-R[8] - Itermax	.89	.85	**.87**	**.13**	.86	.73	.79	.21	.84	.63	**.72**	**.28**	.91	.93	.92	.08	.79	.49	.61	**.39**	.87	.61	.71	.29
		fastText - Argmax	.86	.59	.70	.30	.81	.48	.60	.40	.75	.38	.50	.50	.85	.71	.77	.22	.75	.36	.49	.52	.77	.34	.47	.53
		mBERT[8] - Argmax	.95	.80	**.87**	**.13**	.92	.69	.79	.21	.88	.54	.67	.33	.97	.91	**.94**	**.06**	.84	.39	.54	.47	.90	.50	.64	.36
		XLM-R[8] - Argmax	**.96**	.80	**.87**	**.13**	**.93**	.68	.79	.22	**.91**	.57	.70	.30	.96	.91	.93	**.06**	**.88**	.45	.59	.41	**.94**	.56	.70	.30
	Subword	fastText - Itermax	.61	.57	.59	.41	.63	.54	.58	.42	.20	.07	.11	.90	.70	.76	.73	.28	.14	.05	.07	.93	.56	.38	.45	.55
		mBERT[8] - Itermax	.84	**.89**	.86	.14	.83	**.80**	**.81**	**.19**	.76	.65	.70	.30	.91	**.96**	.93	.08	.71	.49	.58	.42	.79	.62	.69	.31
		XLM-R[8] - Itermax	.84	**.89**	.86	.14	.83	.78	.80	.20	.79	**.67**	**.72**	**.28**	.89	.94	.92	.09	.75	**.52**	**.62**	**.39**	.83	.64	.72	.28
		fastText - Argmax	.72	.48	.58	.42	.75	.45	.56	.44	.27	.06	.09	.91	.80	.67	.73	.26	.14	.02	.04	.96	.67	.31	.43	.58
		mBERT[8] - Argmax	.92	.81	.86	.14	.92	.72	**.81**	**.19**	.85	.56	.67	.33	.96	.92	**.94**	**.06**	.81	.41	.55	.45	.88	.51	.65	.35
		XLM-R[8] - Argmax	.92	.83	**.87**	**.13**	.92	.72	**.81**	**.19**	.87	.59	.71	.30	.95	.91	.93	.07	.86	.47	.61	**.39**	.91	.59	.71	.29

Table 7: Comparison of word and subword levels. Best overall result per column in bold.

Emb.	Method	ENG-DEU Prec.	Rec.	F_1	AER	ENG-CES Prec.	Rec.	F_1	AER	ENG-HIN Prec.	Rec.	F_1	AER
fastText	Argmax	.75	.45	.56	.44	.72	.48	.58	.42	.14	.02	.04	.96
	+Dist	**.79**	**.51**	**.62**	**.38**	**.77**	**.58**	**.66**	**.34**	**.16**	**.04**	**.06**	**.94**
	+Null	.76	.43	.55	.45	.74	.47	.57	.42	.14	.02	.04	.96
	Itermax	.63	.54	.58	.42	.61	.57	.59	.41	.14	.05	.07	.93
	+Dist	**.67**	**.60**	**.64**	**.36**	**.63**	**.66**	**.65**	**.36**	**.15**	**.07**	**.09**	**.91**
	+Null	.64	.52	.57	.43	.62	.56	.59	.41	.14	.04	.07	.93
	Match	.51	.58	.54	.46	.44	.61	.52	.49	**.10**	.08	**.09**	**.91**
	+Dist	**.59**	**.66**	**.62**	**.38**	**.54**	**.71**	**.61**	**.39**	**.10**	**.09**	**.09**	**.91**
	+Null	.52	.57	.54	.46	.46	.60	.52	.48	**.10**	.08	**.09**	**.91**
mBERT[8]	Argmax	.92	**.72**	**.81**	**.19**	**.92**	**.81**	**.86**	**.14**	.81	**.41**	**.55**	**.45**
	+Dist	.90	.70	.79	.21	.91	.80	.85	.15	.65	.30	.41	.59
	+Null	**.93**	.70	.80	.20	**.92**	.78	.85	.15	**.82**	.40	.54	.47
	Itermax	.83	**.80**	**.81**	**.19**	.84	**.89**	**.86**	**.14**	.71	**.49**	**.58**	**.42**
	+Dist	.81	.77	.79	.21	.82	.87	.84	.16	.53	.35	.42	.58
	+Null	**.85**	.77	**.81**	.20	**.84**	.86	.85	.15	**.72**	.47	.57	.43
	Match	.75	**.80**	**.78**	**.23**	.76	**.90**	**.82**	**.18**	.64	**.52**	**.58**	**.43**
	+Dist	.72	.77	.75	.26	.74	.88	.80	.20	.45	.37	.40	.60
	+Null	**.77**	.78	**.78**	**.23**	**.77**	.88	**.82**	.19	**.65**	.51	.57	**.43**

Table 8: Comparison of methods for inducing alignments from similarity matrices. All results are subword-level. Best result per embedding type across columns in bold.

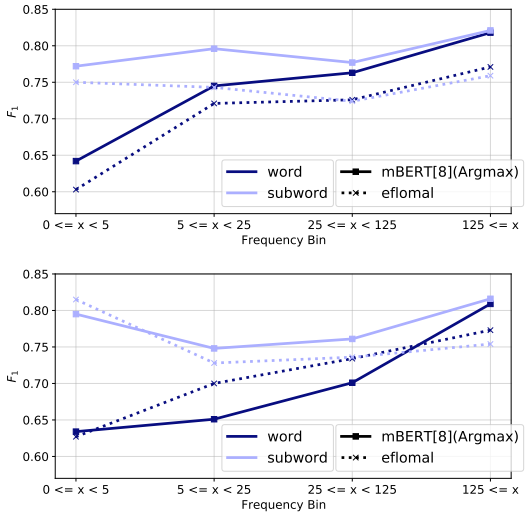

Figure 8: Results for different frequency bins. An edge in S, P, or A is attributed to exactly one bin based on the minimum frequency of the involved words (denoted by x). Top: Eflomal trained and frequencies computed on 100k parallel sentences. Bottom: 1000k parallel sentences.

whereas GDFA is typically used in statistical machine translation.

A.4 Alignment Examples for Different Methods

We show examples in Figure 10, Figure 11, Figure 12, and Figure 13. They provide an overview how the methods actually affect results.

B Hyperparameters

B.1 Overview

We provide a list of customized hyperparameters used in our computations in Table 10. There are three options how we came up with the hyperparameters: a) We simply used default values of 3rd party software. b) We chose an arbitrary value.

Method	Symm.	ENG-CES Prec.	Rec.	F_1	AER	ENG-DEU Prec.	Rec.	F_1	AER	ENG-FAS Prec.	Rec.	F_1	AER	ENG-FRA Prec.	Rec.	F_1	AER	ENG-HIN Prec.	Rec.	F_1	AER	ENG-RON Prec.	Rec.	F_1	AER
eflomal	Inters.	**.95**	.79	**.86**	**.14**	**.91**	.66	.76	.24	**.88**	.43	.58	.42	**.96**	.90	**.93**	**.07**	**.81**	.37	**.51**	**.49**	**.91**	.56	.70	.31
	GDFA	.84	**.86**	.85	.15	.80	**.75**	**.77**	**.23**	.68	**.55**	**.61**	**.39**	.91	**.94**	**.93**	.08	.61	**.44**	**.51**	**.49**	.81	**.63**	**.71**	**.29**
fast-align	Inters.	**.89**	.69	**.78**	**.22**	**.87**	.60	**.71**	**.29**	**.78**	.43	.55	.45	**.93**	.84	**.88**	**.11**	**.55**	.22	.31	.69	**.89**	.50	.64	.36
	GDFA	.71	**.81**	.76	.25	.70	**.73**	**.71**	**.29**	.60	**.54**	**.57**	**.43**	.81	**.93**	.86	.15	.34	**.33**	**.34**	**.66**	.69	**.67**	**.68**	**.33**
GIZA++	Inters.	**.95**	.60	.74	**.26**	**.92**	.62	.74	.26	**.89**	.26	.40	.60	**.97**	.89	**.93**	**.06**	**.82**	.25	.38	.62	**.95**	.47	.63	.37
	GDFA	.71	**.79**	**.75**	**.26**	.79	**.75**	**.77**	**.23**	.55	**.48**	**.51**	**.49**	.90	**.95**	.92	.09	.47	**.43**	**.45**	**.55**	.74	**.64**	**.69**	**.31**

Table 9: Comparison of symmetrization methods at the word level. Best result across rows per method in bold.

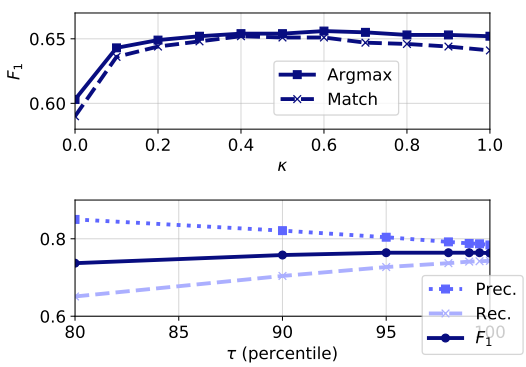

Figure 9: Top: F1 for ENG-DEU with fastText at word-level for different values of κ. Bottom: Performance for ENG-DEU with mBERT[8] (Match) at word-level when setting the value of τ to different percentiles. τ can be used for trading precision against recall. F_1 remains stable although it decreases slightly when assigning τ the value of a smaller percentile (e.g., 80).

Usually we fell back to well-established and rather conventional values (e.g., embedding dimension 300 for static embeddings). c) We defined a reasonable but arbitrary range, out of which we selected the best value using grid search. Table 10 lists the final values we used as well as how we came up with the specific value. For option c) the corresponding analyses are in Figure 4 and Table 3 in the main paper as well as in §B.2 in this supplementary material.

B.2 Null and Distortion Extensions

In Figure 9 we plot the performance for different values of κ. We observe that introducing distortion indeed helps (i.e., $\kappa > 0$) but the actual value is not decisive for performance. This is rather intuitive, as a small adjustment to the similarities is sufficient while larger adjustments do not necessarily change the argmax or the optimal point in the matching algorithm. We choose $\kappa = 0.5$.

For τ in null-word extension, we plot precision, recall and F_1 in Figure 9 when assigning τ different percentile values. Note that values for τ depend on the similarity distribution of all aligned edges. As expected, when using the 100th percentile no edges are removed and thus the performance is not changed compared to not having a null-word extension. When decreasing the value of τ the precision increases and recall goes down, while F_1 remains stable. We use the 95th percentile for τ .

C Reproducibility Information

C.1 Computing Infrastructures, Runtimes, Number of Parameters

We did all computations on up to 48 cores of Intel(R) Xeon(R) CPU E7-8857 v2 with 1TB memory and a single GeForce GTX 1080 GPU with 8GB memory.

Runtimes for aligning 500 parallel sentences on ENG-DEU are reported in Table 12. mBERT and XLM-R computations are done on the GPU. Note that fast-align, GIZA++ and eflomal usually need to be trained on much more parallel data to achieve better performance: this increases their runtime.

All our proposed methods are **parameter-free**. If we consider the parameters of the pretrained language models and pretrained embeddings then fastText has around 1 billion parameters (up to 500k words per language, 7 languages and embedding dimension 300), mBERT has 172 million, XLM-R 270 million parameters.

Method	Runtime[s]
fast-align	4
GIZA++	18
eflomal	5
mBERT[8] - Argmax	15
XLM-R[8] - Argmax	22

Table 12: Runtime (average across 5 runs) in seconds for each method to align 500 parallel sentences.

C.2 Data

Table 11 provides download links to all data used.

System	Parameter	Value
fastText	Version	0.9.1
	Code URL	https://github.com/facebookresearch/fastText/archive/v0.9.1.zip
	Downloaded on	11.11.2019
	Embedding Dimension	300
mBERT,XLM-R	Code: Huggingface Transformer	Version 2.3.1
	Maximum Sequence Length	128
fastalign	Code URL	https://github.com/clab/fast_align
	Git Hash	7c2bbca3d5d61ba4b0f634f098c4fcf63c1373e1
	Flags	-d -o -v
eflomal	Code URL	https://github.com/robertostling/eflomal
	Git Hash	9ef1ace1929c7687a4817ec6f75f47ee684f9aff
	Flags	–model 3
GIZA++	Code URL	http://web.archive.org/web/20100221051856/http://code.google.com/p/giza-pp
	Version	1.0.3
	Iterations	5 iter. HMM, 5 iter. Model 1, 5 iter. Model3, 5 iter. Model 4 (DEFAULT)
	p0	0.98
Vecmap	Code URL	https://github.com/artetxem/vecmap.git
	Git Hash	b82246f6c249633039f67fa6156e51d852bd73a3
	Manual Vocabulary Cutoff	500000
Distortion Ext.	κ	0.5 (chosen ouf of $[0.0, 0.1, \ldots, 1.0]$ by grid search, criterion: F_1)
Null Extension	τ	95th percentile of similarity distribution of aligned edges (chosen out of [80, 90, 95, 98, 99, 99.5] by grid search, criterion: F_1)
Argmax	Layer	8 (for mBERT and XLM-R, chosen out of $[0, 1, \ldots, 12]$ by grid search, criterion: F_1)
Vecmap	α	0.9 (chosen out of [0.9, 0.95, 1] by grid search, criterion: F_1)
	Iterations $n_{\max}$	2 (chosen out of [1,2,3] by grid search, criterion: F_1)

Table 10: Overview on hyperparameters. We only list parameters where we do **not** use default values. Shown are the values which we use unless specifically indicated otherwise.

Lang.	Name	Description	Link
ENG-CES	(Mareček, 2008)	Gold Alignment	http://ufal.mff.cuni.cz/czech-english-manual-word-alignment
ENG-DEU	EuroParl-based	Gold Alignment	www-i6.informatik.rwth-aachen.de/goldAlignment/
ENG-FAS	(Tavakoli and Faili, 2014)	Gold Alignment	http://eceold.ut.ac.ir/en/node/940
ENG-FRA	WPT2003, (Och and Ney, 2000),	Gold Alignment	http://web.eecs.umich.edu/ mihalcea/wpt/
ENG-HIN	WPT2005	Gold Alignment	http://web.eecs.umich.edu/ mihalcea/wpt05/
ENG-RON	WPT2005 (Mihalcea and Pedersen, 2003)	Gold Alignment	http://web.eecs.umich.edu/ mihalcea/wpt05/
ENG-CES	EuroParl (Koehn, 2005)	Parallel Data	https://www.statmt.org/europarl/
ENG-DEU	EuroParl (Koehn, 2005)	Parallel Data	https://www.statmt.org/europarl/
ENG-DEU	ParaCrawl	Parallel Data	https://paracrawl.eu/
ENG-FAS	TEP (Pilevar et al., 2011)	Parallel Data	http://opus.nlpl.eu/TEP.php
ENG-FRA	Hansards (Germann, 2001)	Parallel Data	https://www.isi.edu/natural-language/download/hansard/index.html
ENG-HIN	Emille (McEnery et al., 2000)	Parallel Data	http://web.eecs.umich.edu/mihalcea/wpt05/
ENG-RON	Constitution, Newspaper	Parallel Data	http://web.eecs.umich.edu/ mihalcea/wpt05/
All langs.	Wikipedia (downloaded October 2019)	Monolingual Text	download.wikimedia.org/[X]wiki/latest/[X]wiki-latest-pages-articles.xml.bz2

Table 11: Overview of datasets. “Lang.” uses ISO 639-3 language codes.

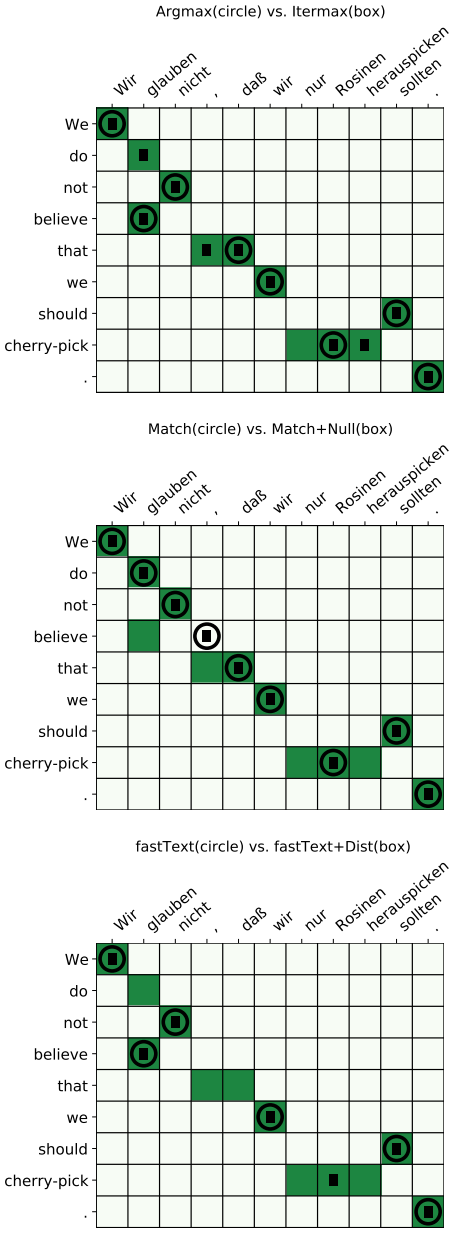

Figure 10: Comparison of alignment methods. Dark/light green: sure/possible edges in the gold standard. Circles are alignments from the first mentioned method in the subfigure title, boxes alignments from the second method.

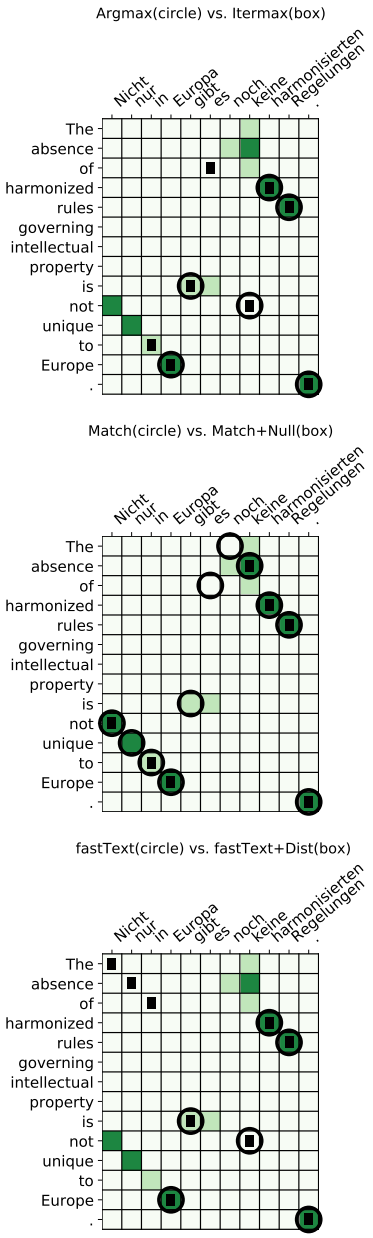

Figure 11: More examples.

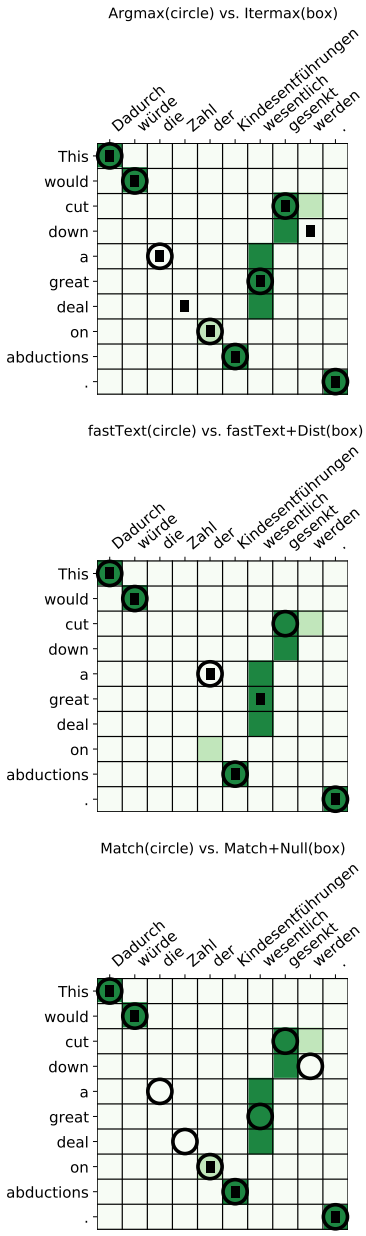

Figure 12: More examples.

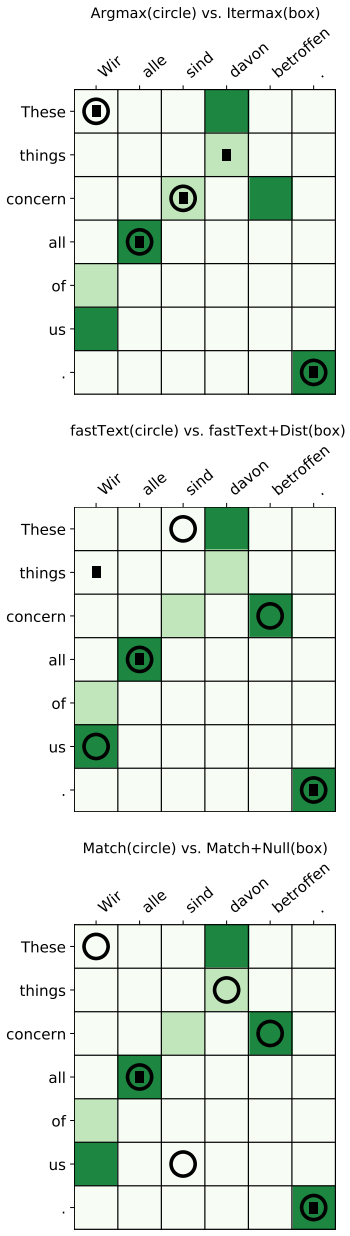

Figure 13: More examples.

Chapter 3

ParCourE: A Parallel Corpus Explorer for a Massively Multilingual Corpus

ParCourE: A Parallel Corpus Explorer for a Massively Multilingual Corpus

Ayyoob Imani[1], **Masoud Jalili Sabet**[1], **Philipp Dufter**[1],
Michael Cysouw[2], **Hinrich Schütze**[1]
[1]Center for Information and Language Processing (CIS), LMU Munich, Germany
[2]Research Center Deutscher Sprachatlas, Philipps University Marburg, Germany.
{ayyoob, masoud, philipp}@cis.lmu.de

Abstract

With more than 7000 languages worldwide, multilingual natural language processing (NLP) is essential both from an academic and commercial perspective. Researching typological properties of languages is fundamental for progress in multilingual NLP. Examples include assessing language similarity for effective transfer learning, injecting inductive biases into machine learning models or creating resources such as dictionaries and inflection tables. We provide ParCourE, an online tool that allows to browse a word-aligned parallel corpus, covering 1334 languages. We give evidence that this is useful for typological research. ParCourE can be set up for any parallel corpus and can thus be used for typological research on other corpora as well as for exploring their quality and properties.

1 Introduction

While ≈7000 languages are spoken (Eberhard et al., 2020), the bulk of NLP research addresses English only. However, multilinguality is an essential element of NLP. It not only supports exploiting common structures across languages and eases maintenance for globally operating companies, but also helps save languages from digital extinction and fosters more diversity in NLP techniques.

There are extensive resources that can be used for massively multilingual typological research, such as WALS (Dryer and Haspelmath, 2013), Glottolog (Hammarstrm et al., 2020), BabelNet (Navigli and Ponzetto, 2012) or http://panlex.org. Many of them are manually created or crowdsourced, which guarantees high quality, but limits coverage, both in terms of content and languages.

We work on the Parallel Bible Corpus (PBC) (Mayer and Cysouw, 2014), covering 1334 languages. More specifically, we provide a word-aligned version of PBC, created using state-of-the-art word alignment tools. As word alignments themselves are only of limited use, we provide an interactive online tool[1] that allows effective browsing of the alignments.

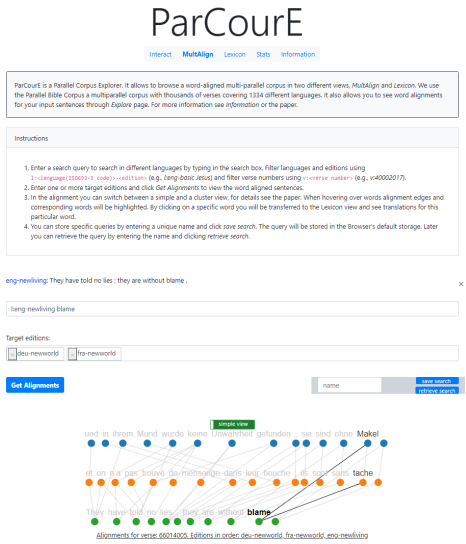

Figure 1: Screenshot of the ParCourE interface. It provides a word-aligned version of the Parallel Bible Corpus (PBC) spanning 1334 languages. Users can search for sentences in any language and see their alignments in other languages from MULTALIGN page. Alternatively they can feed their parallel sentences to INTERACTIVE view and see their word level alignments. They can look up translations of words in other languages, automatically induced from word alignments, from the LEXICON view (This page is interconnected with MULTALIGN). Statistics of the corpus is calculated and shown in the Stats view.

The main contributions of this work are: **i)** We provide a word-aligned version of the Parallel Bible Corpus (PBC) spanning 1334 languages and a total of 20M sentences ('verses'). For the alignment we use the state-of-the-art alignment methods SimAlign (Jalili Sabet et al., 2020) and Eflomal (Östling and Tiedemann, 2016a). **ii)** We release ParCourE,

[1]http://parcoure.cis.lmu.de/

58
Proceedings of the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations, pages 63–72, August 1st - August 6th, 2021.

a user interface for browsing word alignments, see the MULTALIGN view in Figure 1. We demonstrate the usefulness of ParCourE for typological research by presenting use cases in §6. **iii)** In addition to browsing word alignments, we provide an aggregated version in a LEXICON view and compute statistics that support assessing the quality of the word alignments. The two views (MULTALIGN and LEXICON views) are interlinked, resulting in a richer user experience. **iv)** ParCourE has a generic design and can be set up for any parallel corpus. This is useful for analyzing and managing parallel corpora; e.g., errors in an automatically mined parallel corpus can be inspected and flagged for correction.

2 Related Work

Word Alignment is an important tool for typological analysis (Lewis and Xia, 2008) and annotation projection (Yarowsky et al., 2001; Östling, 2015; Asgari and Schütze, 2017). Statistical models such as IBM models (Brown et al., 1993), Giza++ (Och and Ney, 2003), fast-align (Dyer et al., 2013) and Eflomal (Östling and Tiedemann, 2016b) are widely used. Recently, neural models were proposed, such as SimAlign (Jalili Sabet et al., 2020), Awesome-align (Dou and Neubig, 2021), and methods that are based on neural machine translation (Garg et al., 2019; Zenkel et al., 2020). We use Eflomal and SimAlign for generating alignments.

Resources. There are many online resources that enable typological research. WALS (Dryer and Haspelmath, 2013) provides manually created features for more than 2000 languages. We prepare a multiparallel corpus for investigating these features on real data. http://panlex.org is an online dictionary project with 2500 dictionaries covering 5700 languages and BabelNet (Navigli and Ponzetto, 2012) is a large semantic network covering 500 languages, but their information is generally on the type level, without access to example contexts. In contrast, ParCourE supports the exploration of word translations across 1334 languages in context.

Another line of work uses the **Parallel Bible Corpus (PBC)** for analysis. Asgari and Schütze (2017) investigate tense typology across PBC languages. Xia and Yarowsky (2017) created a multiway alignment based on fast-align (Dyer et al., 2013) and extracted resources such as paraphrases for 27 Bible editions. Wu et al. (2018) used alignments to extract names from the PBC.

One of the first attempts to index the Bible and align words in multiple languages were Strong's numbers (Strong, 2009[1890]); they tag words with similar meanings with the same ID. Mayer and Cysouw (2014) created an inverted index of word forms. Östling (2014) align massively parallel corpora simultaneously. We use the Eflomal word aligner by the same authorsostling2016efficient.

Finally, we review work on **Word Alignment Browsers.** Gilmanov et al. (2014)'s tool supports visualization and editing of word alignments. Akbik and Vollgraf (2017) use co-occurrence weights for word alignment and provide a tool for the inspection of annotation projection. Aulamo et al. (2020)'s filtering tool increases the quality of (mined) parallel corpora. Graën et al. (2017) rely on linguistic preprocessing, target corpus and word alignment exploration, do not show the graph of alignment edges and do not provide a dictionary view. While there is commonality with this prior work, ParCourE is distinguished by both its functionality and its motivating use cases: an important use case for us are typological searches; linguistic preprocessing is not available for many PBC languages; ParCourE can be used as an interactive explorer (but is not a fully-automated pipeline for a specific use case); our goal is not annotation; we use state-of-the-art word alignment methods. However, much of the complementary functionality in prior work would be useful additions to ParCourE. Another source of useful additional functionality would be work on embedding learning (Dufter et al., 2018; Kurfal and Östling, 2018) and machine translation (Tiedemann, 2018; Santy et al., 2019; Mueller et al., 2020) for PBC.

3 Features

ParCourE's user facing functionality can be divided into three main parts: MULTALIGN and LEXICON views and interconnections between the two.

3.1 Multiparallel Alignment Browser: MULTALIGN

ParCourE allows the user to search through the parallel corpus and check word alignments in a multiparallel corpus. An overview of MULTALIGN is shown in Figure 2.

In the **search field** (a(1)), the user can enter a text query and select (a(2)) multiple sentences for alignment. For narrowing the search scope, the

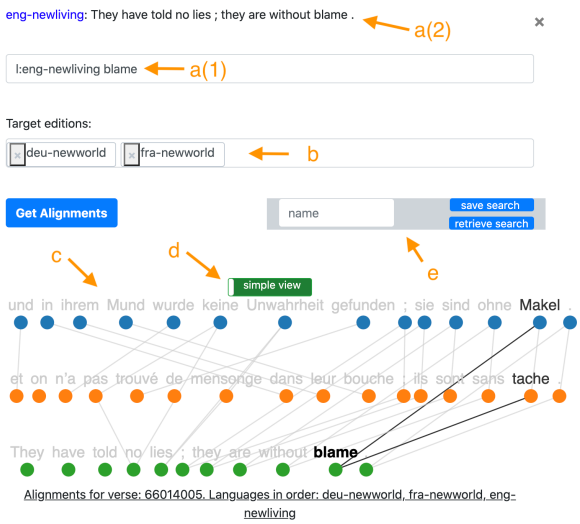

Figure 2: An overview of the MULTALIGN view. a) Search field for selecting sentences [a(1)] and the list of selected sentences [a(2)]. Any language can be used for the source sentence – in this case, it is English. b) Search bar for selecting the target languages. c) The alignment graph for the selected sentences in the source and the target languages. d) Switch button for simple view / cluster view. e) Save and retrieve search results

language and edition of the text segment can be specified in the beginning, e.g., by typing *l:eng-newworld2013*. Similarly, *v:40002017* specifies a verse ID.

PBC has 1334, so showing alignments for all translations of a sentence is difficult. We provide a drop-down (b) to select a subset of target languages for display.

For each sentence, a graph of alignment edges between selected languages is shown (c). By hovering over a word, the alignments of that word will be highlighted. Above each alignment graph, there is a button to switch between **Simple view** and **Cluster view** (d). In the simple view, when hovering over a word, only the alignment edges connected to that word are highlighted; in the cluster view, all words in a cluster (neighbors of neighbors) that are aligned together will be highlighted. We do not actually run any clustering algorithm on the alignment graph. Instead we simply highlight words that are up to two hops away from the hovered word. This helps spot a group of words across languages that have the same meaning.

Creating queries for typology research can take time. Thus, MULTALIGN allows the user to **save and retrieve** (e) queries.

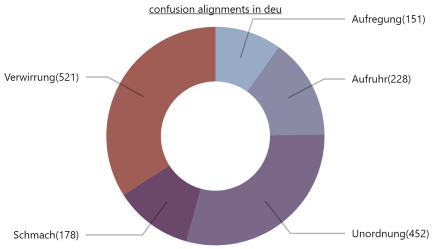

Figure 3: LEXICON view example: for the English word "confusion", there are five frequent translations in German. "Unordnung" literally means "disorder" and "Verwirrung" means "bewilderment".

3.2 Lexicon View: LEXICON

The MULTALIGN view allows the user to focus on word alignments on the sentence level and study the typological structure of languages in context. The LEXICON view focuses on word translations. The user can specify a source language by selecting the language code. This is to distinguish words with the same spelling in different languages. The user can search for one or multiple word(s) and specify target language(s). A pie chart for each target language depicting translations of the word is generated. Figure 3 shows German translations of "confusion" and the number of alignment edges for each. Word alignments are not perfect, so pie charts may also contain errors.

3.3 Interconnections

Both MULTALIGN and LEXICON views provide important features to the user for exploring the parallel corpus. For many use cases (cf. §6), the user may need to go back and forth between the views. For example, if she notices an error in the word alignment, she may want to check the LEXICON statistics to see if one of the typical translations of an incorrectly aligned word occurs in the sentence.

Thus, the two views are interconnected. In the MULTALIGN view, the user will be transferred to the LEXICON statistics of a word by clicking on it. This will open the LEXICON view, showing the search results for the selected word. Conversely, if the user clicks on one of the target translations in the LEXICON view, the MULTALIGN view will show sentences where this correspondence is part of the word alignment between source and target translation.

# editions	1758	# verses	20,470,892
# languages	1334	# verses / # editions	11,520
		# tokens / # verses	28.6

Table 1: PBC corpus statistics

3.4 Alignment Generation View: INTERACTIVE

The views mentioned so far provide the ability to search over the indexed corpus. This is useful when the main corpus of interest is fixed and the user has generated its alignments.

The INTERACTIVE view allows the user to study the alignments between arbitrary input sentences that are not necessarily in the corpus. Since the input sentences are not part of a corpus, INTERACTIVE uses SimAlign to generate alignments for all possible pairs of sentences. Similar to MULTALIGN, the INTERACTIVE view shows the alignment between the input sentences.

4 Experimental Setup

Corpus. We set up ParCourE on the PBC corpus provided by Mayer and Cysouw (2014). The version we use consists of 1758 editions (i.e., translations) of the Bible in 1334 languages (distinct ISO 639-3 codes). Table 1 shows corpus statistics. We use the PBC tokenization, which contains errors for a few languages (e.g., Thai). We extract word alignments for all possible language pairs. Since not all Bible verses are available in all languages, for each language pair we only consider mutually available verses.

PBC aligns Bible editions on the verse level by using verse-IDs that indicate book, chapter and verse (see below). Although one verse may contain multiple sentences, we do not split verses into individual sentences and consider each verse as one sentence.

Retrieval. Elasticsearch[2] is a fast and scalable open source search engine that provides distributed fulltext search. The setup is straightforward using an easy-to-use JSON web interface. We use it as the back-end for ParCourE's search requirement. We find that a single instance is capable of handling the whole PBC corpus efficiently, so we do not need a distributed setup. For bigger corpora, a distributed setup may be required. We created two types of inverted indices for our data: an edge-ngram index to support search-as-you-type capability and a standard index for normal queries.

Alignment Generation. SimAlign (Jalili Sabet et al., 2020) is a recent word alignment method that uses representations from pretrained language models to align sentences. It has achieved better results than statistical word aligners. For the languages that multilingual BERT (Devlin et al., 2019) supports, we use SimAlign to generate word alignments. For the remaining languages, we use Eflomal (Östling and Tiedemann, 2016a), an efficient word aligner using a Bayesian model with Markov Chain Monte Carlo (MCMC) inference. The alignments generated by SimAlign are symmetric. We use atools[3] and the grow-diag-final-and heuristic to symmetrize Eflomal alignments.

Lexicon Induction. We exploit the generated word alignments to induce lexicons for all 889,111 language pairs. To this end, we consider aligned words as translations of each other. For a given word from the source language, we count the number of times a word from the target language is aligned with it. The higher the number of alignments between two words, the higher the probability that the two have the same meaning. We filter out translations with frequency less than 5%.

5 Backend Design

An overview of our architecture can be found in Figure 4. The code is available online.[4]

Parallel Data Format. We use the PBC corpus format (Mayer and Cysouw, 2014): each verse has a unique ID across languages / editions, the *verse-ID*. The verse-ID is an 8-digit number, consisting of two digits for the book (e.g., 41 for the Gospel of Mark), three digits for the Chapter, and two digits for the verse itself. There are separate files for each edition. In each edition file, a line consists of the ID and the verse, separated by a tab.

Indexing. We identify a PBC verse using the following format: {verse-ID}@{language-code}-{edition-name}. We use this identifier to save and retrieve sentences with Elasticsearch. In addition, we store all metadata identifiers within Elasticsearch. Thus, we can search for a sentence by keyword, sentence number (= verse-ID), language code, or edition name.

ParCourE also supports the Corpus Alignment

[2] https://www.elastic.co/

[3] https://github.com/clab/fast_align

[4] https://github.com/cisnlp/parcoure

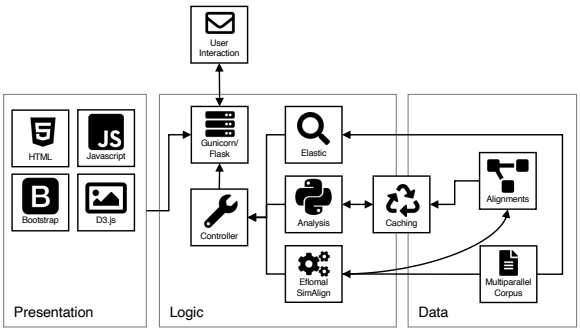

Figure 4: Overview of the system architecture. We use a standard front-end stack with d3.js for visualization. The backend is written in Python, which we use for computing alignments and performing analyses such as lexicon induction. We use Elasticsearch for search. The input is a multiparallel corpus for which all alignments are precomputed. For speeding up the system we use smart caching algorithms for our analyses. Icons taken without changes from https://fontawesome.com/license.

Encoding (CES)[5] format. One can download parallel corpora in CES format and use our tools to adapt them to ParCourE's input format.

Alignment Computation. Since Eflomal's performance depends on the amount of data it uses for training, we concatenate all editions to create a bigger training corpus for languages that have more than one edition. If language l_1 has two, and language l_2 three different editions, then the final training corpus for this language pair will contain six aligned edition pairs.

System Architecture. ParCourE is built on top of modern open source technologies, see Figure 4. The back-end uses the Flask web framework,[6] Gunicorn web server,[7] and Elasticsearch.[8] The front-end utilizes the Bootstrap CSS framework,[9] and the d3 visualization library.[10] Since all these tools are free and open-source, there is no restriction on setting up and releasing a new ParCourE instance. To extract word alignments, one can use any tool, such as Eflomal, fast_align or SimAlign.

Performance Improvements. For good runtime performance, we precompute the word alignments. Regarding LEXICON, given a query word and a target language, ParCourE first looks for a precomputed lexicon file; if it does not exist, ParCourE obtains the translations for the query word online. To accelerate the translation process, ParCourE employs Python's multiprocessing library. The number of CPU cores is decided online based on the number of editions available for source and target languages.

For a corpus with 1334 languages, we will end up with 890,445 alignment files and the same number of lexicon files. We cache alignment / lexicon files to speed up access. We use the Last Recently Used (LRU) cache replacement algorithm.

6 ParCourE Use Cases

Languages differ in how they encode meanings/functions. There are various aspects that make such differences an interesting problem when dealing with a dataset that has good coverage of the entire variation of the world's languages. (i) Many such differences between languages are not widely acknowledged in linguistic theory, so to document the extent of variation becomes a discovery of sorts. For example, the fact that interrogative words might distinguish between singular and plural (Figure 6) turns out to be a typologically salient differentiation (Mayer and Cysouw, 2012). (ii) The variation of linguistic marking is even stronger in the domain of grammatical function, like the differentiation between the interrogative and relative pronoun in Figure 6. (iii) In lexical semantics, ParCourE supports the investigation of how languages carve up the meaning space differently (cf. Figure 5), especially when it comes to the ≈1000 low-resource languages covered in PBC. Massively parallel texts are an ideal resource to investigate such variation (Haspelmath, 2003).

Grammatical differences between languages, like differences in word order, have a long history in research on worldwide linguistic variation (Greenberg, 1966; Dryer, 1992). However, being able to look at the usage of word order in specific contexts (and being able to directly compare exactly the same context across languages) is only possible by using parallel texts. For example, specific orders of more than two elements can be directly extracted from the parallel texts, like the order of demonstrative, numeral and noun "these two commandments" in Figure 7 (Cysouw, 2010).

For lack of space, we describe four more use cases only briefly: grammatical markers vs. morphology as devices to express grammatical features (Figure 8); differences in how languages use gram-

[5] https://www.cs.vassar.edu/CES/
[6] https://flask.palletsprojects.com
[7] https://gunicorn.org/
[8] https://www.elastic.co/
[9] https://getbootstrap.com/
[10] https://d3js.org/

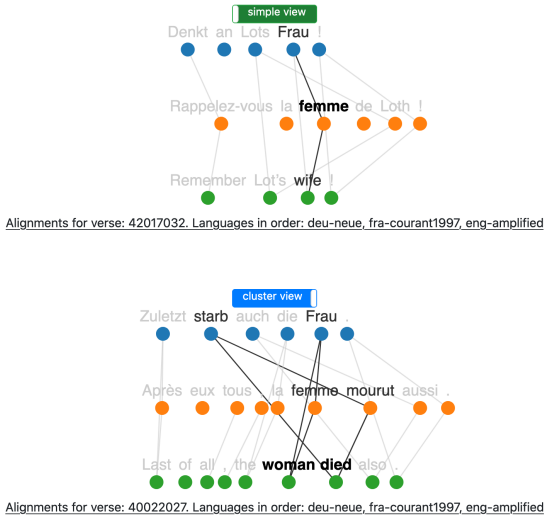

Figure 5: Use case 1, *lexical differentiation*. French "femme" has two different translations in English ("wife" and "woman") whereas German also conflates the two different meanings.

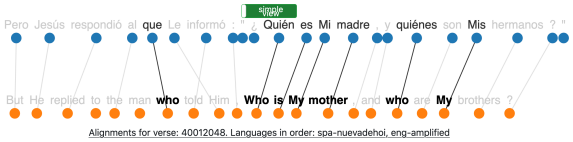

Figure 6: Use case 2, *grammatical differentiation*. English "who" has three different translations in this Spanish example: relative pronoun ("que"), and singular ("quién)" and plural ("quiénes") interrogative pronoun.

matical case (Figure 9, ablative/dative in Latin can correspond to five different cases in Croatian); and exploration of paraphrases (Figure 10). See the captions of the figures for more details.

7 Extension to Other Corpora

Our code is available on GitHub and can be generically applied: you can create a ParCourE instance for your own parallel corpus. Parallel corpora are essential for machine translation (MT); ParCourE's functionality is useful for analyzing the quality of a parallel corpus and the difficulty of the translation problem it poses. We give three examples **i)** Incorrect sentence alignments can be identified, e.g., cases in which a target sentence is matched with the merger of two sentences in the source: cf. Figure 11 where a short sentence in English is aligned with German and French sentences that also contain a second sentence that is missing in English. This functionality is particularly helpful for mined parallel corpora that tend to contain erroneous sentence pairs. **ii)** Suppose an MT system trained on the parallel corpus makes a lexical error in a particular context c by mistranslating source word w_s with target word w_t. The LEXICON view can be consulted for w_s and the user can then click on the erroneous target word w_t to get back to a MULTALIGN view of aligned sentence pairs containing w_s and w_t. She can then analyze why the MT system mismatched c with these contexts. Examples of the desired translation are easy to find and inspect to support the formation of hypotheses as to the source of the error. **iii)** For multi-source approaches to MT (Zoph and Knight, 2016; Firat et al., 2016; Libovický and Helcl, 2017; Crego et al., 2010), ParCourE supports the inspection of all input sentences together. The MT system output can also be loaded into ParCourE for a view that contains all input sentences and the output sentence. Since any of the input sentences can be responsible for an error in multi-source MT, this facilitates analysis and hypothesis formation as to what caused a specific error.

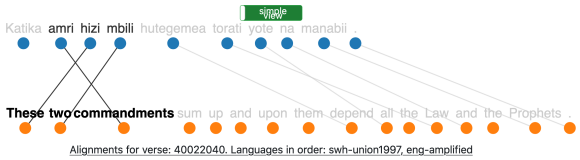

Figure 7: Use case 3, *word order variation*. The English order is demonstrative, numeral, noun whereas Swahili has noun, demonstrative, numeral.

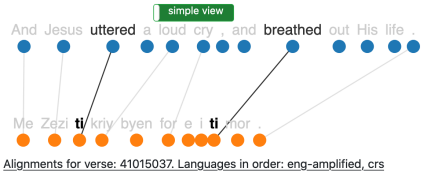

Figure 8: Use case 4, *grammatical markers*. In contrast to English, Seychelles Creole does not inflect verbs for tense and uses the past tense marker "ti" instead.

7.1 Computing Infrastructure and Runtime

We did all computations on a machine with 48 cores of Intel(R) Xeon(R) CPU E7-8857 v2 with 1TB memory. In this experiment only one core was used.

We created a corpus of 5 translations in 4 languages, with around 31k parallel sentences (overally 155k sentences) and applied the ParCourE pipeline to it. Runtimes for different parts of the

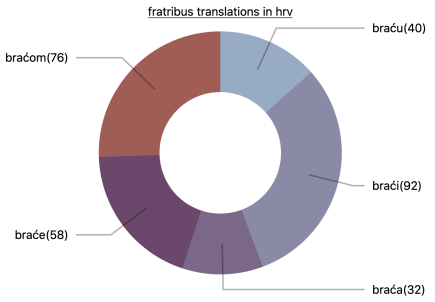

Figure 9: Use case 5, *morphology*. The Latin ending "ibus" in "fratribus" (dative/ablativ plural) corresponds to five different cases in Croatian: accusative, locative/dative, nominative, genitive, instrumental (clockwise starting from "braću").

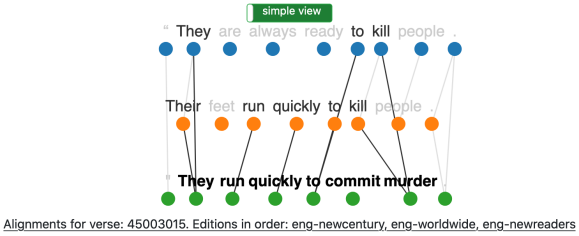

Figure 10: Use case 6, *paraphrases*. PBC is a rich source of paraphrases since high-resource languages have several translations (32 for English). ParCourE can be used to explore these paraphrases. Here, the paraphrases "kill" and "murder" are correctly aligned, "always ready" and "run quickly" are not.

pipeline are reported in Table 2. The installation of the package is straightforward and as shown in the table, it takes around 12 minutes to initiate ParCourE on a small corpus with 4 languages.

Method	Runtime
Conversion from CES to ParCourE format	153
Indexing with Elasticsearch	14
Alignment generation with Eflomal	537
Stats calculation	22
Overall	726

Table 2: Runtime in seconds for each part of the pipeline to initiate a ParCourE instance on a corpus with 4 languages and 31K parallel sentences.

8 Conclusion

Progress in multilingual NLP is an important goal of NLP and requires researching typological properties of languages. Examples include assessing language similarity for effective transfer learning, injecting inductive biases into machine learning models and creating resources such as dictionaries and inflection tables. To serve such use cases, we have created ParCourE, an online tool for browsing a word-aligned parallel corpus of 1334 languages, and given evidence that it is useful for typological research. ParCourE can be set up for any other parallel corpus, e.g., for quality control and improvement of automatically mined parallel corpora.

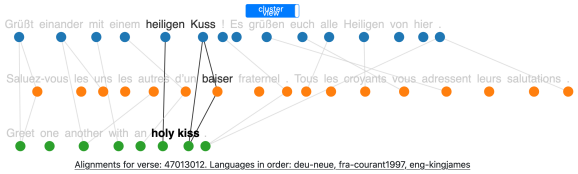

Figure 11: Use case 7, *quality analysis*. ParCourE makes it easy to analyze the quality of the parallel corpus. For this sentence, part of a Bible verse present in German and French is missing in English. Note that the alignment of *holy, heiligen* to French *fraternel* is not discovered.

Acknowledgments

This work was supported by the European Research Council (ERC, Grant No. 740516) and the German Federal Ministry of Education and Research (BMBF, Grant No. 01IS18036A). The third author was supported by the Bavarian research institute for digital transformation (bidt) through their fellowship program. We thank the anonymous reviewers for their constructive comments.

9 Ethical Considerations

Word alignments and lexicon induction as tasks themselves may not have ethical implications. However, working on a biblical corpus requires special consideration of the following issues.

i) The Bible is the central religious text of Christianity and the Hebrew Bible that of Judaism. It contains strong opinions and world views (e.g., on divorce and homosexuality) that are not generally shared. We would like to emphasize that we treat the PBC simply as a multiparallel corpus, and the corpus does not necessarily reflect the opinions of the authors nor of the institutions funding the authors. **ii)** In a similar vein, while the PBC has great language coverage and allows for typological analysis, we need to be aware that languages might not be accurately and completely reflected in the PBC. The language used in the PBC might be outdated and is restricted to a relatively small subset of topics and thus cannot be considered a balanced and complete view of the language. **iii)** We also need to

be aware of selection bias. The PBC only covers a subset of the world's languages. The selection criteria are unknown and may be based on historical and cultural biases that we are not able to assess.

References

Alan Akbik and Roland Vollgraf. 2017. The projector: An interactive annotation projection visualization tool. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 43–48, Copenhagen, Denmark. Association for Computational Linguistics.

Ehsaneddin Asgari and Hinrich Schütze. 2017. Past, present, future: A computational investigation of the typology of tense in 1000 languages. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 113–124, Copenhagen, Denmark. Association for Computational Linguistics.

Mikko Aulamo, Sami Virpioja, and Jörg Tiedemann. 2020. OpusFilter: A configurable parallel corpus filtering toolbox. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 150–156, Online. Association for Computational Linguistics.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2).

Josep Maria Crego, Aurélien Max, and François Yvon. 2010. Local lexical adaptation in machine translation through triangulation: SMT helping SMT. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 232–240, Beijing, China. Coling 2010 Organizing Committee.

Michael Cysouw. 2010. Dealing with diversity: towards an explanation of NP word order frequencies. *Linguistic Typology*, 14(2):253–287.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Zi-Yi Dou and Graham Neubig. 2021. Word alignment by fine-tuning embeddings on parallel corpora. *CoRR*, abs/2101.08231.

Matthew S. Dryer. 1992. The Greenbergian word order correlations. *Language*, 68(1):80–138.

Matthew S. Dryer and Martin Haspelmath, editors. 2013. *WALS Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

Philipp Dufter, Mengjie Zhao, Martin Schmitt, Alexander Fraser, and Hinrich Schütze. 2018. Embedding learning through multilingual concept induction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1520–1530, Melbourne, Australia. Association for Computational Linguistics.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia. Association for Computational Linguistics.

David M. Eberhard, F. Simons Gary, and D. Fennig (eds.) Charles. 2020. *Ethnologue: Languages of the World*, 23rd edition. SIL International.

Orhan Firat, Baskaran Sankaran, Yaser Al-onaizan, Fatos T. Yarman Vural, and Kyunghyun Cho. 2016. Zero-resource translation with multi-lingual neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 268–277, Austin, Texas. Association for Computational Linguistics.

Sarthak Garg, Stephan Peitz, Udhyakumar Nallasamy, and Matthias Paulik. 2019. Jointly learning to align and translate with transformer models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4453–4462, Hong Kong, China. Association for Computational Linguistics.

Timur Gilmanov, Olga Scrivner, and Sandra Kübler. 2014. SWIFT aligner, a multifunctional tool for parallel corpora: Visualization, word alignment, and (morpho)-syntactic cross-language transfer. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2913–2919, Reykjavik, Iceland. European Language Resources Association (ELRA).

Johannes Graën, Dominique Sandoz, and Martin Volk. 2017. Multilingwis2 ėxtendash explore your parallel corpus. In *Proceedings of the 21st Nordic Conference on Computational Linguistics, NODALIDA 2017, Gothenburg, Sweden, May 22-24, 2017*, volume 131 of *Linköping Electronic Conference Proceedings*, pages 247–250. Linköping University Electronic Press / Association for Computational Linguistics.

Joseph H. Greenberg. 1966. *Language Universals: with special reference to feature hierarchies*. Janua Linguarum, Series Minor. Mouton, The Hague.

Harald Hammarstrm, Robert Forkel, Martin Haspelmath, and Sebastian Bank. 2020. Glottolog 4.3. Max Planck Institute for the Science of Human History.

Martin Haspelmath. 2003. The geometry of grammatical meaning: Semantic maps and cross-linguistic comparison. In Michael Tomasello, editor, *The New Psychology of Language: Cognitive and Functional Approaches to Language Structure (Volume 2)*, pages 211–242. Erlbaum, Mahwah, NJ.

Masoud Jalili Sabet, Philipp Dufter, François Yvon, and Hinrich Schütze. 2020. SimAlign: High quality word alignments without parallel training data using static and contextualized embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1627–1643, Online. Association for Computational Linguistics.

Murathan Kurfal and Robert Östling. 2018. Word embeddings for 1250 languages through multi-source projection. In *Seventh Swedish Language Technology Conference*.

William D. Lewis and Fei Xia. 2008. Automatically identifying computationally relevant typological features. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II*.

Jindrich Libovický and Jindrich Helcl. 2017. Attention strategies for multi-source sequence-to-sequence learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 2: Short Papers*, pages 196–202. Association for Computational Linguistics.

Thomas Mayer and Michael Cysouw. 2012. Language comparison through sparse multilingual word alignment. In *Proceedings of the EACL 2012 Joint Workshop of LINGVIS & UNCLH*, pages 54–62, Avignon, France. Association for Computational Linguistics.

Thomas Mayer and Michael Cysouw. 2014. Creating a massively parallel Bible corpus. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3158–3163, Reykjavik, Iceland. European Language Resources Association (ELRA).

Aaron Mueller, Garrett Nicolai, Arya D. McCarthy, Dylan Lewis, Winston Wu, and David Yarowsky. 2020. An analysis of massively multilingual neural machine translation for low-resource languages. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3710–3718, Marseille, France. European Language Resources Association.

Roberto Navigli and Simone Paolo Ponzetto. 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artif. Intell.*, 193:217–250.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1).

Robert Östling. 2014. Bayesian word alignment for massively parallel texts. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2014, April 26-30, 2014, Gothenburg, Sweden*, pages 123–127. The Association for Computer Linguistics.

Robert Östling. 2015. Word order typology through multilingual word alignment. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 205–211, Beijing, China. Association for Computational Linguistics.

Robert Östling and Jörg Tiedemann. 2016a. Efficient word alignment with Markov Chain Monte Carlo. *Prague Bulletin of Mathematical Linguistics*, 106:125–146.

Robert Östling and Jörg Tiedemann. 2016b. Efficient word alignment with Markov Chain Monte Carlo. *The Prague Bulletin of Mathematical Linguistics*, 106(1).

Sebastin Santy, Sandipan Dandapat, Monojit Choudhury, and Kalika Bali. 2019. INMT: Interactive neural machine translation prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 103–108, Hong Kong, China. Association for Computational Linguistics.

James Strong. 2009[1890]. *Strong's exhaustive concordance of the Bible*. Hendrickson Publishers.

Jörg Tiedemann. 2018. Emerging language spaces learned from massively multilingual corpora. In *Proceedings of the Digital Humanities in the Nordic Countries 3rd Conference, DHN 2018, Helsinki, Finland, March 7-9, 2018*, volume 2084 of *CEUR Workshop Proceedings*, pages 188–197. CEUR-WS.org.

Winston Wu, Nidhi Vyas, and David Yarowsky. 2018. Creating a translation matrix of the Bible's names across 591 languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Patrick Xia and David Yarowsky. 2017. Deriving consensus for multi-parallel corpora: an English Bible study. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 448–453, Taipei, Taiwan. Asian Federation of Natural Language Processing.

David Yarowsky, Grace Ngai, and Richard Wicentowski. 2001. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of the First International Conference on Human Language Technology Research*.

Thomas Zenkel, Joern Wuebker, and John DeNero. 2020. End-to-end neural word alignment outperforms GIZA++. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1605–1617, Online. Association for Computational Linguistics.

Barret Zoph and Kevin Knight. 2016. Multi-source neural translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 30–34, San Diego, California. Association for Computational Linguistics.

Chapter 4

Graph Algorithms for Multiparallel Word Alignment

Graph Algorithms for Multiparallel Word Alignment

Ayyoob Imani[*1], **Masoud Jalili Sabet**[*1], **Lütfi Kerem Şenel**[1],
Philipp Dufter[1], **François Yvon**[2], **Hinrich Schütze**[1]
[1]Center for Information and Language Processing (CIS), LMU Munich, Germany
[2]Université Paris-Saclay, CNRS, LISN, France
{ayyoob, masoud, lksenel, philipp}@cis.lmu.de,
francois.yvon@limsi.fr

Abstract

With the advent of end-to-end deep learning approaches in machine translation, interest in word alignments initially decreased; however, they have again become a focus of research more recently. Alignments are useful for typological research, transferring formatting like markup to translated texts, and can be used in the decoding of machine translation systems. At the same time, massively multilingual processing is becoming an important NLP scenario, and pretrained language and machine translation models that are truly multilingual are proposed. However, most alignment algorithms rely on bitexts only and do not leverage the fact that many parallel corpora are multiparallel. In this work, we exploit the multiparallelity of corpora by representing an initial set of bilingual alignments as a graph and then predicting additional edges in the graph. We present two graph algorithms for edge prediction: one inspired by recommender systems and one based on network link prediction. Our experimental results show absolute improvements in F_1 of up to 28% over the baseline bilingual word aligner in different datasets.

1 Introduction

Word alignment is a challenging NLP task that plays an essential role in statistical machine translation and is useful for neural machine translation (Alkhouli and Ney, 2017; Alkhouli et al., 2016; Koehn et al., 2003). Other applications of word alignments include bilingual lexicon induction, annotation projection, and typological analysis (Shi et al., 2021; Rasooli et al., 2018; Müller, 2017; Lewis and Xia, 2008). With the advent of deep learning, interest in word alignment initially decreased. However, recently a new wave of publications has again drawn attention to the task (Jalili Sabet et al., 2020; Dou and Neubig, 2021; Marchisio et al., 2021; Wu and Dredze, 2020).

[*] Equal contribution.

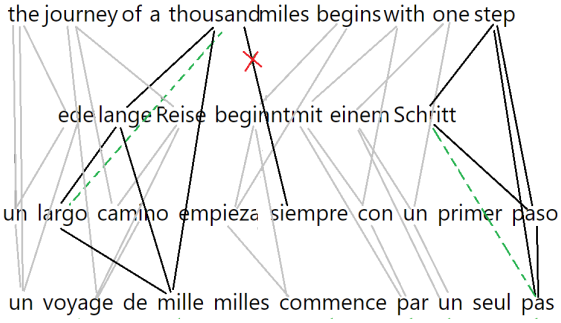

Figure 1: Bilingual alignments of a verse in English, German, Spanish, and French. Two of the alignment edges not found by the bilingual method are German "Schritt" to French "pas" and Spanish "largo" to English "thousand miles". By looking at the structure of the entire graph, one can infer the correctness of these two edges.

In this paper we propose *MPWA* (MultiParallel Word Alignment), a framework that employs graph algorithms to exploit the information latent in a multiparallel corpus to achieve better word alignments than aligning pairs of languages in isolation. Starting from translations of a sentence in multiple languages in a multiparallel corpus, MPWA generates bilingual word alignments for all language pairs using any available bilingual word aligner. MPWA then improves the quality of word alignments for a target language pair by inspecting how they are aligned to other languages. *The central idea is to exploit the graph structure of an initial multiparallel word alignment to improve the alignment for a target language pair.* To this end, MPWA casts the multiparallel word alignment task as a link (or edge) prediction problem. We explore standard algorithms for this purpose: Adamic-Adar and matrix factorization. While these two graph-based algorithms are quite different and are used in different applications, we will show that MPWA effectively leverages them for high-performing word alignment.

Link prediction methods are used to predict whether there should be a link between two nodes in a graph. They have various applications like movie recommendations, knowledge graph completion, and metabolic network reconstruction (Zhang and Chen, 2018). We use the Adamic-Adar index (Adamic and Adar, 2003); it is a second-order link prediction algorithm, i.e., it exploits the information of neighbors that are up to two hops aways from the starting target nodes (Zhou et al., 2009). We use a second-order algorithm since a set of aligned words in multiple languages (representing a concept) tends to establish a clique (Dufter et al., 2018). This means that exploring the influence of nodes at a distance of two in the graph provides informative signals while at the same time keeping runtime complexity low.

Matrix factorization is a collaborative filtering algorithm that is most prominently used in recommender systems where it provides users with product recommendations based on their interactions with other products. This method is especially useful if the matrix is sparse (Koren et al., 2009). This is true for our application: Given two translations of a sentence with lengths M and N, among all possible alignment links ($M \times N$), only a few ($O(M + N)$) are correct. This is partly due to fertility: words in the source language generally have only a few possible matches in the target language (Zhao and Gildea, 2010).

A multiparallel corpus provides parallel sentences in more than two languages. This type of corpus facilitates the study of multiple languages together, which is especially important for research on low resource languages. As far as we know, out of all available multiparallel corpora, the Parallel Bible Corpus (Mayer and Cysouw, 2014) (PBC) provides the highest language coverage, supporting 1334 different languages, many of which belong to categories 0 and 1 (Joshi et al., 2020) – that is, they are languages for which no language technologies are available and that are severely underresourced.

MPWA has especially strong word alignment improvements for distant language pairs for which existing bilingual word aligners perform poorly. Much work that addresses low resource languages relies on the availabiliy of monolingual corpora. Complementarily, MPWA assumes the existence of a very small (a few 10,000s of sentences in our case) parallel corpus and then takes advantage of information from the other languages in the parallel corpus. This is an alternative approach that is especially important for low resource languages for which monolingual data often are not available.

The PBC corpus does not contain a word alignment gold standard. To conduct the comparative evaluation of our new method, we port three existing word alignment gold standards of Bible translations to PBC, for the language pairs English-French, Finnish-Hebrew and Finnish-Greek. We also create artificial multiparallel datasets for four widely used word alignment datasets using machine translation. We evaluate our method with all seven datasets. Results demonstrate substantial improvements in all scenarios.

Our main contributions are:

1. We propose two graph-based algorithms for link prediction (i.e., the prediction of word alignment edges in the alignment graph), one based on second-order link prediction and one based on recommender systems for improving word alignment in a multiparallel corpus and show that they perform better than established baselines.

2. We port and publish three word alignment gold standards for the Parallel Bible Corpus.

3. We show that our method is also applicable, using machine translation, to scenarios where multiparallel data is not available.

4. We publish our code[1] and data.

2 Related Work

Bilingual Word Aligners take different approaches. Some are based on statistical analysis, like IBM models (Brown et al., 1993), Giza++ (Och and Ney, 2003a), fast-align (Dyer et al., 2013) and Eflomal (Östling and Tiedemann, 2016). Another more recent group, including SimAlign (Jalili Sabet et al., 2020) and Awesome-align (Dou and Neubig, 2021), utilizes neural language models. The last group is based on neural machine translation (Garg et al., 2019; Zenkel et al., 2020). While neural models outperform statistical models, for cases where only a small parallel dataset is available, statistical models are still superior. In this paper we use PBC, a corpus with 1334 languages, of which only about two hundred are supported by multilingual language models like Bert and XLM-R (Devlin et al., 2019; Conneau et al., 2020). MPWA can

[1] https://github.com/cisnlp/graph-align

leverage multiparallelism on top of any bilingual word aligner; in this paper, we use Eflomal and SimAlign.

Multiparallel corpus alignment. Most work on word alignment has focused on bilingual corpora. To the best of our knowledge, only one method specifically designed for multiparallel corpora was previously proposed: (Östling, 2014).[2] However this method is outperformed by a "biparallel" method by the same author, Eflomal (Östling and Tiedemann, 2016). We compare with Eflomal in our experiments.

Cohn and Lapata (2007) make use of multiparallel corpora to obtain more reliable translations from small datasets. Kumar et al. (2007) show that multiparallel corpora can be of benefit to reach better performance in phrase-based statistical machine translation (SMT). Filali and Bilmes (2005) present a multilingual SMT-based word alignment model, extending IBM models, based on HMM models and a two step alignment procedure. Since the goal of this research is to tackle word alignment directly without considering machine translation, these works are not considered here.

In another line of research, Lardilleux and Lepage (2008a) introduce a corpus splitting method to come up with a perfect alignment of multiwords. Lardilleux and Lepage (2008b), and Lardilleux and Lepage (2009) suggest to rely only on low frequency terms for a similar purpose: sub-sentential alignment. These methods solve a somewhat different problem than what is addressed by us. Other usages of multiparallel corpora are language comparison (Mayer and Cysouw, 2012), typology studies (Östling, 2015; Asgari and Schütze, 2017; Imani-Googhari et al., 2021) and SMT (Nakov and Ng, 2012; Bertoldi et al., 2008; Dyer et al., 2013)

Matrix factorization and link prediction. Matrix factorization is a technique that factors, in the most typical case, a matrix into two lower-ranked matrices in which the latent factors of the original matrix are represented. Matrix factorization approaches have been widely used in document clustering (Xu et al., 2003; Shahnaz et al., 2006), topic modeling (Kuang et al., 2015; Choo et al., 2013) information retrieval (Zamani et al., 2016; Deerwester et al., 1990) and NLP tasks like word sense disambiguation (Schütze, 1998). In 2009, Netflix's recommender system competition revealed that this technique effectively works for collaborative filtering (Koren et al., 2009). Since then it has been a state of the art method in recommender systems.

Link prediction algorithms are widely used in different areas of science since many social, biological, and information systems can be described as networks with nodes and connecting links (Zhou et al., 2009). Link prediction algorithms compute the likelihood of links based on different heuristics. One can categorize available methods based on the maximum number of hops they consider in their computations for each node (Zhang and Chen, 2018). First order algorithms, such as common neighbors (CN), only consider one hop neighborhoods, e.g., (Barabási and Albert, 1999). Second order methods consider two hops, e.g., (Zhou et al., 2009). Finally, higher order methods take the whole network into account for making predictions (Brin and Page, 1998; Jeh and Widom, 2002; Rothe and Schütze, 2014). In this paper, we use a two-hop method since it offers a good tradeoff between effectiveness and efficiency.

3 Methods

3.1 The MPWA framework

While a bilingual aligner considers each language pair separately, MPWA utilizes the synergy between all language pairs to improve word alignment performance. In Figure 1, Eflomal alignments of a sentence from PBC in four different languages are depicted. Although Eflomal has failed to find the link between German "Schritt" and French "pas", we can easily find this relation by observing that the four nodes "step", "Schritt", "paso", and "pas" are fully connected, except for the edge from "Schritt" to "pas". In this case, the inference amounts to a completion of a clique. However, most cases are not that simple. In the figure, English "thousand miles" is mistakenly aligned to Spanish "siempre" although its alignments to German "lange" and French "mille" are correct. We would like to infer that "thousand miles" should be aligned to "largo", but in this case creating a fully connected subgraph, i.e., a clique (which would include "siempre"), would add many incorrect edges. Given the complexity and error-proneness of initial bilingual alignments, inferring an alignment between two languages from a multiparallel alignment in general is a complex problem.

Starting from a multiparallel corpus, we first generate bilingual alignments for all language pairs.

[2] https://github.com/robertostling/eflomal

MPWA then employs a prediction algorithm to find and add new alignment links. In this paper, we focus on two prediction algorithms: non-negative matrix factorization and Adamic-Adar link prediction.

3.2 Non-negative matrix factorization

Non-negative matrix factorization (NMF) has been used in many different applications. After discovery of its effectiveness for collaborative recommendation (Koren et al., 2009), it was widely accepted as a standard method for recommender systems.

In a standard recommender system with m users and n items, ratings (a number from 1 to 5) from each user for the items they have seen so far are known. The aim is to predict the ratings the user would give to unseen items and, based on these predictions, recommend new items to the user. As described by (Luo et al., 2014), let $W = [w_{u,i}] \in \mathbb{R}^{m \times n}$ be the matrix of ratings. For NMF to work it is essential that the matrix be sparse, thus if a user's rating for an item is unknown, the corresponding cell is zeroed. The matrix W is then decomposed into two low-rank non-negative matrices, $T = [t_{u,k}] \in \mathbb{R}^{m \times r}$ and $V = [v_{k,i}] \in \mathbb{R}^{r \times n}$ such that $TV \approx W$ and $r \ll \min(m, n)$. r is a hyperparameter. By multiplication of these two matrices we end up with a reduced matrix $W' = TV$ in which each zeroed cell $w_{u,i}$ from matrix W is replaced with a value $w'_{u,i}$ that represents a prediction for the rating that user u would give to item i. NMF solves the following optimization program:

$$\begin{aligned} &\underset{T,V}{\operatorname{argmin}} \left(\|W - TV\|^2 \right) \\ &\text{subject to} \quad T, V \geqslant 0 \end{aligned}$$

This optimization problem can be solved by gradient descent using the following updates:

$$t_{u,k} \leftarrow t_{u,k} + \eta_{u,k}((WV^T)_{u,k} - (TVV^T)_{u,k})$$
$$v_{k,i} \leftarrow v_{k,i} + \eta_{k,i}((T^TW)_{k,i} - (T^TTV)_{k,i})$$

In this equation, η is the learning rate. To guarantee non-negativity, it is defined as:

$$\eta_{u,k} = \frac{t_{u,k}}{(TVV^T)_{u,k}}, \quad \eta_{k,i} = \frac{v_{k,i}}{(T^TTV)_{k,i}}$$

Note that the objective function only takes account of non-zero cells. Luo et al. (2014) propose an approach that takes advantage of the sparseness of the matrix for faster computation. In addition, Tikhonov regularization is integrated to improve precision, recall, and convergence rate.

We use the implementaion of NMF provided by the Surprise[3] library, with default hyperparameters ($r = 15$, n_epochs $= 50$).

	I	can	see	ich	kann	es	sehen	je	vois
I	5		1	5		1		5	1
can		5	1		5		1		
see	1	1	5		1		5	1	5
ich	5	1		5			1	5	1
kann	1	5		1	5				
es						5	1		
sehen	1		5		1		5	1	
je	5	1		5		1		5	1
vois	1		5	1			5	1	5

Figure 2: An example of how the original matrix is filled for a sentence in three languages. Zero entries are left blank for readability.

3.2.1 NMF in MPWA framework

We create a separate matrix W for each sentence in the multiparallel corpus. Tokens in the sentence play the role of both users and items, i.e., we consider each token both as a row and as a column. Figure 2 shows an example of a sentence in a toy English-German-French multiparallel corpus. If two tokens are aligned using the bilingual aligner, we fill the corresponding cell with the highest rating (5). To give a few negative examples to the algorithm, if a token x from language L_1 is aligned to token y in language L_2, we pick another random token z from L_2 and fill the corresponding cell of x to z with the lowest rating (1). We zero out all other cells. Next we apply the matrix factorization algorithm to this matrix and then compute the reduced matrix W' from the factors. Now we grab the predicted alignment scores between source and target languages from W'. To extract new alignment edges we apply the Argmax algorithm (Jalili Sabet et al., 2020). Argmax creates an alignment edge between word w_i from language L_1 and word w_j from language L_2 if among all words from L_2, w_i has the highest alignment score with w_j, and among all words from L_1, w_j has the highest alignment score with w_i.

3.3 Link Prediction

A multiparallel sentence annotated with bilingual word alignments can be considered to be a graph with words from all translations as nodes and the

[3] http://surpriselib.com/

word alignments as edges. Link prediction algorithms such as Common Neighbors (CN) and Adamic-Adar (AdAd) estimate the likelihood of having an edge between two nodes x and y in the graph based on the similarity of their neighborhoods. The CN index weights all common neighbors equally. In contrast, AdAd gives higher weight to neighbors with low degrees because sharing a neighbor that in turn has few neighbors is more significant. Therefore, we use the AdAd index. It is defined as:

$$\text{AdAd}_{x,y} = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{\log |\Gamma(z)|} \quad (1)$$

where $\Gamma(x)$ is the neighborhood of x.

If we use a word aligner that produces a score for each alignment edge, we can use Weighted Adamic-Adar (Lü and Zhou, 2010):

$$\text{WAdAd}_{x,y} = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{w(x,z) + w(z,y)}{\log(1 + S(z))} \quad (2)$$

where $w(x,z)$ is the similarity score of x and z generated by the aligner and $S(x) = \sum_{z \in \Gamma(x)} w(x,z)$. For embedding-based aligners we use embedding similarity as the score $w(x,z)$. If an aligner does not provide scores, we set all weights to 1.0.

Given a scored word alignment, we create a multilingual word alignment matrix W for each sentence as shown in Figure 2. Each cell contains 0 or 1 for Adamic-Adar or the alignment score for Weighted Adamic-Adar. We again apply Argmax to extract new alignment edges and then add them to the original alignment.

4 Experimental setup

4.1 PBC

The PBC corpus (Mayer and Cysouw, 2014) contains 1758 editions of the Bible in 1334 languages. The editions are aligned at the verse level and tokenized. A verse can contain more than one sentence, but we treat it as one unit in the parallel corpus since a true sentence level alignment is not available. There are some errors in tokenization (e.g., for Tibetan, Khmer and Chinese), but the overall quality of the corpus is good. For the majority of languages one edition is provided, while a few languages (in particular, English, French and German) contain up to dozens of editions. The verse coverage also differs from language to language. Some languages have translations of both New Testament and Hebrew Bible while others contain only one. Table 2 gives corpus statistics.

4.2 Word alignment datasets

PBC does not provide gold word alignments. To evaluate MPWA, we port two word alignment gold datasets of the Bible to PBC: Blinker (Melamed, 1998) and the recently published HELFI (Yli-Jyrä et al., 2020). We further experiment with bilingual datasets, using Machine Translation (MT) to create multiparallel corpora. Table 1 gives dataset statistics.

The HELFI dataset consists of the Greek New Testament, the Hebrew Bible and translations of both into Finnish. In addition, morpheme alignments are provided for Finnish-Greek and Finnish-Hebrew. We reformatted this dataset to the format used by PBC. In more detail, we added three new editions for the three languages to PBC. We identified the PBC verse identifier for each verse of HELFI to ensure proper verse alignment of these three new editions. The Finnish-Hebrew dataset has 22,291 verses and the Finnish-Greek dataset 7,909. We split these datasets 80/10/10 into train, validation and test.

The Blinker Bible dataset provides word level alignments of 250 Bible verses between English and French. The French side of this dataset matches with the edition Louis Segond 1910 in PBC. However, the tokenizations (Blinker vs PBC) are different. We therefore create a mapping of the tokens using character n-gram matching. For English, we created and added a new edition to PBC.

MT datasets. To more broadly evaluate MPWA, we also create multiparallel datasets for four non-Bible word alignment gold standards; these are listed in Table 1 as "Non-Bible" corpora. For these gold standards, we translate from English to all languages available in Google Translate, using their API.[4] For the added languages, we create alignments for the gold standard sentences using SimAlign.

4.3 Initial word alignments

We compare with two state of the art models, one statistical, one neural. Eflomal (Östling and Tiedemann, 2016) is a Bayesian statistical word aligner using Markov Chain Monte Carlo inference. SimAlign (Jalili Sabet et al., 2020) obtains word align-

[4] https://cloud.google.com/translate/docs/basic/translating-text

	Language Pair	Name	# Sentences (train/valid./test)
Bible	FIN-HEB	HELFI (Yli-Jyrä et al., 2020)	22291 (17832/2229/2230)
	FIN-GRC	HELFI (Yli-Jyrä et al., 2020)	7909 (6327/791/791)
	ENG-FRA	BLINKER (Melamed, 1998)	250
Non-Bible	ENG-DEU	EuroParl-based[a]	508
	ENG-FAS	(Tavakoli and Faili, 2014)	400
	ENG-HIN	WPT2005[b]	90
	ENG-RON	WPT2005[b]	203

[a] www-i6.informatik.rwth-aachen.de/goldAlignment/
[b] http://web.eecs.umich.edu/~mihalcea/wpt05/

Table 1: Overview of datasets. We use ISO 639-3 language codes. # Sentences: the number of available verses (i.e., sentences). FIN-HEB and FIN-GRC datasets split into train, validation and test.

# editions	1758
# languages	1334
# verses	20,470,892
# verses / # editions	11,520
# tokens / # verses	28.6

Table 2: PBC corpus statistics

ments from multilingual pretrained language models with no need for parallel data. For the symmetrization of Eflomal, we use grow-diag-final-and (GDFA) and intersection, and for SimAlign we use Argmax and Itermax. Intersection and Argmax generate accurate alignments while GDFA and Itermax are less accurate but have better coverage (Jalili Sabet et al., 2020).

We evaluate on a *target language pair* parallel sentence as follows: First, we create the matrix (Figure 2) for this sentence for all languages in the multiparallel corpus. Then we run link prediction on the matrix – this accumulates evidence from a set of languages in the multiparallel corpus. Finally, we take the predictions for the target language pair and add them to the original (bilingual) alignment.

NMF works best if it starts with high-accuracy (i.e., non-noisy) bilingual alignments – errors can result in incorrectly predicted alignment edges. We therefore use SimAlign Argmax and Eflomal Intersection, two word alignment methods with high precision, to create the initial alignments that are then fed into NMF. We then add the predictions to any desired original alignments; e.g., NMF (GDFA) uses Eflomal Intersection as the initial alignments and adds the predictions to Eflomal GDFA. See the Appendix for more details.

SimAlign offers high quality word alignments for well-represented languages from pretrained language models; however, our experiments show that its performance is far behind Eflomal for less well resourced languages like Biblical Hebrew and Koine Greek. Also, Eflomal is a better match for MPWA because it can provide word alignments for all languages available in a multiparallel corpus. In contrast, SimAlign is limited to languages supported by pretrained multilingual embeddings.

To feed Eflomal with enough training data for a target language pair, we use all available data from different translations of the language pair. For example if one language has two translations and the other one has three translations, Eflomal's training data will contain six aligned translation pairs for these two languages.

We use the standard evaluation measures for word alignment: precision, recall, F_1 and Alignment Error Rate (AER) (Och and Ney, 2003b; Östling and Tiedemann, 2016; Jalili Sabet et al., 2020).

5 Results

5.1 Multiparallel corpus results

We perform the first set of experiments on the Blinker Bible and the HELFI gold standards in the PBC. The baseline results are calculated on the original language pairs. MPWA can be applied to both Eflomal and SimAlign alignments. Since the default version of SimAlign can only generate alignments for the 84 languages that multilingual BERT supports,[5] for a better comparison, we use the same set of languages in the alignment graph for both SimAlign and Eflomal.

Table 3 shows the results for our methods applied on SimAlign and Eflomal baselines.[6] AdAd, NMF and WAdAd substantially improve the performance for all language pairs. SimAlign generates high-quality alignments for the English-French dataset, but cannot properly align underresourced languages like Biblical Hebrew and Koine Greek.

[5] https://github.com/google-research/bert/blob/master/multilingual.md

[6] We only consider SimAlign IterMax, not SimAlign ArgMax, because IterMax performed better throughout.

	Method	FIN-HEB Prec.	Rec.	F_1	AER	FIN-GRC Prec.	Rec.	F_1	AER	ENG-FRA Prec.	Rec.	F_1	AER
Baseline	Eflomal (intersection)	**0.818**	0.269	0.405	0.595	**0.897**	0.506	0.647	0.353	**0.971**	0.521	0.678	0.261
	Eflomal (GDFA)	0.508	0.448	0.476	0.524	0.733	0.671	0.701	0.300	0.856	0.710	0.776	0.221
	SimAlign	0.190	0.113	0.142	0.858	0.366	0.265	0.307	0.693	0.886	0.692	0.777	0.221
Init SimAlign	AdAd	0.199	0.127	0.155	0.845	0.402	0.289	0.336	0.664	0.878	0.731	0.798	0.200
	WAdAd	0.186	0.165	0.175	0.825	0.353	0.350	0.351	0.649	0.856	0.752	0.801	0.197
	NMF	0.122	0.100	0.110	0.890	0.396	0.337	0.364	0.636	0.835	0.700	0.762	0.236
Init Eflomal	WAdAd (intersection)	0.781	0.612	**0.686**	**0.314**	0.849	0.696	**0.765**	**0.235**	0.938	0.689	0.794	0.203
	NMF (intersection)	0.78	0.576	0.663	0.337	0.864	0.669	0.754	0.248	0.948	0.624	0.753	0.245
	WAdAd (GDFA)	0.546	**0.693**	0.611	0.389	0.707	**0.783**	0.743	0.257	0.831	**0.796**	**0.813**	**0.186**
	NMF (GDFA)	0.548	0.646	0.593	0.407	0.72	0.759	0.739	0.261	0.844	0.767	0.804	0.195

Table 3: Comparison of results from different methods on PBC. The best results in each column are in bold. The three methods exploiting multiparallelism (AdAd, WAdAd, NMF) generally outperform the baselines on F_1 and AER.

In such cases, MPWA uses the accumulated information from all other language pairs in the graph to improve the performance. When starting with the SimAlign alignment ("Init SimAlign"), both methods improve the result for both FIN-HEB and FIN-GRC.

Eflomal generates better alignments for FIN-HEB and FIN-GRC. This means that Eflomal also generates better alignments between FIN, HEB and GRC on the one hand and the other languages in the graph on the other hand and therefore can provide a better signal for MPWA. The improvements of our models applied on Eflomal are higher than the ones applied on SimAlign for these language pairs.

When changing the initial alignments from Eflomal (intersection) to Eflomal (GDFA), we see different behaviors: GDFA improves the results for Blinker while it does not help for HELFI. We believe this is caused by the different ways the two datasets were annotated. In Blinker, many phrases are "exhaustively" aligned: if a phrase DE in English is aligned with FG in French then all four alignment edges (D-F, D-G, E-F, E-G) are given as gold edges.[7]

So Blinker contains a lot of many-to-many links. In contrast, most alignments are one-to-one in HELFI. This partially explains why intersection as initial alignment works much better for HELFI than GDFA and vice versa for Blinker.

In summary, compared to the baselines, we see very large improvements through exploiting multiparallelism for one type of alignment methodology (HELFI, F_1 improved by up to 20% for FIN-HEB) and improvements of up to 3.5% for the other (ENG-FRA).

[7]For example, the alignment of the phrases "trembled violently" and "fut saisi d'und grande, d'une violente émotion" consists of $2 \cdot 8$ gold edges.

5.2 MT dataset results

We perform the second set of experiments on gold standard alignments for language pairs that are not part of a multiparallel corpus such as PBC. To this end, we create artificial multiparallel corpora by translating the English side to all languages available in the Google Translate API. The main goal is to give broader evidence for the effectiveness of our method, beyond the specialized domain of the Bible.

Eflomal's alignments generally have good quality. However, they get worse when less parallel data is available (Jalili Sabet et al., 2020). Since the size of the multiparallel corpus created by machine translation is rather small, we use SimAlign for generating initial alignments. SimAlign has been shown to have good performance even for very small parallel corpora; in fact, it does not need any parallel data at all.

Table 4 shows the results of the experiments. Both NMF and WAdAd, improve the performance of the baseline by using the alignment graph. Improvements range from 0.8% (ENG-DEU) to 3.3% (ENG-HIN). This again demonstrates the utility of exploiting multiparallelism for word alignment. It is worth mentioning that in this case the translations are noisy since they were automatically generated. But even with these noisy translations (instead of a "true" multiparallel corpus), our models effectively leverage multiparallelism.

	Method	ENG-PES Prec.	Rec.	F_1	AER	ENG-HIN Prec.	Rec.	F_1	AER	ENG-RON Prec.	Rec.	F_1	AER	ENG-DEU Prec.	Rec.	F_1	AER
Baseline	SimAlign	**0.756**	0.645	0.696	0.304	**0.709**	0.493	0.582	0.418	**0.807**	0.663	0.728	0.272	**0.829**	0.795	0.812	0.188
Init SimAlign	AdAd	0.751	0.700	**0.725**	**0.276**	0.693	0.544	0.610	0.390	0.799	0.696	**0.744**	**0.256**	0.818	0.823	**0.820**	**0.179**
	WAdAd	0.705	**0.740**	0.722	0.278	0.643	**0.574**	0.607	0.394	0.725	**0.717**	0.721	0.279	0.749	**0.844**	0.794	0.207
	NMF	0.734	0.698	0.716	0.284	0.684	0.559	**0.615**	**0.385**	0.780	0.696	0.736	0.265	0.804	0.827	0.815	0.185

Table 4: Results with gold standards translated into other languages using machine translation. The best results in each column are in bold. The three methods exploiting multiparallelism (AdAd, WAdAd, NMF) outperform the baselines on F_1 and AER.

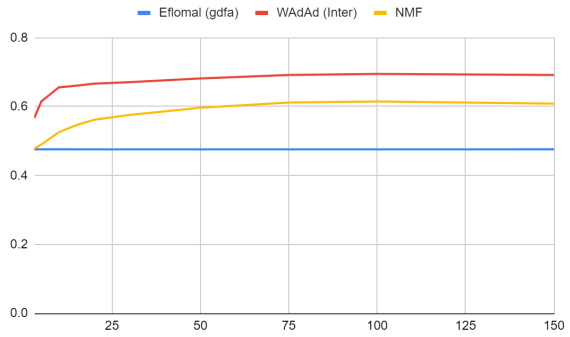

Figure 3: F_1 of MPWA for the target language pair FIN-HEB as a function of the number of additional languages. There is a clear rise initially. The curve flattens around 75.

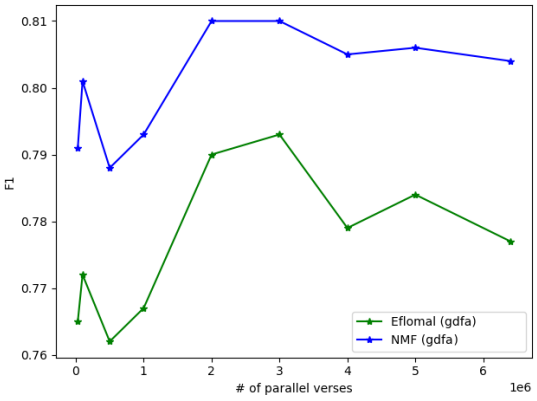

Figure 4: Word alignment F_1 on ENG-FRA as a function of the size of the training set, ranging from 30K to 6.4M training sentence pairs

5.3 Analysis

5.3.1 Effect of number of languages

The effect of adding more languages to the alignment graph is depicted in Figure 3. This plot shows F_1 for FIN-HEB. As seen in the figure, the slope is pretty steep up to 25 languages, but even adding just three languages can still improve the results. For 75 languages we have almost reached the peak and after 100, adding more languages is not improving the results. This means that MPWA can also be helpful for corpora with a smaller number of languages – a massively parallel corpus with thousands of languages is not required.

5.3.2 Size of the training set

To assess the effect of dataset size on the performance of MPWA, we perform a set of experiments on ENG-FRA and NMF. To this end, we take the training data for ENG-FRA and train models on subsets of it. The training data consists of 6.4M sentence pairs – this number is so high because we use the crossproduct of all editions in English and French (§4.3).

The results are shown in Figure 4. Eflomal performance increases with training set size initially and is then less predictable. NMF consistently improves the scores.

5.3.3 Effect of task difficulty

Table 3 shows large improvements for all datasets, and especially for FIN-HEB and FIN-GRC. To get more insight into the reasons for this improvement, we stratify FIN-HEB verses by dividing the interval $[0, 1]$ of initial F_1 performance of Eflomal into five equal-sized subintervals: $[0, 0.2], \ldots, (0.8, 1]$.

Figure 5 indicates that MPWA is most effective for difficult verses, but brings little improvement for easy verses. We attribute this to two reasons:

1. An easy to align verse in a language pair cannot use help from other languages since it already has good alignment links (although the language pair would still be of benefit in improving alignments for the sentence in other languages). So there is no way for MPWA to get better results in this scenario.

2. MPWA only tries to get better results by adding new alignments, and as an easy verse already has many alignment links, adding new links almost inevitably results in a drop in pre-

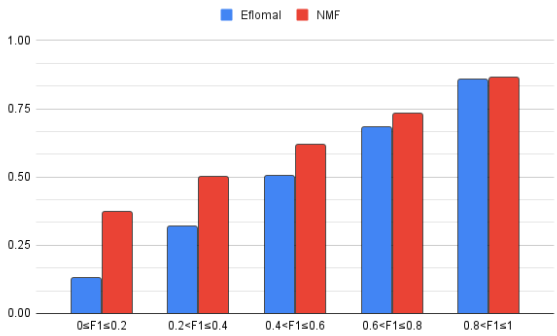

Figure 5: How helpful is MPWA for different difficulty levels? For this analysis, FIN-HEB verses were stratified according to Eflomal F_1 (x-axis). We see that MPWA brings the largest improvements for difficult sentences.

ENG-FRA		FIN-HEB		FIN-GRC	
Lang.	Δ	Lang.	Δ	Lang.	Δ
SPA	+2.0%	TGL	+17.7%	LAT	+7.5%
ITA	+1.9%	FRY,ELL	+17.3%	ELL	+6.6%
DEU	+1.8%	SWE	+17.3%	ENG	+6.1%
NLD	+1.4%	NLD	+16.8%	FRY	+5.8%
AFR	+1.3%	YOR	+14.2%	BEL	+5.7%

Table 5: The five most helpful languages and WAdAd's absolute improvements in F_1 over the initial bilingual aligner SimAlign. For example, MPWA improves the bilingual FIN-GRC alignment by 7.5% if applied to the trilingual corpus FIN-GRC-LAT, i.e., Latin can be viewed as the best bridge between Finnish and Greek.

cision. It may also be possible to inspect and prune existing Eflomal links using MPWA to get better results in this scenario.

5.3.4 Most helpful languages

For each dataset, the five most helpful languages with their corresponding improvements are listed in Table 5. We hypothesize that these languages serve to bridge the typological gap between the two target languages. Table 5 suggests one should be able to achieve excellent results – even for a corpus with a small number of languages – if we utilize an intelligent selection of languages.

5.3.5 Multiple translations in two languages

There are some datasets that contain few languages, but many translations of a text in one language. PBC is one example of such a dataset, many literary works another (e.g., many novels have many translations in English). To see whether MPWA can also help in this scenario, we picked all available 49 English and French editions from PBC and used them as additional translations for the ENG-FRA dataset. The outcome of this experiment is compared with the outcome of the same setup, but with translations from languages other than French and English in Table 6. From this table we can conclude that translations from the target language pair can also assist, but not as much as translations from other languages.

	Prec.	Rec.	F_1	AER
Eflomal (intersection)	0.971	0.521	0.678	0.319
Eflomal (GDFA)	0.856	0.710	0.776	0.221
NMF (target languages)	0.830	0.749	0.787	0.213
NMF (other languages)	0.837	0.753	0.793	0.205

Table 6: F_1 for ENG-FRA. MPWA can exploit a multiparallel corpus with languages different from the target languages ("other languages") better than one that contains additional translations in the target languages ("target languages").

6 Conclusion and Future Work

We presented MPWA, a framework for leveraging multiparallel corpora for word alignment. We used two prediction methods, one based on recommender systems and one based on link prediction algorithms. By adding new alignment edges to the output of bilingual aligners, both methods show large improvements over the bilingual baselines, with absolute improvements of F_1 of up to 20%. We have also ported Blinker and HELFI word alignment gold standards to the Parallel Bible Corpus in the hope that this will help foster more work on exploiting multiparallel corproa.

Future work. In this paper, we have mainly focused on *adding* new alignment edges to baseline word alignments based on properties of the multiparallel alignment graph. This increases recall, but can harm precision. In future work, we plan to expand on the possibility of *deleting* edges based on evidence from the multiparallel alignment graph (cf. 5.3.3), thereby potentially improving both precision and recall.

Acknowledgments

This work was supported by the European Research Council (ERC, Grant No. 740516) and the German Federal Ministry of Education and Research (BMBF, Grant No. 01IS18036A). The fourth author was supported by the Bavarian research institute for digital transformation (bidt) through their fellowship program. We thank the anonymous reviewers for their constructive comments.

References

Lada A Adamic and Eytan Adar. 2003. Friends and neighbors on the web. *Social networks*, 25(3):211–230.

Tamer Alkhouli, Gabriel Bretschner, Jan-Thorsten Peter, Mohammed Hethnawi, Andreas Guta, and Hermann Ney. 2016. Alignment-based neural machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, pages 54–65, Berlin, Germany. Association for Computational Linguistics.

Tamer Alkhouli and Hermann Ney. 2017. Biasing attention-based recurrent neural networks using external alignment information. In *Proceedings of the Second Conference on Machine Translation*, pages 108–117, Copenhagen, Denmark. Association for Computational Linguistics.

Ehsaneddin Asgari and Hinrich Schütze. 2017. Past, present, future: A computational investigation of the typology of tense in 1000 languages. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 113–124, Copenhagen, Denmark. Association for Computational Linguistics.

Albert-László Barabási and Réka Albert. 1999. Emergence of scaling in random networks. *science*, 286(5439):509–512.

Nicola Bertoldi, Madalina Barbaiani, Marcello Federico, and Roldano Cattoni. 2008. Phrase-based statistical machine translation with pivot languages. In *International Workshop on Spoken Language Translation (IWSLT) 2008*.

Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2).

Jaegul Choo, Changhyun Lee, Chandan K Reddy, and Haesun Park. 2013. Utopian: User-driven topic modeling based on interactive nonnegative matrix factorization. *IEEE transactions on visualization and computer graphics*, 19(12):1992–2001.

Trevor Cohn and Mirella Lapata. 2007. Machine translation by triangulation: Making effective use of multi-parallel corpora. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 728–735, Prague, Czech Republic. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Zi-Yi Dou and Graham Neubig. 2021. Word alignment by fine-tuning embeddings on parallel corpora. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2112–2128, Online. Association for Computational Linguistics.

Philipp Dufter, Mengjie Zhao, Martin Schmitt, Alexander Fraser, and Hinrich Schütze. 2018. Embedding learning through multilingual concept induction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1520–1530, Melbourne, Australia. Association for Computational Linguistics.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia. Association for Computational Linguistics.

Karim Filali and Jeff Bilmes. 2005. Leveraging multiple languages to improve statistical MT word alignments. In *IEEE Workshop on Automatic Speech Recognition and Understanding, 2005.*, pages 92–97. IEEE.

Sarthak Garg, Stephan Peitz, Udhyakumar Nallasamy, and Matthias Paulik. 2019. Jointly learning to align and translate with transformer models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4453–4462, Hong Kong, China. Association for Computational Linguistics.

Ayyoob ImaniGooghari, Masoud Jalili Sabet, Philipp Dufter, Michael Cysou, and Hinrich Schütze. 2021. ParCourE: A parallel corpus explorer for a massively multilingual corpus. In *Proceedings of the 59th Annual Meeting of the Association for Computational*

Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations, pages 63–72, Online. Association for Computational Linguistics.

Masoud Jalili Sabet, Philipp Dufter, François Yvon, and Hinrich Schütze. 2020. SimAlign: High quality word alignments without parallel training data using static and contextualized embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1627–1643, Online. Association for Computational Linguistics.

Glen Jeh and Jennifer Widom. 2002. Simrank: a measure of structural-context similarity. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 538–543.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Philipp Koehn, Franz J. Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 127–133.

Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37.

Da Kuang, Jaegul Choo, and Haesun Park. 2015. Nonnegative matrix factorization for interactive topic modeling and document clustering. In *Partitional Clustering Algorithms*, pages 215–243. Springer.

Shankar Kumar, Franz J. Och, and Wolfgang Macherey. 2007. Improving word alignment with bridge languages. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 42–50, Prague, Czech Republic. Association for Computational Linguistics.

Adrien Lardilleux and Yves Lepage. 2008a. A truly multilingual, high coverage, accurate, yet simple, sub-sentential alignment method. In *The 8th conference of the Association for Machine Translation in the Americas (AMTA 2008)*, pages 125–132, Waikiki, Honolulu, United States.

Adrien Lardilleux and Yves Lepage. 2008b. Multilingual alignments by monolingual string differences. In *Coling 2008: Companion volume: Posters*, pages 55–58, Manchester, UK. Coling 2008 Organizing Committee.

Adrien Lardilleux and Yves Lepage. 2009. Sampling-based multilingual alignment. In *Proceedings of the International Conference RANLP-2009*, pages 214–218, Borovets, Bulgaria. Association for Computational Linguistics.

William D. Lewis and Fei Xia. 2008. Automatically identifying computationally relevant typological features. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II*.

Linyuan Lü and Tao Zhou. 2010. Link prediction in weighted networks: The role of weak ties. *EPL (Europhysics Letters)*, 89(1):18001.

Xin Luo, Mengchu Zhou, Yunni Xia, and Qingsheng Zhu. 2014. An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems. *IEEE Transactions on Industrial Informatics*, 10(2):1273–1284.

Kelly Marchisio, Conghao Xiong, and Philipp Koehn. 2021. Embedding-enhanced GIZA++: Improving alignment in low-and high-resource scenarios using embedding space geometry. *arXiv preprint arXiv:2104.08721*.

Thomas Mayer and Michael Cysouw. 2012. Language comparison through sparse multilingual word alignment. In *Proceedings of the EACL 2012 Joint Workshop of LINGVIS & UNCLH*, pages 54–62, Avignon, France. Association for Computational Linguistics.

Thomas Mayer and Michael Cysouw. 2014. Creating a massively parallel Bible corpus. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3158–3163, Reykjavik, Iceland. European Language Resources Association (ELRA).

I. Dan Melamed. 1998. Manual annotation of translational equivalence: The blinker project. *CoRR*, cmp-lg/9805005.

Mathias Müller. 2017. Treatment of markup in statistical machine translation. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 36–46, Copenhagen, Denmark. Association for Computational Linguistics.

Preslav Nakov and Hwee Tou Ng. 2012. Improving statistical machine translation for a resource-poor language using related resource-rich languages. *Journal of Artificial Intelligence Research*, 44:179–222.

Franz Josef Och and Hermann Ney. 2003a. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1).

Franz Josef Och and Hermann Ney. 2003b. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Robert Östling. 2014. Bayesian word alignment for massively parallel texts. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2014, April 26-30, 2014, Gothenburg, Sweden*, pages 123–127. The Association for Computer Linguistics.

Robert Östling. 2015. Word order typology through multilingual word alignment. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 205–211, Beijing, China. Association for Computational Linguistics.

Robert Östling and Jörg Tiedemann. 2016. Efficient word alignment with Markov Chain Monte Carlo. *The Prague Bulletin of Mathematical Linguistics*, 106(1).

Mohammad Sadegh Rasooli, Noura Farra, Axinia Radeva, Tao Yu, and Kathleen McKeown. 2018. Cross-lingual sentiment transfer with limited resources. *Machine Translation*, 32(1):143–165.

Sascha Rothe and Hinrich Schütze. 2014. CoSimRank: A flexible & efficient graph-theoretic similarity measure. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1392–1402, Baltimore, Maryland. Association for Computational Linguistics.

Hinrich Schütze. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123.

Farial Shahnaz, Michael W Berry, V Paul Pauca, and Robert J Plemmons. 2006. Document clustering using nonnegative matrix factorization. *Information Processing & Management*, 42(2):373–386.

Haoyue Shi, Luke Zettlemoyer, and Sida I. Wang. 2021. Bilingual lexicon induction via unsupervised bitext construction and word alignment. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 813–826, Online. Association for Computational Linguistics.

Leila Tavakoli and Heshaam Faili. 2014. Phrase alignments in parallel corpus using bootstrapping approach. *International Journal of Information & Communication Technology Research*, 6(3).

Shijie Wu and Mark Dredze. 2020. Do explicit alignments robustly improve multilingual encoders? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4471–4482, Online. Association for Computational Linguistics.

Wei Xu, Xin Liu, and Yihong Gong. 2003. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 267–273.

Anssi Yli-Jyrä, Josi Purhonen, Matti Liljeqvist, Arto Antturi, Pekka Nieminen, Kari M. Räntilä, and Valtter Luoto. 2020. HELFI: a Hebrew-Greek-Finnish parallel Bible corpus with cross-lingual morpheme alignment. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4229–4236, Marseille, France. European Language Resources Association.

Hamed Zamani, Javid Dadashkarimi, Azadeh Shakery, and W Bruce Croft. 2016. Pseudo-relevance feedback based on matrix factorization. In *Proceedings of the 25th ACM international on conference on information and knowledge management*, pages 1483–1492.

Thomas Zenkel, Joern Wuebker, and John DeNero. 2020. End-to-end neural word alignment outperforms GIZA++. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1605–1617, Online. Association for Computational Linguistics.

Muhan Zhang and Yixin Chen. 2018. Link prediction based on graph neural networks. *Advances in Neural Information Processing Systems*, 31:5165–5175.

Shaojun Zhao and Daniel Gildea. 2010. A fast fertility hidden Markov model for word alignment using MCMC. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 596–605, Cambridge, MA. Association for Computational Linguistics.

Tao Zhou, Linyuan Lü, and Yi-Cheng Zhang. 2009. Predicting missing links via local information. *The European Physical Journal B*, 71(4):623–630.

A Pipeline Details

There are several elements of the MPWA pipeline that can be configured by the user, e.g., depending on whether precision or recall are more important for an application. Here we show in Figures 6 and 7 the two pipeline configurations we used for the results in the paper.

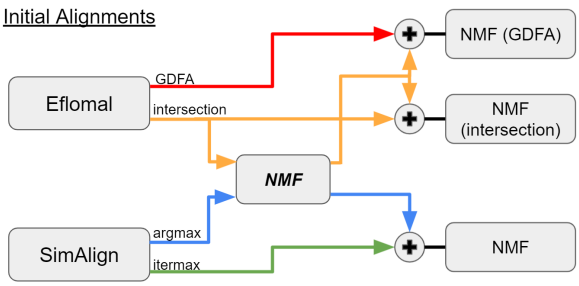

Figure 6: The pipeline for NMF alignments

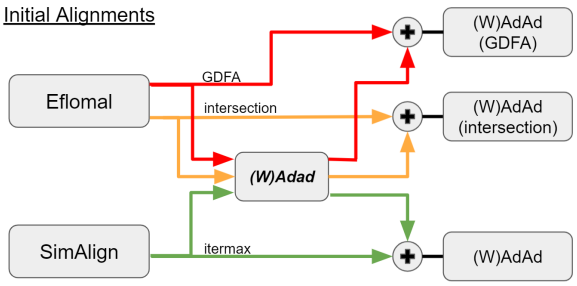

Figure 7: The pipeline for AdAd and WAdAd alignments

Chapter 5

Aligning Very Small Parallel Corpora Using Cross-Lingual Word Embeddings and a Monogamy Objective

Aligning Very Small Parallel Corpora Using Cross-Lingual Word Embeddings and a Monogamy Objective

Nina Poerner, Masoud Jalili Sabet, Benjamin Roth and Hinrich Schütze
Center for Information and Language Processing
LMU Munich, Germany
poerner@cis.uni-muenchen.de

Abstract

Count-based word alignment methods, such as the IBM models or fast-align, struggle on very small parallel corpora. We therefore present an alternative approach based on cross-lingual word embeddings (CLWEs), which are trained on purely monolingual data. Our main contribution is an unsupervised objective to adapt CLWEs to parallel corpora. In experiments on between 25 and 500 sentences, our method outperforms fast-align. We also show that our fine-tuning objective consistently improves a CLWE-only baseline.

1 Introduction

Some parallel corpora, such as the Universal Declaration of Human Rights, are too small to apply count-based word alignment algorithms.

Sabet et al. (2016) show that integrating monolingual word embeddings into IBM Model 1 (Brown et al., 1990) decreases word alignment error rate on a parallel corpus of 1000 sentences. Pourdamghani et al. (2018) exploit monolingual embedding similarity scores to create synthetic training data for Statistical Machine Translation (SMT), and report an increase in alignment F1.

Recent advances have made it possible to create cross-lingual word embeddings (CLWEs) from purely monolingual data (Zhang et al. (2017a), Zhang et al. (2017b), Conneau et al. (2017), Artetxe et al. (2018a)). We propose to leverage such CLWEs for a **similarity-based** word alignment method, which works on corpora as small as 25 sentences. Like Sabet et al. (2016), our method relies only on monolingual data (to train the embeddings) and on the small parallel corpus itself.

Our **CLWE-only baseline** aligns source and target words in a parallel corpus if their CLWEs have maximum cosine similarity. This approach is independent from the size of the parallel corpus, but has the following problems:

- Semantics may differ between the embedding training domain and the parallel corpus.
- CLWEs sometimes fail to discriminate between words with similar contexts, e.g., antonyms.

We therefore propose to **fine-tune** the CLWEs on the small parallel corpus using an **unsupervised embedding monogamy objective**. To evaluate the proposed method, we simulate sparse data settings using Europarl sentences and Bible verses. Our method outperforms the count-based fast-align model (Dyer et al., 2013) for corpus sizes up to 500 (resp., 250) sentences. The proposed fine-tuning method improves over the CLWE-only baseline in terms of both precision and recall.

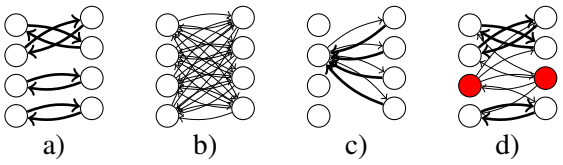

Figure 1: Schematic representation of the monogamy objective. a) one-to-one ("monogamous") alignment: $l(s,t) = 0$, b) many-to-many alignment: $l(s,t) = 1$, c) one-to-many alignment: $l(s,t) = 1$, d) minimizing $l(s,t)$ means making the red nodes more similar to each other, and less similar to the white nodes.

2 Method

2.1 CLWE-only baseline

Our CLWE-only baseline uses a cross-lingual embedding space derived from purely monolingual data (Artetxe et al., 2018a). Let D be our small corpus, and let s (source) and t (target) be parallel sentences from D. Let $\mathrm{clwe}(s_i)$ and $\mathrm{clwe}(t_j)$ be the embedding vectors of tokens s_i and t_j. We align s_i to $\mathrm{argmax}_{t_j \in t}[\cos(\mathrm{clwe}(s_i), \mathrm{clwe}(t_j)]$.

Any ties are broken by proximity to the diagonal of the alignment matrix.

2.2 Fine-tuning method

Intuition. Assume that we have the following sentence pair: *aaa bbb xxx ||| 111 000 222*. Assume further that we know from CLWEs that *aaa* $\approx$ *111* and *bbb* $\approx$ *222*, but we lack informative embeddings for *000* and *xxx*. We may hypothesize that *xxx* $\approx$ *000*, as they are the only tokens that lack translations. We may also hypothesize that *xxx* $\not\approx$ *111*, *xxx* $\not\approx$ *222*, as 111 and 222 already have translations of their own.

In the following, we will refer to this principle as **embedding monogamy**. We assume that in the absence of evidence to the contrary, a source embedding should have

- high similarity to one target embedding
- low similarity to other target embeddings[1]

This principle is related to the IBM Model (Brown et al., 1990), where Expectation Maximization increases $p(f|e)$ if e and f co-occur in sentences where f is not explained by other source words.

Embedding monogamy objective. We define the probability of t_j given s_i as:

$$p(t_j|s_i,t) = \frac{e^{\frac{1}{\tau}\cos(\text{clwe}(s_i),\text{clwe}(t_j))}}{\sum_{j'} e^{\frac{1}{\tau}\cos(\text{clwe}(s_i),\text{clwe}(t_{j'}))}} \quad (1)$$

where τ is a temperature hyperparameter. This definition is similar to the definition of translation probability in Artetxe et al. (2018b) and Lample et al. (2018). But while they normalize over the vocabulary, we normalize over the target sentence. As a consequence, the probability of t_j depends not only on s_i, but also on competitor tokens in t.

With these translation probabilities, we model a two-step random walker $\mathbf{R}^{s\to t\to s}$ that starts at s_i, steps to a random target word and then to $s_{i'}$: $r_{ii'}^{s\to t\to s} = \sum_{j=1}^{\text{len}(t)} p(t_j|s_i,t)p(s_{i'}|t_j,s)$. To maximize monogamy, we maximize the entries on the diagonal of $\mathbf{R}^{s\to t\to s}$, i.e., the probability of the walker returning to its origin. To avoid penalizing long sentences, we minimize the negative logarithm to the base of the source sentence length: $l(s,t) = 1 - \log_{\text{len}(s)} \sum_{i=1}^{\text{len}(s)} r_{ii}^{s\to t\to s}$. This loss has the following properties:

- In a fully "monogamous" situation (see Figure 1 a), $r_{ii}^{s\to t\to s} \to 1 \implies l(s,t) \to 0$.
- In a situation where all source words are equidistant from all target words (see Figure 1 b), $r_{ii}^{s\to t\to s} = \frac{1}{\text{len}(s)} \implies l(s,t) = 1$.

Reversing the roles of source and target results in the following bidirectional loss: $L_{\text{bi}}(s,t) = \frac{1}{2}[l(s,t) + l(t,s)]$. Both terms are necessary, since a given alignment may appear highly monogamous from the perspective of one sentence but not the other (especially when there are left-over words due to a difference in length).

Adding position information. At this point, our objective ignores word positions, which we know to be useful from count-based methods (e.g., Dyer et al. (2013)). Therefore, we add position embeddings inside the translation probability equation:

$$p(t_j|s_i,t) = \frac{e^{\frac{1}{\tau}\cos[\text{clwe}(s_i)+\text{a}(i),\text{clwe}(t_j)+\text{a}(j)]}}{\sum_{j'} e^{\frac{1}{\tau}\cos[\text{clwe}(s_i)+\text{a}(i),\text{clwe}(t_{j'})+\text{a}(j')]}}$$

where $\text{a}(i)$ is a sinusoid embedding vector for position i (Vaswani et al., 2017). As a result, word pairs near the diagonal have higher round trip probabilities initially. Since the monogamy objective aims to strengthen strong links, similar position embeddings act as attractors for non-positional embeddings. Note that we use only the non-positional embeddings for alignment, as the position prior is too strong at test time.

Alignment retention objective. In initial experiments, we found that the monogamy objective increases recall but risks losing precision, relative to the CLWE-only baseline. Therefore, we add an additional objective that aims to increase round trip probability for alignments made by the baseline, but does not influence unaligned words:

$$L_{\text{ret}}(s,t) = \frac{1}{2}[l_{\text{ret}}(s,t) + l_{\text{ret}}(t,s)]$$
$$l_{\text{ret}}(s,t) = -\log \frac{\sum_{i,j} p(t_j|s_i,t)p(s_i|t_j,s)m_{ij}^{st}}{\sum_{i,j} m_{ij}^{st}}$$
$$m_{ij}^{st} = \mathbb{I}[(s_i,t_j) \in \text{align}_0]$$

where align_0 is the intersection of the s-to-t and t-to-s alignments made with the initial CLWEs (see Section 2.1). Our final loss function is: $L(D) = \frac{1}{|D|}\sum_{(s,t)\in D}[L_{\text{bi}}(s,t) + \alpha L_{\text{ret}}(s,t)]$.

[1] Of course, this assumption is over-simplistic, as one-to-n alignments exist (e.g., English *not* should be similar to both French *ne* and *pas*).

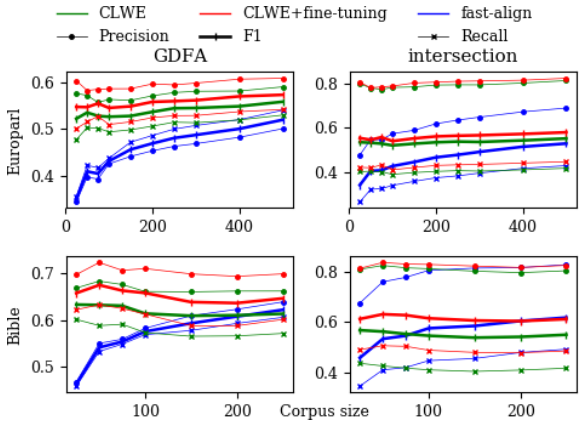

Figure 2: Alignment precision, recall and F1 as a function of corpus size.

3 Evaluation

We evaluate our model on subsets of different sizes from the English-German Europarl gold alignments[2] and French-English Bible gold alignments (Melamed, 1998)[3]. We initialize CLWEs with the unsupervised algorithm of Artetxe et al. (2018a) on monolingual FastText embeddings (Bojanowski et al., 2017)[4]. Fine-tuning is done in `keras`, using the adam optimizer (Kingma and Ba, 2014). We set $\alpha = 1.0$ and $\tau = 0.001$, and apply 50% dropout to the embeddings.

We use fast-align (Dyer et al., 2013) as a count-based baseline, since it outperformed the IBM models in initial experiments. We symmetrize alignments by either intersection or the grow-diag-final-and (GDFA) heuristic (Koehn et al., 2007). We train fast-align and our fine-tuning method for 500 iterations.

4 Discussion

4.1 Corpus size

The performance of fast-align is highly dependent on corpus size, which is not surprising, seeing that it has to infer word semantics from the small corpus alone. The CLWE-only baseline on the other hand is independent from corpus size, resulting in decent performance even on 25 parallel sentences. Importantly, the positive effect of our fine-tuning method seems to be robust to corpus size, as we see improvements in F1 for all sizes.

[2] `www-i6.informatik.rwth-aachen.de/goldAlignment/`

[3] `nlp.cs.nyu.edu/blinker/`. We consider links with inter-annotator agreement as sure, others as possible.

[4] `fasttext.cc`, top-200000 words per language

4.2 Benefits of fine-tuning

We find that the proposed fine-tuning method has a positive effect on alignment precision and recall, relative to the CLWE-only baseline. We assess some sentence pairs qualitatively to find reasons for this improvement:

Resolution of ambiguities. Word embeddings sometimes fail to differentiate between words with similar contexts, such as antonyms. In Figure 3 (top), our fine-tuning method resolves such an ambiguity: Here, the initial CLWE of *answer* is slightly more similar to German *frage* (= *question*) than to the true translation *antwort*. Since *frage* already has a round trip partner, the monogamy objective pushes *answer* away from *frage*, resulting in the addition of a correct alignment between *answer* and *antwort*.

In-domain word translations. Since word embeddings are trained on general-purpose corpora, CLWEs can fail to reflect domain-specific word translations. One such example is the translation of *lord* as French *éternel* (≈ *"eternal one"*) in Figure 3 (bottom). While the translation is common in this particular Bible version, the CLWEs do not reflect it well ($\cos(\textit{lord}, \textit{éternel}) < \cos(\textit{wicked}, \textit{éternel})$). Through fine-tuning, and due to their frequent coocurrence in the small corpus, the similarity between *éternel* and *lord* increases enough for a successful alignment.

5 Use case: Aligning the UDHR

In practice, our method would not be applied to English-German or English-French, as there is no lack of parallel data for these language pairs. For a more realistic use case, we align the 50 articles of the Universal Declaration of Human Rights[5] in Macedonian and Afrikaans. While we do not have gold alignments for an evaluation, a preliminary qualitative analysis suggests that our method finds a reasonable semantic word alignment, while fast-align mainly predicts the diagonal (see Figure 4 for examples).

6 Related Work

Embeddings for word alignment. Sabet et al. (2016) reformulate the IBM 1 model to predict the probability of monolingual target embedding vectors. They report improvements in AER for

[5] https://unicode.org/udhr/

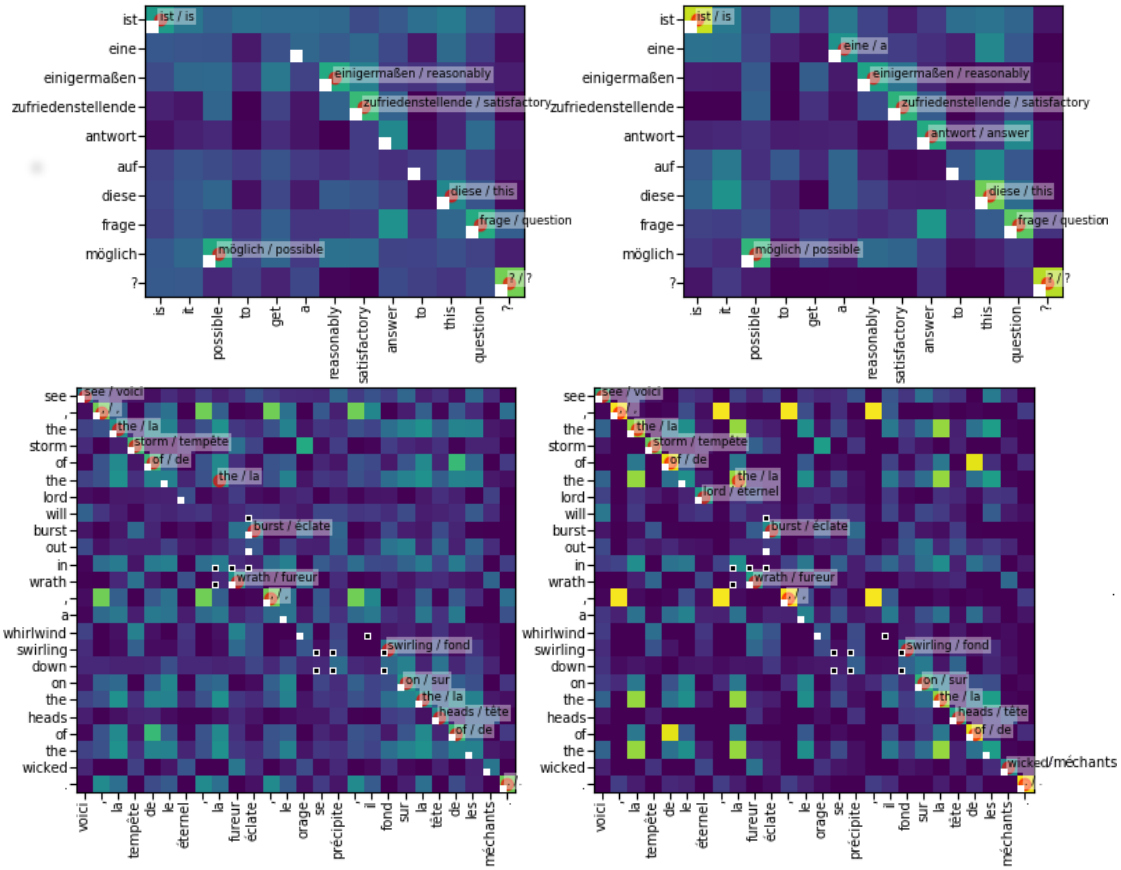

Figure 3: Similarity matrices before (left) and after (right) fine-tuning. Red dots: our alignment (intersection). White squares: sure gold alignments. Empty white squares: possible gold alignments.

English-French on parallel corpora between 1K and 40K sentences, as well as improvements in precision on words with frequency ≤ 20.

Pourdamghani et al. (2018) exploit similarity scores from monolingual embeddings to create synthetic training data for an SMT system. They report improvements for English-Chinese, English-Arabic and English-Farsi alignment ($\Delta F1 = 0.2\%, 0.5\%, 1.7\%$). Their smallest parallel corpus has 500K sentences, while we align a few dozen to hundred sentences.

Two-step round trip objective. Our use of two-step round trips is inspired by Haeusser et al. (2017). They optimize domain adaptation using a random walker that steps from image representations with known labels to image representations with unknown labels and back. While their target is a uniform distribution over images with the same label as the image of origin, ours is to have maximum probability mass on the word of origin.

Low resource CLWEs. Our approach relies on the availability of high-quality CLWEs. Wada and Iwata (2018) report that in settings with little monolingual data ($<$ 1M sentences), mapping approaches like Artetxe et al. (2018a) are not robust. Instead, they propose to learn CLWEs from a language model trained on the union of two small monolingual corpora. Their work is orthogonal to our fine-tuning method, since we make no assumptions about how the CLWEs are created.

7 Conclusion

We have presented a **similarity-based** method to produce word alignments for very small parallel corpora, using monolingual data and the corpus itself. Our **CLWE-only baseline** uses an unsupervised mapping of monolingual embeddings (Artetxe et al., 2018a). Our main contribution is an **unsupervised embedding monogamy objective**, which adapts CLWEs to the small parallel corpus. Our model outperforms count-based fast-align (Dyer et al., 2013) on parallel corpora up to 500 (resp., 250) sentences.

We expect that our method will be useful in low-resource settings, e.g., when aligning the Universal Declaration of Human Rights.

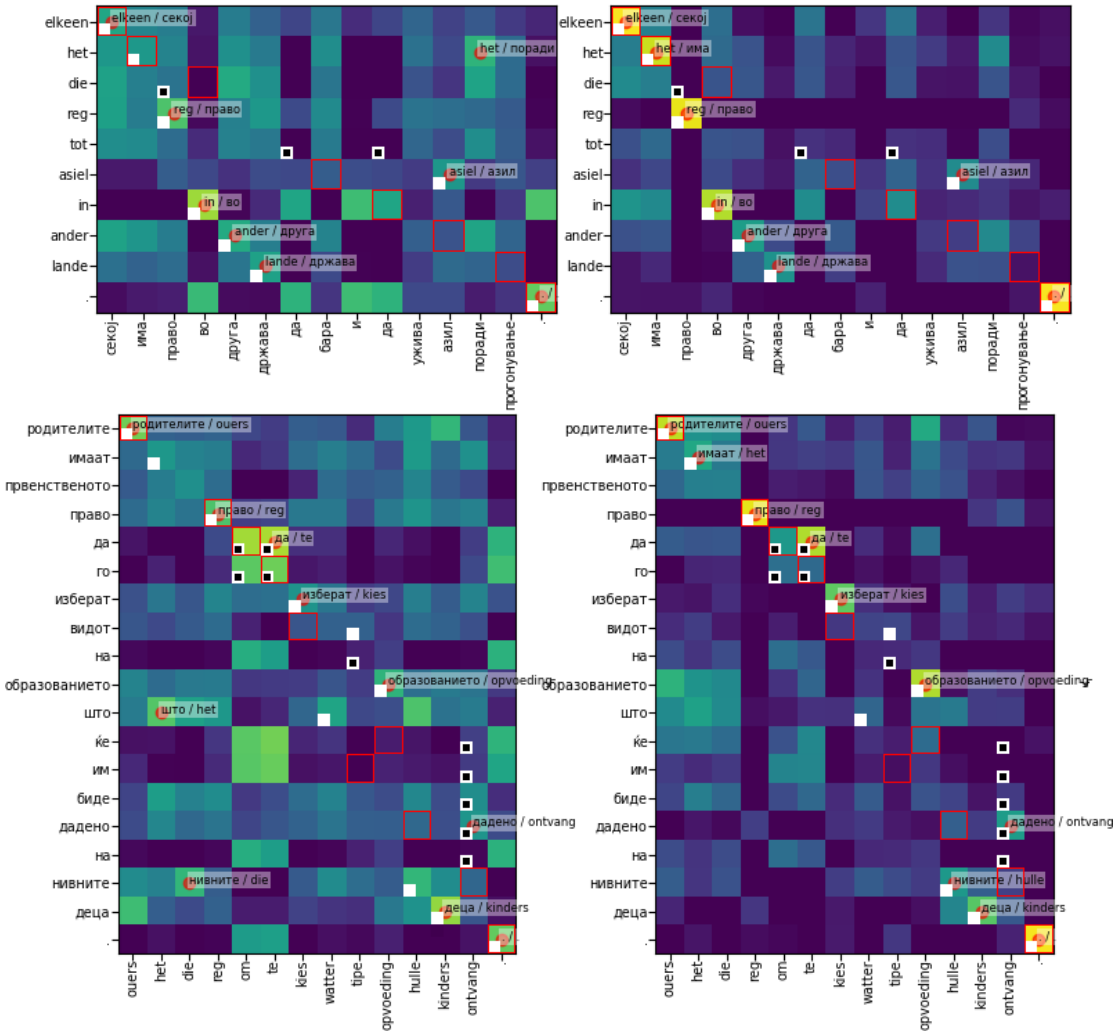

Figure 4: Articles 14(1) and 26(3) from the UDHR. Similarity matrices before (left) and after (right) fine-tuning. Red dots: our alignment (intersection). Red boxes: fast-align (intersection). White squares: sure gold alignments. Empty white squares: possible gold alignments (by the authors).

Acknowledgments. We gratefully acknowledge funding for this work by the European Research Council (ERC #740516).

References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018a. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *ACL*, pages 789–798, Melbourne, Australia.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018b. Unsupervised statistical machine translation. In *EMNLP*, pages 3632–3642, Brussels, Belgium.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association of Computational Linguistics*, 5(1):135–146.

Peter F Brown, John Cocke, Stephen A Della Pietra, Vincent J Della Pietra, Fredrick Jelinek, John D Lafferty, Robert L Mercer, and Paul S Roossin. 1990. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*.

Chris Dyer, Victor Chahuneau, and Noah A Smith. 2013. A simple, fast, and effective reparameterization of IBM Model 2. In *NAACL-HTL*, pages 644–648, Atlanta, USA.

Philip Haeusser, Thomas Frerix, Alexander Mordvintsev, and Daniel Cremers. 2017. Associative domain adaptation. In *ICCV*, pages 2765–2773, Venice, Italy.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *ACL*, pages 177–180, Prague, Czech Republic.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Phrase-based & neural unsupervised machine translation. *arXiv preprint arXiv:1804.07755*.

I Dan Melamed. 1998. Manual annotation of translational equivalence: The Blinker project. Technical report, University of Pennsylvania Institute for Research in Cognitive Science.

Nima Pourdamghani, Marjan Ghazvininejad, and Kevin Knight. 2018. Using word vectors to improve word alignments for low resource machine translation. In *NAACL-HLT*, pages 524–528, New Orleans, USA.

Masoud Jalili Sabet, Heshaam Faili, and Gholamreza Haffari. 2016. Improving word alignment of rare words with word embeddings. In *COLING 2016: Technical Papers*, pages 3209–3215, Osaka, Japan.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*, pages 5998–6008, Long Beach, USA.

Takashi Wada and Tomoharu Iwata. 2018. Unsupervised cross-lingual word embedding by multilingual neural language models. *arXiv preprint arXiv:1809.02306*.

Meng Zhang, Yang Liu, Huanbo Luan, and Maosong Sun. 2017a. Adversarial training for unsupervised bilingual lexicon induction. In *ACL*, pages 1959–1970, Vancouver, Canada.

Meng Zhang, Yang Liu, Huanbo Luan, and Maosong Sun. 2017b. Earth mover's distance minimization for unsupervised bilingual lexicon induction. In *EMNLP*, pages 1934–1945, Copenhagen, Denmark.

Chapter 6

Subword Sampling for Low Resource Word Alignment

Subword Sampling for Low Resource Word Alignment

Ehsaneddin Asgari[*†], **Masoud Jalili Sabet** [*,◇], **Philipp Dufter**[◇], **Christopher Ringlstetter**[†], **Hinrich Schütze**[◇]
[◇] Center for Information and Language Processing, LMU Munich, Germany
[†] NLP Expert Center, Data:Lab, Volkswagen AG, Munich, Germany
ehsaneddin.asgari@volkswagen.de masoud@cislmu.org

Abstract

Annotation projection is an important area in NLP that can greatly contribute to creating language resources for low-resource languages. Word alignment plays a key role in this setting. However, most of the existing word alignment methods are designed for a high resource setting in machine translation where millions of parallel sentences are available. This amount reduces to a few thousands of sentences when dealing with low-resource languages failing the existing established IBM models. In this paper, we propose *subword sampling-based alignment* of text units. This method's hypothesis is that the aggregation of different granularities of text for certain language pairs can help word-level alignment. For certain languages for which gold-standard alignments exist, we propose an iterative Bayesian optimization framework to optimize selecting possible subwords from the space of possible subword representations of the source and target sentences. We show that the subword sampling method consistently outperforms word-level alignment on six language pairs: English-German, English-French, English-Romanian, English-Persian, English-Hindi, and English-Inuktitut. In addition, we show that the hyperparameters learned for certain language pairs can be applied to other languages at no supervision and consistently improve the alignment results. We observe that using $5K$ parallel sentences together with our proposed subword sampling approach, we obtain similar F1 scores to the use of $100K$'s of parallel sentences in existing word-level fast-align/eflomal alignment methods.

1 Introduction

Annotation projection is an important area in Natural Language Processing (NLP), which aims to exploit existing linguistic resources of a particular language for creating comparable resources in other languages (usually low resource languages) using a mapping of words across languages. More precisely, annotation projection is a specific use of parallel corpora, corpora containing pairs of translated sentences from language l_s to l_t. In annotation projection, a set of labels available for language l_s is projected to language l_t via alignment links (the mapping between words in parallel corpora of l_s and l_t). L_s labels can either be obtained through manual annotation or through an analysis module that may be available for l_s, but not for l_t. The label here can be interpreted broadly, including, e.g., part of speech labels, morphological tags and segmentation boundaries, sense labels, mood labels, event labels, syntactic analysis, and coreference (Yarowsky et al., 2001; Diab and Resnik, 2002; Agić et al., 2016).

Language resource creation for low-resource languages, for the purpose of automatic text analysis can create financial, cultural, scientific, and political value. For instance, the creation of a sentiment lexicon for a low resource language would be an excellent help for customer reviews analysis in big corporations having branches all over the world, where 7000 languages are spoken, or such a resource can be used to predict stock market movements from social media in a low resource setting. Furthermore, such resources can contribute to creating knowledge (Östling, 2015; Asgari and Schütze, 2017) for linguists, which pure scientific value aside, the linguistic knowledge can be incorporated into machine learning models for natural language understanding as well.

The mapping between words across languages as a basis for annotation projection is automatically generated using statistical word alignment, modeled on parallel corpora. This means that given parallel corpora for a set of languages and linguis-

[*] Equal contribution

tic resources for only one language, we can automatically create resources for the other languages through annotation projection. One of the main challenges for annotation projection is that corpora are often relatively small for low resource languages. The existing IBM-based alignment models work well for high-resource settings, but they fail in the low-resource case (Poerner et al., 2018). The most popular dataset for low resource alignment, the Bible Parallel Corpus, containing a large number (1000+) of languages, are characteristically low-resource, i.e., having only around 5000-10000 parallel sentences per language pair. This paper aims to introduce a framework to reliably relate linguistic units, words, or subwords, in a low resource parallel corpus, based on sampling from the space of possible subwords.

2 Methods

2.1 Dataset

We work on the word alignment gold standards from the WPT 2003 and 2005 shared tasks on word alignment. Those language pairs include French, Hindi, Romanian, and Inuktitut always paired with English. In addition, we add English-Persian and English-German. As per the standard scenario in the world alignment literature, we compute an alignment model on an independent corpus of training materials. To simulate a low-resource scenario, we sample the number of parallel training sentences down to 5000, except for Hindi with 3000 sentence pairs. This is the order of magnitude in training data when dealing with low-resource languages contained, e.g., in the Bible Parallel Corpus. In addition, we experiment on mid-resource cases when using the complete set of available training sentences in English-Romanian, English-Inuktitut, English-Persian, in which their complete set is neither low-resource nor contain more than 1M parallel sentences. See Table 1 for details on the data.

2.2 Evaluation

We evaluate word alignments with F_1 score computed by

$$\text{prec} = \frac{|A \cap P|}{|A|}, \ \text{rec} = \frac{|A \cap S|}{|S|}, \ F_1 = \frac{2 \ \text{prec} \ \text{rec}}{\text{prec} + \text{rec}},$$

where $|A|$ is the set of predicted alignment edges, $|S|$ the set of sure and $|P|$ the set of possible alignment edges. Note that $S \subset P$, and both are known from the gold standard.

2.3 Sentence subword space

For splitting text into subwords, we use Byte-Pair-Encoding by Sennrich et al. (2016). The BPE algorithm for a certain random seed and a given vocabulary size (analogous to the number of character merging steps) breaks a sentence into a unique sequence of subwords. Continuing the merging steps would result in the enlargement of the subwords, resulting in fewer tokens.

Hypothesis: Let $S_{pq} = \bigcup_{j=1}^{N} (s_p^{(j)}, s_q^{(j)})$ be a collection of N parallel paired sentences in the language pair l_p and l_q. We assume that for a certain S_{pq} there exists an optimal segmentation scheme constructed by accumulation of different granularities of (l_p,l_q), ξ^*, among all possible segmentation schemes (ξ's), which depends on the morphological structures of this language pair.

The space of possible segmentations of a sentence s, denoted as $\Phi_l(s)$ for language l, is created by variations in the segmentation by varying the number of merging steps.

In this notation $\Phi_l(s) = \bigcup_{i=1}^{M_l} \Phi_l^{(i)}(s)$, where $\Phi_l^{(k)}(s)$ refers to a specific vocabulary size selection for the segmentation of s considering the first k merging steps in the BPE algorithm for language l. M_l is the maximum number of merging steps in l. We define Φ_{pq} as the set of all possible segmentation pairs in language pair l_p and l_q:

$$\Phi_{pq} = \bigcup_{i=1}^{M_{l1}} \Phi_p^{(i)} \times \bigcup_{i=1}^{M_{l2}} \Phi_q^{(i)}$$

When we deal with a single language, to explore the possible segmentations, Monte Carlo sampling from Φ_l can be used to have different views on the segmentation, where the likelihood of certain segmentation $\Phi_l^{(k)}$ is proportional to the number of sentences affected in the corpus by introducing the k^{th} merging step, as proposed in (Asgari et al., 2019a; Asgari, 2019; Asgari et al., 2019b) for the segmentation of protein and DNA sequences. However, in the alignment problem, we deal with a 2D space (can be represented as a grid as in Figure 4) of possibilities for the vocabulary sizes ($\approx$ the number of merging steps in BPE) of l_p and l_q. The inclusion of each cell in this grid introduces new instances to the parallel corpus, potentially transferring a low-resource setting to a high-or-mid resource setting. In this high resource setting, the subwords of a certain sentence are assigned in T ways (the number of cells we select from the

Lang.	Gold Standard	# Sentences	$\|S\|$	$\|P \setminus S\|$	Parallel Training Data	# Sentences
English-German	EuroParl-based[a]	508	9612	921	EuroParl (Koehn, 2005)	1920k
English-Persian	(Tavakoli and Faili, 2014)	400	11606	0	TEP (Pilevar et al., 2011)	600k
English-French	WPT2003, (Och and Ney, 2000),	447	4038	13400	Hansards (Germann, 2001)	1130k
English-Hindi	WPT2005[b]	90	1409	0	Emille (McEnery et al., 2000)	3k
English-Inuktitut	WPT2005[b]	75	293	1679	Legislative Assembly of Nunavut[b]	340k
English-Romanian	WPT2005[b]	203	5033	0	Constitution, Newspaper[b]	50k

[a] www-i6.informatik.rwth-aachen.de/goldAlignment/
[b] http://web.eecs.umich.edu/~mihalcea/wpt05/

Table 1: Details on gold standards and training data. $|S|$ is the number of sure edges in the gold standard and $|P \setminus S|$ the number of additional possible edges.

Φ_{pq} grid). Finally, to confirm an alignment link at word-level, we set a threshold λ; λ is the minimum ratio of subword segmentation is required to confirm a word alignment link. Note that not necessarily all cells of the grid improve the alignment, we thus need a strategy to pick a subset of cells $\xi^* \subset \Phi_{pq}$ maximizing the ultimate alignment score. Having language pairs with ground-truth alignment, we can solve this problem via hyperparameter optimization using Bayesian optimization. Subsequently, we investigate whether applying the same hyperparameters, on another language pair yields improvements. To solve this the optimization problem for the supervised case, we propose an iterative greedy subword sampling algorithm.

2.4 Iterative subword sampling algorithm

To maximize the alignment score for the known links (the ground-truth) at the word-level, we are seeking for ξ^* a set of cells in the Φ_{pq} grid, and their corresponding thresholds λ^* satisfying the following equation:

$$\xi^*, \lambda^* = \underset{\xi^i, 0 \leq \lambda \leq 1; i \in \{1,2,..,T\}}{\operatorname{argmin}} -f(\Phi_{pq}, S_{pq}, \mathbf{y_{pq}}),$$

where f refers to the alignment F1 score based on ground-truth, which its underlying alignment model does not have any closed form nor gradient. $\mathbf{y_{pq}}$ is the ground-truth we have for the language pair l_p and l_q, and S_{pq} refers to the parallel sentences, which are going to be segmented in T different schemes (T cells from the Φ_{pq} grid, $0 < i < T$). These T cells can be selected in any order. However, to reduce the search space, we propose a sequential greedy selection of the segmentations (ξ^i, λ), and solve each step in a Bayesian optimization framework. The iterative process is detailed in Algorithm 1. The core computation of this algorithm is $\xi_i, \lambda = \underset{\xi_i, \lambda}{\operatorname{argmin}} -f(\Phi_{pq}, S_{pq}, \mathbf{y_{pq}}, \xi_{0:i-1})$, for which the selected vocabulary sizes up to the current iteration $(\xi_{0:i-1})$ are used for segmentation and the measurement of the alignment score. We perform Bayesian optimization to find the next optimal values for ξ_i and λ. As discussed in §2.3, in the Bayesian optimization, we explore the cells from the grid of Φ_{pq} using logarithmic priors for each of Φ_p and Φ_q. We continue this process until the the moment where introducing more segmentations does not improve the alignment score, setting an early stopping condition.

Algorithm 1: Iterative subword sampling

Result: graph G of word-aligned sentence pairs
$f1_{prev} = 0$; $i = 0$;
ξ = $< empty >$; % history of selected cells
λ = $< empty >$; % history of selected λ's
$\delta = +\infty$; E = early-stopping parameter;
while $\exists \delta > 0$ *in the last* E *iterations* **do**
  $\xi_i, \lambda = \underset{\xi_i, \lambda}{\operatorname{argmin}} -f(\Phi_{pq}, S, y, \xi, \lambda)$;
  $\xi.push(\xi_i)$;
  $\lambda.push(\lambda)$;
  $f1^* = f(S, y, \xi, \lambda)$;
  $\delta = f1^* - f1_{prev}$;
  $f1_{prev} = f1^*$;
  $i = i + 1$;
end
G = $alignment(segment(S, \xi, \lambda))$

2.5 Intuition behind the use of logarithmic priors for the vocabulary size

Figure 1 provides an intuition behind the use of logarithmic priors. This diagram shows that by introducing a new merging step (increasing the vocabulary size by one) in the BPE algorithm, which portion of sequences are affected. As proposed for protein sequences (Asgari et al., 2019a), this can be served as an approximation for the relative likelihood of including a merging step (which is analogous to introducing a new subword).

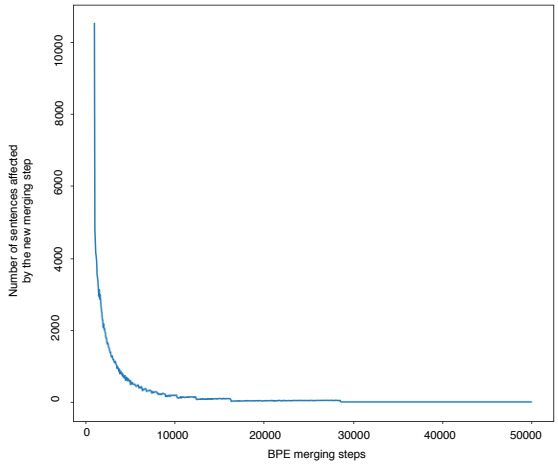

Figure 1: An example of English BPE on a collection of 10000 sentences. This diagram shows that with introducing new merging steps how many sentences are going to be affected .

2.6 Subword sampling in other languages

After training how to choose ξ^*, λ^* for a particular language pair, we apply the same vocabulary settings on new language pairs and evaluate the resulting alignment scores. This would help us in the investigation of the generalizability of a language pair on other language pairs.

2.7 Experimental Setup

Evaluate Low-Resource Alignment

Since the main motivation of subword sampling alignment is for the low-resource scenario[1], we first evaluate the method for an analogous use case, where only a few thousands of parallel sentences are available, similar to the Bible Parallel Corpus of 1000+ languages. To produce a similar scenario for the evaluation, we get samples of 5K aligned sentences from the training datasets of English-German, English-French, English-Romanian, English-Persian, English-Hindi, and English-Inuktitut and concatenate the gold-standard datasets to them. The statistical word aligners generate forward, and backward alignments need a post-processing step of symmetrization (Koehn, 2010). We compared intersection and grow-diag-final-and (GDFA), which produce comparable results in terms of F1 score, and the intersection method having a higher precision. Since the final alignments are produced from the aggregation of all segmentations' alignments, the intersection method with higher precision is a proper candidate. Thus, we use the intersection method throughout the experiments.

For each language pair, we evaluate the word-level alignment, as well as the Bayesian optimization subword sampling. In addition, in order to investigate how the vocabulary size of a particular language pair generalizes to the other language pairs, we also evaluate the optimized $\xi^*_{l_1,l_2}, \lambda^*_{l_1,l_2}$ for each pair on all other language pairs.

Evaluate Mid-Resource Alignment

In addition to the low-resource alignment, we evaluate our approach against the word-level alignment of fast-align and eflomal in the mid-resource scenario (having less than 1M sentence pairs). Therefore, From the six language pairs with gold-standard alignment, we select English-Persian, English-Inuktitut, and English Romanian, containing 600k, 340k, 50k sentence pairs, respectively. For each language pair, we use the vocabulary sizes optimized in the low-resource alignment experiment.

3 Results

3.1 Iterative Subword Sampling

An example space of Φ_{pq} (for English-German) that is explored in the Bayesian optimization to find the ξ^* is shown in Figure 3, a 2D representation of the selected cells and the order of selection by Bayesian optimization on the English-German corpus is provided in Figure 4. We observe that the new segmentation in each iteration consistently improves the alignment scores in the next iteration. Furthermore, as may be expected, the sampled vocabulary sizes are mainly chosen from the lower sizes, i.e., affecting more sentences (Figure 1). All studied language pairs show similar behaviour in selecting subword vocabulary sizes (Figure 2).

3.2 Low-Resource Alignment Results

Table 2 shows F_1 scores of alignment across six language pairs in the low-resource alignment (having a maximum set of 5K aligned sentence pairs). This table compares the word-level and subword-level alignments as well as the generalizability of the $\xi^*_{l_1,l_2}, \lambda^*_{l_1,l_2}$ on the other language pairs. Interestingly, across all language pairs, we observe improvements of 1.4 to 8.0 percentage points in the alignment F1 score in comparison with the word-

[1]The language itself is not necessarily a low-resource language, but the number of sentence pairs is relatively low (less than 10K)

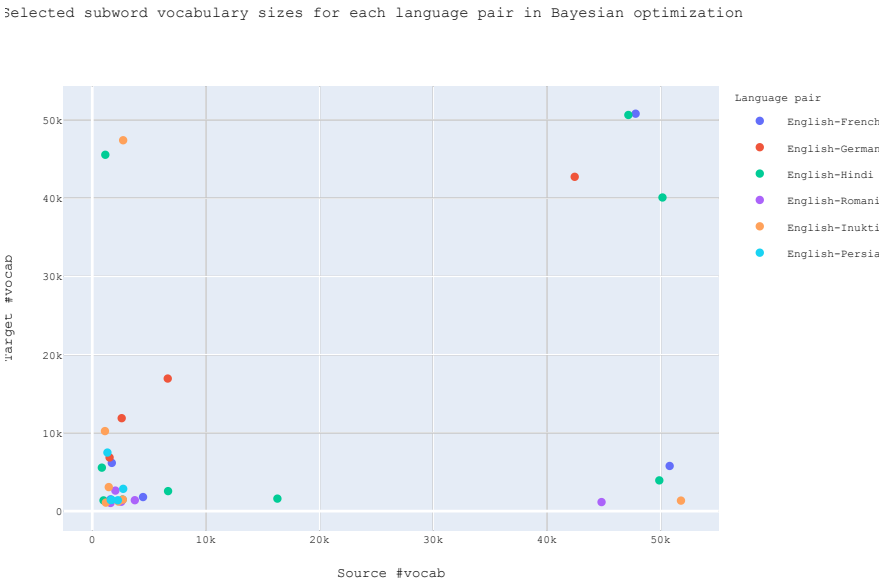

Figure 2: Selected subword vocabulary sizes within Bayesian optimization for each language pairs of English-German, English-French, English-Romanian, English-Persian, English-Hindi, and English-Inuktitut.

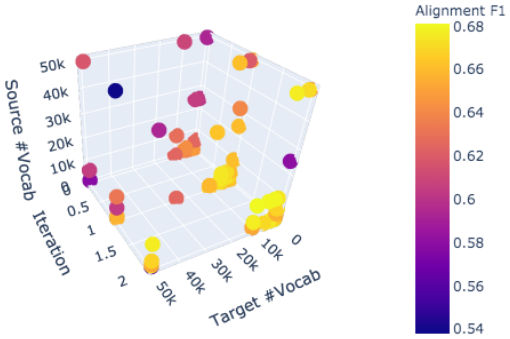

Figure 3: The space of Φ_{pq} that is explored in the Bayesian optimization in the first 3 iterations. The exploring steps are colored with their alignment F1 scores.

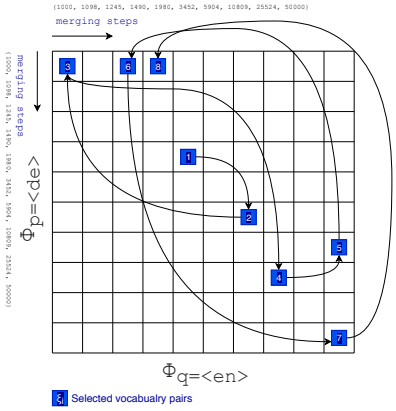

Figure 4: An example of the space Φ_{pq} for English and German and the selected cells by the Bayesian optimization.

level alignment. Subword-sampling optimized on a specific pair consistently improves the word-level alignment also of the other languages. Certain language pairs, like Romanian-English and Hindi-English, proved better generalizability when applied to the other language pairs. This result suggests that although the gold standard is decisive for a significant improvement of the alignment through optimizing the vocabulary sizes, then optimal vocabulary sizes trained on different language pairs (potentially with similar morphological complexity) can be efficiently applied to increase the alignment performance for a new language pair .

3.3 Mid-Resource Alignment Results

F_1 scores of alignment across three language pairs in the mid-resource alignment (having less than 1M aligned sentence pairs) is shown in Table 3. This table compares the word-level and subword-level alignments and the generalizability of the hyper-parameter optimized on other language pairs in low-resource for a mid-resource setting. Again, across all language pairs, we observe improvements of 2.7 to 7 percentage points in the alignment F1 score compared to the word-level alignment. Interestingly, the F1 we achieved, using 5K parallel sentences and subword sampling, is similar to the word-level F1 score of English-Inuktitut

	English-German	English-French	English-Romanian	English-Persian	English-Hindi	English-Inuktitut
word-level	0.685	0.897	0.616	0.527	0.508	0.771
Vocab-size Sampling Optimization	0.746	0.913	0.662	0.579	0.548	0.858*
Apply <English-German>		0.899	0.645	0.560	0.532	0.845*
Apply <English-French>	0.743		0.651	0.541	0.524	**0.853**
Apply <English-Romanian>	**0.745**	0.905		0.579	**0.547**	0.821
Apply <English-Persian>	0.743	0.908	**0.663**		0.521	0.816
Apply <English-Hindi>	0.742	0.904	**0.663**	**0.580**		0.804
Apply <English-Inuktitut>	0.744	**0.914**	0.655	0.530	0.529	

Table 2: The alignment performances (in terms of F1 score) of six language pairs in the low-resource scenario, where the subword sampling and word-level alignments are compared. In addition, the results on applying the hyper-parameters of language pairs on all other pairs are also provided. We experimented systematically on the use of both eflomal and fast-align for every setting. However, for simplicity, in each cell, the best performance of fast-align and eflomal is reported. With the exception of the marked F1 's with *, the best results obtained using eflomal method for all the alignments.

using 240K parallel sentences and the word-level F1 score of English-Persian using 600K parallel sentences.

3.4 Qualitative Analysis

We performed a qualitative analysis for English-German and showed six examples in Figure 5. We observed two sources of improvements: i) Compounds, which are frequent in German, obtain better alignments and ii) As we aggregate alignment edges through a λ-weight vote, we observe an "ensembling" effect which mainly affects fertility. Examples 1, 2 and 3 show the compound effect: "Menschenrechte" is correctly aligned to "Human rights" only when using sampling optimization. Similarly, "Mobiltelefonlizenzen" gets successfully aligned to "mobile phone license" whereas pure eflomal only aligns it to "licenses". In addition, the differently formatted year numbers in Example 2 are easy to align once subword sampling is used. Examples 4 and 5 show the presumed ensembling effect. We hypothesize that "EVP" is aligned to different words in the English sentence across different subword samples. Once aggregated, "EVP-Fraktion" has high fertility, which is useful in this scenario. Similarly, "des" (meaning "of the") receives a better alignment through subword sampling as some models align "des" to "of" and some others to "the". Subword sampling cannot resolve all errors of eflomal and can also be harmful in rare cases. Example 6 shows a case where the word "Abmessungen" ("dimensions" or "measurements") obtains two incorrect alignment edges, presumably because it frequently gets split into subwords like "Ab" or "ungen" which carry only little semantic information.

4 Related Work

Classical models. Statistical word alignment methods (e.g., GIZA++ (Och and Ney, 2000), fast-align (Dyer et al., 2013), eflomal (Östling and Tiedemann, 2016)) are mostly based on *IBM models* (Brown et al., 1993), which are generative models describing how a source language sentence S generates a target language sentence T using alignment latent variables. These models use an expectation maximization (EM) algorithm to train the alignment and only require sentence-aligned parallel corpora.

Neural models. In 2014, seq2seq recurrent neural network (RNN) models introduced for machine translation providing an end-to-end translation framework (Sutskever et al., 2014). Attention was a key component to improve such models (Bahdanau et al., 2014; Luong et al., 2015). Two modifications to attention were proposed to improve the quality of underlying alignment and consequently, the quality of translation. (i) Model guided alignment training is introduced (Chen et al., 2016; Mi et al., 2016; Garg et al., 2019; Stengel-Eskin et al., 2019) where the cross-entropy between attention weights and the alignment coming from an IBM model (GIZA++) or a manual gold standard is used as an additional cost function. Garg et al. (2019) find that operating at the subword-level can be beneficial for alignment models. Note that they only consider a single subword segmentation. (ii) A disadvantage of neural architectures in comparison with IBM models in producing alignments is that in the neural model the attention weights have only observed the previous target words; in contrast, the IBM models benefit from full observation of the target sentence in alignment generation. *Target foresight* (Peter et al., 2017) improves translation by

	Alignment method	English-Romanian (50K)	English-Inuktitut (340K)	English-Persian (600K)
word-level	fast-align	0.643	0.794	0.552
	eflomal	0.692	0.864	0.58
Apply <English-German> parameters	fast-align	0.667	**0.915**	0.525
	eflomal	0.715	0.849	0.638
Apply <English-French> parameters	fast-align	0.664	0.913	0.534
	eflomal	0.709	0.885	0.647
Apply <English-Romanian> parameters	fast-align	0.663	0.873	0.534
	eflomal	0.712	0.826	0.587
Apply <English-Persian> parameters	fast-align	0.677	0.897	0.562
	eflomal	0.711	0.812	0.587
Apply <English-Hindi> parameters	fast-align	0.659	0.865	0.521
	eflomal	**0.719**	0.813	0.606
Apply <English-Inuktitut> parameters	fast-align	0.654	0.911	0.519
	eflomal	0.71	0.898	**0.65**

Table 3: The alignment performances (in terms of F1 score) of three language pairs in mid-resource scenario, where the subword sampling and word-level alignments are compared. In addition, the results on applying the hyper-parameters of language pairs on all other pairs are also provided. For each setting, both eflomal and fast-align results are reported.

considering the target word of the current decoding step as an additional input to the attention calculation. The main purpose of the above-mentioned alignment structures has been to improve translation quality. In contrast, our main motivation is providing a framework to reliably relate linguistic units, words, or subwords in parallel corpora, which can be used in linguistic resource creation (Agić et al., 2016; Asgari et al., 2020) and typological analysis (Östling, 2015; Asgari and Schütze, 2017). The above mentioned methods work well for the large parallel corpora, but they fail when parallel sentences are scarce. Insufficiency of parallel sentences is usually the case for low-resource languages, which are usually the most interesting scenarios for linguistic resource creation and linguistic analysis (Cieri et al., 2016).

Low-resource alignment models. The most popular dataset for low resource alignment is the Bible Parallel Corpus containing a large number (1000+) of languages, but are characteristically low-resource, i.e., have little text per language (Mayer and Cysouw, 2014). Some recent work touched upon this problem using unsupervised cross-lingual embeddings and a monogamy objective (Poerner et al., 2018). However, this method could not improve the fast-align results for the parallel corpora containing more than 250 sentences. We showed that our method improves the fast-align and eflomal on six language pairs consistently on the size of 5000K parallel sentences, in the range of parallel sentences of 1000+ languages in BPC, the most interesting parallel corpora for the low-resource scenario (in terms of the number of covered languages). Our proposed method improved the mid-resource alignments (50K-600K parallel sentences) as well.

Subword sampling. The use of multiple subword candidates has improved the machine translation performance (Kudo, 2018). BPE-Dropout (Provilkov et al., 2020) followed the same idea, introducing dropout in the merging steps of a fixed BPE to create multiple segmentations. The probabilistic use of multiple subword candidates has been proposed to segmentation protein sequences (Asgari et al., 2019a). We use the inspiration from the latter approach for the word-alignment of parallel sequences of language pairs, using a multitude of possible subword segmentations.

5 Conclusion

Motivated by the important NLP area of annotation projection, used to create linguistic resources/knowledge in the low-resource languages, we proposed subword sampling-based alignment of text units. This method's hypothesis is that the aggregation of different granularities of text for specific language pairs can help with word-level alignment. For individual languages where a gold-standard alignment corpus exists, we proposed an iterative Bayesian optimization framework to optimize selecting subwords from the space of possible BPE representations of the source and target sentences. We showed that the subword sampling method consistently outperforms the pure word-level alignment on six language pairs of English-German, English-French, English-Romanian, English-Persian, English-Hindi, and English-Inuktitut in a low-resource scenario. Although the subword samples are selected in a super-

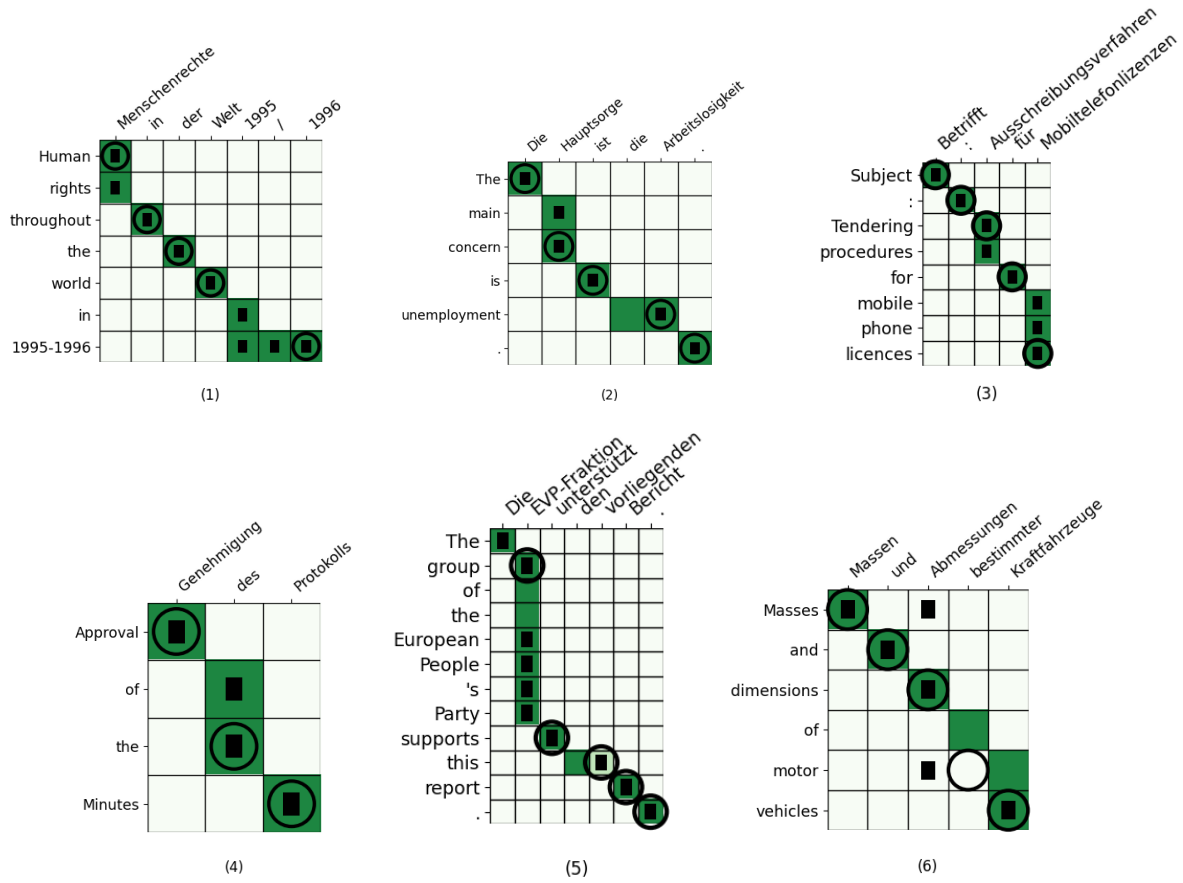

Figure 5: Examples from the English-German gold standard. Dark green represent sure alignment edges. Light green possible edges (only on edge in Example 5). Circles are edges predicted by word-level eflomal, boxes are predicted when applying our proposed subword sampling with eflomal.

vised manner, we show that the hyperparameters can fruitfully be used for other language pairs with no supervision and consistently improve the alignment results. We showed that using 5K parallel sentences together with our proposed subword sampling approach, we obtain similar F1 scores to the use of 340K and 600K parallel sentences and word-level alignment in English-Inuktitut and English-Persian, respectively. The proposed method can efficiently improve the creation of linguistic resources (POS tagging, sentiment lexicon, etc.) for low-resource languages, where only a few thousand parallel sentences are available.

Acknowledgment

This work has been funded by the German Federal Ministry of Education and Research (BMBF) under Grant No. 01IS18036A. The authors of this work take full responsibility for its content.

References

Željko Agić, Anders Johannsen, Barbara Plank, Héctor Alonso Martínez, Natalie Schluter, and Anders Søgaard. 2016. Multilingual projection for parsing truly low-resource languages. *Transactions of the Association for Computational Linguistics*, 4:301–312.

Ehsaneddin Asgari. 2019. *Life Language Processing: Deep Learning-based Language-agnostic Processing of Proteomics, Genomics/Metagenomics, and Human Languages*. Ph.D. thesis, UC Berkeley.

Ehsaneddin Asgari, Fabienne Braune, Benjamin Roth, Christoph Ringlstetter, and Mohammad Mofrad. 2020. UniSent: Universal adaptable sentiment lexica for 1000+ languages. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4113–4120, Marseille, France. European Language Resources Association.

Ehsaneddin Asgari, Alice C McHardy, and Mohammad RK Mofrad. 2019a. Probabilistic variable-length segmentation of protein sequences for discriminative motif discovery (dimotif) and sequence embedding (protvecx). *Scientific reports*, 9(1):1–16.

Ehsaneddin Asgari, Philipp C Münch, Till R Lesker, Alice C McHardy, and Mohammad RK Mofrad. 2019b. Ditaxa: Nucleotide-pair encoding of 16s rrna for host phenotype and biomarker detection.

Bioinformatics, 35(14):2498–2500.

Ehsaneddin Asgari and Hinrich Schütze. 2017. Past, present, future: A computational investigation of the typology of tense in 1000 languages. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 113–124, Copenhagen, Denmark. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Peter F Brown, Vincent J Della Pietra, Stephen A Della Pietra, and Robert L Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311.

Wenhu Chen, Evgeny Matusov, Shahram Khadivi, and Jan-Thorsten Peter. 2016. Guided alignment training for topic-aware neural machine translation. *arXiv preprint arXiv:1607.01628*.

Christopher Cieri, Mike Maxwell, Stephanie Strassel, and Jennifer Tracey. 2016. Selection criteria for low resource language programs. In *LREC*.

Mona Diab and Philip Resnik. 2002. An unsupervised method for word sense tagging using parallel corpora. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 255–262. Association for Computational Linguistics.

Chris Dyer, Victor Chahuneau, and Noah A Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. Association for Computational Linguistics.

Sarthak Garg, Stephan Peitz, Udhyakumar Nallasamy, and Matthias Paulik. 2019. Jointly learning to align and translate with transformer models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4453–4462, Hong Kong, China. Association for Computational Linguistics.

Ulrich Germann. 2001. Aligned Hansards of the 36th parliament of Canada.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Machine Translation Summit*, volume 5.

Philipp Koehn. 2010. *Statistical Machine Translation*. Cambridge University Press.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Thomas Mayer and Michael Cysouw. 2014. Creating a massively parallel bible corpus. *Oceania*, 135(273):40.

Anthony McEnery, Paul Baker, Rob Gaizauskas, and Hamish Cunningham. 2000. Emille: Building a corpus of South Asian languages. *VIVEK-BOMBAY-*, 13(3).

Haitao Mi, Zhiguo Wang, and Abe Ittycheriah. 2016. Supervised attentions for neural machine translation. *arXiv preprint arXiv:1608.00112*.

Franz Josef Och and Hermann Ney. 2000. Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 440–447, Hong Kong. Association for Computational Linguistics.

Robert Östling. 2015. Word order typology through multilingual word alignment. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 205–211.

Robert Östling and Jörg Tiedemann. 2016. Efficient word alignment with markov chain monte carlo. *The Prague Bulletin of Mathematical Linguistics*, 106(1):125–146.

Jan-Thorsten Peter, Arne Nix, and Hermann Ney. 2017. Generating alignments using target foresight in attention-based neural machine translation. *The Prague Bulletin of Mathematical Linguistics*, 108(1):27–36.

Mohammad Taher Pilevar, Heshaam Faili, and Abdol Hamid Pilevar. 2011. TEP: Tehran English-Persian parallel corpus. In *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer.

Nina Poerner, Masoud Jalili Sabet, Benjamin Roth, and Hinrich Schütze. 2018. Aligning very small parallel corpora using cross-lingual word embeddings and a monogamy objective. *arXiv preprint arXiv:1811.00066*.

Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. 2020. BPE-dropout: Simple and effective subword regularization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1882–1892, Online. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Elias Stengel-Eskin, Tzu-ray Su, Matt Post, and Benjamin Van Durme. 2019. A discriminative neural model for cross-lingual word alignment. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 910–920, Hong Kong, China. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Leila Tavakoli and Heshaam Faili. 2014. Phrase alignments in parallel corpus using bootstrapping approach. *International Journal of Information & Communication Technology Research*, 6(3).

David Yarowsky, Grace Ngai, and Richard Wicentowski. 2001. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of the first international conference on Human language technology research*, pages 1–8. Association for Computational Linguistics.

Bibliography

Željko Agić, Anders Johannsen, Barbara Plank, Héctor Martínez Alonso, Natalie Schluter, and Anders Søgaard. 2016. Multilingual projection for parsing truly low-resource languages. *Transactions of the Association for Computational Linguistics*, 4:301–312.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 789–798, Melbourne, Australia. Association for Computational Linguistics.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *CoRR*, abs/2004.05150.

Damián E. Blasi, Antonios Anastasopoulos, and Graham Neubig. 2021. Systematic inequalities in language technology performance across the world's languages. *CoRR*, abs/2110.06733.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language*

Processing, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

P. Brown, J. Cocke, S. Della Pietra, V. Della Pietra, F. Jelinek, R. Mercer, and P. Roossin. 1988a. A statistical approach to language translation. In *Coling Budapest 1988 Volume 1: International Conference on Computational Linguistics*.

P. Brown, John Cocke, Stephen Della Pietra, Vincent J. Della Pietra, Frederick Jelinek, Robert L. Mercer, and P. Roossin. 1988b. A statistical approach to French/English translation. In *Proceedings of the Second Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages*, Pittsburgh, USA.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar. Association for Computational Linguistics.

Jonathan H. Clark, Dan Garrette, Iulia Turc, and John Wieting. 2021. CANINE: pre-training an efficient tokenization-free encoder for language representation. *CoRR*, abs/2103.06874.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel P. Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12:2493–2537.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020a. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 7059–7069. Curran Associates, Inc.

BIBLIOGRAPHY

Alexis Conneau, Shijie Wu, Haoran Li, Luke Zettlemoyer, and Veselin Stoyanov. 2020b. Emerging cross-lingual structure in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6022–6034, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Zi-Yi Dou and Graham Neubig. 2021. Word alignment by fine-tuning embeddings on parallel corpora. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2112–2128, Online. Association for Computational Linguistics.

Philipp Dufter. 2021. *Distributed representations for multilingual language processing*. Ph.D. thesis, Ludwig-Maximilians-Universität München.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia. Association for Computational Linguistics.

David M. Eberhard, F. Simons Gary, and D. Fennig (eds.) Charles. 2020. *Ethnologue: Languages of the World*, 23rd edition. SIL International.

Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. 2015. Retrofitting word vectors to semantic lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1606–1615, Denver, Colorado. Association for Computational Linguistics.

Sarthak Garg, Stephan Peitz, Udhyakumar Nallasamy, and Matthias Paulik. 2019. Jointly learning to align and translate with transformer models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4453–4462, Hong Kong, China. Association for Computational Linguistics.

Daniela Gerz, Ivan Vulić, Felix Hill, Roi Reichart, and Anna Korhonen. 2016. SimVerb-3500: A large-scale evaluation set of verb similarity. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2173–2182, Austin, Texas. Association for Computational Linguistics.

Anna Gladkova, Aleksandr Drozd, and Satoshi Matsuoka. 2016. Analogy-based detection of morphological and semantic relations with word embeddings: what works and what doesn't. In *Proceedings of the NAACL Student Research Workshop*, pages 8–15, San Diego, California. Association for Computational Linguistics.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc.

Stephan Gouws, Yoshua Bengio, and Greg Corrado. 2015. Bilbowa: Fast bilingual distributed representations without word alignments. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 748–756, Lille, France. PMLR.

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. SimLex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.

Geoffrey E. Hinton, James L. McClelland, and David E. Rumelhart. 1990. Distributed representations. In *The Philosophy of Artificial Intelligence*, Oxford readings in philosophy, pages 248–280. Oxford University Press.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Valentin Hofmann, Janet Pierrehumbert, and Hinrich Schütze. 2021. Superbizarre is not superb: Derivational morphology improves BERT's interpretation of complex words. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3594–3608, Online. Association for Computational Linguistics.

Masoud Jalili Sabet, Philipp Dufter, François Yvon, and Hinrich Schütze. 2020. SimAlign: High quality word alignments without parallel training data using

static and contextualized embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1627–1643, Online. Association for Computational Linguistics.

Masoud Jalili Sabet, Matteo Negri, Marco Turchi, and Eduard Barbu. 2016. An unsupervised method for automatic translation memory cleaning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 287–292, Berlin, Germany. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Alexandre Klementiev, Ivan Titov, and Binod Bhattarai. 2012. Inducing crosslingual distributed representations of words. In *Proceedings of COLING 2012*, pages 1459–1474, Mumbai, India. The COLING 2012 Organizing Committee.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. *Proceedings of the 10th Machine Translation Summit (MT Summit), 2005*, pages 79–86.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. 2019. Set transformer: A framework for attention-based permutation-invariant neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3744–3753. PMLR.

Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2177–2185. Curran Associates, Inc.

Omer Levy, Anders Søgaard, and Yoav Goldberg. 2017. A strong baseline for learning cross-lingual word embeddings from sentence alignments. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 765–774, Valencia, Spain. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *Computing Research Repository*, arXiv:1907.11692.

Antonis Maronikolakis, Philipp Dufter, and Hinrich Schütze. 2021. Wine is not v i n. on the compatibility of tokenizations across languages. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2382–2399, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Thomas Mayer and Michael Cysouw. 2014. Creating a massively parallel Bible corpus. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3158–3163, Reykjavik, Iceland. European Language Resources Association (ELRA).

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Tomás Mikolov, Quoc V. Le, and Ilya Sutskever. 2013b. Exploiting similarities among languages for machine translation. *CoRR*, abs/1309.4168.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013c. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia. Association for Computational Linguistics.

Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. 2014. Efficient non-parametric estimation of multiple embeddings per word in vector space. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1059–1069, Doha, Qatar. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2000. Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 440–447, Hong Kong. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Robert Östling and Jörg Tiedemann. 2016. Efficient word alignment with Markov Chain Monte Carlo. *The Prague Bulletin of Mathematical Linguistics*, 106(1).

Chanjun Park, Sugyeong Eo, Hyeonseok Moon, and Heuiseok Lim. 2021. Should we find another model?: Improving neural machine translation performance with ONE-piece tokenization method without model modification. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Papers*, pages 97–104, Online. Association for Computational Linguistics.

Peyman Passban, Qun Liu, and Andy Way. 2018. Improving character-based decoding using target-side morphological information for neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 58–68, New Orleans, Louisiana. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew E. Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. 2017. Semi-supervised sequence tagging with bidirectional language models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1756–1765, Vancouver, Canada. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural*

Language Processing (EMNLP-IJCNLP), pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Uma Roy, Noah Constant, Rami Al-Rfou, Aditya Barua, Aaron Phillips, and Yinfei Yang. 2020. LAReQA: Language-agnostic answer retrieval from a multilingual pool. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5919–5930, Online. Association for Computational Linguistics.

Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. 2021. How good is your tokenizer? on the monolingual performance of multilingual language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3118–3135, Online. Association for Computational Linguistics.

Elizabeth Salesky, David Etter, and Matt Post. 2021. Robust open-vocabulary translation from visual text representations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7235–7252, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2009. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80.

Timo Schick and Hinrich Schütze. 2019. Learning semantic representations for novel words: Leveraging both form and context. AAAI'19/IAAI'19/EAAI'19. AAAI Press.

Peter H Schönemann. 1966. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and Korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2012, Kyoto, Japan, March 25-30, 2012*, pages 5149–5152. IEEE.

Hinrich Schütze. 1992. Dimensions of meaning. In *Proceedings Supercomputing '92, Minneapolis, MN, USA, November 16-20, 1992*, pages 787–796. IEEE Computer Society.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Anders Søgaard, Željko Agić, Héctor Martínez Alonso, Barbara Plank, Bernd Bohnet, and Anders Johannsen. 2015. Inverted indexing for cross-lingual NLP. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1713–1722, Beijing, China. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929–1958.

Steinþór Steingrímsson, Hrafn Loftsson, and Andy Way. 2021. CombAlign: a tool for obtaining high-quality word alignments. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 64–73, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Yulia Tsvetkov, Manaal Faruqui, Wang Ling, Guillaume Lample, and Chris Dyer. 2015. Evaluation of word vector representations by subspace alignment. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2049–2054, Lisbon, Portugal. Association for Computational Linguistics.

Alan Turing. 1950. Computing machinery and intelligence. *Mind*, LIX(236):433–460.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you

need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Ivan Vulić and Marie-Francine Moens. 2015. Bilingual word embeddings from non-parallel document-aligned data applied to bilingual lexicon induction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 719–725, Beijing, China. Association for Computational Linguistics.

Ivan Vulić, Sebastian Ruder, and Anders Søgaard. 2020. Are all good word vector spaces isomorphic? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3178–3192, Online. Association for Computational Linguistics.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, volume 32, pages 3266–3280. Curran Associates, Inc.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Chao Xing, Dong Wang, Chao Liu, and Yiye Lin. 2015. Normalized word embedding and orthogonal transform for bilingual word translation. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1006–1011, Denver, Colorado. Association for Computational Linguistics.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2021. Byt5: Towards a token-free future with pre-trained byte-to-byte models. *CoRR*, abs/2105.13626.

Thomas Zenkel, Joern Wuebker, and John DeNero. 2020. End-to-end neural word alignment outperforms GIZA++. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1605–1617, Online. Association for Computational Linguistics.