Data-Efficient Methods for Information Extraction

Inaugural-Dissertation
zur Erlangung des Doktorgrades der Philosophie
an der Ludwig–Maximilians–Universität
München

vorgelegt von
Usama Yaseen
aus Jhelum, Pakistan

2023

Erstgutachter: PD Dr. Stefan Langer
Zweitgutachter: Prof. Dr. Klaus U. Schulz
Tag der mündlichen Prüfung: 15.02.2023

Contents

Zusammenfassung

Strukturierte Wissensrepräsentationssysteme wie Wissensdatenbanken oder Wissensgraphen bieten Einblicke in Entitäten und Beziehungen zwischen diesen Entitäten in der realen Welt. Solche Wissensrepräsentationssysteme können in verschiedenen Anwendungen der natürlichen Sprachverarbeitung eingesetzt werden, z. B. bei der semantischen Suche, der Beantwortung von Fragen und der Textzusammenfassung. Es ist nicht praktikabel und ineffizient, diese Wissensrepräsentationssysteme manuell zu befüllen. In dieser Arbeit entwickeln wir Methoden, um automatisch benannte Entitäten und Beziehungen zwischen den Entitäten aus Klartext zu extrahieren. Unsere Methoden können daher verwendet werden, um entweder die bestehenden unvollständigen Wissensrepräsentationssysteme zu vervollständigen oder ein neues strukturiertes Wissensrepräsentationssystem von Grund auf zu erstellen. Im Gegensatz zu den gängigen überwachten Methoden zur Informationsextraktion konzentrieren sich unsere Methoden auf das Szenario mit wenigen Daten und erfordern keine große Menge an kommentierten Daten.

Im ersten Teil der Arbeit haben wir uns auf das Problem der Erkennung von benannten Entitäten konzentriert. Wir haben an der gemeinsamen Aufgabe von *Bacteria Biotope 2019* teilgenommen. Die gemeinsame Aufgabe besteht darin, biomedizinische Entitätserwähnungen zu erkennen und zu normalisieren. Unser *linguistically informed* Named-Entity-Recognition-System besteht aus einem Deep-Learning-basierten Modell, das sowohl verschachtelte als auch flache Entitäten extrahieren kann; unser Modell verwendet mehrere linguistische Merkmale und zusätzliche Trainingsziele, um effizientes Lernen in datenarmen Szenarien zu ermöglichen. Unser System zur Entitätsnormalisierung verwendet String-Match, Fuzzy-Suche und semantische Suche, um die extrahierten benannten Entitäten mit den biomedizinischen Datenbanken zu verknüpfen. Unser System zur Erkennung von benannten Entitäten und zur Entitätsnormalisierung erreichte die niedrigste Slot-Fehlerrate von $0,715$ und belegte den ersten Platz in der gemeinsamen Aufgabe. Wir haben auch an zwei gemeinsamen Aufgaben teilgenommen: *Adverse Drug Effect Span Detection* (Englisch) und *Profession Span Detection* (Spanisch); beide Aufgaben sammeln Daten von der Social Media Plattform Twitter. Wir haben ein Named-Entity-Recognition-Modell entwickelt, das die Eingabedarstellung des Modells durch das Stapeln heterogener Einbettungen aus verschiedenen Domänen verbessern kann; unsere empirischen Ergebnisse zeigen komplementäres Lernen aus diesen heterogenen Einbettungen. Unser Beitrag belegte den 3. Platz in den beiden gemeinsamen Aufgaben.

Im zweiten Teil der Arbeit untersuchten wir Strategien zur Erweiterung synthetis-

cher Daten, um ressourcenarme Informationsextraktion in spezialisierten Domänen zu ermöglichen. Insbesondere haben wir *backtranslation* an die Aufgabe der Erkennung von benannten Entitäten auf Token-Ebene und der Extraktion von Beziehungen auf Satzebene angepasst. Wir zeigen, dass die Rückübersetzung sprachlich vielfältige und grammatikalisch kohärente synthetische Sätze erzeugen kann und als wettbewerbsfähige Erweiterungsstrategie für die Aufgaben der Erkennung von benannten Entitäten und der Extraktion von Beziehungen dient.

Bei den meisten realen Aufgaben zur Extraktion von Beziehungen stehen keine kommentierten Daten zur Verfügung, jedoch ist häufig ein großer unkommentierter Textkorpus vorhanden. Bootstrapping-Methoden zur Beziehungsextraktion können mit diesem großen Korpus arbeiten, da sie nur eine Handvoll Startinstanzen benötigen. Bootstrapping-Methoden neigen jedoch dazu, im Laufe der Zeit Rauschen zu akkumulieren (bekannt als semantische Drift), und dieses Phänomen hat einen drastischen negativen Einfluss auf die endgültige Genauigkeit der Extraktionen. Wir entwickeln zwei Methoden zur Einschränkung des Bootstrapping-Prozesses, um die semantische Drift bei der Extraktion von Beziehungen zu minimieren. Unsere Methoden nutzen die Graphentheorie und vortrainierte Sprachmodelle, um verrauschte Extraktionsmuster explizit zu identifizieren und zu entfernen. Wir berichten über die experimentellen Ergebnisse auf dem TACRED-Datensatz für vier Relationen.

Im letzten Teil der Arbeit demonstrieren wir die Anwendung der Domänenanpassung auf die anspruchsvolle Aufgabe der mehrsprachigen Akronymextraktion. Unsere Experimente zeigen, dass die Domänenanpassung die Akronymextraktion in wissenschaftlichen und juristischen Bereichen in sechs Sprachen verbessern kann, darunter auch Sprachen mit geringen Ressourcen wie Persisch und Vietnamesisch.

Abstract

The structured knowledge representation systems such as knowledge base or knowledge graph can provide insights regarding entities and relationship(s) among these entities in the real-world, such knowledge representation systems can be employed in various natural language processing applications such as semantic search, question answering and text summarization. It is infeasible and inefficient to manually populate these knowledge representation systems. In this work, we develop methods to automatically extract named entities and relationships among the entities from plain text and hence our methods can be used to either complete the existing incomplete knowledge representation systems to create a new structured knowledge representation system from scratch. Unlike mainstream supervised methods for information extraction, our methods focus on the *low-data* scenario and do not require a large amount of annotated data.

In the first part of the thesis, we focused on the problem of named entity recognition. We participated in the shared task of *Bacteria Biotope 2019*, the shared task consists of recognizing and normalizing the biomedical entity mentions. Our *linguistically informed* named entity recognition system consists of a deep learning based model which can extract both nested and flat entities; our model employed several linguistic features and auxiliary training objectives to enable efficient learning in data-scarce scenarios. Our entity normalization system employed string match, fuzzy search and semantic search to link the extracted named entities to the biomedical databases. Our named entity recognition and entity normalization system achieved the lowest slot error rate of 0.715 and ranked first in the shared task. We also participated in two shared tasks of *Adverse Drug Effect Span detection* (English) and *Profession Span Detection* (Spanish); both of these tasks collect data from the social media platform Twitter. We developed a named entity recognition model which can improve the input representation of the model by stacking heterogeneous embeddings from a diverse domain(s); our empirical results demonstrate complementary learning from these heterogeneous embeddings. Our submission ranked 3rd in both of the shared tasks.

In the second part of the thesis, we explored synthetic data augmentation strategies to address low-resource information extraction in specialized domains. Specifically, we adapted *backtranslation* to the token-level task of named entity recognition and sentence-level task of relation extraction. We demonstrate that backtranslation can generate linguistically diverse and grammatically coherent synthetic sentences and serve as a competitive augmentation strategy for the task of named entity recognition and relation extraction.

In most of the real-world relation extraction tasks, the annotated data is not available, however, quite often a large unannotated text corpus is available. Bootstrapping methods for relation extraction can operate on this large corpus as they only require a handful of seed instances. However, bootstrapping methods tend to accumulate noise over time (known as *semantic drift*) and this phenomenon has a drastic negative impact on the final precision of the extractions. We develop two methods to constrain the bootstrapping process to minimise semantic drift for relation extraction; our methods leverage graph theory and pre-trained language models to explicitly identify and remove noisy extraction patterns. We report the experimental results on the TACRED dataset for four relations.

In the last part of the thesis, we demonstrate the application of domain adaptation to the challenging task of multi-lingual acronym extraction. Our experiments demonstrate that domain adaptation can improve acronym extraction within scientific and legal domains in 6 languages including low-resource languages such as Persian and Vietnamese.

Chapter 1

Introduction

1.1 Motivation

In the past few decades, the amount of unstructured text has grown tremendously across scientific and industrial domains; this phenomenon of information growth was most prominent in the general domain i.e. on the *internet*. To extract *knowledge* from the unstructured plain text, it must first be converted into a structured representation. Structured knowledge representation often takes the form of a Database (DB), Knowledge Base (KB) or Knowledge Graph (KG). Once such structured knowledge representation is available, the insights hidden in the unstructured, raw and plain text can be revealed by a simple query or a lookup. The structured knowledge representation systems have enormous applications in various Natural Language Processing (NLP) systems such as semantic search, question answering and virtual assistants.

The knowledge graph is one of the widely used structured representation systems for encoding knowledge about a domain. Typically, the KG represents entities which occur in the domain in a graph structure such that each entity is a node in the graph and edges between entities signify the relationship between the connected entities. Named Entity Recognition (NER) refers to the task of identifying entities in the text and is often the primitive component of an Information Extraction (IE) pipeline. NER is often followed by the step of Entity Normalization (entity linking) to resolve ambiguous entities by mapping entities to a list of known entities based on the context. The final step of the IE pipeline is to identify relations among the extracted entities, referred as Relation Extraction (RE). The KG can be populated with the extracted normalized entities and the semantic relationship between the entities to complete the conversion of unstructured plain text to a structured knowledge representation.

Most recently, Deep Learning (DL) based methods for natural language processing have achieved state-of-the-art performance on various information extraction tasks including named entity recognition, entity linking and relation extraction. One of the caveats of deep learning based NLP methods is that they rely on the availability of large datasets to achieve this improved performance and avoid overfitting. However, in many real-world settings, it

is not feasible to collect large annotated datasets, especially for specialized domains such as the material science, biomedical, legal or financial domain etc., where annotating data requires expert knowledge and is usually time-consuming and expensive. Due to the lack of sufficient annotated datasets, rule-based systems dominate industrial information extraction technologies (Chiticariu et al., 2013a).

In this thesis, we focus on different components of information extraction pipeline including (nested) named entity recognition, entity linking and relation extraction. One of the central focus of this thesis is *data efficient learning* and we propose methods to circumvent reliance of existing deep learning based methods for information extraction on large annotated datasets.

1.2 Main Contributions

In this thesis, we contribute to the state of the art data efficient methods for information extraction research as described below.

Named Entity Recognition. We develop state-of-the-art NER systems for various domains. Our system was ranked first in the official shared task evaluations 2019. Our system also addressed nested named entity recognition which is often ignored by the mainstream NER systems; where we define a nested entity as an entity or sub-concept which is part of a longer entity (i.e., a parent). In particular, our system consists of two BiDirectional Long Short Term Memory Networks (LSTMS) with Conditional Random Fields (CRF) as output layer to detect parent and nested/child entities. We further improve our NER system by incorporating *stacked heterogeneous embeddings* to enhance vector representation of words; our NER system demonstrates competitive performance on several shared task evaluations across multiple domains and languages. To address *low-resource scenarios* (i.e. when not enough annotated training data is available), we develop data augmentation methods for NER based on *backtranslation* techniques to automatically generate linguistically diverse and coherent data for the underlying deep learning based NER models. Our experiments demonstrate that proposed data augmentation methods can significantly improve performance in low-resource scenarios by improving the generalization capabilities of the underlying NER model.

Entity Normalization. We develop methods based on exact, fuzzy and semantic search to resolve ambiguities of the NER system by aligning noisy predicted entities to a list of pre-defined entities defined in the database. Our entity normalization system ranked first in the official shared task evaluation 2019.

Relation Extraction. We develop a state-of-the-art RE method based on linguistically informed features with Support Vector Machines (SVM) as a classifier. Our system ranked first in the official shared task evaluation of 2019. To address low-resource scenarios in domain-specific RE, we develop data augmentation methods using *backtranslation* to

automatically generate linguistically diverse and coherent entity mention contexts, enabling RE models to learn from richer semantic contexts and thus improving the overall generalization capabilities of the model. In cases when no annotated data is available, we develop semi-supervised learning methods based on Bootstrapping to automatically identify semantic relationships between entity pairs using only a handful of seed instances (a seed instance refers to an entity pair expressing a particular relationship).

Domain Adaptation. Most recently, pre-trained language models have shown state-of-the-art performance in several NLP tasks as these models have been trained on a huge corpus (typically obtained from the internet). However, they tend to perform poorly in domains which differ significantly from the pre-training corpus i.e. general domain. We explored Domain Adaptation (DA) methods to adapt pre-trained embeddings to the domain-specific embeddings to achieve optimal performance on domain-specific problems; to demonstrate the effectiveness of domain adaptation methods we report results on the task of Acronym Extraction (AE) for the scientific and legal domains.

1.3 Structure

The rest of this thesis is structured as follows:

Chapter 2 covers the background material for the remaining chapters of this thesis. Section 2.1 argues the need of structured knowledge representation and knowledge graph in general. Section 2.2 gives an overview of the information extraction pipeline; section 2.3 further narrows the focus of the information extraction methods in low-resource (limited data) scenarios. Section 2.4 provides an overview of the various neural network architectures.

Chapter 3 presents our work on the task of named entity recognition. Section 3.1 describes the architecture of our nested named entity recognition and entity linking model, along with a detailed discussion on the experimental setup, results and analysis. Section 3.2 describes the architecture of our NER model which employs a stack of heterogeneous embeddings to exploit complementary representations from various language models from general and specific domains. Our NER model with stacked heterogeneous embeddings achieves competitive performance across three datasets in English and Spanish (section 3.2.5).

Chapter 4 describes our work on low-resource domain-specific NER, where we explored synthetic data augmentation to deal with data scarcity. Specifically, our method uses backtranslation to generate linguistically diverse and coherent augmentation instances, we also analyse and compare our method with the existing rule-based data augmentation strategies for NER. Section 4.4.3 discuss the experimental setup with results and analysis on two domain-specific datasets for NER.

Chapter 5 details our work on the semi-supervised relation extraction to perform RE in scenarios when no labelled data is available. In particular, we addressed the problem of *semantic drift* in the bootstrapping process, the semantic drift refers to the inclusion of noise during the bootstrapping iterations, this added noise has a snowball effect over the

bootstrapping lifecycle and greatly reduces the precision of the bootstrapping system. In section 5.3, we proposed two methods to constrain the bootstrapping system to mitigate noise and thus steer the bootstrapping system away from the semantic drift. Section 5.4 describe the datasets, comparison of constrained and unconstrained bootstrapping and analysis of our work.

In Chapter 6, we describe our experiments on low-resource domain-specific relation extraction. In particular, we employ *backtranslation* using multiple pivot languages to create diverse linguistic variations of the training instances (see section 6.3). Section 6.4.3 describes the evaluation setup including the datasets (section 6.4.1), supervised RE model (section 6.4.2) and the results (section 6.4.3).

In Chapter 7, we demonstrate the effectiveness of *domain adaptation* on the complex multi-lingual acronym extraction task. Specifically, we perform domain adaptation using the multi-lingual *XLM-RoBERTa* to steer the XLM-RoBERTa's representation from the general domain to the specific scientific and legal domains (section 7.3.1). We frame the task of acronyms and their corresponding long-form extraction as a sequence labelling problem (section 7.2) and report the results of our experiments on six diverse languages including the low-resource Persian and Vietnamese. The description of the dataset and the evaluation results are reported in section 7.4.

Chapter 8 concludes the thesis with a summary of the data-efficient information extraction and directions for future work.

Chapter 2

Foundations and Background

This chapter provides an overview of the relevant background for this thesis. The first section motivates the need for structured knowledge representation and knowledge base population. The second section describes the information extraction pipeline i.e. named entity recognition, entity linking and relation extraction. The third section elucidates the information extraction in data-scarce scenarios with a special focus on transfer learning, distant supervision, synthetic data generation and domain adaptation. The fourth section provides an overview of various neural architectures.

2.1 Structured Knowledge Representation

The past decade has seen the rise of "Big Data" and the implication of this phenomenon was also observed on the unstructured text across the scientific, industrial and public (the internet) domains. To extract *knowledge* and *insights* from this tremendous amount of unstructured plain text, it must first be converted into a structured representation. The structured knowledge representation often takes the form of a Database (DB), Knowledge Base (KB) or a Knowledge Graph (KG). This structured representation can reveal the hidden insights in the plain text with a simple query or a lookup. These structured knowledge representation systems play an important part in several natural language processing tasks across all the domains and hence have attracted immense research efforts to develop and populate these knowledge representation systems.

The traditional approaches to (structured) data management systems employ (relational) databases; the relational databases got very popular as they define this simpler and intuitive "tabular" approach to organizing and accessing data. Relational databases use *tables* as the fundamental block to model and represent data; data is placed into predefined categories in a series of tables. Each table consists of columns and rows, where columns refer to the data category and rows refer to the data instances for the data categories. The data can be created, accessed, managed, modified and deleted using the structured query language (SQL). As the need for gathering and generating insights from data intensified, it was realised that real-world knowledge is situational (depends on a situation), inter-

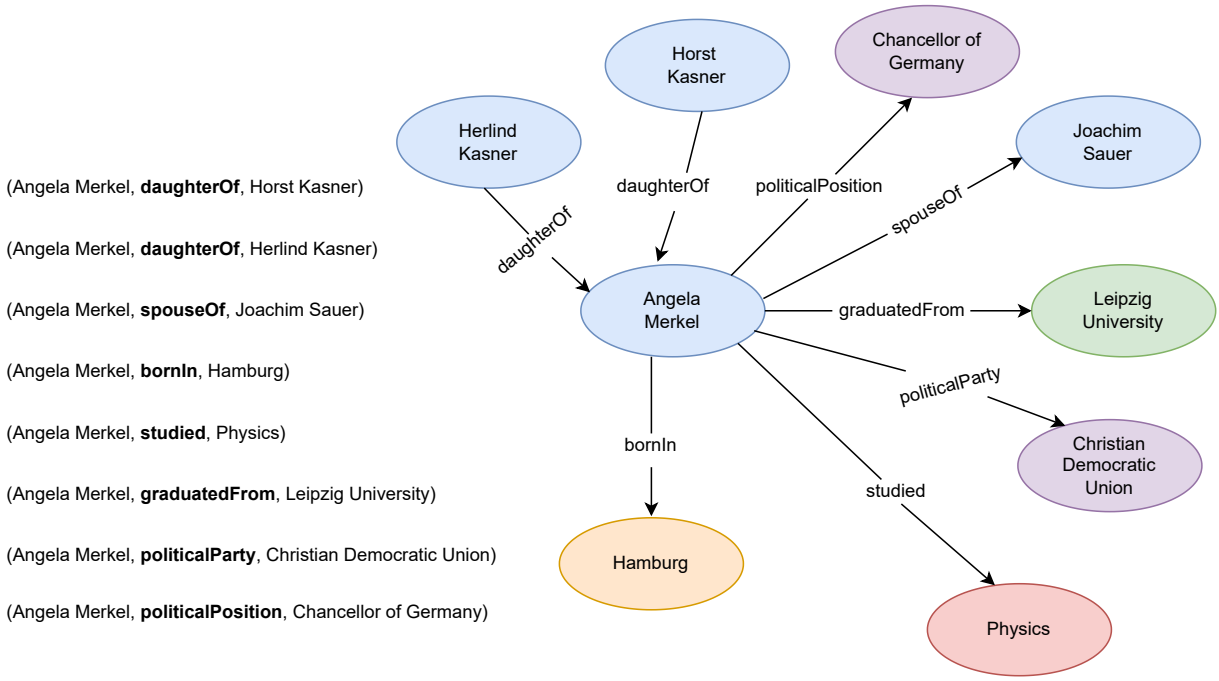
connected (association between concepts and events) and dynamic (concepts evolve and change meaning) (Natarajan). These aspects of knowledge represent the "context" which is often missing from the (raw) data itself and traditional data management systems failed to capture this context.

The aforementioned limitations of databases lead to the development of *knowledge graphs*, a structured representation of facts: defined by entities, relationships and semantic descriptions. The entities can be real-world objects or abstract concepts, relationships represent the (semantic) relation between entities, and semantic descriptions of entities and their relationships contain types and properties with a well-defined meaning(Ji et al., 2022). The term knowledge graph is synonymous with the knowledge base with a minor difference that a knowledge graph can be viewed as as a graph when considering its graph structure (Stokman and Vries, 1988) but when it involves formal semantics, it can be taken as a knowledge base for interpretation and inference over facts (Bordes et al., 2011; Ji et al., 2022). Figure 2.1 illustrates an example of a knowledge graph and a knowledge base. The resource description framework (RDF) enables facts to be expressed in the form of (head, *relation*, tail) or (subject, predicate, object), for example, (Angela Merkel, *born in*, Hamburg). This factual information can be also expressed using a directed graph, such that the nodes represent the entities and the edges represent the relationship between the entities. In this thesis, we use the terms knowledge graph and knowledge base interchangeably.

The prominent large-scale knowledge bases include Freebase (Bollacker et al., 2008), Wikidata (Tanon et al., 2016), DBpedia (Lehmann et al., 2015), YAGO (Suchanek et al., 2007) and the Google Knowledge Graph (Singhal, 2012). The DBpedia and YAGO automatically extract the facts from Wikipedia, however, Freebase and Wikidata rely on a manual and collaborative effort. The Google knowledge graph has been developed based on the information stored in Freebase, Wikipedia and the CIA World Factbook; the Google knowledge graph has been augmented at a large scale with 500 million entities and 3.5 billion facts about and relationships between them (Singhal, 2012). The data dumps are available for download for Freebase, Wikidata, DBpedia and YAGO, however, the Google Knowledge Graph only provides a search API for accessing its information. Table 2.1 reports the statistics about the information stored in different knowledge bases. Tanon et al. note that the number of entities (items, instances), relation instances (facts) or labels (properties) is not directly comparable since various knowledge bases have different criteria for which entity they store and different ways to handle inverse relations.

	Freebase	Wikidata	YAGO2	DBpedia (en)
# entities	48M	14.5M	9.8M	4.6M
# facts	2997M	66M	447.5M	152.9M
# labels	68M	82M	365.5K	61.8K

Table 2.1: Statistics of different knowledge bases.



(a) Facts (triples) in knowledge base. (b) Entities and relations in a knowledge graph.

Figure 2.1: An example of knowledge base and knowledge graph.

2.1.1 Knowledge Base Population

In spite of a large number of facts stored in knowledge bases (see Table 2.1), the knowledge bases are still incomplete. According to Min et al., 93.8% of persons in Freebase have no place of birth, 98.8% have no parents, 96.6% have no places of residence and 78.5% have no nationality. West et al. report that 99% of persons have no ethnicity information in Freebase. It is impractical to manually complete a knowledge base as it will be extremely expensive and slow. Hence, the natural language processing community have dedicated research efforts to develop automatic methods to enable creating a knowledge base from scratch or filling up missing information into an existing knowledge base. The two dominant trends in this respect include: extending existing knowledge bases by reasoning over them, inferring missing links, and extracting new structured information from the unstructured text. The latter is usually referred to as knowledge base population (KBP) (Glass and Gliozzo, 2018). This thesis focus on the knowledge base population, we will now describe the complete information extraction pipeline in detail.

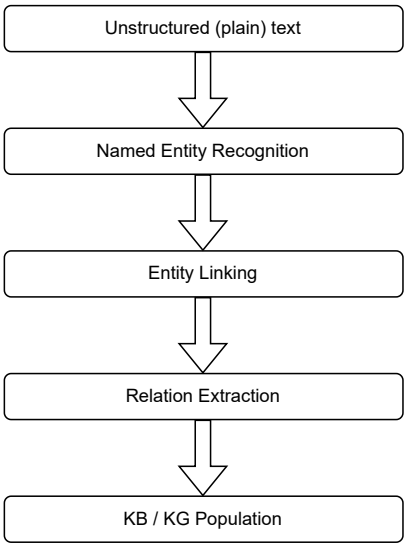


Figure 2.2: A standard information extraction pipeline to convert unstructured text into a structured representation. Here, KB refers to knowledge base and KG refers to Knowledge Graph.

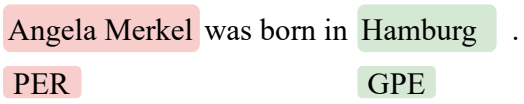


Figure 2.3: An example of named entity recognition in the general domain.

2.2 Information Extraction Pipeline

Figure 2.2 illustrates the standard information extraction pipeline with steps to transform the unstructured plain text into a structured representation. Now, we will formalise each step of the information extraction pipeline and briefly describe the mainstream methods for each step.

2.2.1 Named Entity Recognition

Named entity recognition is a critical primitive step in several NLP pipelines (including the information extraction pipeline), as the subsequent tasks such as relation extraction (Lin et al., 2016; Zhang et al., 2017a), question answering (Ji and Grishman, 2011; Xu et al., 2016a) etc., depend on it. The task of named entity recognition aims to detect *named entities*[1] of interest mentioned in the plain text. Informally, a named entity is defined as a real-world object with an abstract or physical existence. The definition of named entities may vary across the domain(s), in the case of general or public domain the typical example of entities include names of a person, cities, companies, etc., (see the example in figure

[1]In this work we use the term entity and named entity interchangeably.

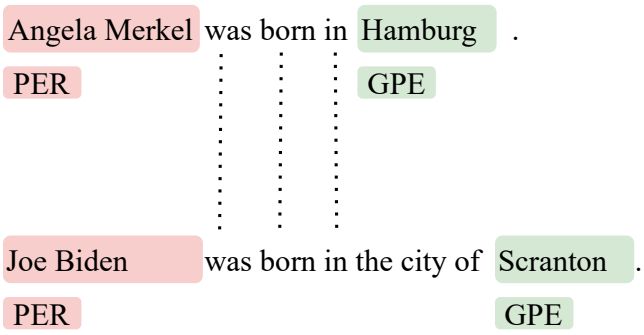


Figure 2.4: An illustration of similar semantic contexts signalling the occurrence of an entity mention. Here, PER refers to Person and GPE refers to Geopolitical entity (cities, countries, etc.,).

2.3). A named entity can be a single word or a multi-word expression[2], can be denoted with a proper noun, and is usually specified with an entity type. The task of named entity recognition consists of two steps: a) named entity detection b) named entity classification. The *named entity detection* aims to identify the sequence of one or more words which belong to an entity but without specifying the entity type; the task of *named entity classification* assigns a category to the detected named entity. Most modern NER methods perform both of these steps together without any explicit distinction.

The task of named entity recognition has both the *syntactic* and *semantic* aspects. The syntactic aspect refers to the fact that an entity mention is a named entity even without any context, the sequence of words "Angela Merkel" or "Joe Biden" refers to the person even without any textual context. On the other hand, similar semantic contexts in the text signal the occurrence of an entity mention. Consider the two sentences in the figure 2.4, the textual contexts ("was born in" and "was born in the city of") around the entities of type person and GPE in both of these two sentences is very similar semantically. The underlying NER model should exploit this semantic similarity and learn that if this specific context appears in a sentence then before this context there is an entity mention of type person and after this context, there is an entity mention of type geopolitical entity. The syntactic aspect can be captured by (shallow) word features such as part-of-speech tags, orthographic, capitalization, etc., whereas, understanding the semantic aspect requires capturing (semantic) features such as dependency parse tree, word representations (embeddings), etc.

The approaches to address the task of NER can be classified as follows (Yadav and Bethard, 2018):

Knowledge-based systems: These methods rely on existing lexicon resources, domain knowledge and manually curated dictionaries and therefore, do not require annotated training data (Zamin and Oxley, 2011). These methods only work for domains where the linguistic resources are available; thus, it requires domain experts to construct and maintain

[2]The discontinuous entities are the exception, where an entity is not a continuous sequence of words but rather entity tokens are spread across the text.

these linguistic resources. The knowledge-based systems tend to have high precision but low recall, as linguistic resources are often not extensive and therefore, incomplete.

Unsupervised and bootstrapped systems: These methods rely on shallow syntactic knowledge and can be used to construct a gazetteer. Collins and Singer use a handful of labelled seeds, orthographic features, and context of entity words to extract and classify named entities. Etzioni et al. use 8 generic patterns to extract named entities from the Web. Zhang and Elhadad employed syntactic knowledge (noun phrase chunking) and corpus statistics (inverse document frequency vectors) to extract named entities from the biomedical text.

Feature-engineered supervised systems: These methods use the supervised machine learning algorithms to replace the rule-based systems, manually defined rules are replaced with manually selected features; the typical features include orthographic features, trigger words for named entities, a list of words in the gazetteers (if available), capitalisation features etc. The commonly employed machine learning algorithms include Hidden Markov Models (HMM) (Rabiner and Juang, 1986), Support Vector Machines (SVM) (Cortes and Vapnik, 1995), Conditional Random Fields (CRF) (Lafferty et al., 2001b), and decision trees (Breiman et al., 1984). Unsurprisingly, the performance of these methods depends on the quality of selected features (Zhou and Su, 2002; Malouf, 2002; Carreras et al., 2002; Li et al., 2004; Liu et al., 2015).

Deep learning based systems: The modern deep learning based methods can automatically construct abstract features from the textual input, these methods typically represent words in the input text as either words, characters, sub-words or any combination of these. Collobert and Weston developed a neural network architecture based on the convolution and max-pooling layer, the neural model is simultaneously jointly trained on six tasks including part-of-speech tagging, chunking, NER, semantic role labelling, semantically similar words and language modelling in a *multitask learning* setup. Huang et al. employed word-level Bidirectional LSTM with CRF layer to perform NER. Kim et al. employed LSTM to perform NER but with an added twist that word representation input to LSTM is being computed using a convolutional neural network, the whole model is trained end to end. Gillick et al. frame the task of NER as sequence to sequence such that an LSTM (encoder) reads text as bytes and the second LSTM (decoder) directly outputs span annotations in the input text; one of the advantages of using bytes as text representation is the manageable vocabulary size which enables smoothly extending the NER model to multiple languages. Ma and Hovy employ word-level Bidirectional LSTM with CRF layer to perform NER; the input to the BiLSTM is the concatenation of word embeddings and the character embeddings which are computed using a CNN. The architecture of Lample et al. is very similar to that of Ma and Hovy with the only difference being that the character representations are computed using a bidirectional LSTM instead of a CNN.

2.2.2 Entity Linking

Entity Linking or Entity Disambiguation refers to the task of linking an entity mention to a unique entry in the database or a knowledge base. Entity linking is crucial to resolve

the lexical ambiguity of entity mentions and determines their meaning in context (Sevgili et al., 2022). For example, "Müller" may refer to a German retail store company or the name of the person "Thomas Müller" depending on the context. Most of the traditional entity linking systems consist of two components: a *candidate generation module* and a *mention disambiguation module*. The candidate generation module takes the ambiguous entity mention as the input and returns a list of possible suitable named entities that are plausible in the given textual context, this list is usually selected using a knowledge source such as Wikipedia[3]. The mention disambiguation module finds the most relevant entity mention out of the list of possible named entities by employing manually designed features such as entity popularity, local context compatibility, and document-level global coherence of referring entities (Shen et al., 2021). The traditional entity linking systems suffer from two problems: i) it is cumbersome to select robust features for the mention disambiguation module ii) the domain-specific knowledge base and manual feature selection prohibit the generalizing capabilities of a trained entity linking model to other knowledge base or domains (Kulkarni et al., 2009; Ratinov et al., 2011; Shen et al., 2012; Guo et al., 2013). The recent deep learning based methods for entity linking promise a remedy for these problems, as these methods can automatically learn robust features which can be transferred across the domains (Zwicklbauer et al., 2016; Francis-Landau et al., 2016; Gupta et al., 2017; Mueller and Durrett, 2018; Gillick et al., 2019). At the base level, the input features include various kinds of pre-trained or learned embeddings such as word embeddings, entity embeddings, and context embeddings.

2.2.3 Relation Extraction

Relation Extraction is a fundamental step of the information extraction pipeline and is crucial to enabling the transformation of unstructured text into a structured representation such as a knowledge graph or a knowledge base. Relation Extraction aims to find the semantic relationships among entities in a textual context, the most common textual context for relation extraction is a sentence but it can also be a paragraph or a document. It is important to note that a relationship can exist between any number of entities, an entity can have a relation with itself (unary or 1-ary), two entities can participate in a relation (binary or 2-ary), three entities participate in a relation (ternary or 3-ary) and so on. Also, the literature on relation extraction makes a distinction based on the localisation of entities i.e. if the textual context of the participating entities is a sentence or a paragraph/document. If the participating entities for a relationship do not occur in the same sentence then this is referred to as *cross-sentence relation extraction*. The most common setting for performing relation extraction is *multi-class binary relation extraction at sentence-level*, we also follow this setting for our work on relation extraction in this thesis.

Since the relation extraction step comes after the named entity recognition step in the information extraction pipeline (see figure 2.2), the RE systems assume that named entities

[3] `https://www.wikipedia.org/`

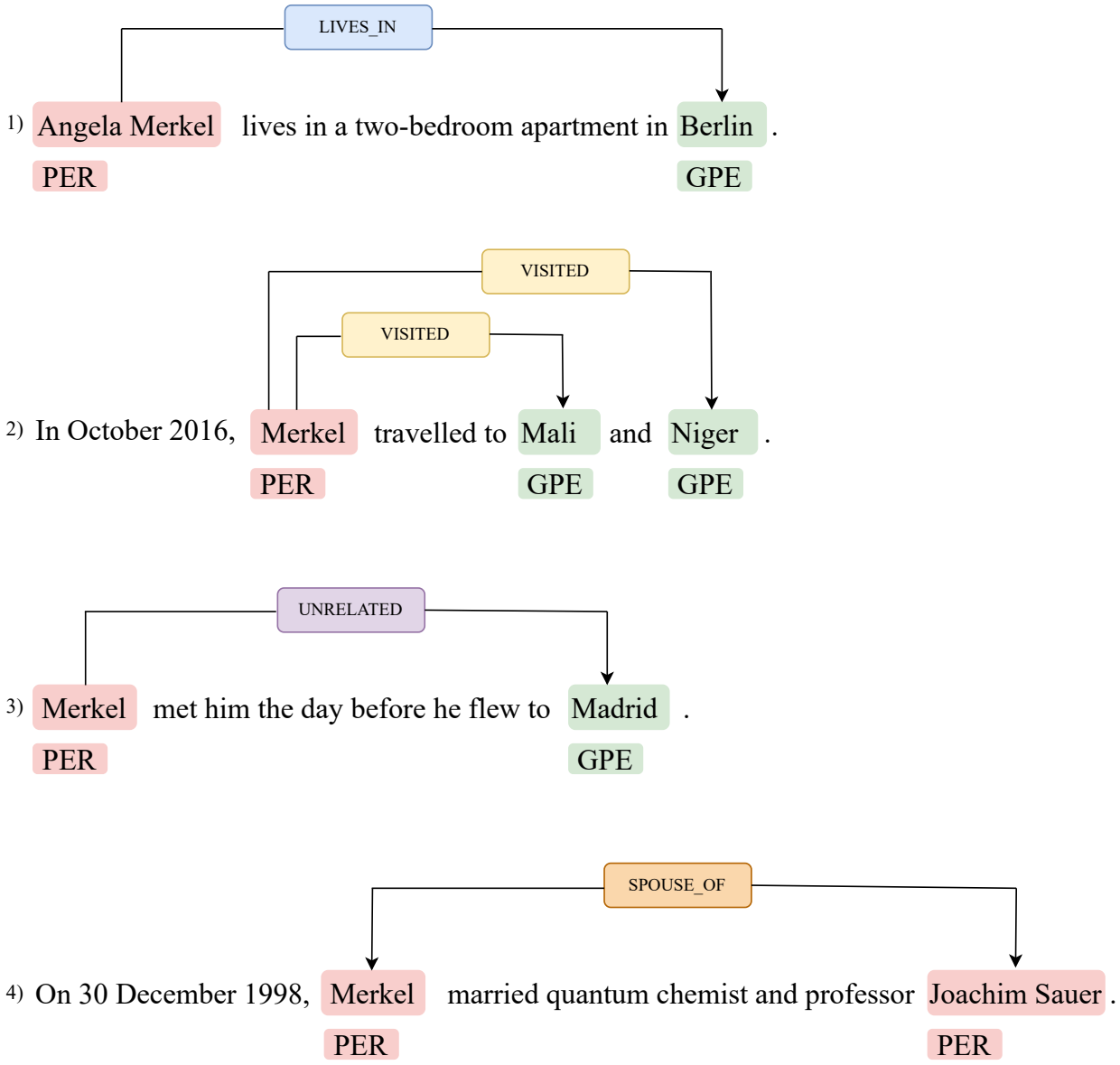


Figure 2.5: Relation extraction examples in the general/public domain.

are already tagged in the text [4]. The figure 2.5 illustrates a few example sentences for relation extraction, the relations of type *LIVES_IN* and *VISITED* are unidirectional and can only exist between the entities of type person and GPE, in contrast, the *SPOUSE_OF* relation is bidirectional and can only exist between entities of type person.

The relation extraction methods can be broadly classified into four categories based on

[4] *Joint Named Entity Recognition and Relation Extraction* is an exception as these methods perform NER and RE simultaneously.

the availability of annotated data and structured knowledge sources such as a KB:

- **Rule-based RE systems**: These methods rely on carefully hand-crafted rules to detect a set of lexico-syntactic patterns which signal the occurrence of a relation; these lexico-syntactic patterns are identified after the manual inspection of the text corpus. On one hand, rules are interpretable, generally have high precision and can be modified for various domains and relations, but on the other hand, rules require a lot of manual effort to ensure acceptable coverage across the domain and various kinds of relations, rules are often very brittle and cumbersome to maintain and rule-based systems tend to have low recall as they are too constrained to handle the diversity of a natural language (Hearst, 1992).

- **Weakly Supervised RE systems**: These systems use a handful *seeds* of hand-crafted patterns or entity pairs to automatically find new patterns that express the relation iteratively, these methods are effective to discover relation instances in a large text corpus. Weakly supervised methods typically have higher recall than rule-based systems and only require a handful of seeds to operate, but these methods are prone to accumulate noise over subsequent iterations (*semantic drift*) (Brin, 1998; Agichtein and Gravano, 2000; Pantel and Pennacchiotti, 2006; Ravichandran and Hovy, 2002). We discuss weakly supervised bootstrapping in detail in Chapter 5.

- **Distant Supervision RE systems**: The distant supervision employ *heuristic(s)* to automatically create annotated training data for RE at scale using an external knowledge source such as a knowledge base. The most familiar heuristic is that "if a relation exists for an entity pair in a knowledge base than all the sentences mentioning this entity pair also expresses that same relation". The distant supervision methods do not require much manual effort to create training data but it is easy to see that *unconstrained heuristic(s)* introduces noise (false positives) in the training data which complicates learning for the supervised RE model. It is important to note that the distant supervision methods assume the existence of an external knowledge source, this prohibits its applicability to specialized domains (Mintz et al., 2009; Riedel et al., 2010; Zeng et al., 2015).

- **Supervised RE systems**: The supervised RE methods use the annotated data to train the supervised RE models. The modern deep learning based methods for RE have achieved impressive performance on several RE datasets, this improved performance is attributed to the availability of large-scale training data (Goodfellow et al., 2015). The existing supervised methods for RE employ a variety of supervised machine learning models including support vector machines (Minard et al., 2011; Hong, 2005), convolutional neural networks (Nguyen and Grishman, 2015; Lin et al., 2016; Jiang et al., 2016), recurrent neural networks (Ebrahimi and Dou, 2015; Miwa and Bansal, 2016), and transformer models (Soares et al., 2019; Yang et al., 2021).

- **Open Information Extraction**: These methods employ a set of very general constraints and heuristics to create extractions that represent relations in the text,

typically at the sentence level. Recently, open information extraction systems have been popular (Fader et al., 2011; Mausam et al., 2012; Angeli et al., 2015).

2.3 Information Extraction for Low Resource Scenarios

In the past few years, researchers have proposed several deep-learning based methods for named entity recognition, entity linking and relation extraction. These methods have achieved compelling performance on various information extraction benchmarks and proved to be critical to enabling the knowledge base population reliably and efficiently. However, one of the major limitations of these deep learning methods is that they rely on a large amount of annotated data. This reliance on huge training datasets prohibits the applicability of these methods in specialized domains including industrial, financial, legal and scientific domains. In the general domain, few datasets are available and have been very popular however, they do not suffice the need of most real-world information extraction applications. Due to the unavailability of enough annotated training data, rule-based systems still dominate industrial information extraction technologies (Chiticariu et al., 2013b). In the past few years, there has been quite some interest to overcome this bottleneck of unavailability of annotated data using two distinct approaches: i) developing methods which can learn in low-data scenarios ii) methods to automatically generate synthetic training data.

Dimensions of Low Resource

The existing work in data-efficient information extraction makes certain assumptions about the low-resource scenario. Hence, it is important to formally define the low-resource scenario, we categorize the low-resource scenarios along the following three dimensions about the target language or domain (Hedderich et al., 2021): i) availability of task-specific annotations ii) availability of unlabelled language or domain-specific text (required to train (word) representations) iii) availability of external sources of information such as gazetteers or knowledge base.

One important aspect is to quantify the scale of low-resource, this is especially relevant across the dimension of availability of task-specific annotations. The existing work uses different thresholds to define low-resource. The threshold also depends on the complexity of the task as a more complex task might also increase the resource requirements. Garrette and Baldridge employ annotation time as a proxy to constrain the threshold value for part-of-speech (POS) tagging, specifically they restrict the annotation time to 2 hours which resulted in up to annotations of 1-2k tokens. Yang et al. treat their work on the challenging task of text generation as low-resource even with 350K labelled training examples. It is worth noting that the resource requirement also depends on the language, Plank et al. reports varying performance across the languages for the same amount of annotated training data in a low-resource setup.

2.3.1 Distributed Representations

The natural language consists of discrete symbols, which form the basis of the representation and communication of human knowledge. Most (modern) text processing methods can only operate on numerical representations of these discrete symbols, typically a vector. The classical methods to transform a string of words into a vector of numbers consist of *categorical word representation* and *weighted word representation* (Naseem et al., 2021).

1. *Categorical Word representation*: This method expresses the occurrence or absence of a word token in a context (typically a sentence, paragraph or document) by the symbolic representation of "1" or "0".

   (a) One hot encoding: This encoding scheme represent every word token with a vector (so-called one hot vector) such that, the dimension of the vector is equal to the (unique) terms in the vocabulary. Every term in the vocabulary has a unique index in the one hot vector, and to represent a word in this one hot vector, the index of the corresponding word is marked as 1, whereas the index for all the other words in the vocabulary is marked as 0. The addition of a new word in the vocabulary will increase the dimension of one-hot-vector by 1.

   (b) Bag-of-Words (BoW): BoW is the extension of one hot encoding such that it adds up the one-hot vectors for words in a sentence, paragraph or document. BoW, therefore, has a non-zero value for every word that occurred in the textual context.

2. *Weighted Word representation*: The methods in this category assign weights to the feature vector based on the frequency of words in the textual context. The methods in this category are extensively employed in the area of information retrieval.

   (a) Term Frequency (TF): *TF* calculates the frequency of a word in a document. The frequency of a word in a document also depends on the size of the document, a word is more likely to appear in a large document than in a small document; therefore, the term frequency of a term is computed by dividing the frequency of the term with the total number of words in a document.

   (b) Term Frequency-Inverse Document Frequency (TF-IDF): The document often contains common words such as "the", "is", "an" etc., there terms will have a higher term frequency and can distort the feature representation in the context of the underlying NLP task. To minimize the impact of these common terms, TF-IDF was introduced (Jones, 2004). IDF assigns a higher weight to the rare terms, as rare terms can be more informative than the frequent terms. The product of term frequency and inverse document frequency represents TF-IDF and is computed with the equation below:

$$TF - IDF(t, d, N) = TF(t, d) * log(\frac{N}{df_t})$$

where TF(t,d) is the frequency of a term in the document, t denotes the term, d represents the document, N represent the number of documents and df_t denotes the frequency of a term t in all the documents.

It is important to note that both categorical and weighted word representation methods fail to capture word order, exhibit no notion of the syntax of the language or the semantic attributes of the words, and also suffer from the *curse of dimensionality*. The input feature representation has a profound impact on the performance of the underlying machine learning model employed for a certain NLP task (Bengio et al., 2013). Therefore, this motivated the development of distributional representation methods which address the above-mentioned shortcomings of the classical categorical and weighted representation methods. The central idea of distributional representation methods is based on the *distributional hypothesis*, according to which the words that occur in the same context tend to have similar meanings (Harris, 1954).

(Static) Word Representations

The word embedding methods map words in a corpus to a low-dimensional vector, such that the value of each dimension corresponds to a feature (often referred to as *word feature*) and might even have a grammatical or semantic interpretation. The embedding vectors are usually dense and learned using co-occurrence statistics from the text corpus. We briefly provide an overview of the popular methods to learn word embeddings, that are relevant to this thesis.

Skip-gram: The skip-gram was introduced by Mikolov et al. to efficiently estimate (static) word vectors from the unlabelled text. The basic idea behind the skip-gram model is to build representations to predict the surrounding words in a sentence using the reference (main) word. Formally, given a sequence of word tokens $\{w_i\}_{i=1}^{N}$, the skip-gram model aims to maximise the objective:

$$\frac{1}{N}\sum_{i=1}^{N}\sum_{-c\leq j\leq c, j\neq 0} logp(w_{i+j}|w_i)$$

where c is the hyper-parameter which refers to the size of the context window. Mikolov et al. later provided an efficient implementation of skip-gram called *word2vec*.

Continuous Bag of Words (CBOW): The *Continuous Bag of Words* (CBOW) was introduced by Mikolov et al., the algorithm of CBOW is similar to skip-gram but it is trained to predict the a single word from a fixed size of context words. Formally, given a sequence of word tokens $\{w_i\}_{i=1}^{N}$, the CBOW model aims to maximise the objective:

$$\frac{1}{N}\sum_{i=1}^{N}\sum_{-c\leq j\leq c, j\neq 0} logp(w_i|w_{i+j})$$

where c is the hyper-parameter which refers to the size of the context window. Mikolov et al. reports that skip-gram works better with a small amount of training data and can also decently represent rare words or phrases, on the other hand, the CBOW is several times faster to train than skip-gram and can slightly better represent word vectors for the frequent words. In practice, both skip-gram and CBOW models employ shallow *feedforward neural network* to compute embeddings, trained using stochastic gradient descent and backpropagation.

GloVe: *GloVe* (Pennington et al., 2014) belongs to the family of methods which employ matrix factorization techniques to generate low-dimensional word representations. The GloVe model constructs a matrix (words x context) of global co-occurrence statistics; the model computes the frequency of each word (the rows) appearing in some context (the columns) in a large corpus. Since the number of contexts can be very large, the GloVe model factorizes this large matrix to achieve a lower dimension matrix (word x feature); each row yields a vector representation for the corresponding word. The GloVe uses the weighted least-square to minimise the reconstruction loss to find the lower-dimensional representation which can explain maximum variance in the text corpus.

fastText: One significant disadvantage of previous word representation approaches is that they operate at the (word) n-gram level and ignore the internal structure of a word. For example, cat and cats will have distinct vectors that will be learned independently of each other. Bojanowski et al. proposed *fastText*, an extension of the skip-gram model which takes into account the subword information. The fastText algorithm represents each word as a bag of character n-grams and learns representations for each character n-gram, the word representations are computed by summing up the corresponding character n-grams. The representations capturing sub-word information are critical for morphologically rich languages, such as Turkish or Finnish. The authors argue that fastText representations are also effective for languages with large vocabularies and many rare words. Also, since the representations are learned for character n-grams, the vector for unknown or out-of-vocabulary words can still be computed by summing up the corresponding vectors of the character n-grams. Bojanowski et al. demonstrate the superiority of fastText representations over skip-gram representations for several word analogy tasks.

It is important to note that the above described word representation models learn a global word embedding matrix. One interesting attribute of the learning (global) word representations is that vector arithmetic operations can be performed on these learned embedding vectors to derive meaning (see figure 2.6). In terms of actual performance on the downstream NLP tasks, Mikolov et al. reports superiority of skip-gram over the CBOW method for tasks involving exploiting semantic regularities, the GloVe and skip-gram performs more or less similar and fastText achieves superior performance as compared to skip-gram and GloVe (David and Renjith, 2021; Yaseen and Langer, 2021b)[5]. The

[5]The exact performance difference among different word representation methods can vary across the datasets for various NLP tasks.

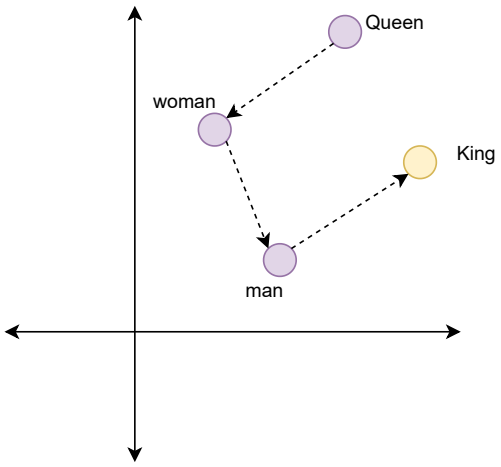


Figure 2.6: An illustration of vector arithmetics using word2vec embeddings: *Queen - woman + man = King*. Figure inspired by Mikolov et al..

pre-trained vectors for word2vec [6], GloVe [7] and fastText [8] can be downloaded from their respective webpages.

Contextual Word Representations

The static representation methods map a word to a fix-sized vector, each word is always assigned the exact same vector, irrespective of the context in which it appears. Consider the following sentences: a) *Heavy rain causes the river to overflow banks in France.* b) *He must borrow money from the bank to pay his tuition.* The word *bank* clearly has two different meanings in both the sentences, it is not obvious if fusing multiple meaning of a word in a single vector is optimal.

The contextualized representation methods explicitly take into account the context in which a word appears. Thus, the contextualized representations for a word depend on all other words in the sentence, the contextualized representation for the word *bank* will be different in the above example sentences. We will briefly discuss few popular approaches to learning contextualized word representations below:

ELMO: The ELMo model (Peters et al., 2018b) was among the initial methods to introduce the idea of learning contextualized representations using unsupervised pre-training, the model extracts context-dependent representations from a *bidirectional language model*.

A language model models the probability distribution over a sequence of tokens. Given a sequence of N word tokens, $(w_1, w_2, ..., w_N)$, a forward language model computes the probability of token w_k given the history $(w_1, , ..., w_{k-1})$:

[6] `https://code.google.com/archive/p/word2vec/`
[7] `https://nlp.stanford.edu/projects/glove/`
[8] `https://fasttext.cc/docs/en/crawl-vectors.html`

$$p(w_1, w_2, ..., w_N) = \prod_{k=1}^{N} p(w_k|w_1, w_2, ..., w_{k-1})$$

A backward language model is similar to a forward language model, except it runs over the sequence in reverse order, predicting the previous token given the future context:

$$p(w_1, w_2, ..., w_N) = \prod_{k=1}^{N} p(w_k|w_{k+1}, w_{k+2}, ..., w_N)$$

The forward L-layer LSTM encodes the left context and a backward L-layer LSTM encodes the right context. For a sequence of length N, the contextualized representations are obtained by the concatenation of the hidden representations from the forward and backward LSTMs at each layer j, obtaining N hidden representations, $(h_{1,j}, h_{2,j}, ..., h_{N,j})$.

The training objective jointly maximizes the log-likelihood of the forward and backward language models:

$$\sum_{k=1}^{N} (logp(t_k|t_1, ..., t_{k-1}; \Theta_x, \overrightarrow{\Theta}_{LSTM}, \Theta_s) + logp(t_k|t_{k+1}, ..., t_N; \Theta_x, \overleftarrow{\Theta}_{LSTM}, \Theta_s))$$

where Θ_x represents the parameters of the token representation, Θ_s denote the parameters of the softmax layer, $\overrightarrow{\Theta}_{LSTM}$ and $\overleftarrow{\Theta}_{LSTM}$ refer to the parameters of the forward and backward LSTM respectively. For efficiency reasons, the token representation and softmax parameters are shared across both LSTMs.

Peters et al. demonstrate the effectiveness of ELMo on six NLP tasks including question answering, named entity extraction and sentiment analysis.

GPT, GPT2: GPT (Radford et al., 2018) aims to learn universal representations which can be transferred to a diverse range of language understanding tasks. GPT follows the familiar two-step learning framework: a) unsupervised pre-training using a language modelling objective, b) supervised fine-tuning on the downstream task. The architecture of the GPT model is based on the Transformer (Vaswani et al., 2017) model (refer to section 2.4.6 for a detailed discussion on the Transformer architecture), it consists of 12-layer decoder blocks with masked self-attention to train language model. The transformer architecture has been shown to better capture global dependencies between input and output as compared to alternate models, e.g. recurrent neural networks and convolutional neural networks, and have achieved superior performance on various sequence processing tasks such as machine translation (Vaswani et al., 2017), text summarization (Liu and Lapata, 2019) and document generation (Liu et al., 2018). In order to support various types of downstream tasks, the GPT pre-processes each text input as a single contiguous sequence of tokens with special tokens including *[START]* (the start of a sequence), *[END]* (the end of a sequence) and *DELIM* (delimiting two sequences from the input text). GPT uses the BookCorpus dataset [9] (Zhu et al., 2015) to train the language model, the dataset

[9] `https://yknzhu.wixsite.com/mbweb`

contains more than 7,000 books from various genres. GPT outperforms discriminately trained task-specific architectures in 9 out of 12 tasks studied.

GPT2 (Radford et al., 2019) follows the architecture of its parent model GPT but uses a different pre-training corpus. GPT2 uses Byte Pair Encoding (BPE) (Sennrich et al., 2016) as the text representation to build up its vocabulary. Radford et al. creates a new dataset named WebText, the dataset consists of millions of scrapped webpages collected by outbound links from Reddit[10]. The dataset was collected with the intention of creating a very large corpus and diverse corpus comprising diverse domains and contexts. The authors hypothesized that pre-training a language model on large-scale unlabelled diverse corpora will enable the language model to learn some common supervised language understanding tasks such as machine translation, question answering and summarization without explicit supervision signal. The authors validate this hypothesis by testing GPT2 on ten datasets of varying complexity in a zero-shot setup. GPT2 achieves strong performance on some tasks, on the CoQA dataset (Reddy et al., 2019), GPT outperforms the performance of 3 out of 4 baselines without using any labelled data.

BERT: Devlin et al. argues that the standard contextualized representation models are not truly bidirectional. In the case of ELMo, the representations from the forward and backward LSTMs are concatenated but this concatenation does not take into account the interactions between left and right contexts; in the case of GPT (Radford et al., 2018) and GPT2 (Radford et al., 2019), the left-to-right decoder is employed which can only attend to the left context. This inability to truly model the bidirectional context is limiting for sentence-level tasks, this effect is aggravated especially for token-level tasks such as named entity recognition or sentence-level tasks such as sentiment analysis, where it is crucial to incorporate contexts from both directions.

BERT alleviate the limitation of unidirectionality constraint by introducing a novel pre-training objective called *masked language modelling* (MLM). The basic idea behind the masked language model is to randomly mask some of the tokens in the input sequence, and the objective is to predict these masked positions taking the corrupted sequence as input. In addition to the masked language modelling objective, BERT also employs a next-sentence-prediction (NSP) objective; given two input sentences, NSP predicts if the second sentence is the actual next sentence of the first sentence. The authors suggest that the NSP objective is helpful for tasks such as question answering and natural language inference, as these tasks require understanding the relation between the two sentences. BERT uses the subword tokenization algorithm called *WordPiece* with a 30,000 token vocabulary; the WordPiece and BPE both use this intuition that frequent words should not be decomposed as their meaningful representations can be learned, but on the other hand, rare words should be split into smaller meaningful subwords to be able to learn meaningful representations for these subwords.

Similar to GPT, BERT also employs special tokens to form a contiguous sequence of tokens for each input sequence. In particular, the first token is always a special classification token [CLS], sentence pairs are packed together into a single sequence and separated

[10] `https://www.reddit.com/`

using a special token [SEP]. BERT follows the pre-training followed by the fine-tuning strategy. For sentence-level tasks, the final hidden state of the [CLS] tokens is used; whereas, for token-level tasks, the final hidden state of each token is used. The pre-training corpus for BERT consists of BooksCorpus (800M words) (Zhu et al., 2015) and English Wikipedia (2,500M words). BERT achieves new state-of-the-art performance on eleven natural language understanding tasks including the famous GLUE (Wang et al., 2019) and SQuAD (Rajpurkar et al., 2016) benchmarks.

RoBERTa: Liu et al. argue that BERT is undertrained and proposed an improved recipe for training BERT models, they refer to their training procedure as *RoBERTa*. The specific changes include: (1) training the model longer with bigger batches and more data (2) removing the next sentence prediction objective (3) training on longer sequences (4) dynamically changing the masked token positions during training. These changes resulted in significant performance gains on the three text comprehension and question answering datasets. RoBERTa is pre-trained on five English-language corpora of varying domains, totalling over 160GB of uncompressed text.

XLM, XLM-R: *XLM* (Conneau and Lample, 2019) extend the generative pre-training using transformer-based models to multiple languages. XLM employs three pre-training methods for learning cross-lingual representations: (1) the causal language modelling objective, the model is asked to predict $p(w_i|w_1, w_2, ..., w_{i-1})$ (2) masked language modelling (MLM) objective, (3) translation language modelling (TLM); TLM is an extension of MLM, is supervised and aims to leverage parallel data when it is available, tokens in both source and target sentences are masked to enable learning cross-lingual association. XLM demonstrates competitive performance on unsupervised machine translation, supervised machine translation and cross-lingual classification.

XLM-R (Conneau et al., 2020) scales up XLM by training a transformer-based masked language model on one hundred languages, using more than two terabytes of filtered CommonCrawl data. XLM-R demonstrates that large-scale multilingual pre-training can achieve compelling performance across a variety of cross-lingual benchmarks.

It is worth noting that it is straightforward to incorporate the pre-trained contextual representation models into a task-specific architecture for improving the performance. Most supervised models use global word representations x_k in their lowest layers, the contextual representations can be concatenated with these global word representations before feeding them to the higher layers.

The pre-trained static word representations in general and contextualized word representations, in particular, have a profound impact on the natural language processing research and are part of most state-of-the-art models; the pre-trained word representations are thus a very successful story of *transfer learning*, demonstrating fruitful transfer of knowledge across various natural language processing tasks.

2.3.2 Distant Supervision

To address the unavailability of sufficient annotated training data for relation extraction in order to train a performant supervised machine learning classifier, Mintz et al. proposed an

alternative method called "distant supervision" or "weak supervision". The idea of distant supervision is similar to the concept of weakly labelled examples introduced by Craven and Kumlien for the biomedical domain. The distant supervision relies on the availability of an external source of information, such as a knowledge base or gazetteers to automatically generate training data. The distant supervision assumes that if two entities participate in a relation, then all the sentences which mention these two entities express the same relation as well (see Figure 2.7). It is easy to see that this relaxed and naive assumption will lead to too many false positive training samples, hence the data generated using distant supervision is often very noisy. Consider the fact triplet, (*Angela Merkel*, bornIn, *Hamburg*), now all the sentences which mention Angela Merkel and Hamburg will be assigned the positive label for the bornIn relation:

1. "Angela Merkel was born in 1954, in Hamburg." → True
2. "Angela Merkel, native of Hamburg, ..." → True
3. "In the afternoon Chancellor Angela Merkel visited the DESY research centre in Hamburg." → False
4. "German Chancelor Angela Merkel is seen during the welcoming ceremony at the G20 summit in Hamburg on 7 July, 2017." → False

Riedel et al. report the precision of aligning three relations in Freebase using distant supervision to the New York Times corpus to be only 70%, this amounts to 30% false positives i.e. 30% of the sentences mention the entity pair but do not express the target relation. Training supervised classifiers on this proportion of noisy data will lead to unreliable relation extraction models with poor performance on the test set. The researchers have proposed various approaches to circumvent the impact of noise from the distant supervision assumption. One of the straightforward approaches involves employing *label refinement* using rules or patterns to ignore noisy labels, this is typically performed as part of the (final) post-processing step (Min et al., 2012; Takamatsu et al., 2012).

Riedel et al. constrains the default relaxed distant supervision assumption such that if two entities participate in a relation, then at least one sentence that mentions these two entities might express that relation. With this constrained assumption, the problem of detecting noise can be framed as a multi-instance learning problem. Multi-instance learning collects all sentences mentioning a certain entity pair in a bag and assigns a relation label to the bag under the assumption that at least one of the sentences actually expresses the relation (Bunescu and Mooney, 2007; Riedel et al., 2010).

The extension of multi-instance learning include *MultiR* (Hoffmann et al., 2011) and MIML (multi-instance multi-label) (Surdeanu et al., 2012), these methods allow entity pairs to participate in multiple relations. Pershina et al. propose "guided distant supervision", an approach to incorporate labelled data in the MIML model. Grave propose a convex formulation of the weakly supervised relation extraction and learns an instance-level classifier using entity-relation triples in a knowledge base to assign labels to the text mentioning

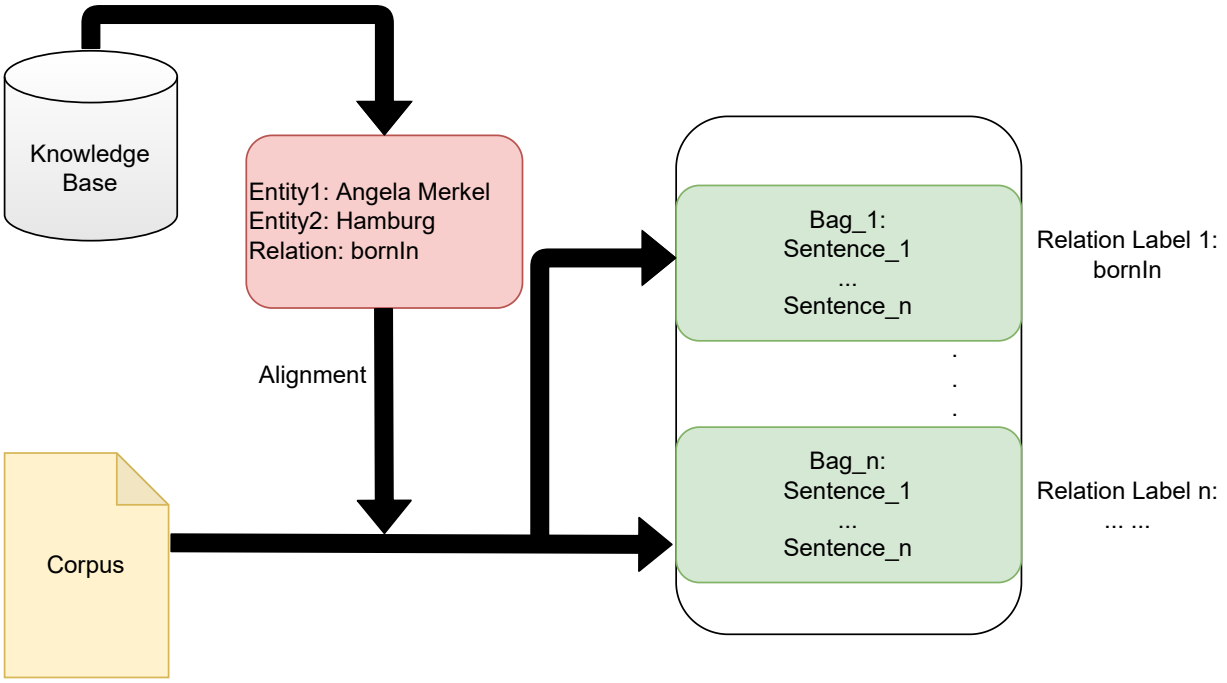


Figure 2.7: An illustration of the distant supervision process. The process involves aligning the entity pairs in the knowledge base to the sentences in the corpus and assigning the relation label to the sentences. The bags are created based on the entity pairs to learn the relation classifier in the multi-instance learning setup. Figure inspired by Wang et al..

the entity pairs, instead of using the triples directly. Zeng et al. proposed PCNNs, an approach to incorporating multi-instance learning objectives into the loss function of a neural network, PCNNs do not require any hand-designed features and can automatically learn features without complicated pre-processing. Jiang et al. relaxes the expressed-at-least-once assumption of PCNN and instead assumes, "a relation holding between two entities can be either expressed explicitly or inferred implicitly from all sentences that mention these two entities", their method creates a representation for a bag of instances by cross-sentence max pooling. Lin et al. propose a sentence-level attention based approach to enable neural networks to dynamically learn to weight multiple instances of a bag in order to pay more attention to correctly labelled instances than to wrongly labelled ones.

The above mentioned methods address the problem of false-positive labels due to the relaxed assumption of distant supervision, distant supervision can also result in false negative labels (Xu et al., 2013). In distant supervision, any sentence that mentions an entity pair which does not have a relation in a knowledge base will be labelled as a negative relation label. However, as mentioned previously (see Section 2.1.1), the knowledge bases are not complete, therefore, inferring no relation between the entity pairs from the absence of the entity pair in a knowledge base is not accurate and can lead to false negatives. Xu et al. report the result of manual analysis of 1834 sentences with two entities from the

NYT 2006 corpus; 133 sentences (7.3%) express a Freebase relation but only 32 (1.7%) of these relation triples are included in the Freebase, resulting in 101 (5.5%) false negative labels. It is important to note that the number of false negatives is higher than the number of false positive labels introduced as a result of distant supervision (5.5% vs 2.7%). These results indicate the importance of the knowledge base population. Xu et al. employ learning-to-rank to develop a passage-retrieval method based on pseudo relevance feedback to reduce false negative labels. Zhang et al. get rid of negative relation labels by using the information of other relations the entity pairs participate in. Min et al. propose to leave potential negative sentences unlabelled and propose an extension of MIML to model unlabelled instances. Ritter et al. employ a latent-variable method to model missing data in both the knowledge base and the text.

2.3.3 Synthetic Data Generation

Data augmentation expands the training set by creating synthetic augmentation of the existing training data points. The synthetic augmentation is usually created by applying a transformation to the original data point such that the label semantics of the dataset are preserved. Data augmentation is well-studied in the computer vision and speech domains, however, the discrete nature of language makes it difficult to adapt data augmentation methods from the domain of computer vision and speech to natural language processing. In the case of computer vision, the pre-defined transformations such as rotation, cropping, flipping etc. can be trivially applied to the images while preserving the underlying label semantics, however, the manipulation of a single word can change the meaning of the sentence. Despite these difficulties, in recent years, researchers have proposed several methods to perform data augmentation for various natural language processing tasks. Broadly, the data augmentation methods can be categorised into two categories:

a) *Rule-based*: The rule-based methods apply a set of pre-defined rules and heuristics to manipulate the data point such that the original label of the data point is preserved. Zhang et al. randomly replace a word by its synonym to generate the augmentation of the training instance; the synonyms are retrieved from an English thesaurus WordNet (Miller, 1994). In a similar direction, Wei and Zou randomly select n words that are not stop words and replaces them with their synonyms. Wang et al. extend the word-replacement technique to generate augmented parallel sentence pairs for machine translation by replacing words in both the source and target sentence pairs. Wei and Zou randomly swaps two words in a sentence to generate an augmented text classification example. Zhao et al. address gender bias in coreference resolution systems by replacing mentions of male entities with mentions of female entities. Fadaee et al. employed RNNs to search for contexts to replace high-frequency common words with low-frequency words. We discuss rule-based methods for sequence labelling tasks in detail in Chapter 4.

b) *Generative models*: The data augmentation methods based on generative models use generative models to generate either a part of the training example or altogether the complete training example. The two dominant trends in this category are to sample from a language model or use backtranslation to generate a paraphrase of the original sentence.

Shleifer employ backtranslation for low-resource text classification. Kobayashi propose *context-aware* augmentation, their method replaces a word with another word which is predicted by a language model at that word position. It is important to note that the language model prediction may not always respect the original label semantics of the data point, consider the example sentence from the task of sentiment analysis with label *positive*: *the movie was great*, the language model can potentially change the positive sentiment word "great" with negative sentiment words such as "bad" or "terrible", this will distort the true label of the sentence and can potentially act as noise for the machine learning model. Li et al. use backtranslation to generate diverse training data for machine translation for low-resource language. Papanikolaou and Pierleoni fine-tune GPT2 to generate synthetic training examples for relation extraction.

2.4 Neural Networks

In this section, we describe the neural network architectures and training strategies which are relevant to this thesis. Neural networks have attracted tremendous interest in recent years due to their impressive performance and ability to model various kinds of tasks across diverse data modalities including computer vision and natural language processing. Generally speaking, neural networks can be viewed as a composition of functions, the *linear perceptron* (Rosenblatt, 1958) is the simplest form of a neural network and consists of one layer with the output y computed as (Bishop, 1995):

$$y = g(w^T \mathbf{x} + b)$$

where $x \in \mathbb{R}^d$ denotes the input, $W \in \mathbb{R}^d$ a weight vector, $b \in \mathbb{R}^d$ a bias vector, g is the threshold *activation function*:

$$g(a) = \begin{cases} -1 & a < 0 \\ 1 & a \geq 0 \end{cases}$$

The value of d represents the dimensionality of the input feature vector. The values of the weight vector and bias vector are learned as part of the training procedure. As the name suggests, the linear perceptron can only model a linear function, thus, it can only classify linearly separable data (Bishop, 1995).

However, most of the real-world NLP problems are non-linear in nature, therefore, require non-linear classifiers, that is why the neural networks which are typically employed for NLP tasks are non-linear classifiers and consist of several layers; namely an input layer, one or more hidden layers and an output layer. The hidden layers typically employ non-linear activation functions, they extract abstract features from the data, they are called *hidden* because their values are not given in the data, instead the model must determine which concepts (captured by the values of the hidden layer(s)) are useful for explaining the relationships in the observed data (Goodfellow et al., 2016). Most modern neural networks

typically have several hidden layers [11], therefore, they are referred to as "deep" and training these deep networks is called "deep learning" (Bengio, 2009; Goodfellow et al., 2016).

2.4.1 Feedforward Networks

In the Feedforward neural networks information flows in the forward direction, from the input layer, through the hidden layer and to the output layer; there are no feedback connections in which the output of the model is fed back into itself (Goodfellow et al., 2016). The hidden layer of feedforward network manipulates the the input vector $x \in \mathbb{R}^d$ by multiplying it with a weight matrix $W_l \in \mathbb{R}^d$ and adding a bias vector $b_l \in \mathbb{R}^d$; as a last step a non-linear activation function f is applied (Bishop, 1995):

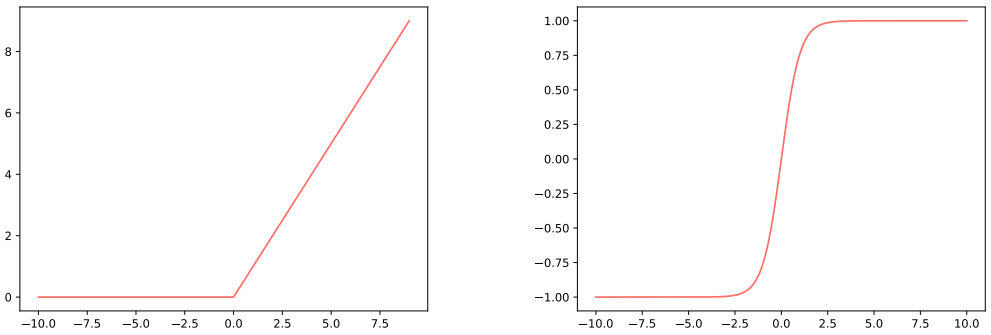
$$f_1 = W_1 x + b_1$$
$$h_1 = a_1(f)$$

where a_1 is the activation function of the first hidden layer, each layer is parameterized by its own weight matrix W_l and bias vector b_l, h_l is commonly referred as the hidden state of the network at layer l. From the above equations, it can be seen that the hidden state h_l is a composition of two functions f and a, where $f(.)$ is an affine function and $a(.)$ is the activation function, a deep feedforward network (also referred to as multilayer perceptron (MLP)) typically is a composition of multiple of these functions. Also, it can be seen that every neuron in the input layer is connected to every other neuron in the hidden layer, this type of network architecture is also referred to as "fully connected feedforward neural network". The typical choice for non-linear activation functions are hyperbolic tangent tanh or rectified linear units ReLu (Nair and Hinton, 2010) (see figure 2.8):

$$ReLU(a) = max(0, a)$$
$$tanh(a) = \frac{exp(a) - exp(-a)}{exp(a) + exp(-a)}$$

At the output layer, first the input $h \in \mathbb{R}^H$ is mapped with a linear transformation $W^{hq} \in \mathbb{R}^{CxH}$ to a vector $z \in \mathbb{R}^c$ of the number of output classes C and then softmax activation function is used to obtain a probability distribution over classes:

$$z = W_{hq} h + b$$
$$P(y = k) = \frac{exp(z_k)}{\sum_j exp(z_j)}$$



(a) ReLu activation function. (b) Hyperbolic tangent (tanh) function.

Figure 2.8: An illustration of activation functions for a hidden layer.

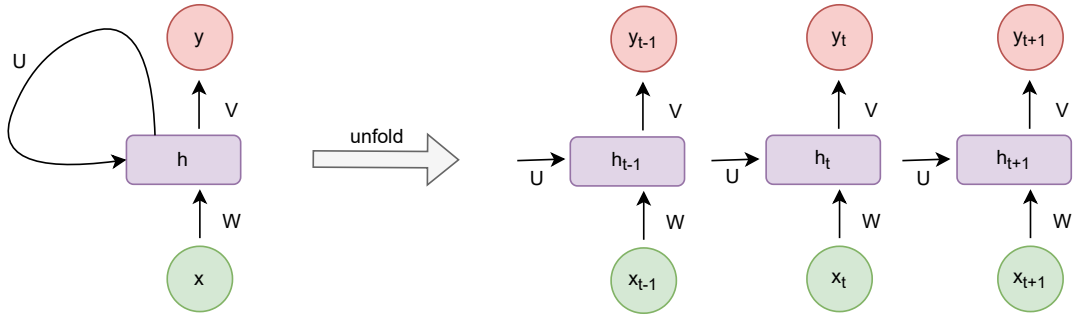


Figure 2.9: An illustration of a recurrent neural network. Figure inspired by Goodfellow et al..

2.4.2 Recurrent Networks

The natural language has this aspect of *sequentiality*, a document is essentially a sequence of paragraphs, a paragraph is a sequence of sentences and a sentence is a sequence of words. In order to model various tasks for natural language text, we require models that can process sequential input. The recurrent neural networks (RNNs) (Rumelhart et al., 1986; Elman, 1990) are a family of neural network architectures which can process sequential data. A recurrent neural network can be viewed as a feedforward neural network unrolled over a sequence of time steps. At each time step t, the network consumes the current input x_t at the current time step and the hidden state h_{t-1} from the previous time step to update hidden state h_t at the current time step; the hidden state can be viewed as a *memory* and should capture the important information about the input sequence till the time step t. Specifically, at every time step the RNN computes the following operation:

$$h_t = a_h(W_h x_t + U_h h_{t-1} + b_h) \tag{2.1}$$

[11] A neural network with one hidden layer is also called *shallow neural network*.

$$y_t = a_o(V_o h_t + b_o) \tag{2.2}$$

where W_h weights the current input x_t, U_h weights the hidden state h_{t-1} from the previous time step, b_h is a bias term for the hidden state, a_h is the activation function for the hidden state; V_o weights the hidden state from the current time step t, b_o is a bias term for the output computation y_t at the current time step, a_o is the activation function for the output computation.

It is important to note that the weight matrices W_h, U_h and U_o are shared across the time steps, this sharing of parameters enables training the recurrent neural networks feasible; if there would be separate weight matrices for individual time steps than it will be very difficult to effectively train RNN, also with this modelling setup the RNN will not generalize to sequence lengths not seen during training.

Note that the equation 2.1 translates to creating a loop of hidden layers as shown in figure 2.9. While training the RNNs using backpropagation through time, this loop is unfolded, which can effectively be viewed as a neural network with a large number of hidden states. When the error term is propagated from the last hidden layer to the first hidden layer, the gradients are multiplied with the same value at each time step; generally speaking for large time steps if this value is greater than 1 the value of the gradient will become very large i.e. explode, contrary if this value is less than 1 than the value of gradients will become very small i.e. vanish. Both of these situations will inhibit the effective training of the model (Hochreiter, 1991; Bengio et al., 1994). Pascanu et al. proposed adding a soft constraint for the problem of vanishing gradients, in order to address the problem of exploding gradients, they propose a simple gradient norm clipping strategy which clips the value of the gradients above a certain threshold. The mainstream approach to enable RNNs to learn long-term dependencies is to employ *gating mechanisms* to the vanilla RNN, we discuss two of these popular methods below.

Figure for the unfolding of RNN (inspirations from the goodfellow's book and heike's thesis)

2.4.3 Long Short Term Memory

Long Short Term Memory networks (LSTMs)(Hochreiter and Schmidhuber, 1997) are extensions of RNNs which are capable of learning long-term dependencies and thus can retain information over longer time horizons. This is especially important for language processing as it is common in natural language to have long-range dependencies. LSTMs employ a gating mechanism to regulate which information should be retained or forgotten. LSTMs have three gates, input gate i_t, forget gate f_t, and output gate g_t; all these gates are a function of the input x_t at current time step and the hidden state h_{t-1} at the previous time step. These gates interact with the current input x_t, the cell state of the previous time step c_{t-1}, and the current cell state c_t to selectively decide which information should be retained or overwritten:

$$
\begin{aligned}
i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i) \\
f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_f) \\
o_t &= \sigma(W_o x_t + U_o h_{t-1} + b_o) \\
c^{\sim}{}_t &= \sigma(W_c x_t + U_c h_{t-1} + b_c) \\
c_t &= f_t \odot c_{t-1} + i_t \odot c^{\sim}{}_t \\
h_t &= o_t \odot \sigma c_t
\end{aligned}
$$

where $\odot$ is the element-wise multiplication, and σ is the sigmoid function.
Say something about bidirectional LSTM ...

2.4.4 Gated Recurrent Units

Chung et al. introduced Gated Recurrent Units (GRUs), GRUS are a simplification to the architecture of LSTMs as it combines the input and forget gate into a single "*update gate*", GRUs also merge cell state and hidden state. Since GRUs have fewer parameters as compared to LSTMs, they can be trained more efficiently. The functions to update the hidden state of GRU are given below:

$$
\begin{aligned}
z_t &= \sigma(W_z x_t + U_z h_{t-1}) \\
r_t &= \sigma(W_r x_t + U_r h_{t-1}) \\
h^{\sim}{}_t &= \sigma(W_h x_t + U_h(r_t \odot h_{t-1})) \\
h_t &= z_t \odot h_{t-1} + (1 - z^{\mathrm{t}}) \odot h^{\sim}{}_t
\end{aligned}
$$

where z_t is the update gate, r_t is the reset gate and σ is the sigmoid function. The update gate z_t decides if the hidden state h_t is updated, and the reset gate controls how much the previous hidden state h_{t-1} should be ignored.

2.4.5 Attention Mechanism

When RNNs are applied to tasks such as text classification or machine translation, the standard practice is to use the last hidden state of the RNN for predicting the class or generating the translated sequence; the assumption is that the last hidden state has accumulated the necessary information from the whole input sequence. However, in practice, it is extremely challenging for the network to subsume all the relevant information for prediction in a fixed-sized hidden state vector (Conneau et al., 2018), even when using gating mechanisms as in LSTMs or GRUs. The attention mechanism was introduced to address this limitation by enabling the model to focus on the intermediate hidden states based on their relevance to the prediction. The attention mechanism is realised by assigning weights to the intermediate hidden states, and the value of the weights signifies the importance of a particular hidden state for the prediction task. The attention mechanism

can be especially helpful in cases when the input sequence is quite large. The attention mechanism was initially introduced for the task of machine translation to perform automatic alignment with RNNs (Bahdanau et al., 2015); they quickly got popular and were employed in a variety of computer vision, speech recognition and NLP tasks including abstractive summarization (Rush et al., 2015), machine reading comprehension (Hermann et al., 2015), question answering (He and Golub, 2016), document classification (Yang et al., 2016), image captioning (Xu et al., 2015), speech recognition (Chan et al., 2016), and conversational modelling (Vinyals and Le, 2015).

2.4.6 Transformers

One of the major limitations of recurrent neural networks is their inherent sequential nature which prohibits parallel computation, hence, recurrent neural networks generally take longer to train. Vaswani et al. proposed the Transformer model that only relied on self-attention mechanisms without any recurrence, therefore, can be parallelised efficiently, Transformer model demonstrated superior performance as compared to recurrent neural networks. Most of the recent state-of-the-art methods for various natural language processing tasks employ the transformer architecture, the popular methods include the OpenAI GPT (Radford et al., 2018) and BERT (Devlin et al., 2019).

The transformer model essentially consists of stacks of "encoder" layers, which further consists of two sub-layers, the *multi-head self-attention* mechanism and *position-wise feed-forward network*. Around each of the two sub-layers, residual connections (He et al., 2016) is employed, followed by a layer normalization component (Ba et al., 2016). The central component is an *attention function* which maps a query vector $\mathbf{Q}$ and a set of key-value vector pairs $(\mathbf{K}, \mathbf{V})$ to an output. The output is essentially a weighted sum of the values, where the value of weight is computed based on the compatibility function (usually a dot product) between the query and the corresponding key. In order to optimise computation, the attention is typically computed on multiple queries at once, Queries, keys, and values are packed together into matrices $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$. The output matrix is computed as follows:

$$attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

where d_k is the dimensionality of queries and keys. In the above equation, the output can be interpreted as the attention distribution over values. *Multi-head self-attention* is the application of several attention layers ("attention heads") in parallel, it is computed by applying h different learned linear projections to queries, keys and values before the attention computation, h here refers to the number of attention heads. The intuition behind multi-head self-attention is to enable the attention layer access to richer representation spaces, this in-effect improves the representation capabilities of the model as now each attention head can focus on one important aspect of the downstream task.

$$multi\_head(Q, K, V) = concat(head_1, ..., head_h)W^O$$
$$head_i = attention(QW_i^Q, KW_i^K, VW_i^V)$$

where $W_i^Q \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{model} \times d_v}$, and $W_i^O \in \mathbb{R}^{hd_v \times d_{model}}$; d_{model} is the dimensionality specified for the model. It is important to note that the self-attention mechanism in the transformer model does not respect the order of the input, therefore, for sequential tasks such as text processing it is usual to add position embeddings to the input.

2.4.7 Training Neural Architectures

The use of non-linear functions enables the neural architectures to achieve expressive representational capabilities which are essential to achieve optimal performance on most real-world datasets, but this comes at the cost of difficulties in the optimization procedures as closed-form solution for optimizing the neural networks on the training set is not possible (Bishop, 1995). The usual practice to train neural networks is with the optimization algorithm called *stochastic gradient descent*, the general steps include: a) sample a batch of data points from the training set b) feed the training examples to the network c) Evaluate the network's performance on the training examples using *loss function* (also known as a cost function) d) compute the gradients of the loss function e) use the gradient descent to guide the network (updating the parameters) in the direction of decreasing value of loss function. In practice, most of the NLP tasks employ *cross-entropy* as the loss function, the cross-entropy quantifies the difference between the estimated distribution from the network and the true distribution. The cross-entropy is computed by:

$$L = -\mathbb{E}_{(x,y)\ p_{data}} log P_\theta(y|x)$$

where θ refers to the parameters of the model, p_{data} is the empirical distribution defined by the training set, y is the true labels and $p(y|x)$ is the probability of the model for output y given the input x (Goodfellow et al., 2016).

The modern neural networks consist of millions of parameters and computing the value of gradients of the error term w.r.t to each parameter can be computationally infeasible, *backpropagation* enable efficient computation of gradient values using *chain rule*(Rumelhari et al., 1986). The actual parameter update is performed by the gradient descent, the vanilla version updates the network's weights as follows:

$$\theta_i = \theta_i - \alpha \frac{\partial L}{\partial \theta_i}$$

where α refers to the learning rate which determines the step size in the negative direction of the gradient. One value of step size for all the network's parameters is limiting as different parts of the network are not optimised equally, this is resolved by employing a per-parameter learning rate which dynamically assigns higher learning rates to less frequent

updates weight parameters and lower learning rates to frequently updates parameters (Duchi et al., 2010; Zeiler, 2012; Kingma and Ba, 2015).

When deep neural networks with large numbers of parameters are trained on limited training data, they achieve a high-performance score on the training data but they perform poorly on the new data, this phenomenon is called *overfitting*; this happens because instead of learning the patterns in the data the model has instead learned the noise in the data (i.e. superficial patterns) or simply memorized the training data. In order to avoid overfitting different strategies are employed, such as regularization, early stopping and dropout. The *regularization* discourages the network to rely solely on a handful of features, this is realized by constraining the weight values by adding the l_2 norm term to the objective function of the neural network (Bishop, 1995; Goodfellow et al., 2016). The early stopping makes sure that the model performs optimally not only on the training set but also on the held-out set (development set) thus encouraging the generalization of the network (Prechelt, 1996). The *dropout* (Srivastava et al., 2014) uses this very simple idea of randomly dropping the neurons/units of the network, this simple trick has two positive effects: a) it leads to co-adaptations i.e. neuron(s) adapts to fix the mistakes of other neurons (s) b). it makes the activations of the hidden units sparse, thus, achieving the effect of regularization.

Chapter 3

Neural Architectures for (Nested) Named Entity Recognition

This chapter covers work already published at peer-reviewed workshops in international conferences. The relevant publications are Yaseen et al. (2019) and Yaseen and Langer (2021b). In Yaseen et al. (2019), Dr Pankaj Gupta proposed the initial idea; I refined the idea, implemented the model, performed the evaluation experiments, and wrote the initial draft of the paper. Pankaj was involved in the continuous weekly discussions and reviewed the paper before the submission, the remaining authors acted as advisors. In Yaseen and Langer (2021b), I conceived the original research, implemented the model, performed the evaluation experiments and wrote the initial draft of the paper. My supervisor, Stefan Langer, contributed through discussions in our bi-weekly meetings and by reviewing the paper before submission.

3.1 Linguistically Informed Named Entity Recognition and Entity Normalization

Extracting knowledge from scientific articles is a challenging but very important problem. This becomes especially critical for biomedical literature which is growing at an increasing rate of at least 4% per year, as of June 2019 there are 30 Million documents in PubMed (Lu, 2011b). The **BioNLP Open Shared Tasks** (BioNLP-OST) intends to aid the development of computational tasks and solutions for biomedical text mining. The focus of the tasks is on information extraction i.e., named entity recognition, entity normalization and relation extraction. The ultimate objective is to gather information about entities and relationships between entities from a large amount of unstructured biomedical text data to fill this information into either an empty or incomplete knowledge base. In this section, we only focus on the task of named entity recognition and entity normalization. As defined in Section 2.2.1, the formal description of the task of NER is to find entities $(e_1, e_2, ...e_n)$

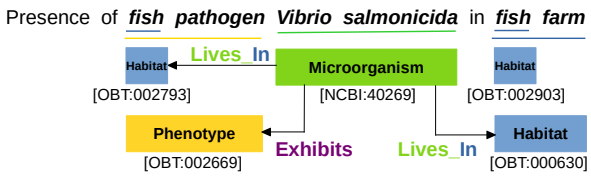


Figure 3.1: An illustration of (nested) Named Entity Recognition, Entity Normalization and Relation Extraction in Biomedical domain. Each rectangular box spans an entity, where the overlapping spans indicate nested entities. E.g., *fish* is a nested entity (a sub-concept) of type *Habitat* within the parent entity *fish pathogen* of type *Phenotype*. The identifiers (e.g. OBT:002669, NCBI:40269, etc.) refer to unique IDs in Biomedical databases (i.e., OBT → OntoBiotope Ontology and NCBI → NCBI Taxonomy), used to perform entity normalization (i.e., entity linking). The arrows indicate binary relationships.

from free text-based on the evidence in a text corpus. We treat the problem of extracting entities in a text corpus or a set of documents as a *sentence level NER* and therefore, all of our proposed models operate at the sentence level.

3.1.1 Tasks

We participate in the following shared tasks:

Bacteria Biotope 2019

The field of Biology has produced immense heterogeneous information about the microbial strains that have been experimentally identified in a given environment (*habitat*) and their properties (*phenotype*). This knowledge of microbial diversity is crucial for studying the interaction mechanisms of bacteria with their environment from genetic, phylogenetic and ecology perspectives. The BB Task consists of: a) recognizing mentions of microorganisms, habitat and phenotype entities in scientific and textbook text b) normalizing entity mentions according to domain knowledge resources (NCBI taxonomy and OntoBiotope Ontology).

PharmaCoNER 2019

The recognition of drugs, medications and chemical entities is critical to understanding the subsequent interactions of chemicals with other biomedically relevant entities. This information is relevant for biomedical researchers, clinicians and the pharmaceutical industry. Most of the existing biomedical and clinical NLP research has been conducted on the English language, with very little emphasis on non-English texts. It is important to note that there is a considerable amount of biomedically relevant content published in non-English languages and particularly clinical texts entirely written in the native language of each country, with a few exceptions. The PharmaCoNER shared task attempts to address these issues and consists of automatic extraction of chemicals, pharmaceutical drugs, and gene/proteins mentioned from clinical case studies written in *Spanish*.

3.1.2 Challenges

The BB 2019 and PharmaCoNER 2019 tasks pose several challenges for the task of named entity recognition and entity normalization, we briefly discuss the most important ones below:

Misspellings

The misspellings of entity mentions or their semantic contexts add considerable difficulties not only in identifying and classifying entity mentions but also in normalizing the predicted entity mentions. Especially in non-scientific texts, misspellings are unfortunately prevalent and therefore complicate NER and entity normalization in non-scientific domains including clinical NLP.

Abbreviations

With the increase of scientific papers published every year (Bornmann and Mutz, 2015), the use of abbreviations as a tool to make technical terms less verbose increased at a significant rate as well. This poses a challenge to the NER system as abbreviations may not always be standard written (the first letter of every word in a term/phrase) and can be ambiguous.

Alternate Names for the Same Entity

The bacteria *Escherichia coli* is also referred to by alternate names such as *E. coli*, *Bacillus coli*, *Bacteriumcoli* etc., this adds complications to the NER model as the model should not only identify all the alternate names but should also assign the correct entity types. In the case of entity normalization, the task is much more complicated as the system should resolve all the alternate names to a single standard entity in the database.

Nested Entities

We define *nested* entities as an entity or sub-concept(s) which is completely contained by another entity mention, we refer to the longer entity mention as a *parent* entity. It is important to note that parent and child entity mentions may have the same or different entity types. In figure 3.1, there are two nested entities: a) *fish* in **fish pathogen**, b) *fish* in **fish farm**. The conventional NER models (Lample et al., 2016; Ma and Hovy, 2016; Akbik et al., 2018) assume flat entities in text and ignore nested entities. The efficient extraction of nested entities may require a change in the annotation tagging format or altogether the development of specialized novel models.

Discontinuous Entities

The *discontinuous* or *disconnected* entity mentions consists of discontinuous sequence of tokens. The discontinuous entities are quite common in the biomedical domain. The

BB 2019 dataset contains *3.6% discontinuous entities*[1]. It is very challenging to detect discontinuous entities and most of the mainstream NER models ignore disconnected entities. Consider the text, *[...] oral, vaginal, and intestinal regions [...]*, there are three entities: "oral region", "vaginal region" and "intestinal regions"; intestinal regions is a continuous entity whereas oral region and vaginal region are discontinuous entities.

Ambiguous Entity Identifiers

In the figure 3.1, there are two occurrences of *fish*; the first occurrences refer to *marine fish* while the second refers to a *farm fish*, both of these entities are linked (or normalized) to different identifiers (e.g., OBT:002793 and OBT:002903) in the biomedical database (e.g., OntoBiotope Ontology). The correct attribution of entity identifiers requires a precise understanding of the semantic context and hence makes entity normalization a challenging task.

3.1.3 Neural Architectures

We adopt a modular approach and develop separate systems for named entity recognition and entity normalization. Our developed systems address most of the challenges mentioned in Section 3.1.2 (except for discontinuous entities). Also, each individual component of named entity recognition and entity normalization further consists of sub-modules to perform various tasks. The rationale behind the modular design of our system is to enable extensibility, improved debugging and modular development.

Figure 3.2 describes the architecture of our developed system which consists of two sub-systems including named entity recognition and entity normalization.

Named Entity Recognition. The named entity recognition system consists of two modules including *Level1 NER* and *Level2 Nested NER*. Level1 NER and Level2 NER are both BiLSTM-CRF based sequence taggers but each is assigned to detect different kinds of entities. Level1 NER is responsible to extract parent entities while Level2 NER is responsible to detect nested entities. Both sequence taggers use word embeddings (w_e) and character embeddings (c_e) to optimally represent a word token and corresponding character(s). Note that Level2 Nested NER only operates on the parent entities detected by Level1 NER. Therefore, if a parent entity is not detected by Level1 NER the corresponding nested entity (if exists) will not be detected by Level2 Nested NER. This error propagation is a result of the pipeline nature of our named entity recognition system i.e., the output of Level1 NER is consumed by Level2 NER. To minimise this effect, we develop three strategies to improve both precision and recall of Level1 NER: a) Level1 NER uses additional *linguistically informed features* including: part-of-speech tags (POS), word shape, capitalization and orthographic features b) Level1 NER performs entity detection and language modelling as auxiliary tasks during training of the standard entity tagging task to improve NER generalization c) Level1 NER employs an additional *ranking loss* to enable the model to better disambiguate among confusing named entities.

[1] `https://groups.google.com/g/bb-2019/c/A2MuFYiPQIY/m/9YtMmakeBQAJ`

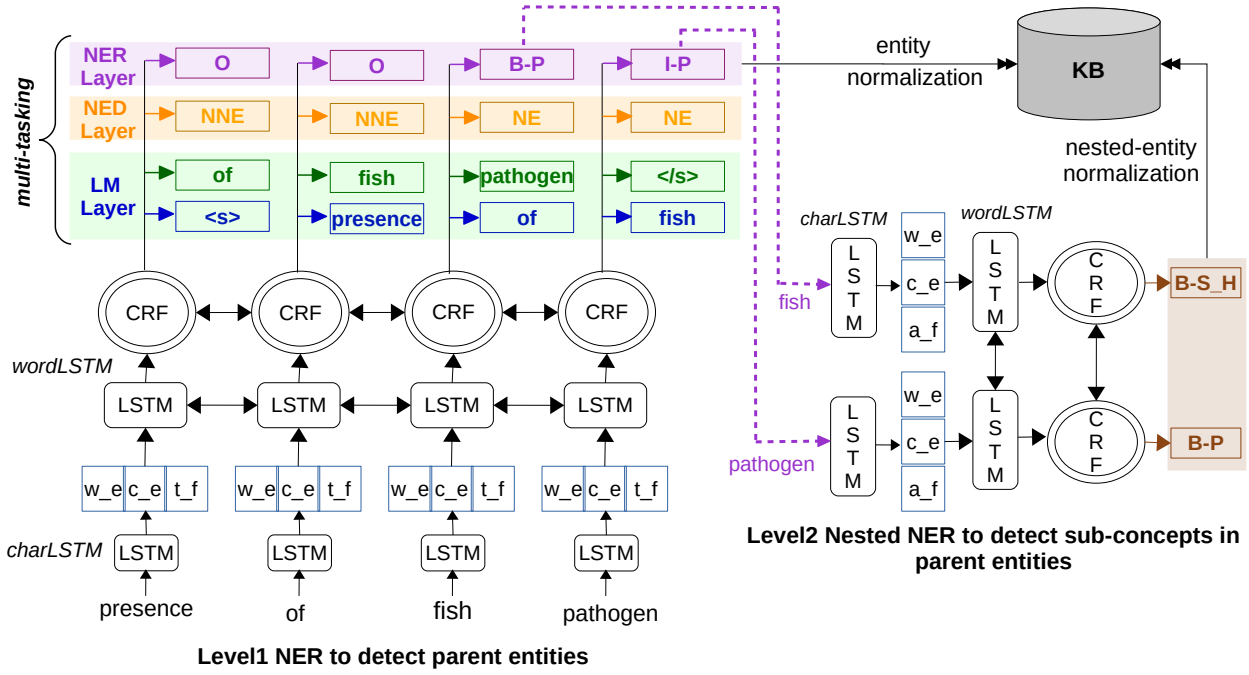


Figure 3.2: System Architecture for the NER task consists of two bi-LSTM-CRF architectures: Level1 NER to detect parent entities and Level2 Nested NER to detect sub-concepts within the parent entities (output of Level1 NER). Here, *w_e*: a word embedding vector; *c_e*: an embedding vector for a word computed using character-level bidirectional LSTM; *t_f*: a vector of additional linguistic features; B_P: B_Pathogen; B-S_H: a sub-concept of type Habitat detected by the Level2 Nested NER run over the parent entity.

Entity Normalization. The parent and nested entities detected by Level1 NER and Level2 Nested NER respectively are then normalized to unique identifiers in KB by our entity normalization system. The entity normalization system consists of individual modules to perform a dictionary-based exact, fuzzy and semantic search. The exact search is based on *string match*, fuzzy search uses the *levenshtein distance* and semantic search employs *distributional similarity* based on cosine distance to perform an entity mention lookup in the KB.

BiLSTM-CRF

The input to LSTM is a sequence of word features $(\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n)$ and they compute a hidden state for each element in the sequence $(\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n)$. This hidden state can be used to jointly model tagging decisions using CRF (Lafferty et al., 2001a). CRF imposes ordering constraints on the tagging decisions e.g. I_Habitat should always be preceded by B_Habitat. For an input sentence,

$$\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n),$$

Features	Description
word-cap	capitalization features
POS	parts-of-speech tags
ortho	orthographic features e.g. *Egg Pulp, 97* encoded as *Ccc Ccccp nn*
tri-gram	tri-gram as features
five-gram	five-gram as features
length	length of the word
sdp-rel	dependency relation tag
alpha-features	detect if certain linguistic pattern occurred in the current word or the next word

Table 3.1: Word-level features for NER. The features are encoded as embeddings, except the *alpha* features that are represented as one-hot vector.

we consider a matrix $\mathbf{P}$ of scores output by the bidirectional LSTM. The size of $\mathbf{P}$ is $n \times k$, where k is the number of distinct tags, and $P_{i,j}$ corresponds to the score of the j^{th} tag of the i^{th} word in a sentence. For a sequence of predictions

$$\mathbf{y} = (y_1, y_2, \ldots, y_n),$$

we define its score to be

$$s(\mathbf{X}, \mathbf{y}) = \sum_{i=0}^{n} A_{y_i, y_{i+1}} + \sum_{i=1}^{n} P_{i, y_i}$$

where the matrix $\mathbf{A}$ express transition scores such that $A_{i,j}$ represents the score of a transition from the tag i to tag j. We add *start* and *end* tag to the set of possible tags, therefore, the size of A is $k + 2$. During training, we minimize the negative log-probability of the correct tag sequence:

$$\begin{aligned} \log(p(\mathbf{y}|\mathbf{X})) &= s(\mathbf{X}, \mathbf{y}) - \log \left(\sum_{\widetilde{\mathbf{y}} \in \mathbf{Y_X}} e^{s(\mathbf{X}, \widetilde{\mathbf{y}})} \right) \\ &= s(\mathbf{X}, \mathbf{y}) - \underset{\widetilde{\mathbf{y}} \in \mathbf{Y_X}}{logadd}\ s(\mathbf{X}, \widetilde{\mathbf{y}}), \end{aligned} \tag{3.1}$$

$$loss_{CRF} = -\log(p(\mathbf{y}|\mathbf{X})) \tag{3.2}$$

Hybrid Loss Function

We use a variant of the ranking loss function proposed by dos Santos et al.. Ranking maximizes the distance between the true label y^+ and the most competitive label c^-:

$$loss_{ranking} = -max(0, 1 + (\gamma * (m^{+} - y^{+})) + (\gamma * (m^{-} + c^{-}))$$

where γ is the scaling factor that penalizes the predictions, m^{+} and m^{-} are margins for correct and incorrect labels respectively. Note that the ranking loss "maximizes" the distance between correct prediction and incorrect confident predictions, therefore, to make it compatible with the $loss_{CRF}$ we add a negative sign in the beginning of the equation to convert the maximization objective to a minimization objective.

The hybrid loss function hence is the sum of CRF tagging loss and ranking loss:

$$loss_{hybrid} = loss_{CRF} + (-\alpha \ \cdot \ loss_{ranking})$$

where $\alpha \in [0, 1]$, weighs the contribution of ranking loss in the overall loss value. During training we minimize the hybrid loss and found it to improve the F1 score for both *Bacteria Biotope* and *PharmaCoNER* tasks.

Multi-Tasking of Named Entity Recognition, Detection and Language Modelling

In order to improve the detection of parent entities, the Level1 NER employs auxiliary objectives of *named-entity-detection* (NED) (Aguilar et al., 2017) and *bidirectional language modelling (LM)* (Rei, 2017). Both of these objectives are employed at the token level. The NED objective aims to identify if the current word token is part of an entity mention or not, it is important to note that the NED objective does not take into account the actual entity type of the current token. The bidirectional LM objective forces the model to predict next and previous words using past and future contexts respectively. LM and NED layers in figure 3.2 realizes NED and LM objectives respectively. With these multi-tasking objectives, for each word token, our model predicts the NED tag, next word, previous word and the NER tag. The existing work has shown that these auxiliary objectives act as regularizers (Collobert and Weston, 2008a) and improves the overall performance of the trained model. It is important to note that the added auxiliary objective requires no additional labelling and their annotations can be generated automatically from existing annotations of the dataset(s). The multi-tasking is only enabled at train time and hence does not add any computational burden at the time of inference.

Nested Entities

The dataset of *Bacteria Biotope* task contains 17.4% nested entities [2] which cannot be extracted by the mainstream standard BiLSTM-CRF based sequence models as they can only extract flat entities. We employed two BiLSTM-CRF models: *L*evel1 NER model to detect parent entities and *L*evel2 Nested NER model to detect nested entities. Figure 3.2 (right) shows the architecture of Level2 Nested NER. The parent entities detected by Level1 NER are fed as input to Level2 Nested NER to detect nested entities within the

[2] `https://groups.google.com/d/msg/bb-2019/A2MuFYiPQIY/9YtMmakeBQAJ`

General Features	Description	Entity Features	Description
bow	bag-of-words (bow) representation of the complete sentence	entity-pos	position of entity in the bow representation
bow-partial	bow representation of the between context (i.e. word tokens between target entities) including three words to the target entities	entity-type	type of the entity mentions
bow-lemma	bow representation of the lemmatized tokens in the between context	dist-entities-cat	distance between target entities as categorical
pos-tags	part-of-speech tags	dist-entities	distance between target entities
sdp	shortest dependency path as bow	entity-count	count of entities in between context
sdp-len	length of shortest dependency path as scalar	entity-count-cat	count of entities in between context as categorical
sdp-rel	dependency relation tag	e1 type = e2 type	if type of e1 and e2 is same
emb-sdp	average embeddings of sdp	sdp-entity	sdp with entity as bow
keyword-vec	if current word is part of feature list of relations	entity-patterns	check if certain linguistic patterns occur in the vicinity of target entities

Table 3.2: General and Entity features used in Relation Extraction

parent entities. Level2 Nested NER has the same architecture as Level1 NER but without additional linguistic features and auxiliary multi-tasking objectives. The final output of the NER system is the aggregation of extracted parent and nested entities. It is easy to see that our NER architecture requires a BiLSTM-CRF to extract entity at one level, since our model consists of only two BiLSTM-CRFs, it can only detect parent entities and nested entities at level 2. If a nested entity occurs at level 3, it will be ignored by our system.

Entity Normalization

The goal of entity normalization (entity linking) is to map noisy predicted entities in the text to canonical entities in the knowledge base. This is challenging because: (1) not all variations of textual forms for a canonical entity exist in the KB, (2) syntactic variations in the predicted entity mentions due to misspellings, abbreviations, acronyms and NER boundary errors.

For *Bacteria Biotope* task, we used two Biomedical databases *OntoBiotope Ontology* and *NCBI Taxonomy*. OntoBiotope Ontology contains $3,602$ canonical forms of type *Habitat* and *Phenotype*. NCBI Taxonomy contains $1,082,401$ records for type *Microorganism*. We employed exact, fuzzy and semantic (embedding) search in an informed ordered manner to perform entity normalization. Algorithm 1 illustrates the detailed steps of our algorithm, it is important to note that the entity type of the predicted entity mention influences the type and order of search sub-module(s). Note that different kind of search operations involve numerous pairwise comparisons and results in reduced run-time performance. We circumvent these repeated comparisons by employing the *caching* mechanism and hence improving the overall run-time efficiency.

Post-processing for NER & Entity Normalization

Our NER model (see Figure 3.2) employs CRF at the decoding step to impose boundary ordering constraints on the predicted named entity types e.g. *I* should always be preceded

Algorithm 1 Entity Normalization

```
    Input: NE, NE_Type
    Output: RF_ID
    Output: NE_PRED (Optional)
 1: RF_ID = None
 2: IF NE_Type == "Microorganism":
 3:    found, RF_ID = exact_match(NE, NCBI)
 4:    if not found:
 5:       found, RF_ID = fuzzy_match(NE, NCBI)
 6:    return RF_ID
 7: ELSE
 8:    found, RF_ID = exact_match(NE, NCBI)
 9:    if not found:
10:       found, RF_ID = fuzzy_match(NE, NCBI)
11:    if found:
12:       # LABEL UPDATE !
13:       NE_PRED = "Microorganism"
14:       return RF_ID, NE_PRED
15:    found, RF_ID = exact_match(NE, OBT)
16:    if not found:
17:       found, RF_ID = semantic_search(NE, OBT)
18: return RF_ID
```

by a *B* token. But our model does not always respect such ordering constraints and therefore, we resolve boundary inconsistencies at inference time to make the NER labels consistent. *Post-processing* column in the Table 3.3 illustrates the post-processing resolving inconsistent labels after the voting on majority labels, consider row *r3* where post-processing correctly imposes the semantics of boundary ordering by changing *I-Habitat* to *B-Habitat*.

3.1.4 Ensemble Strategy

Bagging is a helpful technique to reduce variance without impacting the bias of the learning algorithm. We employed a variant of *Bagging* (Breiman, 1996) which makes sure that every sample in the training set is part of the development set at least once and vice versa. We created three data folds and trained the model using optimal configuration on each fold, prediction on the test involves majority voting among the *three* trained models.

The commonly used tagging schemes (BIO, BIOES etc.,) for NER contain information about the boundary of an entity along with the class of an entity, which is spitted by the model at each time step. Due to this dual information in a single output, maximum voting is not trivial as models can not only disagree on the class but also on the boundary of an entity. Empirically we found that our model is better at predicting the class of an entity rather than the boundary of an entity, therefore, we followed the strategy *class determines the boundary*. In cases when voting results in a tie, we take the prediction of the

	Tokens	Models			Voting	Post-processing
		M1	*M2*	*M3*		
r1	Presence	O	O	O	O	O
r2	of	O	O	B-H	O	O
r3	fish	I-H	B-H	I-H	I-H	B-H
r4	pathogen	I-H	I-P	I-P	I-P	B-P
r5	Vibrio	B-M	B-M	B-M	B-M	B-M
r6	salmonicida	I-M	O	I-M	I-M	I-M
r7	in	B-H	O	O	O	O
r8	fish	B-H	O	B-H	B-H	B-H
r9	farm	I-H	O	I-M	I-H	I-H
r10	.	O	O	O	O	O

Table 3.3: NER: Ensembling and Post-processing correcting individual models mistakes. Here, B, P and M refer to Habitat, Phenotype and Microorganism, respectively.

confident model, and we treat the model trained on the original train/development split as the confident model. We also experimented with an extreme version of ensembling where we aggregate the output of every model with distinct spans, as expected this improves the recall but with the cost of reduced precision. One possible optimization to this ensemble strategy is to only aggregate the non-overlapping spans to control reduction in precision without much decrease in recall.

Table 3.3 shows the ensemble correcting individual model's erroneous predictions. Consider row *r4*, where Model M1 incorrectly predicts the token "fish" as Habitat with tag I-H (both the class and boundary are wrong!), however, the remaining models (M2 and M3) correctly predict the class as Phenotype. The majority voting selects Phenotype as the token label and post-processing fixes the remaining boundary error by converting I-P to B-P.

3.1.5 Datasets

We employed bagging (discussed in section 3.1.4) to split the annotated corpus into 3-folds. We used *BIOES* tagging scheme as an alternative to the commonly used BIO scheme as the BIOES scheme increases the amount of information related to the boundaries of entity mentions. The acronym BIOES refers to beginning (B), inside (I), outside (O), end (E) and single-token entities (S).

PharmaCoNER: The dataset consists of four entity types with very few mentions of type *UNCLEAR* and *NO_NORMALIZABLES* as shown in table 3.4. Entities of type *UNCLEAR*

Task	Train	Dev	Test
Sentence Counts			
PharmaCo	8068	3748	3930
SeeDev	644	308	466
BB-norm+ner	822	413	735
PharmaCoNER Entities			
NORMALIZABLES	2304	1121	859
PROTEINAS	1405	745	973
UNCLEAR	89	44	34
NO_NORMALIZABLES	24	16	10
BB-norm+NER Entities			
Habitat	1118	610	-
Microorganism	739	402	-
Phenotype	369	161	-

Table 3.4: Dataset statistics for NER

are ignored in the evaluation of this shared task but we still treat them as regular entities.

Bacteria Biotope: We used pre-processed versions of datasets for Bacteria Biotope[3] provided by the organizers. This pre-processed version comes with sentence splitting, word tokenization and POS tagging. The dataset consists of three entity types with few mentions of type *Phenotype* (see table 3.4). The dataset also contains 3.6% *disconnected entities*[4], we did not employ any strategy to handle disconnected entities and instead treat them as separate (regular) entities.

3.1.6 Experimental Setup

We found sub-word information to be very helpful in identifying entities and relations in the biomedical domain and all our experiments used word embeddings trained using fastText (Bojanowski et al., 2017a). For tasks in the English language, we used fastText embeddings trained on PubMed (Zhang et al., 2019b). We don't employ any strategy for handling the imbalanced classes. Table 3.5 lists the best configuration of hyper-parameters for both datasets.

PharmaCoNER: We used *SPACCC_POS-TAGGER* (Soares and gonzalez agirre, 2019) for sentence splitting, word tokenization and POS tagging. We trained fastText embeddings on the following corpora: IBECS (Rodríguez, 2002), IULA-Spanish-English-Corpus (Marimon et al., 2017), MedlinePlus (Miller et al., 2000), PubMed (Lu, 2011a), ScIELO (Goldenberg

[3] `https://sites.google.com/view/bb-2019/supporting-resources`
[4] `https://groups.google.com/d/msg/bb-2019/A2MuFYiPQIY/9YtMmakeBQAJ`

Hyper-parameter	Value
NER	
learning rate	0.005
character (char) dimension	25
hidden unit::char LSTM	25
POS dimensions	25^*, 50^+
Ortho dimension	25^*, 50^+
hidden unit::word LSTM	200^*, 100^+
word embeddings dimension	200^*, 100^+
length dimension	10
sdp_rel	10
alpha_features	2
ranking loss::α	1.0
ranking loss::γ	1.0

Table 3.5: Hyper parameter settings for NER, * and + denote the optimal parameters for Bacteria Biotope and PharmaCoNER respectively.

et al., 2007) and PharmaCoNer (Gonzalez-Agirre et al., 2019). We trained embeddings on two variants of corpora: (1) Include train and development set of PharmaCoNER (2) Include complete dataset of PharmaCoNER. We concatenated these two embeddings to enable complementary information fusion and found them to empirically work better than the embeddings trained on individual corpora variants. We compute micro-F1 using the script provided by the organizers on the development set[5].

Bacteria Biotope: For training NER model we compute macro-F1[6] (Tsai et al., 2006) on the development set. NER and Entity normalization together are evaluated using *Standard Error Rate* (SER) (Bossy et al., 2015). The SER indicates the proportions of errors in a prediction in comparison to the ground truth, the lower the SER, the better the prediction. During the entity normalization step, the fuzzy and semantic search can resolve an entity mention to multiple normalization identifiers. Our algorithm returns the top 5 matched identifiers, however, we empirically found selecting only the top most identifier gives superior performance.

3.1.7 Results

Result on development Set

To investigate the impact of various features we incrementally enabled them and observe the effect on the performance of the development set.

[5] `https://github.com/PlanTL-SANIDAD/PharmaCoNER-CODALAB-Evaluation-Script`

[6] evaluation measure with strict boundary detection

	Configuration	PharmaCoNER			Bacteria Biotope			
		P	**R**	**F1**	**P**	**R**	**F1**	**SER**
		Fold=1			**Fold=1**			
r1	*BiLSTM-CRF*	.884	.773	.824	.809	.474	.598	.576
r2	*+ word-emb*	.892	.857	.874	.831	.526	.644	.524
r3	*+ ortho*	.909	.846	.877	.823	.515	.633	.533
r4	*+ POS*	.906	.851	.877	.827	.523	.641	.526
r5	*+ multi-task*	.907	.851	.878	.806	.528	.638	.531
r6	*+ length*	-	-	-	**.842**	.487	.617	.545
r7	*+ ranking*	**.912**	**.860**	**.885**	.827	.535	.650	.520
r8	*+ search*	-	-	-	.810	**.600**	**.690**	**.489**
		Fold=2			**Fold=2**			
r9	*BiLSTM-CRF*	.915	.890	.902	.630	.400	.489	-
r10	*all features*	**.934**	**.889**	**.911**	**.719**	**.513**	**.599**	-
		Fold=3			**Fold=3**			
r11	*BiLSTM-CRF*	.899	.873	.886	.784	.699	.739	-
r12	*all features*	**.917**	**.877**	**.896**	**.813**	**.764**	**.788**	-

Table 3.6: Scores on the development set using different features on *PharmaCoNER* and *Bacteria Biotope* tasks. Here, + signifies feature accumulation to the last row.

NER: Table 3.6 shows the score on the development set for *PharmaCoNER* and *Bacteria Biotope*. Observe that *fastText* embeddings (row r2) outperform randomly initialized embeddings (row r1) and contribute to biggest performance boost for both datasets. Subsequently, *Orthographic* (row r3) and *POS* (row r4) features improve the scores for PharmaCoNER but surprisingly lower the score for Bacteria Biotope. In row r5, we perform multi-tasking with the auxiliary task of NED leading to improvement only for PharmaCoNER. Next, we incorporate hybrid loss including ranking (row r7) which consistently improves the score on both datasets. In row r8, we employed Brute Force Search (discussed in section 3.1.8) that significantly reduce SER for BB-norm+NER. Finally, we create an ensemble of (r7, r10, r12) and (r8, r10, r12) on the test set for PharmaCoNER and Bacteria Biotope respectively.

3.1.8 Analysis

Bacteria Biotope: We also explored approaching the problem of NER and entity normalization in a reverse manner by matching every entity mention from the biomedical databases (i.e. *NCBI Taxonomy* and *Ontobiotope*) in every sentence. This matching is indeed an exhaustive search, we refer to it as *Brute-force search*. Figure 3.3 shows the comparison

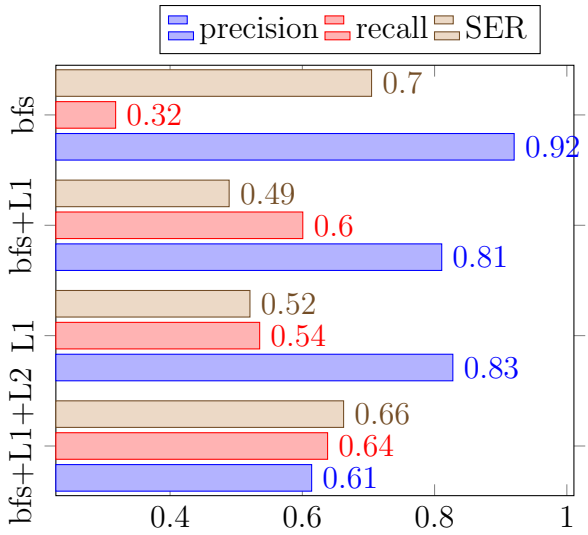


Figure 3.3: *Bacteria Biotope*: Impact of brute-force search, Level1 NER and their aggregation on SER. Here bfs, L1 and L2 refer to *brute-force search*, *Level1 NER* and *Level2 Nested NER* respectively.

of: (1) *brute-force search* (2) *Level1 NER* (3) aggregation of *brute-force search* and *Level1 NER* (4) aggregation of *brute-force search*, *Level1 NER* and *Level2 NER*. Brute-force search yields high precision but a moderately low recall with an SER value of 0.7. In comparison, Level1 NER has significantly higher recall with a little reduction in precision yielding an SER value of 0.52. The aggregation of brute-force search and Level1 NER improves recall and lowers the SER value to 0.49. Finally, aggregation of brute-force search, Level1 NER and Level2 NER results in balanced precision and recall values but an overall higher value of SER. Our submission on the test set employed aggregation of *brute-force search* and *Level1 NER*.

3.1.9 Comparison with Participating Systems

Bacteria Biotope: Table 3.7 shows the comparison of performance among participating teams on Bacteria Biotope test set. Our two submissions (MIC-CIS-1, MIC-CIS-2) ranked first and second with a standard error rate of **0.7159** and *0.7867* respectively. The second submission employed Level2 NER to extract nested entities and hence has higher recall but with reduced precision. *MIC-CIS-1* has the highest precision of 0.6242 and *MIC-CIS-2* has the recall close to the best recall of *BLAIR_GMU-1* with score *0.4676*. Precision and recall are not balanced, we hypothesize improvement in nested entities extraction and modelling of discontinuous entities will improve the system recall.

Team	*P* / *R* / *SER*
MIC-CIS-1	**.624** / .433 / **.715**
MIC-CIS-2	.560 / .449 / .786
BLAIR_GMU-1	.496 / **.467** / .793
BLAIR_GMU-2	.499 / .466 / .805
baseline-1	.572 / .327 / .823

Table 3.7: Comparison of our system (MIC-CIS) with top-5 participants: Scores on Test set for SeeDev and BB-norm+NER

3.2 Stacked Heterogeneous Embeddings for Named Entity Recognition

The user base of social media platforms has tremendously increased since they were first introduced in the early 2000s; this global ubiquitous usage of social media has led to massive user-generated content across various platforms. Twitter is a popular micro-blogging platform where users can publish tweets up to 280 characters. The common public actively uses Twitter to share life-related personal and professional experiences with others. Personal experiences may also include health-related incidents including mentions of adverse drug effects (ADE); this information is crucial to studying Pharmacovigilance, disease tracking and patient-centred outcomes. In the context of the COVID-19 pandemic, the professional experiences may include information about professions and occupations which are vulnerable due to either direct exposure to the virus or due to the associated mental health issues; detecting vulnerable occupations is critical to adopting necessary preventive measures and facilitating intervention strategies.

3.2.1 Tasks

We participate in the following shared tasks:

Task 1b - ADE Span Detection

The task consists of identifying spans of adverse effects mentioned in tweets, the tweets only consist of English language.

Task 7b - Profession Span Detection

The task requires IE systems to detect the spans of expressed professions and occupations in the user tweets. This task only considers tweets in the Spanish language.

3.2.2 Challenges

Twitter has a uniquely distinctive style of communication and this poses several unique challenges to information extraction and language understanding systems. We briefly mention few of these challenges below:

Brief Text

Tweets are short messages and even have an upper limit on the character counts, hence they cannot provide large contexts and is, therefore, difficult to analyze.

Colloquial Expressions.

Colloquial expressions are prevalent on Twitter as tweets are generally written in an informal and casual style, this adds difficulty for the information extraction systems.

Misspellings

The character limit on tweets and informal style of writing lead to prevalent misspellings in the Twitter data and this significantly complicates NER and entity normalization.

Noisy text

Tweets occasionally include non-standard words which must be converted to one or more canonical words to make sense of them. As an example consider the tweet, "Jst read a tweet lol and l o v e it", this should be normalized to "Just read a tweet laughing out loud and love it". Without the accurate normalization of the tweet, it would be very difficult to analyze it.

Data sparsity

Natural language is sparse and this effect is pronounced in the context of social media content such as tweets, due to this it is challenging to learn representative word embeddings in the short text context.

Multilinguality

The ubiquity of social media including Twitter is a global phenomenon and therefore, tweets are written in various languages. Thus, the information extraction systems should be robust enough to extract information in a multilingual context.

3.2.3 Stacked Embeddings

Recently, word embeddings (Mikolov et al., 2013a; Pennington et al., 2014; Bojanowski et al., 2017b) have become the default choice to vectorize words for various natural language

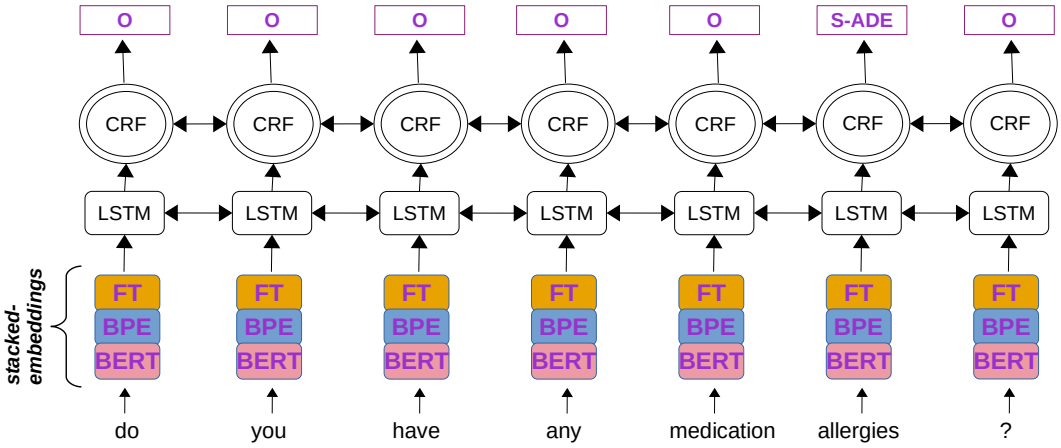


Figure 3.4: System architecture for NER task, consisting of BiLSTM-CRF with stacked heterogeneous embeddings. Here, *FT*: fastText embedding vector; *BPE*: Byte-Pair embedding vector; *BERT*: BERT embedding vector; *S_ADE*: S_Adverse Drug Effect.

processing tasks. More recently, contextualized word embeddings (Devlin et al., 2019; Peters et al., 2018a) have gained popularity due to their ability to adapt the embedding vector of a word based on the context around the word. The existing research has shown that various kinds of embeddings capture different representation aspects, motivated by this observation and success of various types of word and sub-word embeddings we proposed stacking several kinds of heterogeneous word embeddings to enable *complimentary learning*. Figure 3.4 describes the architecture of our model, where we design a sequence tagger to extract entities. The architecture of our model is a standard BiLSTM-CRF (Lample et al., 2016) model with stacked heterogeneous embeddings and linguistic features as input. In particular, we stacked context-independent Byte-Pair subword embeddings (Heinzerling and Strube, 2018), fastText subword embeddings (Bojanowski et al., 2017b) and contextualized BERT embeddings (Devlin et al., 2019).

3.2.4 Datasets

Data: We employed bagging (discussed in section 3.1.4) to split the annotated corpus into 3-folds such that every sample in the training set is part of the development set at least once and vice versa. As a pre-processing step for ADE span detection (Task 1b) and Profession span detection (Task 7b), we perform sentence splitting, word tokenization, computing orthographic features and POS tagging.

ADE Span Detection (Task 1b): The dataset consists of only one entity type *ADE*. The train set contains 1717 entity mentions of *ADE* (see Table 3.8).

Profession Span Detection (Task 7b): The dataset consists of four entity types with few mentions of type *FIGURATIVA* as shown in Table 3.8. Entities of type ACTIVIDAD and FIGURATIVA are ignored in the evaluation of this shared task but we still treat them as regular entities.

Task	Train	Dev
Sentence Counts		
Task 1b	34142	1775
Task 7b	14755	4959
Task 1b Entities		
ADE	1713	87
Task 7b Entities		
PROFESION	1597	566
SITUACION_LABORAL	264	85
ACTIVIDAD	45	16
FIGURATIVA	16	8

Table 3.8: Dataset statistics for NER.

3.2.5 Experimental Setup

We found contextualized embeddings to be very helpful in identifying entities and all our submissions to the official shared task evaluation used pre-trained contextualized embeddings. We employed *RoBERTa* (Gururangan et al., 2020) for ADE span detection (Task 1b) and *Spanish BERT* (Cañete et al., 2020) for Profession detection (Task 7b). We don't employ any strategy for handling imbalanced classes for NER. Table 3.9 lists the best configuration of hyperparameters for all the tasks.

3.2.6 Results

Result on development Set

We perform various experiments to investigate the impact of features on performance on the development set.

NER: Table 3.10 shows the score on the development set for Task 1b and Task 7b. Observe that fastText embeddings (row r2) outperform glove embeddings (row r1) for Task 1b. Subsequently, fastText embeddings with BytePair embeddings (row r4) provide an improvement over only fastText (row r2) and the combination of fastText with Character embeddings (row r3). The contextualized embeddings (row r5) provide an improvement over the combination of fastText with BytePair embeddings. In row r6, we employ BERT, fastText and BytePair embeddings in a stacked format leading to the best f1-score for both Task 1b and Task 7b.

Hyper-parameter	**Value**
learning rate	0.1
optimizer	SGD
hidden size	256
POS dimensions	50
Ortho dimension	50
batch size	32
epochs	150

Table 3.9: Hyper parameter settings for Task 1b and Task 7b.

	Features	**Task 1b** P/R/F1	**Task 7b** P/R/F1
r1	*glove*	.5/.18/.26	-
r2	*fastText*	.89/.28/.43	.84/.64/.73
r3	*fastText + Char*	.64/.28/.39	.83/.67/.74
r4	*fastText + BytePair*	.62/.34/.44	.82/.74/.78
r5	*BERT*	.68/.35/.46	.84/.76/.80
r6	*BERT + fastText + BytePair*	**.61/.52/.56**	**.86/.77/.81**
		Fold=2	**Fold=2**
r7	*BERT + fastText + BytePair*	**.80/.21/.34**	**.85/.79/.82**
		Fold=3	**Fold=3**
r8	*BERT + fastText + BytePair*	**.77/.37/.50**	**.84/.78/.81**

Table 3.10: Scores on dev set using different features for *BiLSTM-CRF* on *Task 1b* and *Task 7b*.

3.2.7 Comparison with Participating Systems

Table 3.11 shows the comparison of our submissions with the arithmetic median of the participating teams for all the tasks. Our submissions achieve the overall best F1-score than the arithmetic median for all the tasks showing a compelling advantage. For Task 1a, the precision of our system is lower than the arithmetic median but this is compensated by the improvement in recall. For all the tasks, the precision is higher than the recall but overall precision and recall are balanced.

Task 1b: ADE		**Task 7b: ProfNER**	
Team	*P* / *R* / *F*1	**Team**	*P* / *R* / *SER*
Team 26	**.510** / .514 / **.514**	Recognai	.840 / **.840** / **.840**
MIC-NLP	.500 / **.555** / .459	**MIC-NLP**	**.850** / .800 / .820
TensorFlu	.500 / .493 / .505	Lasige-BioTM	.810 / **.660** / .730
arithmetic median	.493 / .458 / .420	arithmetic median	.842 / .726 / .760

Table 3.11: Comparison of our system (MIC-NLP) with top-3 participants: Scores on Test set for Task 1b (ADE) and Task 7b (ProfNER). The mapping from team identifiers to system description is mentioned in Table 3.12.

3.3 Related Work

In this section, we describe different approaches employed by participating teams for named entity recognition and entity normalization in the shared tasks of Bacteria Biotope, PharmaCoNER, Adverse Drug Effect span detection and Profession span detection. We use participant identifiers for a better view to describe the participating systems. A mapping from team identifiers to system description papers is given in Table 3.12.

3.3.1 Bacteria Biotope

The Bacteria Biotope shared task has been introduced in 2011 with the goal to promote large-scale information extraction from scientific documents in order to automatically populate knowledge bases in the microbial diversity (Bossy et al., 2012). The shared task addresses the IE tasks of entity extraction, entity normalization and relation extraction. We participated in the *bb-norm+ner* track of shared task in 2019. A total of 10 teams participated in the shared task, however, only 3 teams participated in the track of bb-norm+ner.

The existing work ranges from rule-based systems to transformer-based models. Cook et al. proposed *TagIt*, a dictionary-based entity tagger which is followed by a rule-based expansion system to identify bacteria strain names and habitats and resolve them to the closest match possible in the NCBI taxonomy and the OntoBiotope ontology respectively. Grouin proposed a combination of machine learning and a rule-based system, in particular, they used CRF with several linguistic features and post-processing rules to identify mentions of bacteria and biotopes, a rule-based approach to normalizing the concepts in the ontology and the taxonomy. The team whunlp employed neural networks with linguistic features for named entity recognition. Mao and Liu employed transformer-based architectures for their two submissions. Their first submission employed BERT-Large Cased (Devlin et al., 2019) with CRF as a tag decoder. Their second submission builds on XLNET (Yang et al., 2019b); XLNET improves on BERT by integrating ideas from autoregressive models to accommodate long text and also enables learning bidirectional context using permutation language modelling as the language modelling objective function. Zhang et al. employs

team name	reference
Bacteria Biotope	
TagIt	(Cook et al., 2016)
LIMSI	(Grouin, 2016)
whunlp	(Wuhan University)
MIC-CIS	(Yaseen et al., 2019)
BLAIR_GMU	(Mao and Liu, 2019)
AmritaCen_healthcare	-
AliAI	(Zhang et al., 2019a)
UTU	- (University of Turku)
PharmaCoNER	
VSP	(Suárez-Paniagua, 2019)
IxaMed	(Lahuerta et al., 2019)
NLNDE	(Lange et al., 2019)
xiongying	(Xiong et al., 2019)
sohrab	(Sohrab et al., 2019)
chaanim	(Hakala and Pyysalo, 2019)
MIC-CIS	(Yaseen et al., 2019)
Adverse Drug Effect	
Team 24	(Elkaref and Hassan, 2021)
KFU NLP Team	(Sakhovskiy et al., 2021)
CASIA_Unisound	(Zhou et al., 2021)
TensorFlu	(Ramesh et al., 2021)
MIC-NLP	(Yaseen and Langer, 2021b)
Team 26	(Dima et al., 2021)
Profession Detection	
Recognai	(Carreto Fidalgo et al., 2021)
Lasige-BioTM	(Ruas et al., 2021)
MIC-NLP	(Yaseen and Langer, 2021b)
Troy&AbedInTheMorning	(Santamaría, 2021)
SINAI	(Mesa Murgado et al., 2021)
RACAI	(Pais and Mitrofan, 2021)

Table 3.12: Mapping from participant team name to system description papers. This table includes participants of the Bacteria Biotope, PharmaCoNER, Adverse Drug Effect span detection and Profession Detection shared tasks. Note that this table does not include all participants of all years of shared tasks but only those systems mentioned in this section.

a multi-tasking objective of NER and relation extraction on BERT, they also introduced an alternate tagging scheme to address overlapping and discontinuous entities which are prevalent in the Bacteria Biotope corpus. Their system achieved competitive performance on both NER and relation extraction tasks.

3.3.2 PharmaCoNER

To address the growing amount of clinical records written in Spanish [7], the first shared task on detecting drug and chemical entities in Spanish medical documents was introduced in 2019. The task also includes a concept-indexing sub-track to return SNOMED-CT identifiers related to drugs/chemicals in the documents. We participated in the named entity recognition track (drug and chemical entities) in 2019. A total of 21 teams participated in the named entity recognition track with a total of 77 submissions (each team was allowed to make 5 submissions per task).

The participants employed a diverse set of models and training strategies to address the task of named entity recognition. The VSP team (Suárez-Paniagua, 2019) employed the mainstream BiLSTM with CRF (Lample et al., 2016) to detect named entities, their system used character BiLSTM to create the respective word representation; one of the interesting aspects of their method is that they did not employ pre-trained word embeddings or any hand-crafted features. IxaMed (Lahuerta et al., 2019) also employed the standard BiLSTM with CRF model for NER, however, they employed the word embeddings (Mikolov et al., 2013a) trained on Electronic Health Records (50M words), together with pre-trained Wikipedia2Vec embeddings (Yamada et al., 2020) (word and entity embeddings trained on Wikipedia text). IxaMed's submission obtained a competitive performance with F1-score of 0.8681. NLNDE (Lange et al., 2019) explored attention over different kinds of embeddings to optimally select the best embedding for each word, they also employed training using noisy data; the base architecture of their model was BiLSTM-CRF, NLNDE obtained competitive F1-score of 0.88610. xiongying (Xiong et al., 2019) employed BERT with a CRF layer and obtained F1-score of 0.9105, they observed that the model was quite sensitive to the correct tokenization and stressed the importance of correct word tokenization to obtain improved NER performance; their system obtained the best score on the PharmaCoNER task. Sohrab et al. proposed an exhaustive model that implements a contextual exhaustive approach considering all possible contextual spans in a sentence using the BiLSTM-CRF model; this model can also detect nested entities by enumerating all possible contextual spans, and their system achieved the F1-score of 0.8676. Hakala and Pyysalo employed multi-lingual BERT and achieved the F1-score of 0.87378.

3.3.3 Adverse Drug Effect Span Detection

The Adverse Drug Effect task was introduced to develop automated methods to extract adverse drug effects from tweets containing mentions of drugs for social media pharmacovigilance. One subtask of the shared task was to extract spans of reported adverse drug effects in tweets. In total, 7 teams participated in ADE span detection.

Team 24 proposed a BiLSTM model with contextualized representations derived from the BERT encoder along with linguistic feature representations such as POS embeddings and character embeddings; their model employed a joint modelling strategy to perform ADE span detection as well as ADE normalization. KFU NLP Team employed a multilingual cased

[7] Spanish is spoken by more than 572 million people worldwide (Gonzalez-Agirre et al., 2019).

BERT [8] pre-trained on the English collection of consumer comments on drug administration; they use two additional training corpus of CSIRO Adverse Drug Event Corpus (CADEC) (Karimi et al., 2015) and COMETA corpus (Basaldella et al., 2020) to train the NER models. CASIA_Unisound employ BiLSTM-CRF as a baseline model and report experiments with 7 different BERT variants for token encoding. TensorFlu employs *roberta-base* and obtained a competitive F1-score of 0.50. Team 26 employs BioBERT (Lee et al., 2020) in a multi-task learning setup to jointly perform ADE text classification, ADE span detection and ADE span normalization; the first 11 layers of BioBERT (out of 12) were frozen, whereas the last layer was kept as a shared trainable encoder.

3.3.4 Profession Span Detection

ProfNER is the first shared task introduced to address the recognition of professions and occupational entities from Twitter data in the Spanish language (Miranda-Escalada et al., 2021). Since Twitter is often used to express personal and professional life experiences, the profession detection can enable to characterize health-related issues to identify potential risk groups which not only include health care workers, but also professionals such as caregivers, taxi drivers, etc.; detecting vulnerable occupations is necessary to adopt necessary preventive measures. In total 29 teams participated in the profession span detection task.

Recognai paid special attention to pre-processing to make sure entity tokens are not incorrectly split and employed BERT with CRF as the backbone model; they perform hyperparameter optimization using Parzen Estimator as search algorithm and ASHA trial scheduler to terminate low-performing trials. Recognai obtained the 1st rank with F1-score of 0.93. Lasige-BioTM employed the standard BiLSTM-CRF as a backbone model but they used a stack of three distinct embeddings trained on Spanish text; to improve the robustness of the model they use random character replacement and word replacement using wordnet, and their system obtained the F1-score of 0.73, 9 points lower than the best performing system. RACAI used LSTM with CRF as a base model, the input to LSTM is a concatenation of three different embedding types including, Wikipedia, Twitter and Medical embeddings (all trained on Spanish text). Troy&AbedInTheMorning employed a rich set of features including character embeddings, Syllables embeddings, POS tag embeddings, and a set of word embeddings (Twitter, Medical and Spanish BERT) with BiLSTM-CRF as the backbone model; their system obtained a competitive F1-score of 0.82. SINAI employed a more straightforward approach of employing CRF trained using the L-BFGS method.

3.4 Summary

In this chapter, we described our named entity recognition and entity normalization systems with which we participate in several shared tasks. We start with the description of two shared tasks: 1) *Bacteria Biotope*: the task consists of recognizing and normalizing mentions of microorganisms, habitat and phenotype entities in the scientific and textbook text in

[8] `https://huggingface.co/cimm-kzn/endr-bert`

English, 2) *PharmaCoNER 2019*: the task aims towards automatic extraction of chemicals, pharmaceutical drugs and gene-proteins mentioned in the clinical studies written in Spanish. We highlight the specific challenges these shared tasks pose including misspellings, nested entities, discontinuous entities etc. Afterwards, we introduced our *linguistically informed NER and entity normalization model* to address most of the above challenges. To address nested entities, our NER systems employed two BiLSTM-CRF models, the first BiLSTM-CRF (*Level1 NER*) detects parent entities while the second BiLSTM-CRF (*Level2NER*) detects the nested entities. Since the Level2 NER only operates on the output of Level1 NER it is crucial that Level1 NER has high recall and precision, we employed additional strategies such as auxiliary language modelling objective, auxiliary named entity detection objective, hybrid ranking loss and several linguistic features to improve the performance of Level1 NER. We further improved the performance of our NER model by employing bagging, we trained three models on three different data splits and employed an ensemble strategy based on majority voting to make the final prediction on the test set. To address the entity normalization part of the problem, we employed dictionary-based exact, fuzzy and semantic search. The exact search uses string match, fuzzy search employs Levenshtein distance and semantic search is based on distributional similarity. Our system ranked 1st in the Bacteria Biotope entity recognition and entity normalization challenge.

In the second part of this chapter, we start with the description of the remaining shared tasks including, *Adverse Drug Effect (ADE) span detection* and *Profession Span Detection*. The ADE span detection aims to identify the adverse effects of drugs in the English language, whereas, the profession span detection aims to detect professions and occupations in the Spanish language. Both of these tasks use Twitter as the underlying data source around the theme of social media for health. We also discuss the specific challenges of performing NER on the social media text, these challenges include brief text, misspellings, noisy text, colloquial expression etc. Afterwards, we present our NER model (BiLSTM-CRF) based on the idea of *Stacked Heterogenous Embeddings*. The basic idea behind our model is that different kinds of word embeddings capture different aspects of the language; for instance, Word2Vec or GloVe can learn representative global embeddings of the words, fastText is really good at capturing sub-word embeddings, BERT can capture contextualized embeddings etc.; instead of choosing one over another embedding, we can actually combine different embeddings by stacking or concatenating these heterogeneous embeddings to enable *complimentary learning*. It is important to note that this stack of embeddings does not necessarily need to be of different architecture to enable complementary learning, but they can also be of different data sources for a stack of embeddings trained on domain-specific data and general-purpose data. We experimentally demonstrate that using the stack of embeddings is better than only using a single embedding source for the NER model. Our NER system ranked 3rd for both ADE span detection and profession span detection shared tasks.

Chapter 4

Data Augmentation for NER

This chapter covers work already published at the international peer-reviewed conference. The relevant publication is Yaseen and Langer (2021a). I conceived the original research, implemented the model, performed the evaluation experiments and wrote the initial draft of the paper. My supervisor, Stefan Langer, contributed through discussions in our bi-weekly meetings and by reviewing the paper before submission.

Data augmentation addresses the data scarcity problem by expanding the training dataset with synthetic training instances, the synthetic instances are usually created by transforming the original training instances in such a manner that the underlying label semantics of the dataset are preserved. Data augmentation is well studied in the field of computer vision and speech. The discrete nature of language makes it difficult to adapt data augmentation strategies from computer vision and speech to natural language processing. In computer vision, the hardcoded transformations such as rotation, masking, cropping etc. can be easily applied without changing the label semantics, however, the manipulation of a single word in a sentence could change its meaning. Despite these challenges, several approaches have been proposed for sentence-level NLP tasks. Motivated by these efforts we explore *backtranslation* to generate high-quality and linguistically diverse synthetic data for low-resource named entity recognition (a token level NLP task). We empirically demonstrate the effectiveness of our proposed augmentation strategy on two English datasets from the biomedical and material science domains in the low and high resource scenarios.

4.1 Introduction

Most recently, deep learning based methods have achieved state-of-the-art performance for many natural language processing tasks such as text classification, relation extraction and named entity recognition. The availability of large training datasets is required to achieve this improved performance (Conneau et al., 2017; Zhang et al., 2017a; Mohan and Li, 2019). However, in many real-world scenarios collecting such large training data is not feasible. This is especially true for specialized domains, such as the material science or biomedical

domain, where annotating data requires expert knowledge and is usually expensive.

Different methods have been proposed to address low-resource scenarios. For example, Bootstrapping methods exploit clustering techniques to expand the initial seed set with novel extractions (Gupta and Manning, 2014b; Zhang et al., 2020a; Batista et al., 2015a), Self-supervised learning methods train an initial machine learning model on limited training data and augment original training data with trained model's confident predictions on the unlabelled instances (Qiu et al., 2009). Data augmentation methods expand the original training set with novel synthetic training instances respecting the underlying label semantics of the dataset (Shorten and Khoshgoftaar, 2019; Feng et al., 2021).

Recent work explores the development of data augmentation methods for natural language processing tasks. Most augmentation methods focus on sentence-level tasks such as sentiment analysis (Liesting et al., 2021), text classification (Wei and Zou, 2019; Xie et al., 2019) and sentence-pair tasks such as natural language inference (Min et al., 2020) and machine translation (Wang et al., 2018a). There are two dominant trends in the proposed augmentation methods: (1) employ simple heuristics such as word replacement (Zhang et al., 2015; Wang et al., 2018a; Cai et al., 2020), word swap (Sahin and Steedman, 2018; Min et al., 2020) or random deletion (Wei and Zou, 2019) to generate augmented instances by manipulating a few words in the original sentence (2) *generates* artificial instances via sampling from generative models such as language models (Schick and Schütze, 2021), variational autoencoders (Yoo et al., 2019; Mesbah et al., 2019) or backtranslation models (Yu et al., 2018; Iyyer et al., 2018).

The sequence labelling tasks such as named entity recognition (NER) and part-of-speech tagging (POS) involve prediction at the word/token level. This complicates applying token-level transformations as such manipulations may change the corresponding token level label and hence distort the underlying label semantics of the dataset. The existing DA methods for sequence labelling can broadly be categorized into three categories: (1) apply pre-defined heuristics such as dependency tree morphing (Sahin and Steedman, 2018), label-wise token and synonym replacement (Dai and Adel, 2020) (2) use MIXUP (Zhang et al., 2018) to generate queried samples by combining pairs of examples and their labels in the active learning scenario (Zhang et al., 2020b) (3) sample novel sequences from a pre-trained language model (Ding et al., 2020). It is difficult to apply the existing sequence labelling DA methods in a low-resource domain(s) and language(s) because they: a). require linguistics resources like dependency parser or WordNet b). involves training a language model which could be expensive and not always feasible c). generate grammatically incoherent sequences d). cannot generate linguistically diverse sentences.

In the past few years, there have been significant advancements in machine translation systems which led to the availability of high-quality machine translation systems (He, 2015; Wu et al., 2016; Junczys-Dowmunt, 2019). Motivated by these developments, we adapt *backtranslation* to the task of NER. Backtranslation (BT) can automatically generate diverse

paraphrases of a sentence or a phrase by naturally injecting linguistic variations. One of the appealing characteristics of backtranslation is the ability to diversify the injected linguistic variations by introducing layers of intermediate language translations. In this work, we employ backtranslation to generate paraphrases of one or several phrases in a sentence. We empirically demonstrate the effectiveness of our proposed method on two domain-specific NER datasets in low-resource scenarios.

4.2 Related Work

The recent interest in data augmentation has resulted in plenty of proposed DA methods for various NLP tasks. In this section, we narrow our focus to proposed DA methods for sequence labelling tasks like NER and POS. We categorize existing DA methods for sequence labelling into two categories:

Rule-based: DA primitives, which use predefined easy-to-compute transformations. We briefly describe six of such transformations proposed in the existing work:

(a) *NER::Label-wise token replacement (LwTR):* Replace a token with another token of the same entity type at random (Dai and Adel, 2020).

(b) *NER::Synonym replacement (SR):* Replace a token with one of its synonyms retrieved from WordNet at random (Dai and Adel, 2020).

(c) *NER::Mention replacement (MR):* Replace an entity mention with another entity mention of the same entity type at random (Dai and Adel, 2020).

(d) *NER::Shuffle within segments (SiS):* Divide the sequence of tokens into segments of the same label and then randomly shuffle the order of segments (Dai and Adel, 2020).

(e) *POS::Crop Sentences:* Given a dependency tree of the sentence, "crop" a sentence by removing dependency links (Sahin and Steedman, 2018).

(f) *POS::Rotate Sentences:* Given a dependency tree of the sentence, "rotate" a sentence by moving the tree fragments around the root (Sahin and Steedman, 2018).

Generative models: The existing work employs pre-trained language models to generate either part of the sequence or the entire sequence with the corresponding NER tags. Kang et al. proposed *Filtered BERT* which randomly masks one or several tokens in the original sentence and let BERT (Devlin et al., 2019) predict the masked token. The augmentation is only accepted if the cosine similarity of the word embeddings (computed using fastText embeddings (Bojanowski et al., 2017b)) of the original token and the predicted masked token is above a certain threshold. Ding et al. propose a two-step DA process DAGA. First, a shallow language model is trained over linearized sequences of tags and words. Second, sequences are sampled from this language model and delinearized to create new examples.

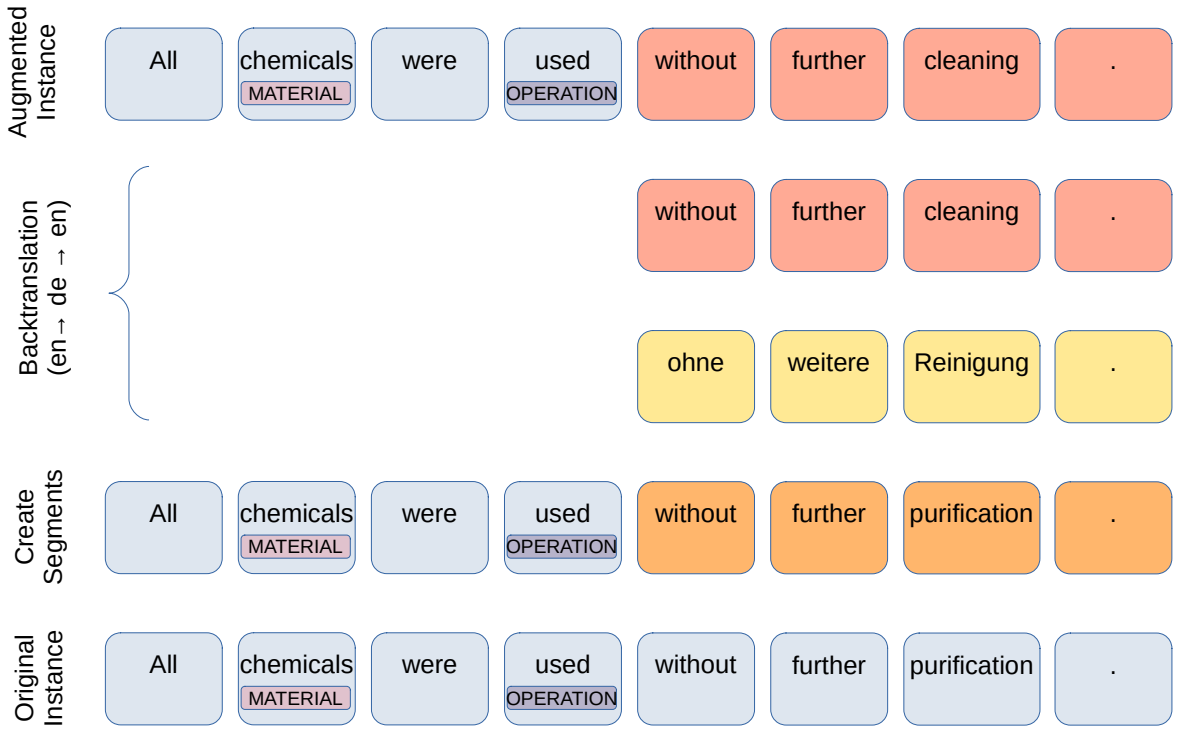


Figure 4.1: An illustration of data augmentation via *backtranslation* for NER. Note that backtranslation is only applied to the context around the entity mentions. Here the entity mention context is first translated to German and then back to English using an off-the-shelf machine translation system. The backtranslation results in a paraphrase of the original entity mention context. The original entity mention context is replaced with backtranslated context to create the augmented data instance.

4.3 Data Augmentation via Backtranslation

Figure 4.1 illustrates an example of our proposed data augmentation method for NER using *backtranslation* with *German* as a pivot/intermediate language. In a nutshell, our algorithm consists of three steps. First, the input token sequence is split into segments of the same label; thus, each segment corresponds to either the entity mention or the context around the entity mention. In order to avoid the alignment issues and preserve the token-level entity labels we only consider the context around the entity mention as a candidate for the backtranslation. Second, the validity of the segment is determined based on the length of the segment, we only consider segments with three or more tokens as a valid segment for backtranslation. As a final step, the segment tokens are translated to the pivot language(s) and finally back to the source language, the original segment tokens are replaced with the backtranslated tokens and thus we obtain the augmentation of the original input token sequence. In practice, we use a binomial distribution to randomly decide whether the segment should be backtranslated. Since only the context around the

entity mention is backtranslated, it is straightforward to adjust the corresponding BIO-label sequence accordingly for the backtranslated text.

Data augmentation with backtranslation augments the original training set with diverse paraphrases of the entity mention contexts to help the supervised NER model generalize beyond the standard training set.

4.4 Evaluation

We empirically evaluate our proposed data augmentation strategy for NER using backtranslation as described in Section 4.3 on two English datasets from the materials science and biomedical domains: MaSciP (Mysore et al., 2019)[1] and S800 (Pafilis et al., 2013)[2].

We use a BiLSTM-CRF model (Lample et al., 2016) as the underlying supervised NER model, and we investigate the impact of applying data augmentation on training data of different sizes.

4.4.1 Datasets

MaSciP contains synthesis procedures annotated with synthesis operations and their typed arguments. S800 consists of PubMed abstracts annotated for mentions of organisms. We use the original train-dev-test split of both datasets provided by the authors. The descriptive statistics of the datasets are reported in Table 4.1.

We follow Dai and Adel to simulate a low-resource setting. We select 50, 150, and 500 sentences from the training set to create the corresponding small, medium and large training sets (denoted as S, M, L in Table 4.2 and Table 4.3, whereas the complete training set is denoted as F). Data augmentation is only applied to the training set without altering the development and test set.

	MaSciP			**S800**		
	Train	Dev	Test	Train	Dev	Test
Number of sentences	1,899	112	162	5,733	830	1,630
Number of mentions	18,896	1,190	1,259	2,557	384	767
Number of unique mentions	4,707	590	605	1,070	194	3781
Number of entity types	21	20	21	1	1	1

Table 4.1: The descriptive statistics of the datasets.

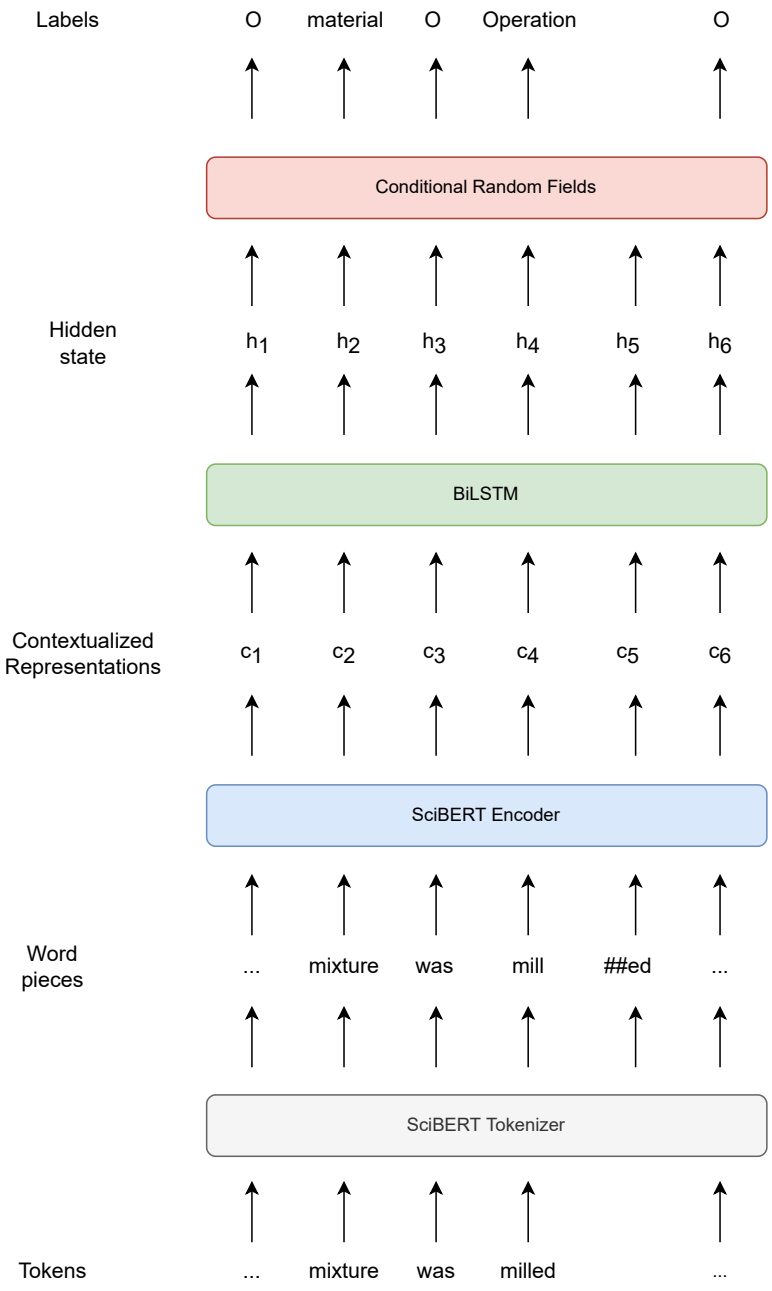


Figure 4.2: High level overview of the BILSTM-CRF model with contextualized SciBERT embeddings.

Emb	DA	S	M	L	F	
Glove	None	48.52± 3.5	67.98± 0.5	73.02± 0.8	75.37± 0.3	
	LwTR	61.95± 1.3	68.04± 0.7	75.05± 0.3	75.32± 0.2	2.9
	SR	**63.91± 1.6**	69.44± 0.7	75.10± 0.4	76.95± 0.8	4.6
	MR	63.46± 0.3	69.64± 0.7	75.08± 0.4	76.33± 1.0	4.6
	SiS	63.63± 1.1	69.60± 0.3	73.35± 0.2	**77.36± 0.3**	4.6
	BT	63.66± 0.6	**69.67± 0.1**	**75.22± 0.2**	76.85± 0.4	**4.6**
SciBERT	None	61.89± 1.3	71.76± 0.6	78.52± 0.1	79.91± 0.1	
	LwTR	66.88± 1.4	73.40± 1.1	77.83± 0.1	77.51± 3.0	0.9
	SR	67.07± 0.8	74.56± 0.3	78.47± 0.4	79.71± 0.3	1.9
	MR	67.65± 1.0	74.60± 1.3	78.04± 1.1	79.57± 0.6	1.9
	SiS	66.87± 2.9	73.40± 1.5	**78.95± 0.6**	79.79± 0.5	1.7
	BT	**70.11± 0.8**	**75.86± 0.8**	78.92± 0.2	**80.30± 0.5**	**3.3**

Table 4.2: F1-score on test sets on **MaSciP** dataset using different subsets of the training set. Here: **S**, **M**, **L** and **F** refer to *small* (50 instances), *medium* (150 instances), *large* (500 instances) and *full* (all instances) set. We repeat all experiments three times with different seeds. Mean values and standard deviations are reported. Δ column shows the averaged improvement due to data augmentation for each embedding type across the datasets.

4.4.2 Supervised NER Model

We follow the standard approach of modelling the NER task as a sequence labelling task on a token-level. Every token in a sentence is assigned a tag/label that enables to determine if a token is part of an entity mention or not. In particular we employed *BIO* tagging scheme, where *B* refers to the *beginning* of an entity mention, *I* refers to *inside* of an entity mention and *O* refers to *outside* of an entity mention.

Figure 4.2 illustrates the architecture of our BiLSTM-CRF model. The encoder of our model consists of a bidirectional LSTM (Hochreiter and Schmidhuber, 1997) with a conditional random field layer (Lafferty et al., 2001b) as the output tagging component, CRF is employed to model dependencies between the neighbouring labels. Our model also takes advantage of a BERT-based encoder which is pre-trained on large-scale text in a language modelling setup to obtain contextualized representation of each token in the sentence.

BERT encoder. Given a sentence, the tokenizer of the pre-trained BERT first converts the sentence tokens into subwords or word pieces (Wu et al., 2016). This conversion of the token into word pieces only happens if the token does not exist in the vocabulary, the actual conversion involves segmenting the token into several pieces from the vocabulary. The word pieces are then mapped via a lookup table to dense vectors or token embeddings. To encode positional information, positional embeddings are employed to indicate the position of each

[1] `https://github.com/olivettigroup/annotated-materials-syntheses`
[2] `https://github.com/spyysalo/s800`

Emb	DA	S	M	L	F	Δ
	None	12.24± 1.6	21.61± 0.7	49.99± 2.6	60.44± 1.4	
	LwTR	17.37± 0.4	41.19± 1.3	50.93± 1.8	62.46± 1.2	6.9
	SR	17.83± 1.3	43.86± 1.1	57.76± 0.2	65.28± 0.5	10.1
Glove	MR	17.86± 2.4	43.90± 0.8	56.70± 0.9	65.34± 0.6	9.9
	SiS	17.17± 1.7	44.36± 0.2	56.80± 0.9	64.93± 0.2	9.7
	BT	**31.06± 1.7**	**47.82± 1.2**	**58.86± 1.0**	**66.89± 0.3**	**15.1**
	None	39.78± 1.6	51.15± 1.6	64.08± 0.8	72.73± 0.9	
	LwTR	41.37± 0.4	51.76± 1.0	64.97± 1.6	71.34± 0.1	0.4
	SR	40.24± 1.2	**53.68± 0.4**	62.98± 1.4	71.77± 0.6	0.2
SciBERT	MR	41.89± 1.4	53.24± 1.3	66.56± 1.2	70.87± 0.5	1.2
	SiS	41.57± 1.8	51.83± 0.7	65.16± 1.0	71.20± 0.6	0.5
	BT	**44.60± 1.0**	53.22± 1.3	**66.76± 1.1**	**72.92± 0.2**	**2.4**

Table 4.3: F1-score on test sets on **S800** using different subsets of the training set. Here: **S**, **M**, **L** and **F** refer to *small* (50 instances), *medium* (150 instances), *large* (500 instances) and *full* (all instances) set. We repeat all experiments three times with different seeds. Mean values and standard deviations are reported. Δ column shows the averaged improvement due to data augmentation for each embedding type across the datasets.

token in the sequence. As a final step, the token embeddings and positional embeddings are fed into a stack of multi-head self-attention and fully-connected feed-forward layers (Vaswani et al., 2017) to obtain the representation of the input sequence.
Existing studies demonstrate the superiority of domain-specific BERT embeddings compared to the general-purpose BERT embeddings on the downstream tasks (Gururangan et al., 2020; Dai and Adel, 2020). We employed *SciBERT* (Beltagy et al., 2019), which is based on the BERT model pretrained on scientific publications, our preliminary experiments suggest that SciBERT achieves better performance than BERT. We also perform experiments with context-independent *GloVe* embeddings (Pennington et al., 2014).

We report the micro-average F_1 score as an evaluation metric. We employ early stopping and report the F_1 score on the test set using the best performant model on the development set.

Backtranslation Models. We employed the pretrained English↔German machine translation models (Ng et al., 2019) [3,4] released by Facebook as part of their submission in the WMT19 News Translation Task. We employed the Huggingface Transformers library's (Wolf et al., 2020) port of the released pretrained machine translation models as the underlying backtranslation models for all our experiments.

[3] `https://huggingface.co/facebook/wmt19-en-de`
[4] `https://huggingface.co/facebook/wmt19-de-en`

Hyperparameters. For each augmentation method, we follow the existing work (Dai and Adel, 2020) to tune the number of augmentation instances per training instance from a list of numbers: $\{1, 3, 6, 10\}$. When the complete training set is used, this tuning list is reduced to: $\{1, 2, 3\}$. We also tune the probability value p of the beta distribution which is used to decide if the segment in a sequence should be backtranslated. It is searched over a list of numbers: $\{0.1, 0.3, 0.5, 0.7\}$. We perform a grid search over these two hyperparameters to find their best combination on the development set. Table 4.4 lists the best configuration of hyperparameters for our BiLSTM-CRF model for all the experiments.

Hyperparameter	Value
hidden units	256
embedding	Glove, SciBERT
embedding dimensions	300^{*}, 768^{+}
epochs	100
learning rate	0.1

Table 4.4: The hyperparameter settings for supervised NER BiLSTM-CRF model.* and + denote the value for Glove and SciBERT embeddings respectively.

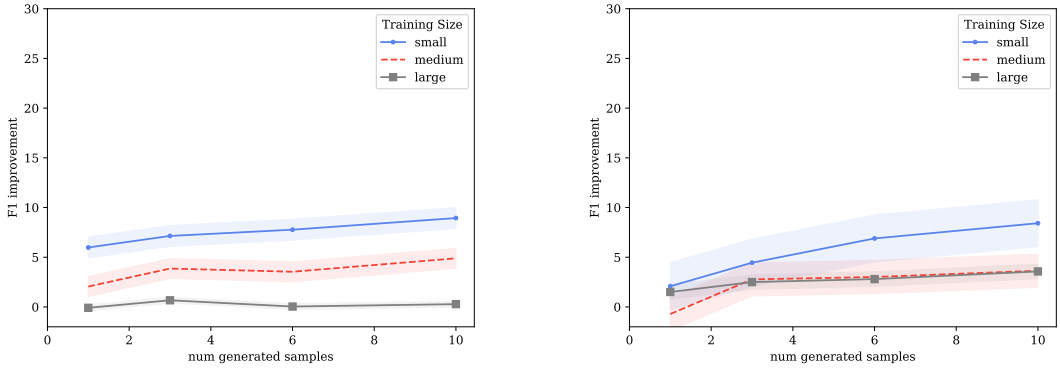
4.4.3 Results and Analysis

We report the evaluation results of various augmentation techniques on the test sets of MaScip and S800 in Table 4.2 and Table 4.3. We use the F1-score to evaluate the performance of the NER models. All experiments are repeated three times with different random seeds. We also report the mean values and standard deviations. The Δ row shows the averaged improvement due to data augmentation for each embedding type across both datasets. For the most part, all data augmentation techniques improve over the baseline; backtranslation results in the biggest average improvement for both context-independent *GloVe* and contextualized *SciBERT* embeddings under different data usage percentiles. We attribute the improved performance of backtranslation to the generation of linguistically diverse and meaning-preserving *entity mention contexts* to enable the improved generalization of the underlying NER model.

In general, we find that the data augmentation techniques contribute to the biggest improvement in performance when the training sets are small. As an example, all data augmentation methods achieve the highest improvements when the training set only contains 50 data points. In contrast, this effect is reduced as the training sets get larger (see columns **S** vs **F** in Table 4.2, Table 4.3). The augmentation on the complete training set even decreases the performances for some augmentation techniques. The performance impact of data augmentation on varying sizes of training sets has also been observed in the existing

work (Fadaee et al., 2017b; Dai and Adel, 2020; Ding et al., 2020).

We also investigate the effectiveness of data augmentation techniques on the mainstream contextualized (pretrained *SciBERT*) embeddings. All the augmentation techniques especially backtranslation result in better performance when compared to the baseline. However, the average performance improvement due to data augmentation with SciBERT embeddings is lower as compared to the GloVe embeddings.



(a) MaSciP - MR

(b) S800 - MR

(c) MaSciP - BT

(d) S800 - BT

Figure 4.3: The impact of the number of generated instances per original training instance on the overall performance. Here, MR: mention replacement, BT: back translation.

The number of augmented instances per training instance is an important hyperparameter and we would like to answer the question of how much augmentation is enough? We present an analysis of backtranslation and mention replacement to provide practical considerations regarding this hyperparameter setting. More specifically, we would like to know how the number of augmented training instances impacts the overall performance of the NER model. Figure 4.3 illustrates the impact of augmented instances per original training instance

on the NER model using SciBERT embeddings across the datasets. We use the absolute improvement in F1-score as the performance gain metric. We observe that a large number of augmented instances results in the highest gains in performance when the training sets are small. The gains are highest when the number of augmented instances is less than or equal to 6. It can be seen that the impact of data augmentation is highest when the training data is scarce and data augmentation does not provide much improvement for medium or large-size training sets.

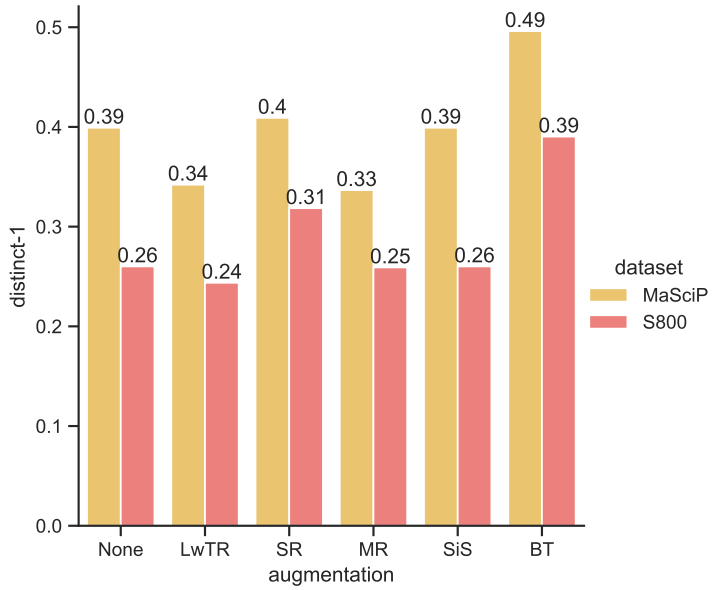


Figure 4.4: The diversity statistics of various augmentation techniques across the datasets.

In order to quantitatively measure the diversity introduced by various augmentation techniques, we report *distinct-1* metric (Li et al., 2016) in Figure 4.4. Distinct-1 quantifies the intra-text diversity by counting the distinct unigrams in each sentence, the count value is scaled by the total number of tokens in the sentence to avoid favouring longer sentences. Backtranslation has the highest level of unigram diversity, this is not very surprising as backtranslation is known to generate diverse linguistic variations.

4.5 Summary

In this chapter, we presented our work on data augmentation for NER. The data augmentation for NER is particularly challenging as NER is a token-level task and the augmentation methods should appropriately assign the correct label to the individual word token for every synthetically generated instance. We propose *backtranslation* as a data augmentation strategy for NER. In a nutshell, our algorithm splits the training sentence into a sequence of entity mention(s) and their contexts, entity mention contexts are paraphrased using

backtranslation. We demonstrate that backtranslation can generate high-quality coherent, linguistically diverse and meaning-preserving synthetic data for the token-level sequence labelling task of NER.

We employed BiLSTM-CRF as the supervised NER model and experimented with context-independent GloVe and contextualized SciBERT embeddings. We report the experimental results on two domain-specific datasets, MaScip (material science domain) and S800 (biomedical domain). We simulate a low-resource setting by using different subsets of training data for training the supervised NER model. Our empirical results demonstrate that backtranslation can improve over the baseline augmentation methods and is a competitive data augmentation strategy for NER. We found that the data augmentation techniques result in the biggest performance gains when the training sets are small, in the case when enough training data is available the data augmentation does not result in performance improvement. We employ the distinct-1 metric to quantify the intra-text diversity introduced by various augmentation techniques. Our analysis found that data augmentation via backtranslation achieves the highest level of unigram diversity. To the best of our knowledge, we are the first to employ backtranslation as the data augmentation strategy for the token-level task of NER.

Chapter 5

Semi-Supervised Bootstrapping for Relation Extraction

The work done in this chapter is the result of a research collaboration with Dr Pankaj Gupta. The initial idea was proposed by Pankaj; I further refined the idea, implemented the model and performed the evaluation experiments. Pankaj was involved in the weekly discussions.

As motivated in Chapter 2, relation extraction is a crucial component of the information extraction pipeline and is necessary to populate knowledge bases to enable fact retrieval. To extract a triplet like (Hasso Plattner, founder_of, SAP) from text, a relation extraction system needs to determine that *founder_of* relation exists between the person *Hasso Plattner* and the company *SAP*. Most RE methods [1] assume that the entities in the text have already been detected by an existing named entity tagger.

The existing mainstream techniques for relation extraction can broadly be classified into five categories:

1. *Rule-based*: The approach consists of carefully creating hand-crafted patterns to identify relation instances in the text. Due to a lack of sufficient annotated data, rule-based systems dominate industrial Information Extraction (IE) technologies (Chiticariu et al., 2013b). Moreover, rules are interpretable, can incorporate domain knowledge, and are easy to maintain by non-experts to debug and fix errors. Rules are usually specified by creating patterns around the entities (Yangarber et al., 2002) or entity pairs (Hearst, 1992).

2. *Supervised RE*: The existing mainstream techniques for relation extraction rely on supervised machine learning methods, mainly neural network based methods (Wang et al., 2016; Zhang et al., 2017b; Guo et al., 2019). One of the major caveats of these supervised approaches is that they are data-hungry and require a large amount of labelled training data to train effective RE models. Collecting large annotated training

[1] Joint Entity and Relation Extraction methods are an exception.

datasets is often not possible in most real-world scenarios. Supervised methods can achieve high performance for the task of RE but their reliance on a large amount of labelled training data makes them difficult to apply to new domains and in many real-world especially industrial use cases and scenarios.

3. *Distant Supervision*: Distant supervision attempts to address the lack of available labelled training data and automates the process of generating training data for relation extraction (Mintz et al., 2009). This automatic annotation is performed due to the assumption that if two entities participate in a relation (determined by a lookup in KB or DB), all sentences that mention these two entities express that relation. Unfortunately, this assumption is too unconstrained and therefore, this automatic labelling results in lots of false positives in the training instances; to circumvent this added noise the RE systems based on distant supervision need to develop special methods to deal with this added noise. Another drawback of the distant supervision methods is the assumption that there exists a KB or DB which is sufficiently populated to enable automatic annotations, this assumption might not be true for all domains and use-cases.

4. *Open Information Extraction*: The methods under the umbrella of Open Information Extraction employ a set of very general constraints and heuristics to create extractions that represent relations in the text (typically at the sentence level). Recently, open information extraction systems (Fader et al., 2011; Mausam et al., 2012; Angeli et al., 2015) have been popular for specialized domain or generalized pattern learning.

5. *Weakly Supervised RE*: The weakly supervised RE systems rely on weak supervisory signals to identify relation instances. Bootstrapping methods for RE is a dominant approach for weakly supervised RE. The bootstrapping method takes as input a set of seed entity pairs of a target relation and iteratively expands it by scanning the corpus for entity pairs belonging to the target relation. One of the biggest advantages of bootstrapping is that it does not require access to labelled data and generally bootstrapping methods have a higher recall, but bootstrapping methods have to deal with the challenge of accumulated noise in each iteration as this added noise can drastically decrease the overall precision of the extractions.

This chapter will focus on weakly-supervised bootstrapping methods for relation extraction and our developed techniques to mitigate noise which is inherently produced as part of the bootstrapping process.

5.1 Bootstrapping for Relation Extraction

Bootstrapping methods for relation extraction provides an attractive alternative to mainstream data-hungry supervised neural network models; they are domain and language independent, require minimal linguistic pre-processing and can be applied to raw text, and

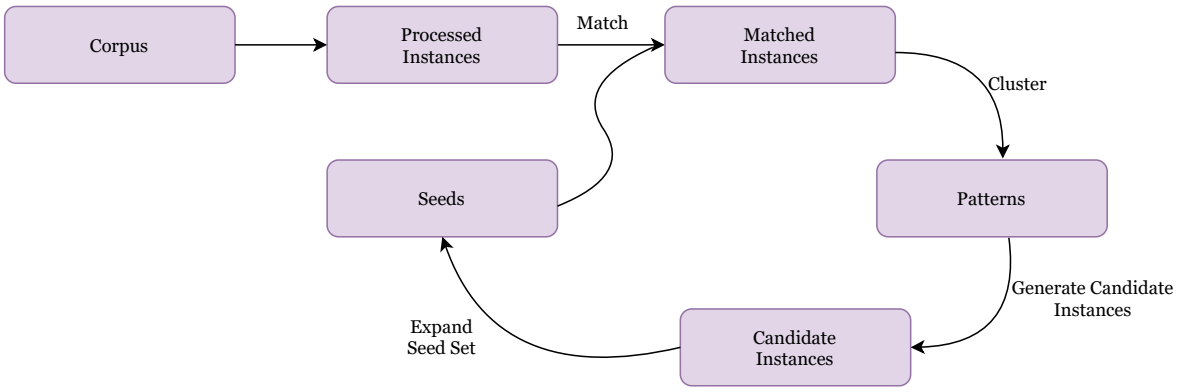


Figure 5.1: The overview of bootstrapping process.

are efficient enough for large-scale extraction (Paşca et al., 2006). Bootstrapping methods require an initial set of seed entity pairs for a target binary relation to operate on a large collection of text documents. Figure 5.1 illustrates the key components of the bootstrapping process. The process involves scanning the text corpus for positive seed occurrences to create extraction patterns [2], where we define the extraction pattern as a cluster of instances generated from the matched seed contexts. The confident extraction patterns iteratively expand the initial seed set by discovering new relationship instances in the corpus. The process is repeated until a stopping criterion is met.

The bootstrapping methods distil implicit supervisory signal from the seed set, the strength of this signal is often very weak and hence these methods are prone to accumulate noise over time. The iterative nature of bootstrapping results in a snowball effect for this added noise, this phenomenon is known as *semantic drift* (see Figure 5.2). The root cause of semantic drift is the confident *noisy patterns* as they are responsible for augmenting the seed set with erroneous relationship extractions (Gupta et al., 2018).

The bootstrapping methods score extraction patterns by their ability to extract more positive extractions and less negative extractions. The pattern scoring mechanism in bootstrapping has to deal with the unobserved information of unknown extractions. It is not obvious how to treat unknown extractions; the conservative criterion of treating them as negative extractions suffer from closed world assumption and incorrectly penalizes non-noisy patterns, whereas treating them as positive extractions may result in boosting the confidence of noisy patterns (Gupta and Manning, 2014a). The former criterion leads to lower recall while the latter result in lower precision. The soft version of the latter criterion is common and hence existing methods are prone to semantic drift.

In this chapter, we discuss two of our proposed methods to explicitly identify and then remove noisy extraction patterns to avoid semantic drift in the subsequent iterations. We take inspiration from the insights that: a). Valid relationship extraction patterns have semantic conformity b). Language Models (LM) (Peters et al., 2018b; Devlin et al., 2019; Clark et al., 2020; Yang et al., 2019b) trained on large text corpus capture relation specific

[2]Throughout the chapter, we use the term extraction pattern and pattern interchangeably.

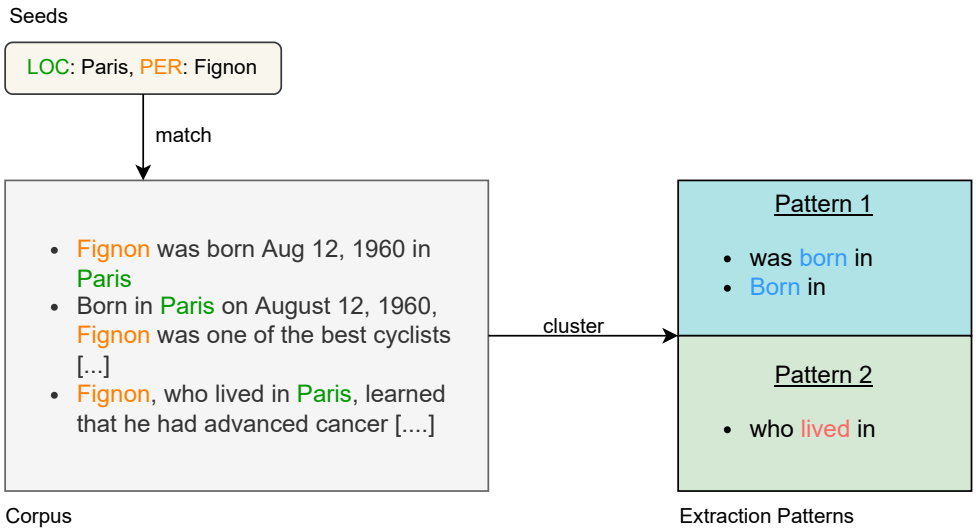


Figure 5.2: An illustration of extraction patterns resulting from the clustering of matched seed instances in the text corpus for *place-of-birth* relation. Observe that Pattern 1 is a valid pattern for *place-of-birth* relation; however, Pattern 2 is an invalid pattern for the target relation, in fact, it corresponds to a different relation i.e. *place-of-residence*. Pattern 2 will cause semantic drift in the bootstrapping process for the target (*place-of-birth*) relation.

knowledge (Petroni et al., 2019; Zhang et al., 2019c; Kassner and Schütze, 2020). Our first method *Maximum Spanning Tree pruning (MST-prune)* encodes extraction patterns in a graph structure to enforce *relatedness constraint* by pruning anomalous patterns. Our second method *Language Model pruning (LM-prune)* exploits encoded knowledge stored in LM to estimate the correctness of extraction patterns. Our constraining methods rely on the value of *pruning threshold* to prune the noisy patterns, we propose a simple yet effective adaptive threshold scheduler to dynamically optimize the pruning threshold parameter.

We empirically demonstrates the effectiveness of our proposed pruning methods on four relationships, our system demonstrate a 17.7% gain in F1 score on average by significantly improving the precision of the unconstrained bootstrapping system.

5.2 Related Work

Bootstrapping has many variants and has been applied to many natural language processing tasks. Here, we mainly discuss bootstrapping methods for relation extraction and existing approaches to address semantic drift in bootstrapping.

The first bootstrapping system for RE was developed by Brin and was called *DIPRE* ("Dual Iterative Pattern Relation Expansion") (Brin, 1998). DIPRE was employed to create a curated list of book titles and their authors from the free text in the webpages across the world wide web, essentially detecting *author-of* relationship between the entities of

type person and book. DIPRE matches occurrences of seed entity pairs in the text using regular expressions and generates extraction patterns by clustering matched contexts. It is important to note that the matched context is represented as three contexts of string: words before the first entity (*before context*), words between the two entities (*between context*), and words after the second entity (*after context*). DIPRE controls semantic drift by limiting the number of instances a pattern can extract. Agichtein and Gravano introduced *Snowball* (Agichtein and Gravano, 2000), a bootstrapping RE system inspired by DIPRE. Snowball improves the DIPRE's context representation by computing a TF-IDF representation of each context. Snowball employ several strategies for estimating the reliability of extraction patterns and tuples. Furthermore, Snowball proposed a scalable evaluation methodology and associated metrics for large-scale system evaluation (a default environment for the bootstrapping methods). Batista et al. introduced *BREDS* (Bootstrapping Relationship Extraction with Distributional Semantics), BREDS improves Snowball by replacing TF-IDF based context representation with the word embeddings (Mikolov et al., 2013a), now each context is represented using a sum of their word vectors. BREDS achieve significant performance improvement in terms of precision and recall over the Snowball system. Gupta et al. developed *BREJ* (Joint Bootstrapping Machines for High Confidence Relation Extraction), BREJ can bootstrap relations using both the seed entity pairs and seed templates (patterns).

Carlson et al. (2010) argue that the default setup of semi-supervised learning algorithms including vanilla bootstrapping is unconstrained and due to this characteristic these systems are often unreliable and have unacceptable accuracy. The limited number of labelled examples or seeds is not sufficient to constrain the learning process, this results in the introduction of noise during training which results in a significant reduction in the precision of the system eventually leading to semantic drift. To address this problem of unconstrained learning, the existing work introduces various kinds of constraints to keep a check on the learning process. One popular learning constraint in prior work is of *mutual exclusion*, mutual exclusion assumes that the semantic classes are mutually exclusive; this constraint is realized by running multiple bootstrapping systems in parallel and ignoring the extractions detected by parallel runs. Mutual Exclusion can be applied among different semantic classes (Thelen and Riloff, 2002; Lin et al., 2003; Curran et al., 2008) or between only a positive and a negative class (McIntosh, 2010). One downside of mutual exclusion is the assumption that one extraction can only belong to one semantic category, overlapping semantic categories violates this assumption.

McIntosh and Curran demonstrates that the choice of seeds greatly impacts the final performance of the bootstrapping algorithm and favourable seeds for one algorithm can perform poorly with others making comparisons unreliable. They exploit this shortcoming to sample from automatically extracted seeds and employ bagging to train multiple bootstrapping variants to reduce semantic drift; they also introduce a weighting scheme in pattern scoring for robust pattern confidence estimation. Komachi et al. (2008) draws a parallel of semantic drift in bootstrapping to topic drift in Hyperlink-Induced-Topic-Search (HITS) algorithm (Kleinberg, 1999) and demonstrate that von Neumann kernels (Kandola et al., 2002) and the regularized Laplacian (Smola and Kondor, 2003) reduce semantic drift in bootstrapping.

type	a named entity type, e.g., `location`
typed entity	a typed entity, e.g., <"Rome",`location`>
entity pair	a pair of two typed entities
context vector	a triple of vectors ($\vec{v}_{-1}$, $\vec{v}_0$, $\vec{v}_1$)
instance	an entity pair and a context vector
γ	instance set extracted from corpus
i	a member of γ, i.e., an instance
$x(i)$	the entity pair of instance i
γ_r	subset of instances γ describing target relationship $\mathcal{R}$
ζ_p	a set of positive seed entity pairs
ζ_n	a set of negative seed entity pairs
ξ_p	a set of positive seed templates
ξ_n	a set of negative seed templates
N	number of iterations
ρ	cluster of instances (*extraction pattern*)
ρ_{NNHC}	Non-Noisy-High-Confidence pattern (True Positive)
ρ_{NNLC}	Non-Noisy-Low-Confidence pattern (True Negative)
ρ_{NHC}	Noisy-High-Confidence pattern (False Positive)
ρ_{NLC}	Noisy-Low-Confidence pattern (False Negative)
λ	a set of extraction patterns

Table 5.1: Notation and definition of key terms

Gupta and Manning (2014a) improve pattern scoring by estimating the semantic class of unknown entities to implicitly mitigate semantic drift; they use various unsupervised features based on contrasting domain-specific and general text, exploiting distributional similarity, TF-IDF scores and edit distances to learned entities. A popular heuristic in prior work to determine the correctness of an extraction is to compare the distributional similarity between novel extraction and the already known confident extractions to avoid augmenting noisy extraction to the seed set (McIntosh and Curran, 2009; McIntosh, 2010; Gupta and Manning, 2015).

We would like to note that the above-mentioned prior work addresses semantic drift in bootstrapping for named entity recognition. In contrast, we focus on mitigating semantic drift for relation extraction.

Algorithm 2 Constrained Bootstrapping

Input: γ, ζ_p, ζ_n
Output: ζ_p^+, γ_r

$\zeta_p^+ \leftarrow \zeta_p$
$\gamma_r \leftarrow \emptyset$
$n \leftarrow 0$
while not_converged(ζ_p^+) and n < N **do**
  $\Theta \leftarrow match\_instances(\gamma, \zeta_p^+)$
  $\lambda^\circ \leftarrow single\_pass\_clustering(\Theta, \tau_{sim})$
  $\lambda \leftarrow$ **filter_noise**(λ°)
  $\gamma_c \leftarrow \emptyset$
  for $i \leftarrow \gamma$ **do**
    $sim \leftarrow 0$
    for $\rho \leftarrow \lambda$ **do**
      $accept, score = attract\_instance(i, \rho)$
      if *accept* **then**
        $update\_confidence(\rho, \zeta_p, \zeta_n)$
        if $score >= sim$ **then**
          $sim \leftarrow score$
    if $sim >= \tau_{sim}$ **then**
      $\gamma_c \leftarrow \gamma_c \cup i$
  for $i \leftarrow \gamma_c$ **do**
    if $instance\_confidence(i) > \tau_{conf}$ **then**
      $\zeta_p^+ \leftarrow add\_entity\_pair(\zeta_p^+, i)$
      $\gamma_r \leftarrow \gamma_r \cup i$
  $n \leftarrow n + 1$

5.3 Method

5.3.1 Notation and Background

We introduce the notation and terminology in Table 5.1, our formulation is inspired from BREJ (Gupta et al., 2018).

Given a sentence expressing a relation like "x_{head} was born in y_{tail}", where x_{head} is the head entity and y_{tail} is the tail entity; the task is to extract entity pairs from a corpus for which the relationship holds. We assume that the arguments of a relation are typed, e.g. x is a person and y is a location. As a preprocessing step, we run a named entity recognizer to tag all the candidate entities in the corpus.

For a sentence in a corpus with a particular type of entity pairs, we define three vectors that represent the context of x and y. $\vec{v}_{-1}$ represents the context before x, $\vec{v}_0$ the context between x and y and $\vec{v}_1$ the context after y. These context vectors are simply the sum of word embeddings of the corresponding word tokens. An *instance* joins an entity pair and context vector.

The first step in bootstrapping is to extract a set of instances from the text corpus. We

refer to this set as γ. We will use i and j to refer to instances. $x(i)$ is the entity pair of instance i.

The input to bootstrapping algorithm is a sets of positive and negative seeds for entity pairs (ζ_p, ζ_n) (Batista et al., 2015b), or templates(ξ_p, ξ_n) (Gupta et al., 2018) or both (Gupta et al., 2018). We run bootstrapping algorithm for N iterations where N is a parameter.

The similarity of two instances is given as a weighted sum of the dot products of their before contexts ($\vec{v}_{-1}$), their between contexts ($\vec{v}_0$) and their after contexts ($\vec{v}_1$) (Batista et al., 2015b; Gupta et al., 2018):

$$sim_{match}(i,j) = \sum_{p \in \{-1,0,1\}} w_p \vec{v}_p(i) \vec{v}_p(j) \tag{5.1}$$

where the weights w are parameters.

The similarity between an instance i and a cluster ρ of instances is defined as the maximum similarity of i with any member of the cluster.

The extraction pattern ρ can be categorized as follows (Gupta et al., 2018):

$$\rho_{NNHC} = \underbrace{\rho \mapsto \mathcal{R}}_{non-noisy} \wedge \text{conf}(\rho, \zeta_p, \zeta_n) \geq \tau_{conf} \tag{5.2}$$

$$\rho_{NNLC} = \rho \mapsto \mathcal{R} \wedge \text{conf}(\rho, \zeta_p, \zeta_n) < \tau_{conf} \tag{5.3}$$

$$\rho_{NHC} = \underbrace{\rho \not\mapsto \mathcal{R}}_{noisy} \wedge \text{conf}(\rho, \zeta_p, \zeta_n) \geq \tau_{conf} \tag{5.4}$$

$$\rho_{NLC} = \rho \not\mapsto \mathcal{R} \wedge \ \text{conf}(\rho, \zeta_p, \zeta_n) < \tau_{conf} \tag{5.5}$$

where $\mathcal{R}$ is the target relation to be bootstrapped. ρ_{NHC} is called as a *noisy-high-confidence* extraction pattern as it does not represents the target relation (i.e., $\rho \not\mapsto \mathcal{R}$), but still (incorrectly) has a confidence score above a certain threshold (τ_{conf}). The extraction patterns of type ρ_{NHC} are the primary culprit behind semantic drift in bootstrapping. In this work we explicitly remove extraction patterns of type ρ_{NHC} and ρ_{NLC}.

5.3.2 Constrained Bootstrapping

Algorithm 2 illustrates the steps of *constrained* bootstrapping for BREDS (Batista et al., 2015b), extension to BRET (Gupta et al., 2018) and BREJ (Gupta et al., 2018) is straightforward. The input to constrained bootstrapping is the set γ of instances extracted from a corpus, a set of positive seeds (ζ_p) and a set of negative seeds (ζ_n). ζ_p^+ collects the entity pairs that bootstrapping extracts in several iterations. In each iteration, the first step is to gather instances which are similar to the positive seeds, where similarity is defined as a weighted sum of the corresponding context vectors ($\vec{v}$) (see equation 5.1).

The collected instances are then clustered using a *single-pass clustering* algorithm (Agichtein and Gravano, 2000; Batista et al., 2015b), which assigns an instance i to the first cluster whose similarity is equal or above the threshold parameter τ_{sim}. The output of single-pass clustering is the set of extraction patterns λ° which also contains the noisy extraction

Algorithm 3 Maximum Spanning Tree Pruning

Input: λ
Output: $\lambda_{filtered}$
1: $V, E \leftarrow construct\_graph(\lambda)$
2: $E \leftarrow -1 \times E$
3: $T \leftarrow Kruskal(V, E)$
4: $T_p \leftarrow prune\_vertices(T, \tau_{prune})$
5: $\lambda_{filtered} \leftarrow flatten\_tree(T_p)$

patterns (see equation 5.4, 5.5), the primary source of semantic drift. We constrain the bootstrapping process by applying a filtering mechanism to detect the noisy extraction patterns to curb semantic drift in the current and subsequent iterations. Note that detection of noisy extraction patterns is particularly challenging as there is no supervisory signal besides initial positive and negative seed entity pairs.

The final step of the bootstrapping process is to find valid relationship instances γ_r for the target relation $\mathcal{R}$, this is performed in two steps. First, we identify the potential candidate relationship instances γ_c from the set of instances γ, every instance i with similarity equal to or above the parameter τ_{sim} with an extraction pattern ρ is a potential candidate relationship instance. Simultaneously, we also update the pattern confidence score based on the instances it extracts using this formula (Batista et al., 2015b):

$$conf(\rho) = \frac{|P|}{|P| + w_n.|N| + w_u.|U|} \tag{5.6}$$

where, $|P|$ is the count of *positive* extractions, $|N|$ is the count of *negative* extractions, $|U|$ is the count of *unknown* extractions and w is the corresponding weight parameter. The extraction is considered positive if it is part of the positive seed set (ζ_p), it is considered negative if it is part of the negative seed set (ζ_n) and it is considered unknown if is not the part of positive and negative seed set.

Finally, we compute the confidence score of the candidate relationship instances based on the pattern which extracted them using this formula (Batista et al., 2015b):

$$conf(i) = 1 - \prod_{j=0}^{|\rho|}(1 - conf(\rho_j) \times sim(i, \rho_j)) \tag{5.7}$$

where $sim(i, \rho_j)$ is the maximum value of similarity between the candidate relationship instance i and any instance of the extraction pattern ρ_j. The candidate relationship instances with the confidence score equal to or above the threshold parameter τ_{conf} are added to the set of relationship instances γ_r. The entity pairs of the relationship instances (γ_r) are appended to the positive seed set ζ_p^+ to expand the search space of bootstrapping for the next iteration.

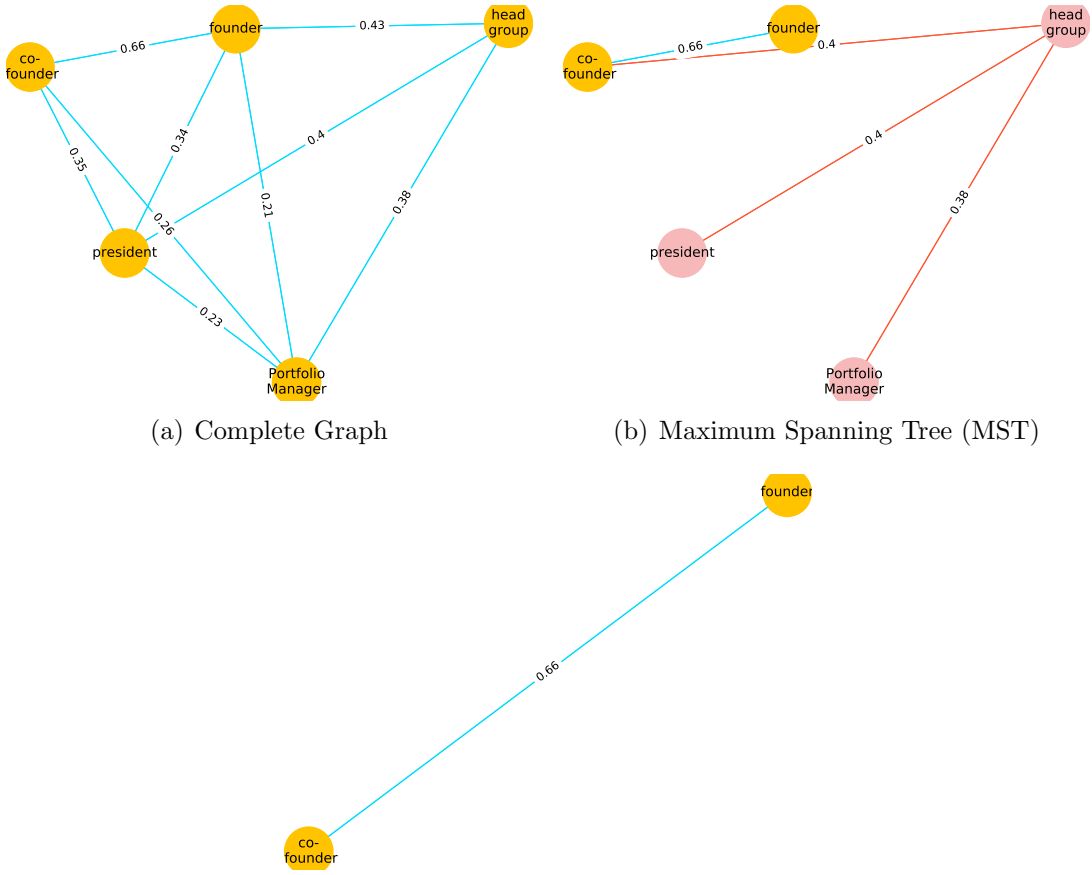
Maximum Spanning Tree Pruning

Our first strategy to constrain the bootstrapping process is called *Maximum Spanning Tree Pruning* (MST-prune) and it leverages the insight that true relationship extraction patterns have semantic coherence and relatedness. *MST-prune* encode extraction patterns in a graph structure and incorporate the relatedness constraint to identify noisy and incoherent patterns. Algorithm 3 illustrates the steps of our MST-prune algorithm. Given a set of extraction patterns λ, we construct an undirected graph $G(V, E)$ such that an individual vertex v corresponds to an extraction pattern ρ and an edge weight e corresponds to the distributional similarity between the adjacent vertices. Note that $G(V, E)$ is a *complete graph* i.e. every vertex is connected to every other vertex. Our goal is to find a sub-graph $T(V, E')$ such that the sum of its edge weights is maximum i.e. T is a *Maximum Spanning Tree* of G. We apply the Kruskal algorithm (Kruskal, 1956) with inverted edge weights to find a maximum spanning tree. We enforce *relatedness constraint* by pruning noisy and uninformative vertices in the maximum spanning tree T with the edge weight below the threshold parameter τ_{prune}. One important detail in MST-prune is the representation of pattern context i.e. vertex in a graph G, as the edge weights, e is directly influenced by the pattern context representation. We represent pattern context by the dominant noun and verb phrase across the instances in a pattern.

Figure 5.3 visually illustrates individual steps of maximum spanning tree pruning for the *founded-by* relation in the first bootstrapping iteration. The text of the vertex is the string representation of pattern context i.e. dominant noun and verb phrase across instances in the pattern. In the first bootstrapping iteration, five extraction patterns are created. Although all the created extraction patterns are related to employment designations but only *founder* and *co-founder* are informative patterns for the target relation *founded-by*. In the *complete graph* (figure 5.3 (left)), it is difficult to identify any useful structure. However, notice that the maximum spanning tree of the complete graph (figure 5.3 (centre)) already reveals an interesting pattern, all the uninformative patterns have significantly lower edge weights. As a final step, relatedness constraint is enforced by pruning uninformative extraction patterns with the edge weight below the threshold parameter τ_{prune} (figure 5.3 (right)).

Language Model Pruning

Language Model Pruning (LM-prune) is our second strategy to constrain the bootstrapping process, which takes inspiration from recent work on exploiting knowledge encoded in deep contextualized pre-trained representation models trained on a large collection of text documents (Petroni et al., 2019; Zhang et al., 2019c; Kassner and Schütze, 2020). Existing research has demonstrated the pre-trained language model's capability to capture relation-specific knowledge. We exploit this intuition to query a pre-trained language model to determine the correctness of an extraction pattern, this is in contrast to existing work which treats the language model as a Knowledge Base (Petroni et al., 2019). Concretely, we construct query string as "*before-context head-entity between-context [MASK] after-context*"; language model is expected to fill in the masked token i.e. tail-entity. Note that we do



(a) Complete Graph

(b) Maximum Spanning Tree (MST)

(c) MST-Pruned

Figure 5.3: An illustration of Maximum Spanning Tree pruning (*MST-prune*) for *founded-by* relation during the first bootstrapping epoch.

not expect the language model to predict the true tail entity mention as understandably rare entities are difficult to predict by the language model; instead, we expect the language model to predict the tail entity mention with the same entity type as that of true tail entity (see Figure 5.4). The extracted patterns which do not align with the language model are treated as unrelated and noisy for the target relation $\mathcal{R}$. In practice we consider the percentage of aligned instances in a pattern to determine its correctness, the threshold parameter τ_{prune}[3] enforce the exact strength of this alignment. Algorithm 4 illustrates the detailed steps of LM-prune. In our work, we employ BERT (Devlin et al., 2019) as the underlying query language model.

[3]Note that τ_{prune} has a different interpretation for LM-prune and MST-prune.

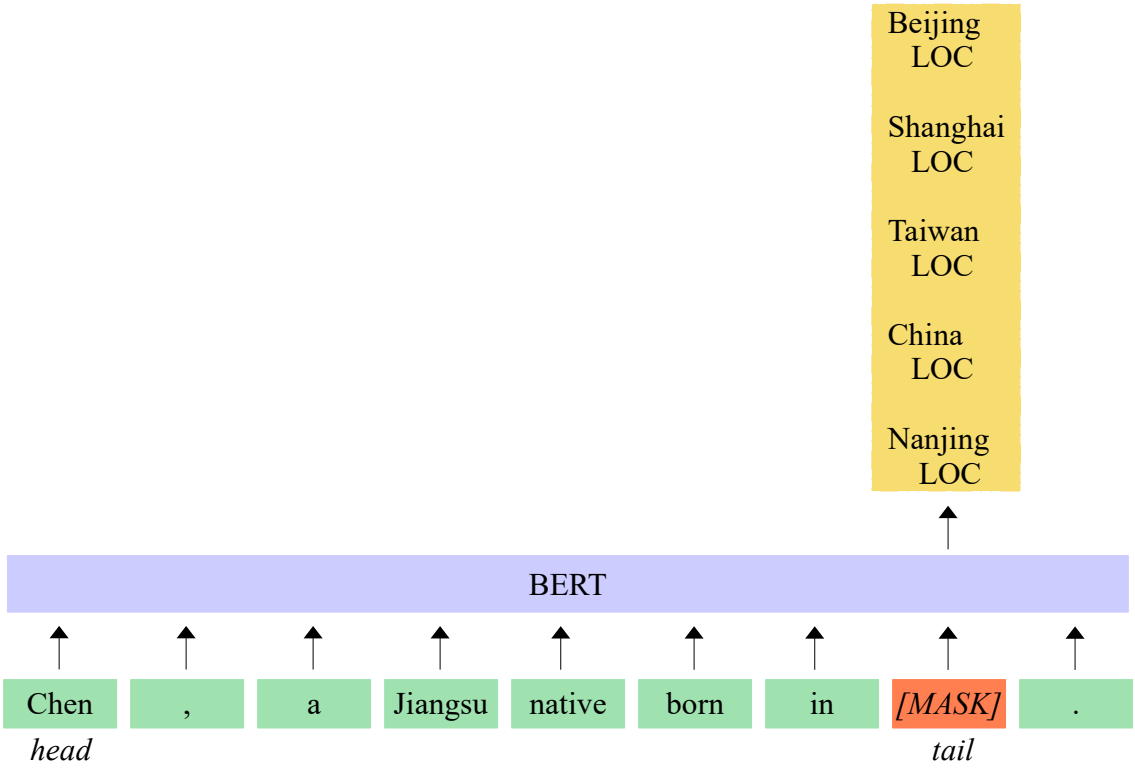


Figure 5.4: An Illustration of querying *masked tail entity* from BERT. Note that instead of matching the tail entity string (from the original text) we match the entity type of top k predictions.

5.3.3 Adaptive Threshold

The bootstrapping constraining methods (MST-prune and LM-prune) rely on the threshold parameter τ_{prune} to identify noisy extraction patterns. The static value of τ_{prune} is limiting because a lower value will *under-prune* the bootstrapping system i.e. it will result in semantic drift and hence lower precision while the higher value will *over-prune* the bootstrapping system i.e. over constrained learning setting and hence lower recall. It is difficult to find the static optimal value, moreover, it may vary depending on the target relation. We propose a simple dynamic update rule for τ_{prune} which take into account the counts of extraction patterns in the adjacent bootstrapping iterations to compute the optimal value.

$$\tau_{prune} = \tau_{prune_{n-1}} + log(\frac{|\lambda_n|}{|\lambda_{n-1}|}) \times \alpha \tag{5.8}$$

where α is the step size, n denotes the current iteration and $|\lambda|$ is the count of extraction patterns in the respective iteration. Note that equation 5.8 only dynamically increase τ_{prune} to constrain the count of extraction patterns, in case τ_{prune} is already high i.e. $|\lambda_n| < |\lambda_{n-1}|$, we decrease the τ_{prune} by a factor of 2 to encourage expansion of extraction patterns in the current iteration.

Algorithm 4 Language Model Pruning

Input: λ
Output: $\lambda_{filtered}$
$\lambda_{filtered} \leftarrow \emptyset$
for $\rho \leftarrow \lambda$ **do**
  $valid\_patterns \leftarrow 0$
  for $i \leftarrow \rho$ **do**
    $q \leftarrow construct\_query(i)$
    $o \leftarrow fill\_mask(q)$
    if $entity\_type(o) == entity\_type(i.object)$ **then**
      $valid\_patterns \leftarrow valid\_patterns + 1$
  if $\frac{valid\_patterns}{len(\rho)} >= \tau_{prune}$ **then**
    $\lambda_{filtered} \leftarrow \lambda_{filtered} \cup \rho$

5.4 Experiment and Results

5.4.1 Dataset and Experimental Setup

In our evaluation, we used public and widely used sentence-level relation extraction dataset TACRED (Zhang et al., 2017b) . We consider four relationships: *place-of-birth* (PER-LOC), *place-of-death* (PER-LOC), *founded-by* (PER-ORG) and *subsidiaries* (ORG-ORG). The original TACRED dataset contains fine-grained annotation at country, state and city level for place-of-birth and place-of-death relations. In our experiments we club the respective country, state, and city level annotations together to create a single respective annotation class i.e. place-of-birth and place-of-death. We report the statistics of the relationships and counts of entity types in table 5.3 and table 5.4 respectively.

We bootstrap relations in unconstrained BREDS (Batista et al., 2015b), BRET (Gupta et al., 2018), BREJ (Gupta et al., 2018) and constrained (LM-prune, MST-prune) fashion using seed entity pairs and templates (Table 5.5). We re-run *BREDS*, *BRET* and *BREJ* as the unconstrained baselines. We used spaCy[4] (Honnibal et al., 2019) to find part-of-speech (POS) tags of tokens and entity type of *masked entity* prediction in LM-prune. We consider the top k LM's predictions for masked object entity. We only consider entity pairs with a maximum of 8 tokens, and a window of 2 tokens for before and after context. When creating extraction patterns, we discard patterns with less than 2 instances. We only consider extracted relationship instances with the confidence score $conf(\rho, \zeta_p, \zeta_n)$ equal or above 0.5. We follow BREDS (Batista et al., 2015b) to identify the presence of passive voice using part-of-speech (PoS) tags to determine the correct order of entities in a relational tuple, where we identify the presence of passive voice by considering any form of the verb *to be*, followed by a verb in the past tense or past participle, and ending in the word *by*. We report micro-averaged precision, recall and F_1 scores as the final performance metric. Table 5.2 lists the optimal values of hyperparameters. In our experiments, we employ GloVe word

[4] `https://spacy.io/`

Parameter	*Description*	*Value*
$\vec{v}_{-1}$	weight of before context	0.0
$\vec{v}_0$	weight of between context	1.0
$\vec{v}_1$	weight of after context	0.0
$\|v_{-1}\|$	token count of before context	2
$\|v_0\|$	token count of between context	8
$\|v_1\|$	token count of after context	2
τ_{sim}	similarity threshold	0.5
τ_{conf}	instance confidence	0.5
w_n	weight of negative extraction	2.0
w_u	weight of unknown extraction	0.0
N	maximum iterations	4
τ_{prune}	prune threshold	0.45
α	step size for τ_{prune}	0.3
k	top k LM predictions	5

Table 5.2: Hyperparameters for *Constrained Bootstrapping* including *MST-prune* and *LM-prune*.

	place-of-birth	*place-of-death*	*founded-by*	*subsidiaries*
# sentences	228	392	268	453

Table 5.3: The sentence counts of the target relationships in the TACRED dataset.

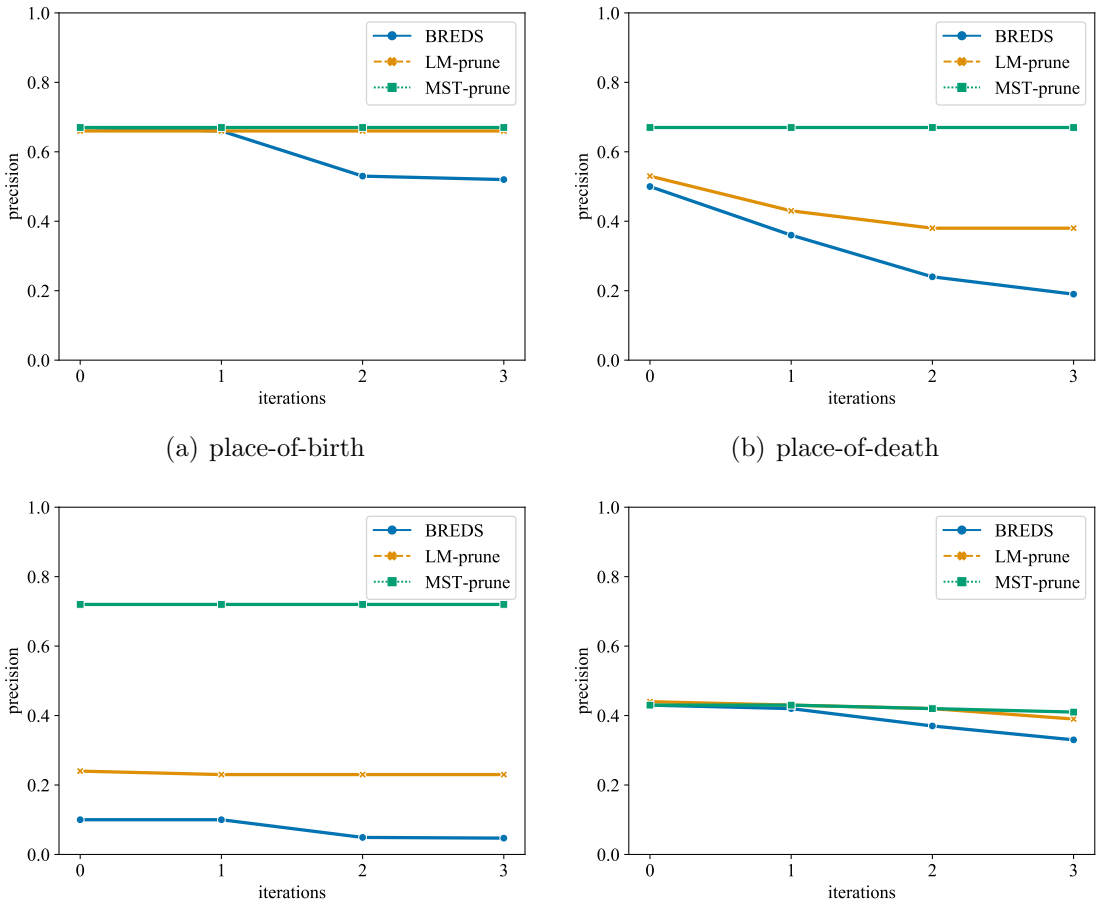
embeddings (Pennington et al., 2014).

	PER-LOC	*PER-ORG*	*ORG-ORG*
count	9995	17681	51800

Table 5.4: The count of sentences with the respective entity-type pair combination in the TACRED corpus.

Relationship	*Seed Entity Pairs*	*Seed Templates*
place-of-birth	{Venezuela, Hugo Chavez}, {Paris, Fignon} {Hong Kong, Chen}, {Potomac, Gross} {Germany, Murat Kurnaz}	[X] born in [Y]
place-of-death	{Paris, Pascal Yoadimnadji}, {Nepal, Girija Prasad Koirala} {Russia, Maria Kaczynska}, {Seoul, Hwang} {Saudi Arabia, Mohammed Sayed Tantawi}	[X] died in [Y]
founded-by	{Galleon Group, Raj Rajaratnam}, {Corporate Library, Nell Minow} {National Action Network, Al Sharpton}, {Focus on the Family, James Dobson} {ShopperTrak, Bill Martin}	[X] is founder of [Y]
subsidiaries	{Alcatel, Lucent}, {USA Network, Burn Notice} {Cunard Line, Carnival Corp.}, {FirstGroup, Laidlaw} {DCC, Fyffes}	[X], subsidiary of [Y]

Table 5.5: Seed Entity Pairs and Templates for each relationship.



(a) place-of-birth

(b) place-of-death

(c) founded-by

(d) subsidiaries

Figure 5.5: Precision plots of BREDS, LM-prune, and MST-prune for the four target relations. MST-prune boosts precision by 34% on average with a max difference of 67%

	Relationship	*Unconstrained Bootstrapping* P/R/F1	*LM-prune* P/R/F1	*MST-prune* P/R/F1
BREDS	place-of-birth	.52/.73/.61	**.67/.71/.69**	**.67/.71/.69**
	place-of-death	.19/.43/.27	.38/.33/.35	**.67/.35/.46**
	founded-by	.04/.39/.08	.23/.27/.25	**.72/.37/.49**
	subsidiaries	.33/.51/.40	.39/.42/.40	**.41/.47/.43**
BRET	place-of-birth	.68/.62/.65	.82/.61/.70	**.83/.62/.71**
	place-of-death	.15/.47/.23	.49/.24/.32	**.39/.31/.34**
	founded-by	.04/.65/.07	.17/.25/.20	**.41/.38/.39**
	subsidiaries	.30/.61/.40	.35/.42/.38	**.36/.49/.41**
BREJ	place-of-birth	.66/.64/.65	.67/.65/.66	**.83/.62/.71**
	place-of-death	.13/.49/.21	.17/.38/.24	**.22/.43/.29**
	founded-by	.04/.68/.07	.16/.30/.20	**.39/.43/.40**
	subsidiaries	.32/.57/.41	.38/.41/.39	**.40/.47/.43**

Table 5.6: Precision (P), Recall (R) and *F1* compared to the unconstrained bootstrapping system BREDS (Batista et al., 2015b), BRET (Gupta et al., 2018) and BREJ (Gupta et al., 2018).

5.4.2 Performance Comparison and Analysis

Table 7.1 shows the experimental results of unconstrained (BREDS, BRET and BREJ) and constrained (LM-prune, MST-prune) bootstrapping for four relationships. Observe that both constrained bootstrapping methods significantly improve precision for all the relationships when compared to the unconstrained baselines, constrained variants also achieve the overall best F1-score. For all the relations, *LM-prune* has lower recall than the respective unconstrained bootstrapping variants which suggests that non-noisy patterns were incorrectly pruned. *MST-prune* on the other hand has (minor) decrease in the recall but with a significant increase of 0.34, 0.20 and 0.17 points in precision on average for BREDS, BRET and BREJ respectively, when compared to the unconstrained baselines for all the relations. One interesting observation is that except for place-of-birth relation MST-prune and LM-prune perform best on BREDS, this is because BRET and BREJ accumulate too much noise for the remaining relations prohibiting effective pruning.

We analyse the extraction patterns extracted by BREDS, LM-prune and MST-prune to demonstrate the mitigation of semantic drift by the constrained bootstrapping variants for the four target relations. We manually label the extraction pattern as positive or negative. Table 5.7 shows the comparison of the characteristics of the extraction patterns. Observe that constraint variants have a significantly lower value of the $|\lambda_{NHC}|$ (*noisy-high-confident* patterns) than BREDS for all the four relations demonstrating the effective pruning of noisy extraction patterns. The lower value of $|\lambda_{NHC}|$ for MST-prune explains the improved precision of MST-prune as compared to LM-prune.

	Bootstrapping	$\|\lambda\|$	$\|\lambda_{NHC}\|$	$\|\lambda_{NLC}\|$	$\|\lambda_{NNHC}\|$	$\|\lambda_{NNLC}\|$
birth	BREDS	11	1	8	2	0
	LM-prune	2	**0**	0	2	0
	MST-prune	2	**0**	0	2	0
death	BREDS	62	51	9	2	0
	LM-prune	21	17	2	2	0
	MST-prune	2	**0**	0	2	0
founded-by	BREDS	32	23	5	4	0
	LM-prune	4	2	0	2	0
	MST-prune	3	**0**	0	3	0
subsidiary	BREDS	53	35	4	12	0
	LM-prune	35	29	1	5	0
	MST-prune	22	**16**	2	6	0

Table 5.7: Comparison of the characteristics of the extraction patterns extracted by the BREDS, LM-prune and MST-prune for the four target relations. Here $|\lambda|$ refers to the count of extraction patterns and $|\lambda_{NHC}|$ denotes the counts of *noisy-high-confidence* patterns (see Table 5.1).

Figure 5.5 shows that the constrained bootstrapping methods (LM-prune and MST-prune) maintain a stable precision as compared to the unconstrained BREDS across the bootstrapping iterations. On the other hand, the unconstrained bootstrapping system *BREDS* has an unstable precision curve for all the four relations; the precision is reduced to half as compared to the first iteration for place-of-death and founded-by relations.

MST-prune is purely driven by distributional (word embeddings) similarity which is effective to identify semantically dissimilar patterns. However, this absolute reliance on distributional similarity can fail in the following cases: a). relation verbs similar in meaning but far away in the embedding space (e.g. $sim(die, murder) = 0.23$) b). relation verbs dissimilar in meaning but close in the embedding space (e.g. $sim(born, married) = 0.64$), c). noisy extraction patterns with high distributional similarity to other noisy extraction patterns will be included in Maximum Spanning Tree with high edge weights and hence will not be pruned. LM-prune on the other hand focuses on the arguments of the relation template to filter the sentences expressing relations with inconsistent entity types. However, LM-prune cannot distinguish between relations involving same entity types, for example, the pre-trained language model may predict object entity of type *LOC* for sentences expressing place-of-birth (PER-LOC) and place-of-death (PER-LOC) relations.

Furthermore, our error analysis of LM-prune indicates the sensitivity of the pre-trained language model, a minor change in input text results in a significantly different prediction of the masked object entity token. In the case of MST-prune we observed that pattern context representation of dominant noun and verb is limiting, also the simple approach of summing the word embeddings makes the resultant embedding vector dense which has a

peculiar impact on the cosine similarity.

5.5 Summary

In this chapter, we explain the semi-supervised method for relation extraction called *bootstrapping*. Bootstrapping is an appealing technique for large-scale relation extraction as it does not require annotated data, and only needs a handful of *seeds* to bootstrap target relation from a large text corpus. In most cases, it is easy to obtain the seed(s) to initiate bootstrapping process. The strength of bootstrapping lies in its ability to operate without labelled data, and bootstrapping methods obtain an implicit supervisory signal from the seeds. Unfortunately, the strength of this supervision is often very weak and bootstrapping methods have difficulty differentiating noise from the valid extractions. Due to this, the noise gets introduced in the extractions and these noisy extractions have a snowball effect as they attract even more noise. This is known as *semantic drift*, and is a significant challenge for semi-supervised bootstrapping methods. The semantic drift significantly reduces the precision of the bootstrapping systems and hence makes the extractions of bootstrapping virtually useless for many real-world use cases.

The core reason behind semantic drift is the lack of supervision and the fact that bootstrapping process is too unconstrained, therefore, it attracts noise. We attempt to address the problem of semantic drift for the bootstrapping methods, we introduce two novel methods to constrain the bootstrapping process in order to minimise semantic drift for relation extraction. Our proposed methods explicitly identify and remove noisy extraction patterns to prevent contamination of subsequent iterations in bootstrapping. Our first constraining method (*MST-prune*) leverages graph theory and distributional similarity to detect anomalous noisy extraction patterns in bootstrapping. Our second method (*LM-prune*) exploit encoded knowledge stored in a large pre-trained language model to recalibrate the confidence score of an extraction pattern such that a low score value is assigned to the noisy extraction patterns.

We report experimental results on the TACRED dataset for four relations including place-of-birth, place-of-death, founded-by and subsidiaries; these results demonstrate that MST-prune and LM-prune can effectively mitigate semantic drift. Both MST-prune and LM-prune improve the overall performance of the bootstrapping system by obtaining significantly high precision, we also empirically demonstrate that our constraining methods maintain a stable precision during the lifecycle of the bootstrapping process.

Chapter 6

Data Augmentation for Relation Extraction

Data augmentation provides a remedy in the scarce data scenarios by augmenting the training dataset with the artificially generated training examples; the synthetic examples are usually created by modifying the original training examples in such a way that the underlying label semantics of the dataset are preserved. In fields like computer vision and speech, data augmentation is well studied. The discrete nature of language makes it challenging to apply data augmentation in the domain of natural language processing. In language, it is challenging to make sure that the manipulation of words or phrases preserves the meaning of the sentence or paragraph and the underlying label semantics are respected. In recent years, there has been significant interest in data augmentation for natural language processing tasks. Inspired by these efforts, we design and compare several data augmentation methods for relation extraction. We perform experiments on two datasets from the biomedical domain and empirically demonstrate that data augmentation can significantly boost the performance of the underlying machine learning models for domain-specific relation extraction in low and high resource scenarios.

6.1 Introduction

In the past few years, deep learning methods have dominated the natural language processing tasks such as sentiment analysis, text classification, named entity recognition and relation extraction; the ease of implementation and state-of-the-art performance contributed to this success and popularity of deep learning based methods. It is worth noting that deep learning based methods can only be trained effectively when sufficient training data is available (Conneau et al., 2017; Zhang et al., 2017a; Mohan and Li, 2019). Unfortunately, in many real-world scenarios, large training datasets are not available and this prohibits the use of deep learning based methods. This especially applies in the specialized domains such as the biomedical domain, as annotations for specialized domains require expert knowledge, and is time-consuming and expensive.

Researchers have proposed various methods to address low-data scenarios. Bootstrapping based methods use iterative clustering to expand the initial seed set with new extractions (Gupta and Manning, 2014b; Zhang et al., 2020a; Batista et al., 2015a). Self-supervised machine learning methods use limited available training data to train an initial machine learning model; this trained model is then used to augment the original (limited) training data with the trained model's confident predictions on the unlabelled examples (Qiu et al., 2009). Data augmentation methods expand the original training dataset with synthetic novel training instances while making sure that the underlying label semantics of the dataset are preserved (Shorten and Khoshgoftaar, 2019; Feng et al., 2021).

In recent years, researchers have proposed and developed data augmentation methods for various natural language processing tasks. The augmentation methods focus on sentence-level tasks such as text classification (Wei and Zou, 2019; Xie et al., 2019), sentiment analysis (Liesting et al., 2021) and sentence-pair tasks such as natural language inference (Min et al., 2020) and machine translation (Wang et al., 2018a). It is particularly challenging to develop data augmentation methods for the sequence labelling tasks such as named entity recognition (NER) and part-of-speech tagging (POS) as they involve prediction at the word/token level (Sahin and Steedman, 2018; Dai and Adel, 2020; Zhang et al., 2018, 2020b; Ding et al., 2020; Yaseen and Langer, 2021a). The proposed data augmentation methods for language processing follow two dominant trends: (1) employ simple heuristics such as word replacement (Zhang et al., 2015; Wang et al., 2018a; Cai et al., 2020), word swap (Sahin and Steedman, 2018; Min et al., 2020) or random deletion (Wei and Zou, 2019) to generate augmented instances by manipulating a few words in the original sentence (2) *generates* artificial instances via sampling from generative models such as language models (Schick and Schütze, 2021), variational autoencoders (Yoo et al., 2019; Mesbah et al., 2019) or backtranslation models (Yu et al., 2018; Iyyer et al., 2018).

Recently, researchers have explored data augmentation approaches for the task of relation extraction as well. Most of the existing DA methods for relation extraction can be broadly categorized into three categories: (1) apply pre-defined heuristics such as random token deletion, random swapping of tokens etc., (Sartakhti et al., 2021); exploit directionality of relationships to create novel training examples (Xu et al., 2016b) (2) use external knowledge base or knowledge graph to create synthetic training examples (Jiang et al., 2021) (3) sample novel sentences from a pre-trained language model (Papanikolaou and Pierleoni, 2020). Unfortunately, most of the existing work cannot be applied in low-resource (technical) domains as existing methods: a) relies on linguistic resources like dependency parser or WordNet b) relies on external knowledge graphs or knowledge bases c) involve training a language model which could be expensive and not always feasible d). generate grammatically incoherent sequences e). cannot generate linguistically diverse sentences.

One of the challenges in data augmentation for NLP tasks is to generate synthetic examples which are *different/diverse enough* from the existing original examples to provide

a meaningful training signal to the underlying machine learning model. It is important to note that the generated training examples should always respect the underlying label semantics of the dataset to be valid augmented samples. Consider this example with a *positive sentiment* from the dataset of sentiment analysis task: *it was a great watch!*; applying word replacement on this example may generate this augmented example: *it is a great watch!*. One needs to ask if this synthetic augmentation would be helpful for the machine learning model?; in this case, the augmented example does not provide any new information to the model, on the contrary, such augmentations will increase the training time of the machine learning model and introduce unexpected biases during the model training. In most cases, the data augmentation method is applied to each available training example in the dataset. Therefore, the data augmentation techniques should generate diverse yet semantically correct augmented examples (a really hard problem!) to truly enable the development of effective machine learning models in low-resource scenarios.

The significant advancements in machine translation research in recent years have led to the availability of high-quality machine translation systems and services (He, 2015; Wu et al., 2016; Junczys-Dowmunt, 2019). Inspired by these efforts and developments, we adapt *backtranslation* to the task of relation extraction. Backtranslation (BT) can automatically generate linguistically diverse paraphrases of a sentence or a phrase by injecting linguistic variations. Further, backtranslation can be applied to diversify the injected linguistic variations by introducing layers of intermediate language translations. In this work, we exploit backtranslation to generate paraphrases of one or several phrases in a sentence. We report the results of our proposed method on two domain-specific RE datasets and empirically demonstrate the effectiveness of the proposed method in low-resource scenarios.

6.2 Related Work

Recently, there has been quite some interest in data augmentation and researchers have proposed many data augmentation methods for various natural language processing tasks. It is worth noting that since relation extraction is often modelled as a text classification task, many of the data augmentation methods for RE are inspired by data augmentation methods for text classification. In this section, we narrow our focus to proposed data augmentation methods for relation extraction. We categorise existing data augmentation methods for RE in three categories:

Rule-based: Heuristics, use predefined transformations to create the augmentation of the original training instance. We briefly discussed a few of the transformations proposed in the existing work.

(a) *Word replacement:* Replace a word with another word at random.

(b) *Word replacement POS:* Replace a word with another word of the same part-of-speech tag at random.

(c) *Word addition:* Add a word anywhere in a sentence at random.

(d) *Word deletion:* Delete a word in a sentence at random.

(e) *Synonym replacement:* Replace a word with one of its synonyms retrieved from WordNet at random.

(f) *Reversing relations:* For a sentence belonging to directional relation, create an augmentation by reversing the order of directionality (Xu et al., 2016b).

Note that some of the above mentioned transformations may break the syntactic or semantic coherence of the sentence.

Exploiting External Knowledge: One dominant trend in data augmentation for RE is to sample subgraphs from knowledge graphs and create synthetic training data (Annervaz et al., 2018; Jiang et al., 2021; Vannur et al., 2021). One of the limitations of methods under the *exploiting external knowledge* umbrella is the assumption that a knowledge graph or a knowledge base exists and is accessible, this limits the applicability of these methods only to certain domains and also for certain kinds of relations, this is especially true for technical, scientific and industrial domains.

Generative models: Recent work employs pre-trained language models to generate part of the relational sentence or the entire relational sentence. Papanikolaou and Pierleoni developed *DARE*, a method to fine-tune GPT-2 (Radford et al., 2019) to generate examples for specific relation types to augment the training data. It is important to note that GPT-2 was fine-tuned separately for each relation type.

6.3 Data Augmentation via Backtranslation

Figure 6.1 illustrates an example of our proposed data augmentation method for RE using *backtranslation* with *German* and *French* as a pivot/intermediate language. In a nutshell, our algorithm consists of three steps. First, the input sentence is split into three segments based on the context of the entity mentions, the *before* context, *between* context and the *after* context. The before context consists of word tokens before the head entity, the between context consists of the word tokens between the head entity and the tail entity, and the after context consists of the word tokens after the tail entity. Since the context(s) around the entities express the semantic relationship between the head entity and the tail entity, we only consider the three contexts as the candidate for the backtranslation. Second, the validity of the entity mention context is determined based on the length of the context, we only consider contexts with three or more tokens as a valid context for backtranslation. As a final step, the context tokens are translated to the pivot language(s) and finally back to

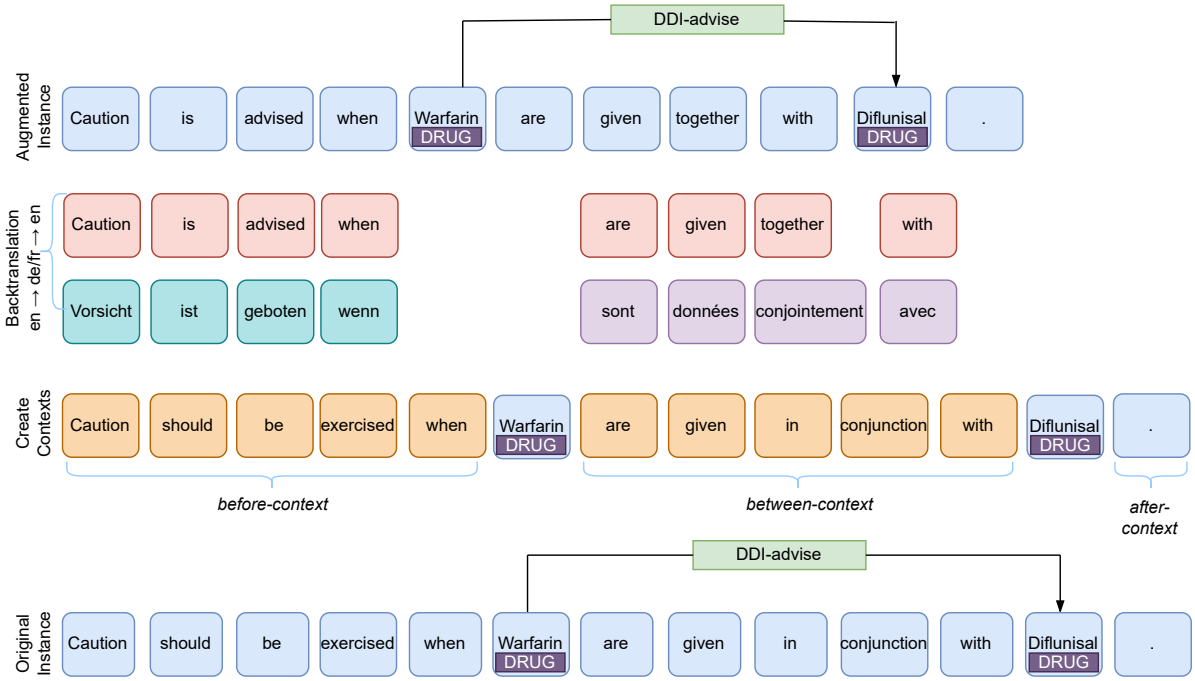


Figure 6.1: An illustration of data augmentation via *backtranslation* for RE. Note that backtranslation is only applied to the context around the entity mentions. Here the entity mention context is first translated to one of the two pivot languages German or French and then back to English using an off-the-shelf machine translation system. The backtranslation results in a paraphrase of the original entity mention context. The original entity mention context is replaced with backtranslated context to create the augmented data instance.

the source language; the original context tokens are replaced with the backtranslated tokens and thus we obtain the augmentation of the original input sentence. In practice, we use a binomial distribution to randomly decide whether the context should be backtranslated; also the choice of pivot language is determined randomly to introduce increased text diversity.

Data augmentation with backtranslation augments the original training set with diverse paraphrases of the entity mention contexts to enable the supervised RE model to generalize beyond the standard training set.

6.4 Evaluation

We empirically evaluate our proposed data augmentation strategy for RE using backtranslation as described in Section 6.3 on two English datasets from the biomedical domains:

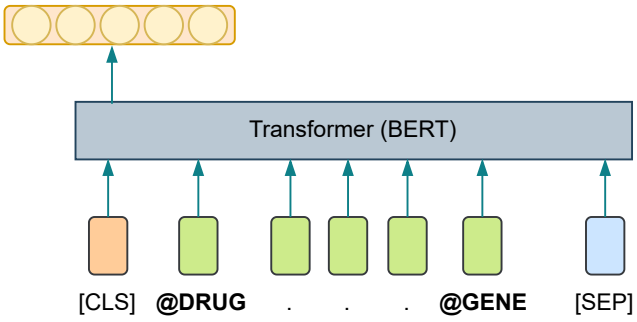


Figure 6.2: An illustration of extracting deep relation representations from transformer network using the [CLS] token's representation.

ChemProt (Kringelum et al., 2016)[1] and DDI (Herrero-Zazo et al., 2013)[2].

The task of relation extraction can essentially be framed as a sequence classification problem, usually specified as a binary or multi-class classification problem. We use the BERT model's (Beltagy et al., 2019) text classification (based on CLS token representation, see figure 6.2) as the underlying supervised RE model for all our experiments, and we investigate the impact of applying data augmentation on training data of varying sizes.

6.4.1 Datasets

ChemProt. The Chemprot (Kringelum et al., 2016) is a publicly available compilation of chemical-protein-disease annotation resources to enable the study of systems pharmacology. The ChemProt RE dataset contains sentences from PubMed abstracts, the dataset classifies the relation between chemicals and proteins within sentences. Sentences are classified into 6 classes including a negative class.

DDI. The DDI (Herrero-Zazo et al., 2013) corpus specifies the task to identify the right types of interactions among the drug pairs. The sentences are collected from the PubMed abstracts and each sentence is classified into four classes for relations including advice, effect, mechanism, and false.

The descriptive statistics of the datasets are reported in Table 6.1. We follow existing work (Dai and Adel, 2020; Yaseen and Langer, 2021a) to simulate a low-resource setting. We select 25%, 50% and 75% of sentences from the training set to create the corresponding small, medium and large training sets (denoted as S, M, L in Table 6.2 and Table 6.3, whereas the complete training set is denoted as F). It is important to note that the data augmentation techniques are only applied to the training set without altering the development and test sets.

[1] https://biocreative.bioinformatics.udel.edu/news/corpora/chemprot-corpus-biocreative-vi/

[2] https://github.com/isegura/DDICorpus

	ChemProt			DDI		
	Train	Dev	Test	Train	Dev	Test
Positive relation instances	4157	2416	3458	3788	-	884
Negative relation instances	11685	7343	9637	22217	-	4381
All relations instances	15842	9759	13095	26005	-	5265

Table 6.1: The descriptive statistics of the RE datasets.

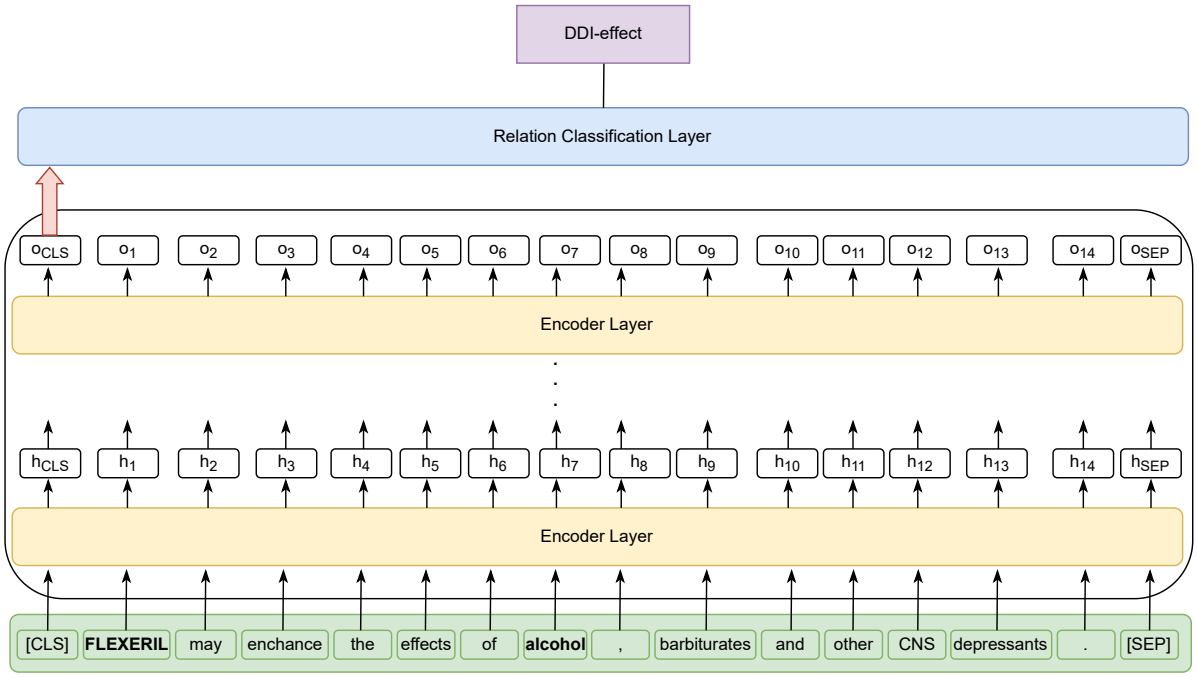


Figure 6.3: All illustration of SciBERT finetuning for relation extraction.

6.4.2 Supervised RE Model

We follow the standard approach of modelling the RE task as a sequence classification problem on a sentence-level, specifically we frame the RE tasks as a multi-class classification problem. In relation extraction, the participating entities are not known in advance, we follow the usual practice to consider and test every valid pair of entities for a relation in a sentence.

Figure 6.3 illustrates the architecture of our model for relation extraction. The encoder of our model is based on SciBERT; the vector representation for the *CLS* token from the final layer of the SciBERT encoder is fed into a relation classification layer. The classification layer applies softmax over the number of relations to predict the relation label. The rationale behind employing SciBERT was its superior performance on natural language processing tasks in the scientific domains including the specialized biomedical domain; this superior performance is attributed to pretraining on the scientific corpus and the capability of transformer based BERT architectures to model contextualized representations for each token in a sentence.

Emb	DA	S	M	L	F	Δ
	None	40.11± 5.0	51.87± 5.6	67.51± 2.8	71.09± 2.4	
	WA	45.86± 2.9	55.12± 5.7	64.55± 5.1	**72.05± 1.0**	1.7
	WD	48.17± 2.2	54.52± 2.0	65.49± 1.5	69.06± 1.5	1.7
SciBERT	WR-POS	41.45± 1.9	55.86± 8.9	63.80± 3.6	71.25± 0.5	0.4
	WSR-POS	42.49± 1.6	54.22± 8.1	65.59± 7.0	70.69± 5.1	0.6
	BT	**49.61± 1.1**	**59.35± 4.0**	**67.82± 1.4**	70.31± 0.9	**4.1**

Table 6.2: F1-score on test sets on **ChemProt** using different subsets of the training set. Here: **S**, **M**, **L** and **F** refer to *small* (10 % sentences), *medium* (25 % sentences), *large* (50 % sentences) and *full* (all sentences) set. We repeat all experiments three times with different seeds. Mean values and standard deviations are reported. Δ column shows the averaged improvement due to data augmentation.

We employ the micro-average F1 score as an evaluation metric and also report micro-average precision and recall. We employ early stopping and report the F1 score on the test set using the best performant model on the development set.

Backtranslation Models. We employed the pre-trained English↔German[3,4] and English↔French[5,6] machine translation models (Tiedemann and Thottingal, 2020) released by *Helsinki-NLP* group. We employed the Huggingface Transformers library's (Wolf et al., 2020) port of the released pretrained machine translation models as the underlying backtranslation models for all our experiments.

Hyperparameters. The two important hyperparameters for our experiments are the number of augmentation instances per training instance and the probability value p of beta distribution to decide if the segment should be backtranslated. However, since we follow the usual practice to consider every valid pair of entities for a relation in a sentence; this results in too many sentences and to optimize the computation costs for performing our experiments, we hardcode these two hyperparameters. We only generate one augmentation sentence per training instance and we fix the value of p to be 0.5. We hypothesize a grid search over a list of optimal values for these two hyperparameters might have resulted in further improved performance, nevertheless, the experimental results demonstrate that our hardcoded values of hyperparameters also significantly improve over the baseline of no augmentation.

6.4.3 Evaluation

We report the evaluation results of various augmentation techniques on the test sets of ChemProt and DDI in Table 6.2 and Table 6.3 respectively. We use the F1-score to evaluate the performance of the RE models. All experiments are repeated three times with different

[3] https://huggingface.co/Helsinki-NLP/opus-mt-en-de
[4] https://huggingface.co/Helsinki-NLP/opus-mt-de-en
[5] https://huggingface.co/Helsinki-NLP/opus-mt-en-roa
[6] https://huggingface.co/Helsinki-NLP/opus-mt-roa-en

Emb	DA	S	M	L	F	Δ
	None	43.97± 4.8	60.75± 4.6	66.27± 4.3	72.14± 1.7	
	WA	47.00± 5.9	66.18± 3.0	68.41± 2.0	74.01± 1.0	3.6
	WD	45.42± 7.0	66.53± 1.2	67.30± 3.0	71.66± 2.5	2.5
SciBERT	WR-POS	45.30± 6.1	63.32± 1.1	70.29± 1.0	74.46± 0.2	3.1
	WSR-POS	41.91± 6.4	61.80± 4.1	68.51± 1.4	73.01± 1.1	1.6
	BT	**56.23**± 5.5	**72.00**± 0.87	**73.21**± 1.7	**75.98**± 1.1	**9.1**

Table 6.3: F1-score on test sets on **DDI** using different subsets of the training set. Here: **S**, **M**, **L** and **F** refer to *small* (10 % sentences), *medium* (25 % sentences), *large* (50 % sentences) and *full* (all sentences) set. We repeat all experiments three times with different seeds. Mean values and standard deviations are reported. Δ column shows the averaged improvement due to data augmentation.

random seeds. We also report the mean values and standard deviations. The Δ row shows the averaged improvement due to data augmentation for both datasets. For the most part, all data augmentation techniques improve over the baseline; backtranslation results in the biggest average improvement for the contextualized SciBERT embeddings under different data usage percentiles. We attribute the improved performance of backtranslation to the generation of linguistically diverse and meaning-preserving before, between and after contexts to enable the improved generalization of the underlying RE model.

In general, we observe that the data augmentation methods result in the highest improvement when the training sets are small. As an example, all data augmentation methods achieve the biggest improvements with training sets of small and medium subsets. In contrast, this effect is reduced as the training sets get larger. The augmentation on the complete training set even decreases the performances for some augmentation techniques. We also observe the similar effects for data augmentation on subsets of training data for named entity recognition 4.4.3.

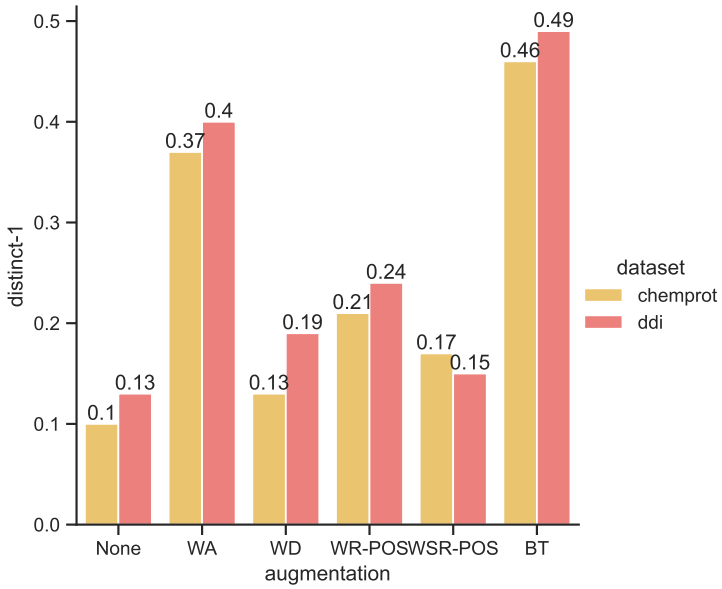


Figure 6.4: The diversity statistics of various augmentation techniques across the datasets.

We employ the distinct-3 metric in Figure 6.4 in order to quantitatively measure the diversity introduced by various augmentation techniques. Distinct-3 quantifies the intra-text diversity by counting the distinct trigrams in each sentence, the count value is scaled by the total number of trigrams in the sentence to avoid favouring longer sentences. Backtranslation has the highest level of unigram diversity, this is not very surprising as backtranslation is known to generate diverse linguistic variations of the text.

6.5 Summary

In this chapter, we presented our work on data augmentation for relation extraction. We consider the task of multi-class binary relation extraction at sentence level and propose *backtranslation* as a data augmentation strategy. In a nutshell, our algorithm splits the training sentence into three segments including before context, between context and after context; each of these contexts is paraphrased using backtranslation resulting in the augmentation of the original training sentence. We demonstrate that backtranslation can generate high-quality coherent and linguistically diverse synthetic data instances for sentence-level RE.

We employed SciBERT encoder with a relation classification layer as the supervised RE model and experimented with the contextualized SciBERT embeddings. We report the experimental results on two biomedical datasets, ChemProt and DDI. We simulate a low-resource setting by using different subsets of training data for training the supervised RE model. Our empirical results demonstrate that backtranslation can improve over the baseline augmentation methods and is a competitive data augmentation strategy for RE.

We found that the data augmentation techniques result in the biggest performance gains when the training sets are small, in the case when enough training data is available the data augmentation does not result in performance improvement. We employ the distinct-3 metric to quantify the intra-text diversity introduced by various augmentation techniques. Our analysis found that data augmentation via backtranslation results in the highest level of trigram diversity.

Chapter 7

Domain Adaptation for Multilingual Acronym Extraction

This chapter covers work already published at the peer-reviewed workshop at an international conference. The relevant publication is Yaseen and Langer (2022). I conceived the original research, implemented the model, performed the evaluation experiments and wrote the initial draft of the paper. My supervisor, Stefan Langer, contributed through discussions in our bi-weekly meetings and by reviewing the paper before submission.

This chapter presents our findings from participating in the multilingual acronym extraction shared task organised as part of the *SDU@AAAI-22* workshop. The task consists of acronym extraction from documents in 6 languages within scientific and legal domains. To address multilingual acronym extraction we employed BiLSTM-CRF with multilingual XLM-RoBERTa embeddings. We pretrained the XLM-RoBERTa model on the shared task corpus to further adapt general purpose XLM-RoBERTa embeddings to the shared task domain(s). Our system (team: SMR-NLP) achieved competitive performance for acronym extraction across all the languages.

7.1 Introduction

Abbreviations (e.g. RNN) are compressed forms of terms and longer phrases, and are used as an alternative to the fully expanded form (e.g., Recurrent Neural Network). Abbreviations are often constructed using a few letters chosen from the longer phrases. As the number of scientific papers published every year is growing at an increasing rate (Bornmann and Mutz, 2015), the amount of abbreviations is enormously increasing as well (Barnett and Doubleday, 2020). This is because the authors of the scientific publications often employ abbreviations as a tool to avoid repeating frequently used long phrases i.e. to make technical terms less verbose. For example, 'LSTM' is often used to refer to 'Long-short term memory' and 'CNN' is often used as an alternative to the long-phrase 'Convolutional Neural Network'. The abbreviations take the form of acronyms or initialisms. We refer to the abbreviated term as "acronym" and we refer to the full term as the "long form". The abbreviations or acronyms

convey the same amount of information with less number of words and therefore, simplify reading and writing. However, acronyms pose a challenge to readers who are not familiar with the domain. This challenge is heightened by the fact that the acronyms are not always standard written, e.g. XGBoost is an acronym of eXtreme Gradient Boosting (Chen and Guestrin, 2016). Acronyms also pose a challenge to various natural language processing and information retrieval systems. In text-processing applications such as question answering, text summarization and machine translation, it is crucial to correctly identify acronyms and their long forms. Similarly, consider the application of search; the (semantic) search engine should be able to retrieve documents only containing the long forms when their respective abbreviation(s) are provided in the query string. Thus, automatic identification of acronyms and their corresponding long forms is crucial for scientific document understanding and language processing tasks.

In recent years, there has been significant progress in acronym extraction, however, most of the existing effort was limited to specific languages and domains. Most prior work focused on the biomedical and scientific texts in English. The recognition of acronyms in other languages and domains is also important and might involve unique challenges which are not addressed in the English biomedical and scientific texts. As an example, many existing AE methods for English use the uppercase information to identify acronyms, however, for many languages like Arabic, Persian and Urdu etc., the concept of a case does not even exist and therefore, most existing AE methods might fail or perform suboptimally. Also, different languages might employ different styles to create acronyms from the longer phrases, for instance, the use of initial letters to create acronyms is common in scientific English, however, it is not very common for legal English and Danish documents (Veyseh et al., 2022a). Thus, it is desirable to study AE in diverse domains and languages to create multi-domain and multilingual acronym extraction systems.

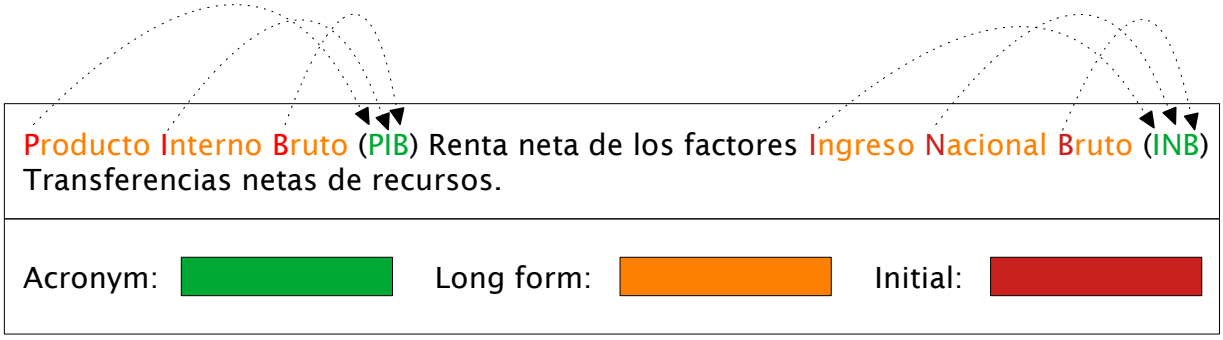


Figure 7.1: An example from the Spanish acronym extraction dataset. In the figure, the Green text represents acronyms, orange text represents long term, and red text represents initials. Also, the black lines indicate the correspondence between initials and acronyms.

7.2 Task Description

The Acronym Extraction shared task (Veyseh et al., 2022b) was organised by the Scientific Document Understanding workshop 2022 (SDU@AAAI-22). The shared task aimed to encourage the creation of a document reading system to recognise acronyms and their correct meanings to lower the barrier to understanding scholarly writing. Since most of the existing AE research is dedicated to English text, this shared task encourages AE research in other languages as well. The task consists of identifying acronyms (short-forms) and their meanings (long-forms) from the documents in six languages including Danish (da), English (en), French (fr), Spanish (es), Persian (fa) and Vietnamese (vi). The task corpus (Veyseh et al., 2022a) consists of documents from the scientific (en, fa, vi) and legal domains (da, en, fr, es).

7.2.1 Challenges

The shared task presents several challenges for the task of acronym extraction, we briefly discuss the most important ones below:

Multilinguality

The appealing aspect of *multilinguality* in the shared task also added the most complication in addressing the problem of acronym extraction. This complication was heightened by the fact that the included languages were very diverse and different from each other. To give an example, case information is often an important indication to recognise the acronyms but since the Persian language does not have any notion of a case; such differences impact the design decisions regarding the model architecture. In the case of a single acronym extraction model for every language, the differences across the languages do not matter directly; but on the other hand for a single multilingual acronym extraction model differences across the languages might confuse the model. It is worth mentioning that a multilingual model can exploit the common structures of syntax and semantics across the languages.

Limited Training Data

In the shared task corpus, the training data for Persian and Vietnamese was quite less compared to the rest of the languages (see figure 7.4). The limited training data adds a distinct challenge to both design choices of one model per language vs one multilingual model for all the languages. In the case of one model per language, the individual models for languages with fewer data can overfit the training data which will lead to poor performance on test data. In the case of one multilingual model for all the languages, the model can become biased towards the languages which have higher amounts of training samples, this will result in the underfitting of the model on the low resource languages and thus will result in a lower score on the test for low resource languages.

Inconsistent annotations

We observed several cases of inconsistent and faulty annotations in the shared task corpus; in many cases, an abbreviation or the long-form was not annotated. This inconsistency in the annotations can provide erroneous signals to the model during training and can harm the performance of the final model.

7.2.2 Task Definition

Sentence:	The	executive	board	of	the	United	Nations	Development	Programme	(	UNDP	)	and	of	...
Labels:	O	O	O	O	O	B-LF	I-LF	I-LF	I-LF	O	B-AC	O	O	O	...

Figure 7.2: An example tagging of a training sentence for acronym and long form extraction. Here, LF refers to the long form and AC refers to the acronym.

Consider a sentence $s = (w_1, w_2, ..., w_n)$ in the corpus, it consists of a sequence of tokens w_n. Our goal is to predict the class label y_t, for each token w_t in the sentence (where t refers to a token at position t in the sentence). Considering the task of determining long forms in a sentence, it is straightforward to understand that the tokens of a long-form have inter-dependence between neighbouring tokens, a token is more likely to be part of the long form if its neighbouring tokens are part of the long-form. This reasoning can also be extended to acronym detection as well, as often acronyms and long forms may appear together in a sentence. Therefore, it is suitable to frame the task of acronym extraction and long-form detection as a sequence labelling problem: given a sentence $s = (w_1, w_2, ..., w_n)$, determine the optimal sequence of token labels $y = (y_1, y_2, ..., y_n)$ for each sentence in the corpus. We adopt the widely used *BIO* tagging scheme to label the training data, an example from the training set in the BIO format is shown in the figure 7.2.

7.3 Methodology

In this section we will describe our proposed method for multilingual acronym extraction.

7.3.1 Multilingual Acronym Extraction

We frame the task of multilingual acronym extraction as a sequence labelling problem and our sequence labelling model follow the well-known architecture (Lample et al., 2016) with a bidirectional long short-term memory (BiLSTM) network and conditional random field (CRF) output layer (Lafferty et al., 2001b). Since the shared task corpus consists of acronym extraction in six languages we decided to create one single model for all the languages. This design decision simplifies the model training process and also makes it practical for

	epochs	all P/R/F1	da P/R/F1	en-sci P/R/F1	en-leg P/R/F1	fr P/R/F1	fa P/R/F1	es P/R/F1	vi P/R/F1
					dev				
r1	0	.841/.868/.854	.825/.833/.829	.727/.750/.738	.758/.784/.771	.738/.742/.740	.619/.539/.576	.820/.871/.845	.375/.547/.445
r2	1	.855/.876/.866	.826/.833/.830	.747/.757/.752	.786/.793/.789	.756/.750/.753	.644/.560/.599	.832/.872/.852	.385/.615/.474
r3	3	**.857/.878/.868**	**.827/.833/.830**	**.750/.759/.755**	**.789/.795/.792**	**.788/.751/.754**	**.665/.557/.606**	**.832/.873/.852**	**.408/.689/.512**
r4	3	-	.77/.773/.775	.617/.703/.650	.677/.677/.677	.715/.733/.724	.864/.294/.439	.823/.850/.836	.623/.074/.132
					test				
r5	3	-	.825/.833/.829	.727/.750/.738	.758/.784/.771	.738/.742/.740	.619/.539/.576	.820/.871/.845	.375/.547/.445

Table 7.1: F1-score on the development set (r1-r4) and test set (r5). Here, *epochs*: number of pretraining epochs for XLM-RoBERTa on the task corpus, *eng-sci*: english scientific domain, *eng-leg*: english legal domain, *all*: all languages combined.

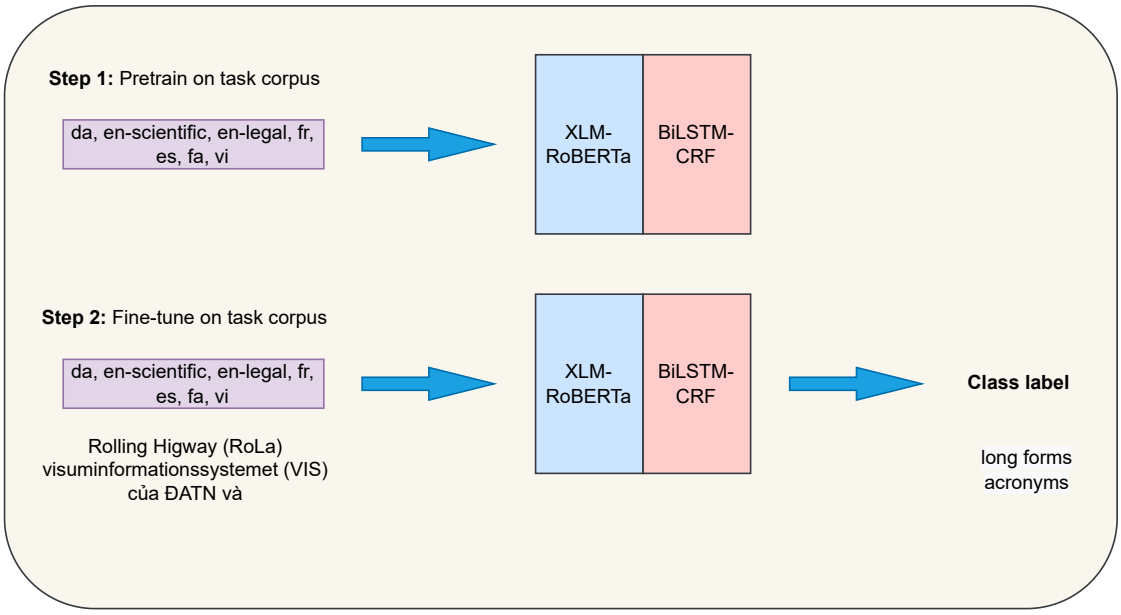


Figure 7.3: An illustration of domain adaptation for multilingual acronym extraction model.

real-world usage (managing one model vs managing a model for every supported language). In order to address the multilingual aspect of the task we employed contextualized cross-lingual language model *XLM-RoBERTa* (Conneau et al., 2020), XLM-RoBERTa is based on RoBERTa (Liu et al., 2019) and is trained on 2.5TB filtered CommonCrawl data in 100 languages. XLM-RoBERTa has demonstrated superior performance in several multilingual natural language understanding tasks (Conneau et al., 2020).

7.3.2 Domain Adaptive Pretraining

The original *XLM-RoBERTa* embeddings (Conneau et al., 2020) are trained on the filtered CommonCrawl data (general domain), whereas the data of the shared task comprises of documents from scientific and legal domains. In order to better adapt the contextualized representation to the target scientific and legal domain, we further pretrained the original XLM-RoBERTa model on the shared task corpus (see figure 7.3); we concatenate the

sentences of all the languages together to create the pretraining corpus. We follow the default setup of pretraining XLM-RoBERTa (Conneau et al., 2020), which uses the Transformer model (Vaswani et al., 2017) trained with multilingual masked language model objective (Devlin et al., 2019). The streams of text are sampled for each language and the model is trained to predict the masked tokens in the input. The subword tokenization is applied directly to the raw text using Sentence Piece (Kudo and Richardson, 2018) with a unigram language model (Kudo, 2018). We refer the readers to Conneau et al. for the complete details regarding the pretraining setup. Our experiments demonstrate improved performance on the task of acronym extraction due to the domain adaptive pretraining across all the languages.

7.4 Experiments and Results

7.4.1 Dataset

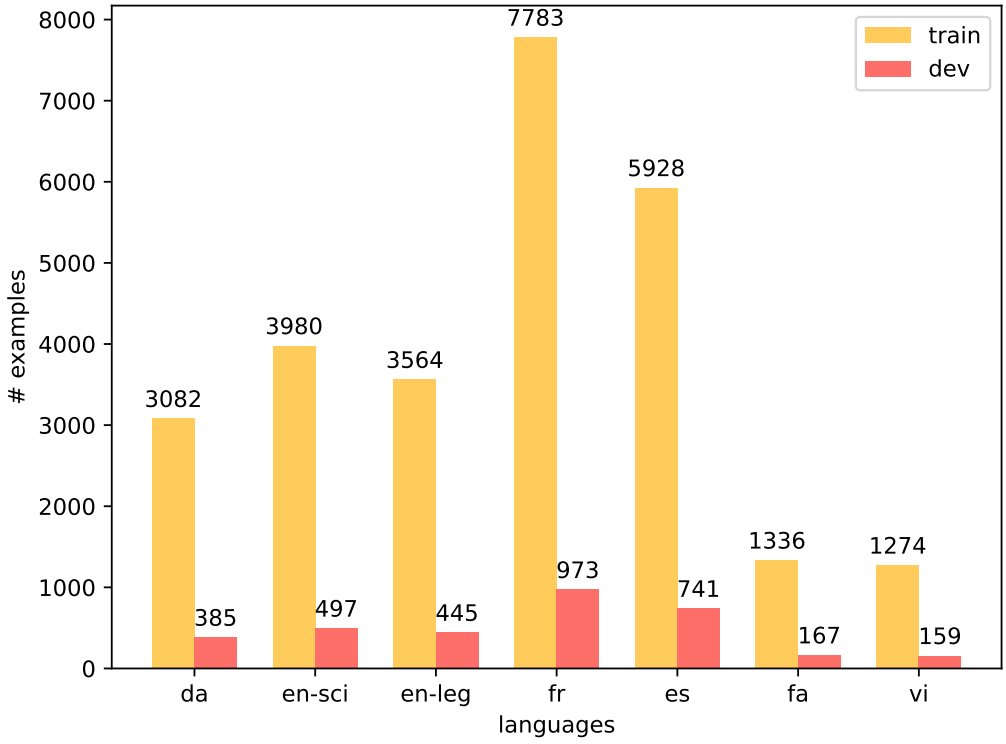


Figure 7.4: Count statistics of train and development set across the languages.

The figure 7.4 shows the example counts of the train and development set for all the languages. Persian and Vietnamese have substantially low examples compared to the rest of the languages in the corpus. As a pre-processing step, we used *spaCy* (Honnibal et al., 2019) to perform word tokenization and POS tagging.

Hyperparameter	Value
hidden size	256
learning rate	$5.0e-6$
training epochs	20
pretraining epochs	3

Table 7.2: Hyperparameter settings for acronym extraction.

We do not apply any strategy to explicitly account for low training data of Persian and Vietnamese. Table 7.2 lists the best configuration of hyperparameters. We compute macro-averaged F1-score using the script provided by the organizers on the development set [1]. We employ early stopping and report the F1-score on the test set using the best performant model on the development set.

7.4.2 Results

Table 7.1 reports the F1-score on the development and test set for all the languages. As a baseline experiment, we combined the training data for all the languages and trained a BiLSTM-CRF model using the pretrained multilingual XLM-Roberta[2] embeddings (row r1). This achieves the overall F1-score of 0.854.

We pretrained the XLM-Roberta model for 1 epoch on the task corpus using a train and development set, which results in 0.1 points improvement in the overall F1-score leading to the F1-score of 0.866 (row r2). Increasing the pretraining epochs to 3 results in an improvement of additional 0.1 points in the overall F1-score (row r3).

We also experimented with training the individual models for each language (including separate models for English scientific and English legal). This results in a significant decrease in F1-score for all the languages (on average 0.12 points in F1-score, see row r4). This demonstrates that BiLSTM-CRF with multilingual XLM-Roberta embeddings performs best when trained with several languages together enabling effective cross-lingual transfer.

The F1-score of our submission on the test set is reported in row r5. Our test submission achieves the F1-score similar to the development set for all the languages demonstrating effective generalization on the test set; Vietnamese is an exception where F1-score on the test set is significantly worse than the F1-score on the development set (see rows r5 vs r3).

7.5 Related Work

The earliest work in acronym extraction consists of carefully crafted rule-based methods. Schwartz and Hearst proposed a two-step approach based on shallow pattern matching to identify abbreviations and their corresponding long forms. Okazaki and Ananiadou

[1] https://github.com/amirveyseh/AAAI-22-SDU-shared-task-1-AE/blob/main/scorer.py

[2] `https://huggingface.co/xlm-roberta-base`

recognise acronyms based on word-sequence co-occurrence; they formalize the long-form recognition as a term extraction problem and modify the C-value method (Frantzi et al., 1998) to extract the long forms. Park and Byrd create a modular rule-based system based on five components to extract abbreviations and their definitions from enterprise documents. Adar developed SARAD, a system to build a dictionary of possible definitions for abbreviations, the clustering of those definitions, and the generation of a classifier for the disambiguation of new definitions. Nadeau and Turney propose search space reduction heuristics for candidate acronyms and candidate definitions; finally, a supervised classifier is employed to identify acronyms and definitions.

The feature-based approach was also dominant in the existing work in acronym extraction. Kuo et al. argues the relatedness of acronym extraction to the NER and developed BIOADI, a system exploiting the string morphological, numerical and contextual features to identify abbreviations and definitions in biomedical literature. Li et al. propose a framework to address the problem of enterprise acronym disambiguation, their framework automatically generates training data from the enterprise corpus via distant supervision to train a supervised model for acronym disambiguation.

Recently, deep learning models have been explored for the task of acronym extraction, however, these methods require large training data to achieve optimal performance. Veyseh et al. employed a deep sequential model based on bidirectional LSTM to acronym expansions; however, the acronym extraction was done based on rules/heuristics. Antunes and Matos used word embeddings along with unigram and bigram features to train decision trees, k-nearest neighbours and linear SVM to address word sense disambiguation. Jaber and Martínez employed several lexical features including word features, POS, position features and word information features to address acronym disambiguation; they report results with SVM, Naive Bayes and K-Nearest neighbours. Li et al. explored several state-of-the-art deep learning models including BiLSTM-CRF (Lample et al., 2016), BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019) and XLNet (Yang et al., 2019b); they employed multi-task learning and adversarial training to improve the model training and their system achieved the second rank in the SDU-2021 shared task for acronym extraction.

One of the major limitations of existing work in acronym extraction is that most prior work only focuses on the English biomedical and scientific texts, leaving non-English texts and other domains less explored.

7.6 Summary

In this chapter, we described our system with which we participate in the multilingual acronym extraction shared task organized by the Scientific Document Understanding workshop 2022 (SDU@AAAI-22). We formulate multilingual acronym extraction in 6 languages and 2 domains as a sequence labelling task and employed the BiLSTM-CRF model with multilingual XLM-RoBERTa embeddings. We pretrained the XLM-RoBERTa model on the target scientific and legal domain to better adapt multilingual XLM-RoBERTa embeddings for the target task. Our system demonstrates competitive performance on the

multilingual acronym extraction task for all the languages.

Chapter 8

Conclusion and Future Work

In this thesis, we develop data-efficient methods for information extraction in low-resource specialized domains. We addressed named entity recognition, relation extraction, data augmentation, semi-supervised bootstrapping and multilingual acronym extraction, our methods achieve state-of-the-art performance on several datasets.

In Chapter 3, we develop methods for named entity recognition and entity normalization. Our named entity recognition method can also extract *nested named entities* which are typically ignored by the mainstream (supervised) named entity recognition methods. To address the complicated problem of nested entities extraction, we employed two BiLSTM-CRF models, the first BiLSTM-CRF (*Level1 NER*) detects parent entities while the second BiLSTM-CRF (*Level2 NER*) detects the nested entities; the Level2 NER only operates on the output of Level1 NER. This pipeline nature of nested entities extraction propagates errors from Level1 NER to Level2 NER, we circumvent this error propagation by employing several strategies such as auxiliary language modelling objective, auxiliary named entity detection objective, several linguistic features, the hybrid ranking loss function and ensembling via bagging. Our entity normalization method employed dictionary-based exact (string match), fuzzy (Levenshtein distance) and semantic search (distributional similarity). Our named entity recognition and entity normalization systems achieved the slot error rate value of 0.715 and ranked 1st in the *Bacteria Biotope Shared task 2019*. In the second part of Chapter 3, we introduced the *stacked heterogeneous embeddings* to improve the representational capabilities of the models. The basic intuition behind stacked heterogeneous embeddings is that various kinds of embeddings capture different aspects of language and domain(s), instead of choosing one over another embedding, combine different heterogeneous embeddings to enable *complimentary learning*. The stacked heterogeneous embeddings demonstrate improved performance over individual embeddings.

One of the shortcomings of our nested NER model is that it can only detect nested entities at level2, it ignores the nested entities at level3 or above. One possible future work would be to dynamically detect nested entities at any level. The Graph Neural Networks (GNN) (Sanchez-Lengeling et al., 2021; Luo and Zhao, 2020) can be explored to dynamically detect nested entities at various levels(s). The dynamic detection of nested entities will also avoid the error propagation which happens due to the pipeline nature of our current

model design i.e. if the level1 NER model fails to detect the parent entity then level2 NER will not be able to detect the nested entity at level2, as level2 NER only operates at the output of level1 NER. In our work on the stacked heterogeneous embeddings for NER we simply *concatenate* the heterogenous embeddings to enrich the input representation of the BiLSTM-CRF model, the plain concatenation increases the dimensionality of the input vector significantly; this increases the model parameters, compute time and can potentially lead to difficulties in the model training. One possible future direction would be to explore dimensionality reduction techniques such as Principal component analysis (PCA) (Shlens, 2014) or auto encoders (Baldi, 2012) to reduce the high dimensionality of the input vector by capturing the most representative input dimensions; this would enable to preserve the expressiveness of the representation space of the input vector without dealing with the difficulty of model training. It would be interesting to develop auxiliary objectives which can improve the (supervised) named entity recognition performance of the recent transformer based models.

In Chapter 4 and Chapter 6, we describe our work on synthetic data augmentation for the task of named entity recognition and relation extraction respectively. The data augmentation for NER is particularly challenging as NER is often framed as a token-level task and hence requires annotation at the token level. We adapt *backtranslation* to generate linguistically diverse, grammatically coherent and meaning-preserving entity mention context(s); we empirically demonstrate the effectiveness of backtranslation as a data augmentation strategy on two domain-specific datasets for both NER and RE. In our experiments, we employed *German*, *German* and *French* as a pivot language for NER and RE respectively. In future work, it would be interesting to explore other languages as pivot language(s) and analyse the impact of pivot languages on the quality of synthetic data generation. We only perform experiments on datasets from the material science and biomedical domain, it would be interesting to explore more datasets of diverse domains to see if the empirical gains can be transferred across the other domains. Also, we only consider the datasets in the English language, additional languages can be explored as part of future work. Finally, to measure the diversity and *soundness* of the generated synthetic data, human evaluation should be performed on a subset of the generated data.

In Chapter 5, we describe the bootstrapping process for large-scale relation extraction; the bootstrapping for RE is particularly attractive as it does not require annotated data and only need a handful of seed instances to bootstrap target relation. However, the vanilla bootstrapping can accumulate noise over time as it cannot always distinguish between the valid extraction pattern and a noisy extraction pattern, this is referred as semantic drift. We propose two methods to constrain the bootstrapping process and minimise semantic drift. Our first constraining method (*MST-prune*) employs maximum spanning tree and distributional similarity to identify anomalous noisy extraction patterns. Our second method (*LM-prune*) use stored knowledge in a pre-trained language model to recalibrate the confidence of an extraction pattern such that a low score value is assigned to the noisy extraction pattern(s). Our experimental results demonstrate significant increase in precision for four target relations on the TACRED dataset. The promising directions of future work would be developing new constraining methods, experiments on additional

datasets including multilingual datasets and combining bootstrapping with deep learning models in a self-supervised learning setup (Qiu et al., 2009; Zhou et al., 2020).

In Chapter 7, we describe our work on the acronym extraction as part of the multilingual acronym extraction shared task 2022. We frame multilingual acronym extraction as a sequence labelling task on the token-level and employed the BiLSTM-CRF model. To address the aspect of *multilinguality*, we use the *multilingual XLM-RoBERTa* embeddings. The shared task corpus consists of 6 languages and 2 domains including the scientific and legal domain. In order to better adapt the multilingual XLM-RoBERTa embeddings from the general domain to the target domain, we pretrain the XLm-RoBERTa embeddings on the task corpus. Our experimental results demonstrate the superiority of domain adaptation via pretraining on the test set. We use the default Masked language modeling (MLM) objective to pretrain the XLM-RoBERTa model, as future work one can explore specialized pretraining objectives for sequence labelling tasks such as *span-boundary objective (SBO)* (Joshi et al., 2020).

List of Figures

List of Tables

Bibliography

Eytan Adar. Sarad: a simple and robust abbreviation dictionary. *Bioinform.*, 20(4):527–533, 2004. URL https://doi.org/10.1093/bioinformatics/btg439.

Eugene Agichtein and Luis Gravano. *Snowball*: extracting relations from large plain-text collections. In *Proceedings of the Fifth ACM Conference on Digital Libraries, June 2-7, 2000, San Antonio, TX, USA*, pages 85–94. ACM, 2000.

Gustavo Aguilar, Suraj Maharjan, Adrián Pastor López-Monroy, and Thamar Solorio. A multi-task approach for named entity recognition in social media data. In *Proceedings of the 3rd Workshop on Noisy User-generated Text, NUT@EMNLP 2017, Copenhagen, Denmark, September 7, 2017*, pages 148–153, 2017.

Alan Akbik, Duncan Blythe, and Roland Vollgraf. Contextual string embeddings for sequence labeling. In Emily M. Bender, Leon Derczynski, and Pierre Isabelle, editors, *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 1638–1649. Association for Computational Linguistics, 2018. URL https://aclanthology.org/C18-1139/.

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. Leveraging linguistic structure for open domain information extraction. 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 344–354, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1034. URL https://aclanthology.org/P15-1034.

K. M. Annervaz, Somnath Basu Roy Chowdhury, and Ambedkar Dukkipati. Learning beyond datasets: Knowledge graph augmented neural networks for natural language processing. In Marilyn A. Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 313–322. Association for Computational Linguistics, 2018. URL https://doi.org/10.18653/v1/n18-1029.

Rui Antunes and Sérgio Matos. Biomedical word sense disambiguation with word embeddings. In Florentino Fdez-Riverola, Mohd Saberi Mohamad, Miguel P. Rocha, Juan F. De

Paz, and Tiago Pinto, editors, *11th International Conference on Practical Applications of Computational Biology & Bioinformatics, PACBB 2017, Porto, Portugal, 21-23 June, 2017*, volume 616 of *Advances in Intelligent Systems and Computing*, pages 273–279. Springer, 2017.

Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016. URL http://arxiv.org/abs/1607.06450.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1409.0473.

Pierre Baldi. Autoencoders, unsupervised learning, and deep architectures. In Isabelle Guyon, Gideon Dror, Vincent Lemaire, Graham W. Taylor, and Daniel L. Silver, editors, *Unsupervised and Transfer Learning - Workshop held at ICML 2011, Bellevue, Washington, USA, July 2, 2011*, volume 27 of *JMLR Proceedings*, pages 37–50. JMLR.org, 2012. URL http://proceedings.mlr.press/v27/baldi12a.html.

Adrian P. A. Barnett and Zoe Doubleday. The growth of acronyms in the scientific literature. *eLife*, 9, 2020.

Marco Basaldella, Fangyu Liu, Ehsan Shareghi, and Nigel Collier. COMETA: A corpus for medical entity linking in the social media. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 3122–3137. Association for Computational Linguistics, 2020. URL https://doi.org/10.18653/v1/2020.emnlp-main.253.

David S. Batista, Bruno Martins, and Mário J. Silva. Semi-supervised bootstrapping of relationship extractors with distributional semantics. In Lluís Màrquez, Chris Callison-Burch, Jian Su, Daniele Pighin, and Yuval Marton, editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 499–504. The Association for Computational Linguistics, 2015a. doi: 10.18653/v1/d15-1056. URL https://doi.org/10.18653/v1/d15-1056.

David S. Batista, Bruno Martins, and Mário J. Silva. Semi-supervised bootstrapping of relationship extractors with distributional semantics. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 499–504, Lisbon, Portugal, September 2015b. Association for Computational Linguistics. doi: 10.18653/v1/D15-1056. URL https://aclanthology.org/D15-1056.

Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: A pretrained language model for scientific text. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of*

the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019, pages 3613–3618. Association for Computational Linguistics, 2019. URL https://doi.org/10.18653/v1/D19-1371.

Yoshua Bengio. Learning deep architectures for AI. *Foundation and Trends in Machine Learning*, 2(1):1–127, 2009. doi: 10.1561/2200000006. URL https://doi.org/10.1561/2200000006.

Yoshua Bengio, Patrice Y. Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Networks*, 5(2):157–166, 1994. doi: 10.1109/72.279181. URL https://doi.org/10.1109/72.279181.

Yoshua Bengio, Aaron C. Courville, and Pascal Vincent. Representation learning: A review and new perspectives. volume 35, pages 1798–1828, 2013. URL https://doi.org/10.1109/TPAMI.2013.50.

Christopher M. Bishop. Neural networks for pattern recognition. 1995.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *TACL*, 5:135–146, 2017a.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomás Mikolov. Enriching word vectors with subword information. *Trans. Assoc. Comput. Linguistics*, 5:135–146, 2017b. URL https://transacl.org/ojs/index.php/tacl/article/view/999.

Kurt D. Bollacker, Colin Evans, Praveen K. Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In Jason Tsong-Li Wang, editor, *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008*, pages 1247–1250. ACM, 2008. URL https://doi.org/10.1145/1376616.1376746.

Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. Learning structured embeddings of knowledge bases. In Wolfram Burgard and Dan Roth, editors, *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, USA, August 7-11, 2011*. AAAI Press, 2011. URL http://www.aaai.org/ocs/index.php/AAAI/AAAI11/paper/view/3659.

Lutz Bornmann and Rüdiger Mutz. Growth rates of modern science: A bibliometric analysis based on the number of publications and cited references. *J. Assoc. Inf. Sci. Technol.*, 66 (11):2215–2222, 2015. URL https://doi.org/10.1002/asi.23329.

Robert Bossy, Julien Jourde, Alain-Pierre Manine, Philippe Veber, Érick Alphonse, Maarten van de Guchte, Philippe Bessières, and Claire Nedellec. Bionlp shared task - the bacteria track. *BMC Bioinform.*, 13(S-11):S3, 2012. URL https://doi.org/10.1186/1471-2105-13-S11-S3.

Robert Bossy, Wiktoria Golik, Zorana Ratkovic, Dialekti Valsamou, Philippe Bessières, and Claire Nedellec. Overview of the gene regulation network and the bacteria biotope tasks in bionlp'13 shared task. In *BMC Bioinformatics*, 2015.

Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

Leo Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth, 1984. ISBN 0-534-98053-8.

Sergey Brin. Extracting patterns and relations from the world wide web. In Paolo Atzeni, Alberto O. Mendelzon, and Giansalvatore Mecca, editors, *The World Wide Web and Databases, International Workshop WebDB'98, Valencia, Spain, March 27-28, 1998, Selected Papers*, volume 1590 of *Lecture Notes in Computer Science*, pages 172–183. Springer, 1998.

Razvan C. Bunescu and Raymond J. Mooney. Learning to extract relations from the web using minimal supervision. In John A. Carroll, Antal van den Bosch, and Annie Zaenen, editors, *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*. The Association for Computational Linguistics, 2007. URL `https://aclanthology.org/P07-1073/`.

Hengyi Cai, Hongshen Chen, Yonghao Song, Cheng Zhang, Xiaofang Zhao, and Dawei Yin. Data manipulation: Towards effective instance learning for neural dialogue generation via learning to augment and reweight. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 6334–6343. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.564. URL `https://doi.org/10.18653/v1/2020.acl-main.564`.

Andrew Carlson, Justin Betteridge, Richard C. Wang, Estevam R. Hruschka Jr., and Tom M. Mitchell. Coupled semi-supervised learning for information extraction. In Brian D. Davison, Torsten Suel, Nick Craswell, and Bing Liu, editors, *Proceedings of the Third International Conference on Web Search and Web Data Mining, WSDM 2010, New York, NY, USA, February 4-6, 2010*, pages 101–110. ACM, 2010.

Xavier Carreras, Lluís Màrquez, and Lluís Padró. Named entity extraction using adaboost. In Dan Roth and Antal van den Bosch, editors, *Proceedings of the 6th Conference on Natural Language Learning, CoNLL 2002, Held in cooperation with COLING 2002, Taipei, Taiwan, 2002*. ACL, 2002. URL `https://aclanthology.org/W02-2004/`.

David Carreto Fidalgo, Daniel Vila-Suero, Francisco Aranda Montes, and Ignacio Talavera Cepeda. System description for ProfNER - SMMH: Optimized finetuning of a pretrained transformer and word vectors. In *Proceedings of the Sixth Social Media Mining for Health (#SMM4H) Workshop and Shared Task*, pages 69–73, Mexico City, Mexico, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.smm4h-1.11. URL `https://aclanthology.org/2021.smm4h-1.11`.

José Cañete, Gabriel Chaperon, Rodrigo Fuentes, Jou-Hui Ho, Hojin Kang, and Jorge Pérez. Spanish pre-trained bert model and evaluation data. In *PML4DC at ICLR 2020*, 2020.

William Chan, Navdeep Jaitly, Quoc V. Le, and Oriol Vinyals. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2016, Shanghai, China, March 20-25, 2016*, pages 4960–4964. IEEE, 2016. doi: 10.1109/ICASSP.2016.7472621. URL https://doi.org/10.1109/ICASSP.2016.7472621.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In Balaji Krishnapuram, Mohak Shah, Alexander J. Smola, Charu C. Aggarwal, Dou Shen, and Rajeev Rastogi, editors, *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 785–794. ACM, 2016. doi: 10.1145/2939672.2939785. URL https://doi.org/10.1145/2939672.2939785.

Laura Chiticariu, Yunyao Li, and Frederick R. Reiss. Rule-based information extraction is dead! long live rule-based information extraction systems! In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 827–832. ACL, 2013a. URL https://aclanthology.org/D13-1079/.

Laura Chiticariu, Yunyao Li, and Frederick R. Reiss. Rule-based information extraction is dead! long live rule-based information extraction systems! In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 827–832, Seattle, Washington, USA, October 2013b. Association for Computational Linguistics. URL https://aclanthology.org/D13-1079.

Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014. URL http://arxiv.org/abs/1412.3555.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

Michael Collins and Yoram Singer. Unsupervised models for named entity classification. In Pascale Fung and Joe Zhou, editors, *Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, EMNLP 1999, College Park, MD, USA, June 21-22, 1999*. Association for Computational Linguistics, 1999. URL https://aclanthology.org/W99-0613/.

Ronan Collobert and Jason Weston. A unified architecture for natural language processing: deep neural networks with multitask learning. In *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008*, pages 160–167, 2008a.

Ronan Collobert and Jason Weston. A unified architecture for natural language processing: deep neural networks with multitask learning. In William W. Cohen, Andrew McCallum, and Sam T. Roweis, editors, *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008*, volume 307 of *ACM International Conference Proceeding Series*, pages 160–167. ACM, 2008b. doi: 10.1145/1390156.1390177. URL https://doi.org/10.1145/1390156.1390177.

Alexis Conneau and Guillaume Lample. Cross-lingual language model pretraining. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 7057–7067, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/c04c19c2c2474dbf5f7ac4372c5b9af1-Abstract.html.

Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann LeCun. Very deep convolutional networks for text classification. In Mirella Lapata, Phil Blunsom, and Alexander Koller, editors, *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 1: Long Papers*, pages 1107–1116. Association for Computational Linguistics, 2017. URL https://doi.org/10.18653/v1/e17-1104.

Alexis Conneau, Germán Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. What you can cram into a single \$&!#* vector: Probing sentence embeddings for linguistic properties. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 2126–2136. Association for Computational Linguistics, 2018. doi: 10.18653/v1/P18-1198. URL https://aclanthology.org/P18-1198/.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8440–8451. Association for Computational Linguistics, 2020. URL https://doi.org/10.18653/v1/2020.acl-main.747.

Helen Cook, Evangelos Pafilis, and Lars Juhl Jensen. A dictionary- and rule-based system for identification of bacteria and habitats in text. In Claire Nedellec, Robert Bossy, and

Jin-Dong Kim, editors, *Proceedings of the 4th BioNLP Shared Task Workshop, BioNLP 2016, Berlin, Germany, August 13, 2016*, pages 50–55. Association for Computational Linguistics, 2016. URL https://doi.org/10.18653/v1/W16-3006.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3): 273–297, 1995.

Mark Craven and Johan Kumlien. Constructing biological knowledge bases by extracting information from text sources. In Thomas Lengauer, Reinhard Schneider, Peer Bork, Douglas L. Brutlag, Janice I. Glasgow, Hans-Werner Mewes, and Ralf Zimmer, editors, *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology, August 6-10, 1999, Heidelberg, Germany*, pages 77–86. AAAI, 1999. URL http://www.aaai.org/Library/ISMB/1999/ismb99-010.php.

James R. Curran, Tara Murphy, and Bernhard Scholz. Minimising semantic drift with mutual exclusion bootstrapping. In *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics*, pages 172–180, 12 2008.

Xiang Dai and Heike Adel. An analysis of simple data augmentation for named entity recognition. In Donia Scott, Núria Bel, and Chengqing Zong, editors, *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 3861–3867. International Committee on Computational Linguistics, 2020. doi: 10.18653/v1/2020.coling-main.343. URL https://doi.org/10.18653/v1/2020.coling-main.343.

Merlin Susan David and Dr. Shini Renjith. Comparison of word embeddings in text classification based on rnn and cnn. volume 1187, 2021.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/v1/n19-1423. URL https://doi.org/10.18653/v1/n19-1423.

George-Andrei Dima, Dumitru-Clementin Cercel, and Mihai Dascalu. Transformer-based multi-task learning for adverse effect mention analysis in tweets. In *Proceedings of the Sixth Social Media Mining for Health (#SMM4H) Workshop and Shared Task*, pages 44–51, Mexico City, Mexico, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.smm4h-1.7. URL https://aclanthology.org/2021.smm4h-1.7.

Bosheng Ding, Linlin Liu, Lidong Bing, Canasai Kruengkrai, Thien Hai Nguyen, Shafiq Joty, Luo Si, and Chunyan Miao. DAGA: Data augmentation with a generation approach for low-resource tagging tasks. In *Proceedings of the 2020 Conference on Empirical Methods*

in Natural Language Processing (EMNLP), pages 6045–6057, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.488. URL https://aclanthology.org/2020.emnlp-main.488.

Cícero Nogueira dos Santos, Bing Xiang, and Bowen Zhou. Classifying relations by ranking with convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 626–634, 2015.

John C. Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. In Adam Tauman Kalai and Mehryar Mohri, editors, *COLT 2010 - The 23rd Conference on Learning Theory, Haifa, Israel, June 27-29, 2010*, pages 257–269. Omnipress, 2010. URL http://colt2010.haifa.il.ibm.com/papers/COLT2010proceedings.pdf#page=265.

Javid Ebrahimi and Dejing Dou. Chain based RNN for relation classification. In Rada Mihalcea, Joyce Yue Chai, and Anoop Sarkar, editors, *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*, pages 1244–1249. The Association for Computational Linguistics, 2015. doi: 10.3115/v1/n15-1133. URL https://doi.org/10.3115/v1/n15-1133.

Mohab Elkaref and Lamiece Hassan. A joint training approach to tweet classification and adverse effect extraction and normalization for SMM4H 2021. In *Proceedings of the Sixth Social Media Mining for Health (#SMM4H) Workshop and Shared Task*, pages 91–94, Mexico City, Mexico, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.smm4h-1.16. URL https://aclanthology.org/2021.smm4h-1.16.

Jeffrey L. Elman. Finding structure in time. *Cogn. Sci.*, 14(2):179–211, 1990. doi: 10.1207/s15516709cog1402_1. URL https://doi.org/10.1207/s15516709cog1402_1.

Oren Etzioni, Michael J. Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. Unsupervised named-entity extraction from the web: An experimental study. *Artif. Intell.*, 165(1):91–134, 2005. doi: 10.1016/j.artint.2005.03.001. URL https://doi.org/10.1016/j.artint.2005.03.001.

Marzieh Fadaee, Arianna Bisazza, and Christof Monz. Data augmentation for low-resource neural machine translation. *ArXiv*, abs/1705.00440, 2017a.

Marzieh Fadaee, Arianna Bisazza, and Christof Monz. Data augmentation for low-resource neural machine translation. In Regina Barzilay and Min-Yen Kan, editors, *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 567–573.

Association for Computational Linguistics, 2017b. doi: 10.18653/v1/P17-2090. URL https://doi.org/10.18653/v1/P17-2090.

Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying relations for open information extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics. URL https://aclanthology.org/D11-1142.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard H. Hovy. A survey of data augmentation approaches for NLP. *CoRR*, abs/2105.03075, 2021. URL https://arxiv.org/abs/2105.03075.

Matthew Francis-Landau, Greg Durrett, and Dan Klein. Capturing semantic similarity for entity linking with convolutional neural networks. In Kevin Knight, Ani Nenkova, and Owen Rambow, editors, *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 1256–1261. The Association for Computational Linguistics, 2016. doi: 10.18653/v1/n16-1150. URL https://doi.org/10.18653/v1/n16-1150.

Katerina T. Frantzi, Sophia Ananiadou, and Junichi Tsujii. The c-value/nc-value method of automatic recognition for multi-word terms. In *ECDL*, 1998.

Dan Garrette and Jason Baldridge. Learning a part-of-speech tagger from two hours of annotation. In Lucy Vanderwende, Hal Daumé III, and Katrin Kirchhoff, editors, *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 138–147. The Association for Computational Linguistics, 2013. URL https://aclanthology.org/N13-1014/.

Dan Gillick, Cliff Brunk, Oriol Vinyals, and Amarnag Subramanya. Multilingual language processing from bytes. In Kevin Knight, Ani Nenkova, and Owen Rambow, editors, *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 1296–1306. The Association for Computational Linguistics, 2016. doi: 10.18653/v1/n16-1155. URL https://doi.org/10.18653/v1/n16-1155.

Daniel Gillick, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldridge, Eugene Ie, and Diego García-Olano. Learning dense representations for entity retrieval. In Mohit Bansal and Aline Villavicencio, editors, *Proceedings of the 23rd Conference on Computational Natural Language Learning, CoNLL 2019, Hong Kong, China, November 3-4, 2019*, pages 528–537. Association for Computational Linguistics, 2019. doi: 10.18653/v1/K19-1049. URL https://doi.org/10.18653/v1/K19-1049.

Michael R. Glass and Alfio Gliozzo. A dataset for web-scale knowledge base population. In Aldo Gangemi, Roberto Navigli, Maria-Esther Vidal, Pascal Hitzler, Raphaël Troncy,

Laura Hollink, Anna Tordai, and Mehwish Alam, editors, *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*, volume 10843 of *Lecture Notes in Computer Science*, pages 256–271. Springer, 2018. URL https://doi.org/10.1007/978-3-319-93417-4_17.

Saul Goldenberg, Regina Célia Figueiredo Castro, and Fernando Redondo Moreira Azevedo. [scielo (scientific electronic library online) statistical data interpretation]. *Acta cirurgica brasileira*, 22 1:1–7, 2007.

Aitor Gonzalez-Agirre, Montserrat Marimon, Ander Intxaurrondo, Obdulia Rabal, Marta Villegas, and Martin Krallinger. Pharmaconer: Pharmacological substances, compounds and proteins named entity recognition track. In *Proceedings of the BioNLP Open Shared Tasks (BioNLP-OST)*, Hong Kong, China, November 2019. Association for Computational Linguistics.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

Ian J. Goodfellow, Yoshua Bengio, and Aaron C. Courville. Deep learning. *Nature*, 521: 436–444, 2015.

Edouard Grave. A convex relaxation for weakly supervised relation extraction. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1580–1590. ACL, 2014. URL https://doi.org/10.3115/v1/d14-1166.

Cyril Grouin. Identification of mentions and relations between bacteria and biotope from pubmed abstracts. In Claire Nedellec, Robert Bossy, and Jin-Dong Kim, editors, *Proceedings of the 4th BioNLP Shared Task Workshop, BioNLP 2016, Berlin, Germany, August 13, 2016*, pages 64–72. Association for Computational Linguistics, 2016. URL https://doi.org/10.18653/v1/W16-3008.

Stephen Guo, Ming-Wei Chang, and Emre Kiciman. To link or not to link? A study on end-to-end tweet entity linking. In Lucy Vanderwende, Hal Daumé III, and Katrin Kirchhoff, editors, *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 1020–1030. The Association for Computational Linguistics, 2013. URL https://aclanthology.org/N13-1122/.

Zhijiang Guo, Yan Zhang, and Wei Lu. Attention guided graph convolutional networks for relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 241–251, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1024. URL https://aclanthology.org/P19-1024.

Nitish Gupta, Sameer Singh, and Dan Roth. Entity linking via joint encoding of types, descriptions, and context. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 2681–2690. Association for Computational Linguistics, 2017. doi: 10.18653/v1/d17-1284. URL https://doi.org/10.18653/v1/d17-1284.

Pankaj Gupta, Benjamin Roth, and Hinrich Schütze. Joint bootstrapping machines for high confidence relation extraction. 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 26–36, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1003. URL https://aclanthology.org/N18-1003.

Sonal Gupta and Christopher Manning. Improved pattern learning for bootstrapped entity extraction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 98–108, Ann Arbor, Michigan, June 2014a. Association for Computational Linguistics. doi: 10.3115/v1/W14-1611. URL https://aclanthology.org/W14-1611.

Sonal Gupta and Christopher D. Manning. Improved pattern learning for bootstrapped entity extraction. In Roser Morante and Wen-tau Yih, editors, *Proceedings of the Eighteenth Conference on Computational Natural Language Learning, CoNLL 2014, Baltimore, Maryland, USA, June 26-27, 2014*, pages 98–108. ACL, 2014b. doi: 10.3115/v1/w14-1611. URL https://doi.org/10.3115/v1/w14-1611.

Sonal Gupta and Christopher D. Manning. Distributed representations of words to guide bootstrapped entity classifiers. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1215–1220, Denver, Colorado, May–June 2015. Association for Computational Linguistics. doi: 10.3115/v1/N15-1128. URL https://aclanthology.org/N15-1128.

Suchin Gururangan, Ana Marasovic, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8342–8360. Association for Computational Linguistics, 2020. URL https://doi.org/10.18653/v1/2020.acl-main.740.

Kai Hakala and Sampo Pyysalo. Biomedical named entity recognition with multilingual BERT. In Jin-Dong Kim, Claire Nédellec, Robert Bossy, and Louise Deléger, editors, *Proceedings of The 5th Workshop on BioNLP Open Shared Tasks, BioNLP-OST@EMNLP-IJNCLP 2019, Hong Kong, China, November 4, 2019*, pages 56–61. Association for Computational Linguistics, 2019. URL https://doi.org/10.18653/v1/D19-5709.

Zellig S. Harris. Distributional structure. volume 10, pages 146–162, 1954.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.90. URL https://doi.org/10.1109/CVPR.2016.90.

Xiaodong He and David Golub. Character-level question answering with attention. In Jian Su, Xavier Carreras, and Kevin Duh, editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1598–1607. The Association for Computational Linguistics, 2016. doi: 10.18653/v1/d16-1166. URL https://doi.org/10.18653/v1/d16-1166.

Zhongjun He. Baidu translate: Research and products. In Bogdan Babych, Kurt Eberle, Patrik Lambert, Reinhard Rapp, Rafael E. Banchs, and Marta R. Costa-jussà, editors, *Proceedings of the Fourth Workshop on Hybrid Approaches to Translation, HyTra@ACL 2015, July 31, 2015, Beijing, China*, pages 61–62. The Association for Computer Linguistics, 2015. doi: 10.18653/v1/w15-4110. URL https://doi.org/10.18653/v1/w15-4110.

Marti A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *COLING 1992 Volume 2: The 14th International Conference on Computational Linguistics*, 1992. URL https://aclanthology.org/C92-2082.

Michael A. Hedderich, Lukas Lange, Heike Adel, Jannik Strötgen, and Dietrich Klakow. A survey on recent approaches for natural language processing in low-resource scenarios. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tür, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 2545–2568. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.naacl-main.201. URL https://doi.org/10.18653/v1/2021.naacl-main.201.

Benjamin Heinzerling and Michael Strube. Bpemb: Tokenization-free pre-trained subword embeddings in 275 languages. In Nicoletta Calzolari, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Kôiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asunción Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga, editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018*. European Language Resources Association (ELRA), 2018. URL http://www.lrec-conf.org/proceedings/lrec2018/summaries/1049.html.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In

Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 1693–1701, 2015. URL https://proceedings.neurips.cc/paper/2015/hash/afdec7005cc9f14302cd0474fd0f3c96-Abstract.html.

María Herrero-Zazo, Isabel Segura-Bedmar, Paloma Martínez, and Thierry Declerck. The DDI corpus: An annotated corpus with pharmacological substances and drug-drug interactions. *J. Biomed. Informatics*, 46(5):914–920, 2013. doi: 10.1016/j.jbi.2013.07.011. URL https://doi.org/10.1016/j.jbi.2013.07.011.

Sepp Hochreiter. Untersuchungen zu dynamischen neuronalen netzen. 1991.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. URL https://doi.org/10.1162/neco.1997.9.8.1735.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S. Weld. Knowledge-based weak supervision for information extraction of overlapping relations. In Dekang Lin, Yuji Matsumoto, and Rada Mihalcea, editors, *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pages 541–550. The Association for Computer Linguistics, 2011. URL https://aclanthology.org/P11-1055/.

Gum-Won Hong. Relation extraction using support vector machine. In Robert Dale, Kam-Fai Wong, Jian Su, and Oi Yee Kwong, editors, *Natural Language Processing - IJCNLP 2005, Second International Joint Conference, Jeju Island, Korea, October 11-13, 2005, Proceedings*, volume 3651 of *Lecture Notes in Computer Science*, pages 366–377. Springer, 2005. doi: 10.1007/11562214_33. URL https://doi.org/10.1007/11562214_33.

Matthew Honnibal, Ines Montani, Matthew Honnibal, Henning Peters, Sofie Van Landeghem, Maxim Samsonov, Jim Geovedi, Jim Regan, György Orosz, Søren Lind Kristiansen, Paul O'Leary McCann, Duygu Altinok, Roman, Grégory Howard, Sam Bozek, Explosion Bot, Mark Amery, Wannaphong Phatthiyaphaibun, Leif Uwe Vogelsang, Björn Böing, Pradeep Kumar Tippa, jeannefukumaru, GregDubbin, Vadim Mazaev, Ramanan Balakrishnan, Jens Dahl Møllerhøj, wbwseeker, Magnus Burton, thomasO, and Avadh Patel. explosion/spaCy: v2.1.7: Improved evaluation, better language factories and bug fixes, August 2019. URL https://doi.org/10.5281/zenodo.3358113.

Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional LSTM-CRF models for sequence tagging. *CoRR*, abs/1508.01991, 2015. URL http://arxiv.org/abs/1508.01991.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. Adversarial example generation with syntactically controlled paraphrase networks. In Marilyn A. Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language*

Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers), pages 1875–1885. Association for Computational Linguistics, 2018. doi: 10.18653/v1/n18-1170. URL https://doi.org/10.18653/v1/n18-1170.

Areej Jaber and Paloma Martínez. Participation of UC3M in sdu@aaai-21: A hybrid approach to disambiguate scientific acronyms. In Amir Pouran Ben Veyseh, Franck Dernoncourt, Thien Huu Nguyen, Walter Chang, and Leo Anthony Celi, editors, *Proceedings of the Workshop on Scientific Document Understanding co-located with 35th AAAI Conference on Artificial Inteligence, SDU@AAAI 2021, Virtual Event, February 9, 2021*, volume 2831 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2021.

Heng Ji and Ralph Grishman. Knowledge base population: Successful approaches and challenges. In Dekang Lin, Yuji Matsumoto, and Rada Mihalcea, editors, *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pages 1148–1158. The Association for Computer Linguistics, 2011. URL https://aclanthology.org/P11-1115/.

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Trans. Neural Networks Learn. Syst.*, 33(2):494–514, 2022. doi: 10.1109/TNNLS.2021.3070843. URL https://doi.org/10.1109/TNNLS.2021.3070843.

Xiaotian Jiang, Quan Wang, Peng Li, and Bin Wang. Relation extraction with multi-instance multi-label convolutional neural networks. In Nicoletta Calzolari, Yuji Matsumoto, and Rashmi Prasad, editors, *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pages 1471–1480. ACL, 2016. URL https://aclanthology.org/C16-1139/.

Zhengbao Jiang, Jialong Han, Bunyamin Sisman, and Xin Luna Dong. Cori: Collective relation integration with data augmentation for open information extraction. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4706–4716. Association for Computational Linguistics, 2021. URL https://doi.org/10.18653/v1/2021.acl-long.363.

Karen Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. volume 60, pages 493–502, 2004. URL https://doi.org/10.1108/00220410410560573.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. Spanbert: Improving pre-training by representing and predicting spans. *Trans. Assoc. Comput. Linguistics*, 8:64–77, 2020. URL https://doi.org/10.1162/tacl_a_00300.

Marcin Junczys-Dowmunt. Microsoft translator at WMT 2019: Towards large-scale document-level neural machine translation. In Ondrej Bojar, Rajen Chatterjee, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno-Yepes, Philipp Koehn, André Martins, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana L. Neves, Matt Post, Marco Turchi, and Karin Verspoor, editors, *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 2: Shared Task Papers, Day 1*, pages 225–233. Association for Computational Linguistics, 2019. URL `https://doi.org/10.18653/v1/w19-5321`.

Jaz S. Kandola, John Shawe-Taylor, and Nello Cristianini. Learning semantic similarity. In Suzanna Becker, Sebastian Thrun, and Klaus Obermayer, editors, *Advances in Neural Information Processing Systems 15 [Neural Information Processing Systems, NIPS 2002, December 9-14, 2002, Vancouver, British Columbia, Canada]*, pages 657–664. MIT Press, 2002.

Min Kang, Kye Lee, and Youngho Lee. Filtered bert: Similarity filter-based augmentation with bidirectional transfer learning for protected health information prediction in clinical documents. *Applied Sciences*, 11:3668, 04 2021. doi: 10.3390/app11083668.

Sarvnaz Karimi, Alejandro Metke-Jimenez, Madonna Kemp, and Chen Wang. Cadec: A corpus of adverse drug event annotations. *J. Biomed. Informatics*, 55:73–81, 2015. doi: 10.1016/j.jbi.2015.03.010. URL `https://doi.org/10.1016/j.jbi.2015.03.010`.

Nora Kassner and Hinrich Schütze. Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7811–7818, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.698. URL `https://aclanthology.org/2020.acl-main.698`.

Yoon Kim, Yacine Jernite, David A. Sontag, and Alexander M. Rush. Character-aware neural language models. In Dale Schuurmans and Michael P. Wellman, editors, *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 2741–2749. AAAI Press, 2016. URL `http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12489`.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL `http://arxiv.org/abs/1412.6980`.

Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5): 604–632, 1999.

Sosuke Kobayashi. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,*

Volume 2 (Short Papers), pages 452–457, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2072. URL https://aclanthology.org/N18-2072.

Mamoru Komachi, Taku Kudo, Masashi Shimbo, and Yuji Matsumoto. Graph-based analysis of semantic drift in espresso-like bootstrapping algorithms. In *2008 Conference on Empirical Methods in Natural Language Processing, EMNLP 2008, Proceedings of the Conference, 25-27 October 2008, Honolulu, Hawaii, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1011–1020. ACL, 2008.

Jens Kringelum, Sonny Kim Kjærulff, Søren Brunak, Ole Lund, Tudor I. Oprea, and Olivier Taboureau. Chemprot-3.0: a global chemical biology diseases mapping. *Database J. Biol. Databases Curation*, 2016, 2016. doi: 10.1093/database/bav123. URL https://doi.org/10.1093/database/bav123.

Joseph Bernard Kruskal. On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. In *Proceedings of the American Mathematical Society, 7*, 1956.

Taku Kudo. Subword regularization: Improving neural network translation models with multiple subword candidates. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 66–75. Association for Computational Linguistics, 2018. URL https://aclanthology.org/P18-1007/.

Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In Eduardo Blanco and Wei Lu, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations, Brussels, Belgium, October 31 - November 4, 2018*, pages 66–71. Association for Computational Linguistics, 2018. URL https://doi.org/10.18653/v1/d18-2012.

Sayali Kulkarni, Amit Singh, Ganesh Ramakrishnan, and Soumen Chakrabarti. Collective annotation of wikipedia entities in web text. In John F. Elder IV, Françoise Fogelman-Soulié, Peter A. Flach, and Mohammed Javeed Zaki, editors, *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July 1, 2009*, pages 457–466. ACM, 2009. doi: 10.1145/1557019.1557073. URL https://doi.org/10.1145/1557019.1557073.

Cheng-Ju Kuo, Maurice H. T. Ling, Kuan-Ting Lin, and Chun-Nan Hsu. BIOADI: a machine learning approach to identifying abbreviations and definitions in biological literature. *BMC Bioinform.*, 10(S-15):7, 2009. URL https://doi.org/10.1186/1471-2105-10-S15-S7.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, June 28 - July 1, 2001*, pages 282–289, 2001a.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Carla E. Brodley and Andrea Pohoreckyj Danyluk, editors, *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, June 28 - July 1, 2001*, pages 282–289. Morgan Kaufmann, 2001b.

Xabier Lahuerta, Iakes Goenaga, Koldo Gojenola, Aitziber Atutxa, and Maite Oronoz. Ixamed at pharmaconer challenge 2019. In Jin-Dong Kim, Claire Nédellec, Robert Bossy, and Louise Deléger, editors, *Proceedings of The 5th Workshop on BioNLP Open Shared Tasks, BioNLP-OST@EMNLP-IJNCLP 2019, Hong Kong, China, November 4, 2019*, pages 21–25. Association for Computational Linguistics, 2019. URL https://doi.org/10.18653/v1/D19-5704.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In Kevin Knight, Ani Nenkova, and Owen Rambow, editors, *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 260–270. The Association for Computational Linguistics, 2016. URL https://doi.org/10.18653/v1/n16-1030.

Lukas Lange, Heike Adel, and Jannik Strötgen. NLNDE: enhancing neural sequence taggers with attention and noisy channel for robust pharmacological entity detection. In Jin-Dong Kim, Claire Nédellec, Robert Bossy, and Louise Deléger, editors, *Proceedings of The 5th Workshop on BioNLP Open Shared Tasks, BioNLP-OST@EMNLP-IJNCLP 2019, Hong Kong, China, November 4, 2019*, pages 26–32. Association for Computational Linguistics, 2019. URL https://doi.org/10.18653/v1/D19-5705.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinform.*, 36(4):1234–1240, 2020. doi: 10.1093/bioinformatics/btz682. URL https://doi.org/10.1093/bioinformatics/btz682.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. Dbpedia - A large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195, 2015. URL https://doi.org/10.3233/SW-140134.

Feng Li, Zhensheng Mai, Wuhe Zou, Wenjie Ou, Xiaolei Qin, Yue Lin, and Weidong Zhang. Systems at SDU-2021 task 1: Transformers for sentence level sequence label. In Amir Pouran Ben Veyseh, Franck Dernoncourt, Thien Huu Nguyen, Walter Chang, and Leo Anthony Celi, editors, *Proceedings of the Workshop on Scientific Document Understanding co-located with 35th AAAI Conference on Artificial Inteligence, SDU@AAAI 2021, Virtual Event, February 9, 2021*, volume 2831 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2021. URL http://ceur-ws.org/Vol-2831/paper26.pdf.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. In Kevin Knight, Ani Nenkova, and Owen Rambow, editors, *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 110–119. The Association for Computational Linguistics, 2016. URL https://doi.org/10.18653/v1/n16-1014.

Yang Li, Bo Zhao, Ariel Fuxman, and Fangbo Tao. Guess me if you can: Acronym disambiguation for enterprises. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1308–1317. Association for Computational Linguistics, 2018. URL https://aclanthology.org/P18-1121/.

Yaoyong Li, Kalina Bontcheva, and Hamish Cunningham. SVM based learning system for information extraction. In Joab R. Winkler, Mahesan Niranjan, and Neil D. Lawrence, editors, *Deterministic and Statistical Methods in Machine Learning, First International Workshop, Sheffield, UK, September 7-10, 2004, Revised Lectures*, volume 3635 of *Lecture Notes in Computer Science*, pages 319–339. Springer, 2004. URL https://doi.org/10.1007/11559887_19.

Yu Li, Xiao Li, Yating Yang, and Rui Dong. A diverse data augmentation strategy for low-resource neural machine translation. *Inf.*, 11(5):255, 2020. doi: 10.3390/info11050255. URL https://doi.org/10.3390/info11050255.

Tomas Liesting, Flavius Frasincar, and Maria Mihaela Trusca. Data augmentation in a hybrid approach for aspect-based sentiment analysis. In Chih-Cheng Hung, Jiman Hong, Alessio Bechini, and Eunjee Song, editors, *SAC '21: The 36th ACM/SIGAPP Symposium on Applied Computing, Virtual Event, Republic of Korea, March 22-26, 2021*, pages 828–835. ACM, 2021. doi: 10.1145/3412841.3441958. URL https://doi.org/10.1145/3412841.3441958.

Winston Lin, Roman Yangarber, and Ralph Grishman. Bootstrapped learning of semantic classes from positive and negative examples. In *Proceedings of the 20th International Conference on Machine Learning: ICML 2003*, 2003.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics, 2016. URL https://doi.org/10.18653/v1/p16-1200.

Peter J. Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. Generating wikipedia by summarizing long sequences. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC,*

Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net, 2018. URL https://openreview.net/forum?id=Hyg0vbWC-.

Shengyu Liu, Buzhou Tang, Qingcai Chen, and Xiaolong Wang. Effects of semantic features on machine learning-based drug name recognition systems: Word embeddings *vs.* manually constructed dictionaries. *Inf.*, 6(4):848–865, 2015. doi: 10.3390/info6040848. URL https://doi.org/10.3390/info6040848.

Yang Liu and Mirella Lapata. Text summarization with pretrained encoders. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3728–3738. Association for Computational Linguistics, 2019. URL https://doi.org/10.18653/v1/D19-1387.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019. URL http://arxiv.org/abs/1907.11692.

Zhiyong Lu. Pubmed and beyond: a survey of web tools for searching biomedical literature. *Database*, 2011, 2011a.

Zhiyong Lu. Pubmed and beyond: a survey of web tools for searching biomedical literature. *Database J. Biol. Databases Curation*, 2011, 2011b. URL https://doi.org/10.1093/database/baq036.

Ying Luo and Hai Zhao. Bipartite flat-graph network for nested named entity recognition. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 6408–6418. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.571. URL https://doi.org/10.18653/v1/2020.acl-main.571.

Xuezhe Ma and Eduard H. Hovy. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics, 2016. URL https://doi.org/10.18653/v1/p16-1101.

Robert Malouf. Markov models for language-independent named entity recognition. In Dan Roth and Antal van den Bosch, editors, *Proceedings of the 6th Conference on Natural Language Learning, CoNLL 2002, Held in cooperation with COLING 2002, Taipei, Taiwan, 2002*. ACL, 2002. URL https://aclanthology.org/W02-2019/.

Jihang Mao and Wanli Liu. Integration of deep learning and traditional machine learning for knowledge extraction from biomedical literature. In Jin-Dong Kim, Claire Nédellec, Robert Bossy, and Louise Deléger, editors, *Proceedings of The 5th Workshop on BioNLP Open Shared Tasks, BioNLP-OST@EMNLP-IJNCLP 2019, Hong Kong, China, November 4, 2019*, pages 168–173. Association for Computational Linguistics, 2019. URL `https://doi.org/10.18653/v1/D19-5724`.

Montserrat Marimon, Jorge Vivaldi, and Núria Bel. Annotation of negation in the IULA Spanish clinical record corpus. In *Proceedings of the Workshop Computational Semantics Beyond Events and Roles*, pages 43–52, Valencia, Spain, April 2017. Association for Computational Linguistics.

Mausam, Michael Schmitz, Stephen Soderland, Robert Bart, and Oren Etzioni. Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 523–534, Jeju Island, Korea, July 2012. Association for Computational Linguistics. URL `https://aclanthology.org/D12-1048`.

Tara McIntosh. Unsupervised discovery of negative categories in lexicon bootstrapping. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 356–365, Cambridge, MA, October 2010. Association for Computational Linguistics. URL `https://aclanthology.org/D10-1035`.

Tara McIntosh and James R. Curran. Reducing semantic drift with bagging and distributional similarity. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 396–404, Suntec, Singapore, August 2009. Association for Computational Linguistics. URL `https://aclanthology.org/P09-1045`.

Alberto Mesa Murgado, Ana Parras Portillo, Pilar López Úbeda, Maite Martin, and Alfonso Ureña-López. Identifying professions & occupations in health-related social media using natural language processing. In *Proceedings of the Sixth Social Media Mining for Health (#SMM4H) Workshop and Shared Task*, pages 141–145, Mexico City, Mexico, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.smm4h-1.31. URL `https://aclanthology.org/2021.smm4h-1.31`.

Sepideh Mesbah, Jie Yang, Robert-Jan Sips, Manuel Valle Torre, Christoph Lofi, Alessandro Bozzon, and Geert-Jan Houben. Training data augmentation for detecting adverse drug reactions in user-generated content. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 2349–2359. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1239. URL `https://doi.org/10.18653/v1/D19-1239`.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In Yoshua Bengio and Yann LeCun, editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013a. URL `http://arxiv.org/abs/1301.3781`.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 3111–3119, 2013b.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 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, June 2013c. Association for Computational Linguistics. URL `https://aclanthology.org/N13-1090`.

George A. Miller. WordNet: A lexical database for English. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*, 1994. URL `https://aclanthology.org/H94-1111`.

Naomi Miller, Eve-Marie Lacroix, and Joyce E. B. Backus. Medlineplus: building and maintaining the national library of medicine's consumer health web service. *Bulletin of the Medical Library Association*, 88 1:11–7, 2000.

Bonan Min, Xiang Li, Ralph Grishman, and Ang Sun. New york university 2012 system for KBP slot filling. In *Proceedings of the Fifth Text Analysis Conference, TAC 2012, Gaithersburg, Maryland, USA, November 5-6, 2012*. NIST, 2012. URL `https://tac.nist.gov/publications/2012/participant.papers/NYU.proceedings.pdf`.

Bonan Min, Ralph Grishman, Li Wan, Chang Wang, and David Gondek. Distant supervision for relation extraction with an incomplete knowledge base. In Lucy Vanderwende, Hal Daumé III, and Katrin Kirchhoff, editors, *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 777–782. The Association for Computational Linguistics, 2013. URL `https://aclanthology.org/N13-1095/`.

Junghyun Min, R. Thomas McCoy, Dipanjan Das, Emily Pitler, and Tal Linzen. Syntactic data augmentation increases robustness to inference heuristics. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10,*

2020, pages 2339–2352. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.212. URL https://doi.org/10.18653/v1/2020.acl-main.212.

Anne-Lyse Minard, Anne-Laure Ligozat, and Brigitte Grau. Multi-class SVM for relation extraction from clinical reports. In Galia Angelova, Kalina Bontcheva, Ruslan Mitkov, and Nicolas Nicolov, editors, *Recent Advances in Natural Language Processing, RANLP 2011, 12-14 September, 2011, Hissar, Bulgaria*, pages 604–609. RANLP 2011 Organising Committee, 2011. URL https://aclanthology.org/R11-1086/.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. Distant supervision for relation extraction without labeled data. In Keh-Yih Su, Jian Su, and Janyce Wiebe, editors, *ACL 2009, Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 2-7 August 2009, Singapore*, pages 1003–1011. The Association for Computer Linguistics, 2009. URL https://aclanthology.org/P09-1113/.

Antonio Miranda-Escalada, Eulàlia Farré-Maduell, Salvador Lima-López, Luis Gascó, Vicent Briva-Iglesias, Marvin Agüero-Torales, and Martin Krallinger. The ProfNER shared task on automatic recognition of occupation mentions in social media: systems, evaluation, guidelines, embeddings and corpora. In *Proceedings of the Sixth Social Media Mining for Health (#SMM4H) Workshop and Shared Task*, pages 13–20, Mexico City, Mexico, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.smm4h-1.3. URL https://aclanthology.org/2021.smm4h-1.3.

Makoto Miwa and Mohit Bansal. End-to-end relation extraction using lstms on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics, 2016. doi: 10.18653/v1/p16-1105. URL https://doi.org/10.18653/v1/p16-1105.

Sunil Mohan and Donghui Li. Medmentions: A large biomedical corpus annotated with UMLS concepts. In *1st Conference on Automated Knowledge Base Construction, AKBC 2019, Amherst, MA, USA, May 20-22, 2019*, 2019. doi: 10.24432/C5G59C. URL https://doi.org/10.24432/C5G59C.

David Mueller and Greg Durrett. Effective use of context in noisy entity linking. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 1024–1029. Association for Computational Linguistics, 2018. doi: 10.18653/v1/d18-1126. URL https://doi.org/10.18653/v1/d18-1126.

Sheshera Mysore, Zach Jensen, Edward Kim, Kevin Huang, Haw-Shiuan Chang, Emma Strubell, Jeffrey Flanigan, Andrew McCallum, and Elsa Olivetti. The materials science procedural text corpus: Annotating materials synthesis procedures with shallow semantic

structures. In Annemarie Friedrich, Deniz Zeyrek, and Jet Hoek, editors, *Proceedings of the 13th Linguistic Annotation Workshop, LAW@ACL 2019, Florence, Italy, August 1, 2019*, pages 56–64. Association for Computational Linguistics, 2019. URL https://doi.org/10.18653/v1/w19-4007.

David Nadeau and Peter D. Turney. A supervised learning approach to acronym identification. In Balázs Kégl and Guy Lapalme, editors, *Advances in Artificial Intelligence, 18th Conference of the Canadian Society for Computational Studies of Intelligence, Canadian AI 2005, Victoria, Canada, May 9-11, 2005, Proceedings*, volume 3501 of *Lecture Notes in Computer Science*, pages 319–329. Springer, 2005. URL https://doi.org/10.1007/11424918_34.

Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In Johannes Fürnkranz and Thorsten Joachims, editors, *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, pages 807–814. Omnipress, 2010. URL https://icml.cc/Conferences/2010/papers/432.pdf.

Usman Naseem, Imran Razzak, Shah Khalid Khan, and Mukesh Prasad. A comprehensive survey on word representation models: From classical to state-of-the-art word representation language models. volume 20, pages 74:1–74:35, 2021. URL https://doi.org/10.1145/3434237.

Maya Natarajan. Knowledge graphs. *Web*. URL https://neo4j.com/use-cases/knowledge-graph/.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. Facebook fair's WMT19 news translation task submission. In Ondrej Bojar, Rajen Chatterjee, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno-Yepes, Philipp Koehn, André Martins, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana L. Neves, Matt Post, Marco Turchi, and Karin Verspoor, editors, *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 2: Shared Task Papers, Day 1*, pages 314–319. Association for Computational Linguistics, 2019. doi: 10.18653/v1/w19-5333. URL https://doi.org/10.18653/v1/w19-5333.

Thien Huu Nguyen and Ralph Grishman. Relation extraction: Perspective from convolutional neural networks. In Phil Blunsom, Shay B. Cohen, Paramveer S. Dhillon, and Percy Liang, editors, *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing, VS@NAACL-HLT 2015, June 5, 2015, Denver, Colorado, USA*, pages 39–48. The Association for Computational Linguistics, 2015. doi: 10.3115/v1/w15-1506. URL https://doi.org/10.3115/v1/w15-1506.

Naoaki Okazaki and Sophia Ananiadou. Building an abbreviation dictionary using a term recognition approach. *Bioinform.*, 22(24):3089–3095, 2006. URL https://doi.org/10.1093/bioinformatics/btl534.

Evangelos Pafilis, Sune P. Frankild, Lucia Fanini, Sarah Faulwetter, Christina Pavloudi, Aikaterini Vasileiadou, Christos Arvanitidis, and Lars Juhl Jensen. The species and organisms resources for fast and accurate identification of taxonomic names in text. *PLoS ONE*, 8(6):1–6, 06 2013.

Vasile Pais and Maria Mitrofan. Assessing multiple word embeddings for named entity recognition of professions and occupations in health-related social media. In *Proceedings of the Sixth Social Media Mining for Health (#SMM4H) Workshop and Shared Task*, pages 128–130, Mexico City, Mexico, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.smm4h-1.27. URL https://aclanthology.org/2021.smm4h-1.27.

Patrick Pantel and Marco Pennacchiotti. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In Nicoletta Calzolari, Claire Cardie, and Pierre Isabelle, editors, *ACL 2006, 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, Sydney, Australia, 17-21 July 2006*. The Association for Computer Linguistics, 2006. doi: 10.3115/1220175.1220190. URL https://aclanthology.org/P06-1015/.

Yannis Papanikolaou and Andrea Pierleoni. DARE: data augmented relation extraction with GPT-2. *CoRR*, abs/2004.13845, 2020. URL https://arxiv.org/abs/2004.13845.

Youngja Park and Roy J. Byrd. Hybrid text mining for finding abbreviations and their definitions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2001, Pittsburgh, PA USA, June 3-4, 2001*. ACL, 2001. URL https://aclanthology.org/W01-0516/.

Marius Paşca, Dekang Lin, Jeffrey Bigham, Andrei Lifchits, and Alpa Jain. Names and similarities on the web: Fact extraction in the fast lane. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 809–816, Sydney, Australia, July 2006. Association for Computational Linguistics. doi: 10.3115/1220175.1220277. URL https://aclanthology.org/P06-1102.

Razvan Pascanu, Tomás Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, volume 28 of *JMLR Workshop and Conference Proceedings*, pages 1310–1318. JMLR.org, 2013. URL http://proceedings.mlr.press/v28/pascanu13.html.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543. ACL, 2014. URL https://doi.org/10.3115/v1/d14-1162.

Maria Pershina, Bonan Min, Wei Xu, and Ralph Grishman. Infusion of labeled data into distant supervision for relation extraction. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 2: Short Papers*, pages 732–738. The Association for Computer Linguistics, 2014. URL https://doi.org/10.3115/v1/p14-2119.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In Marilyn A. Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 2227–2237. Association for Computational Linguistics, 2018a. URL https://doi.org/10.18653/v1/n18-1202.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 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, June 2018b. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. URL https://aclanthology.org/N18-1202.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 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, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1250. URL https://aclanthology.org/D19-1250.

Barbara Plank, Anders Søgaard, and Yoav Goldberg. Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2: Short Papers*. The Association for Computer Linguistics, 2016. URL https://doi.org/10.18653/v1/p16-2067.

Lutz Prechelt. Early stopping-but when? In Genevieve B. Orr and Klaus-Robert Müller, editors, *Neural Networks: Tricks of the Trade*, volume 1524 of *Lecture Notes in Computer Science*, pages 55–69. Springer, 1996. doi: 10.1007/3-540-49430-8_3. URL https://doi.org/10.1007/3-540-49430-8_3.

Likun Qiu, Weishi Zhang, Changjian Hu, and Kai Zhao. SELC: a self-supervised model for sentiment classification. In David Wai-Lok Cheung, Il-Yeol Song, Wesley W. Chu, Xiaohua Hu, and Jimmy Lin, editors, *Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM 2009, Hong Kong, China, November 2-6, 2009*, pages 929–936. ACM, 2009. doi: 10.1145/1645953.1646072. URL https://doi.org/10.1145/1645953.1646072.

L. Rabiner and B. Juang. An introduction to hidden markov models. *IEEE ASSP Magazine*, 3(1):4–16, 1986. doi: 10.1109/MASSP.1986.1165342.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*, 2018.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. , the. *OpenAI Blog*, 1(8), 2019.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100, 000+ questions for machine comprehension of text. In Jian Su, Xavier Carreras, and Kevin Duh, editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2383–2392. The Association for Computational Linguistics, 2016. URL `https://doi.org/10.18653/v1/d16-1264`.

Sidharth Ramesh, Abhiraj Tiwari, Parthivi Choubey, Saisha Kashyap, Sahil Khose, Kumud Lakara, Nishesh Singh, and Ujjwal Verma. BERT based transformers lead the way in extraction of health information from social media. In *Proceedings of the Sixth Social Media Mining for Health (#SMM4H) Workshop and Shared Task*, pages 33–38, Mexico City, Mexico, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.smm4h-1.5. URL `https://aclanthology.org/2021.smm4h-1.5`.

Lev-Arie Ratinov, Dan Roth, Doug Downey, and Mike Anderson. Local and global algorithms for disambiguation to wikipedia. In Dekang Lin, Yuji Matsumoto, and Rada Mihalcea, editors, *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pages 1375–1384. The Association for Computer Linguistics, 2011. URL `https://aclanthology.org/P11-1138/`.

Deepak Ravichandran and Eduard H. Hovy. Learning surface text patterns for a question answering system. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 41–47. ACL, 2002. doi: 10.3115/1073083.1073092. URL `https://aclanthology.org/P02-1006/`.

Siva Reddy, Danqi Chen, and Christopher D. Manning. Coqa: A conversational question answering challenge. *Trans. Assoc. Comput. Linguistics*, 7:249–266, 2019. URL `https://transacl.org/ojs/index.php/tacl/article/view/1572`.

Marek Rei. Semi-supervised multitask learning for sequence labeling. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 2121–2130, 2017.

Sebastian Riedel, Limin Yao, and Andrew McCallum. Modeling relations and their mentions without labeled text. In José L. Balcázar, Francesco Bonchi, Aristides Gionis, and Michèle Sebag, editors, *Machine Learning and Knowledge Discovery in Databases, European Conference, ECML PKDD 2010, Barcelona, Spain, September 20-24, 2010, Proceedings, Part III*, volume 6323 of *Lecture Notes in Computer Science*, pages 148–163. Springer, 2010. URL https://doi.org/10.1007/978-3-642-15939-8_10.

Alan Ritter, Luke Zettlemoyer, Mausam, and Oren Etzioni. Modeling missing data in distant supervision for information extraction. *Trans. Assoc. Comput. Linguistics*, 1: 367–378, 2013. URL https://tacl2013.cs.columbia.edu/ojs/index.php/tacl/article/view/163.

Joaquín Alberto Carballido Rodríguez. [the spanish bibliographic index of the health sciences (ibecs) and actas urológicas españolas]. *Actas urologicas espanolas*, 26 6:381–3, 2002.

Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. volume 65 6, pages 386–408, 1958.

Pedro Ruas, Vitor Andrade, and Francisco Couto. Lasige-BioTM at ProfNER: BiLSTM-CRF and contextual Spanish embeddings for named entity recognition and tweet binary classification. In *Proceedings of the Sixth Social Media Mining for Health (#SMM4H) Workshop and Shared Task*, pages 108–111, Mexico City, Mexico, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.smm4h-1.21. URL https://aclanthology.org/2021.smm4h-1.21.

David E. Rumelhari, Geoffrey E. Hintont, Ronald, J., and Williams. Learning representations by backpropagating errors. volume 323, pages 533–536, 1986. doi: 10.1038/323533a0. URL http://www.nature.com/articles/323533a0.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

Alexander M. Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. In Lluís Màrquez, Chris Callison-Burch, Jian Su, Daniele Pighin, and Yuval Marton, editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 379–389. The Association for Computational Linguistics, 2015. doi: 10.18653/v1/d15-1044. URL https://doi.org/10.18653/v1/d15-1044.

Gözde Gül Sahin and Mark Steedman. Data augmentation via dependency tree morphing for low-resource languages. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 5004–5009. Association for Computational Linguistics, 2018. doi: 10.18653/v1/d18-1545. URL https://doi.org/10.18653/v1/d18-1545.

Andrey Sakhovskiy, Zulfat Miftahutdinov, and Elena Tutubalina. KFU NLP team at SMM4H 2021 tasks: Cross-lingual and cross-modal BERT-based models for adverse drug effects. In *Proceedings of the Sixth Social Media Mining for Health (#SMM4H) Workshop and Shared Task*, pages 39–43, Mexico City, Mexico, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.smm4h-1.6. URL https://aclanthology.org/2021.smm4h-1.6.

Benjamin Sanchez-Lengeling, Emily Reif, Adam Pearce, and Alexander B. Wiltschko. A gentle introduction to graph neural networks. *Distill*, 2021. doi: 10.23915/distill.00033. https://distill.pub/2021/gnn-intro.

Sergio Santamaría. Troy abendinthemorning system code, 2021.

Moein Salimi Sartakhti, Romina Etezadi, and Mehrnoush Shamsfard. Improving persian relation extraction models by data augmentation. In *Proceedings of The Second International Workshop on NLP Solutions for Under Resourced Languages (NSURL 2021) co-located with ICNLSP 2021*, pages 32–37, Trento, Italy, 12–13 November 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.nsurl-1.5.

Timo Schick and Hinrich Schütze. Generating datasets with pretrained language models. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 6943–6951. Association for Computational Linguistics, 2021. URL https://aclanthology.org/2021.emnlp-main.555.

Ariel S. Schwartz and Marti A. Hearst. A simple algorithm for identifying abbreviation definitions in biomedical text. In Russ B. Altman, A. Keith Dunker, Lawrence Hunter, and Teri E. Klein, editors, *Proceedings of the 8th Pacific Symposium on Biocomputing, PSB 2003, Lihue, Hawaii, USA, January 3-7, 2003*, pages 451–462, 2003. URL http://psb.stanford.edu/psb-online/proceedings/psb03/schwartz.pdf.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics, 2016. doi: 10.18653/v1/p16-1162. URL https://doi.org/10.18653/v1/p16-1162.

Özge Sevgili, Artem Shelmanov, Mikhail Y. Arkhipov, Alexander Panchenko, and Chris Biemann. Neural entity linking: A survey of models based on deep learning. *Semantic Web*, 13(3):527–570, 2022. doi: 10.3233/SW-222986. URL https://doi.org/10.3233/SW-222986.

Wei Shen, Jianyong Wang, Ping Luo, and Min Wang. LINDEN: linking named entities with knowledge base via semantic knowledge. In Alain Mille, Fabien Gandon, Jacques Misselis,

Michael Rabinovich, and Steffen Staab, editors, *Proceedings of the 21st World Wide Web Conference 2012, WWW 2012, Lyon, France, April 16-20, 2012*, pages 449–458. ACM, 2012. doi: 10.1145/2187836.2187898. URL https://doi.org/10.1145/2187836.2187898.

Wei Shen, Yuhan Li, Yinan Liu, Jiawei Han, Jianyong Wang, and Xiaojie Yuan. Entity linking meets deep learning: Techniques and solutions. *CoRR*, abs/2109.12520, 2021. URL https://arxiv.org/abs/2109.12520.

Sam Shleifer. Low resource text classification with ulmfit and backtranslation. *CoRR*, abs/1903.09244, 2019. URL http://arxiv.org/abs/1903.09244.

Jonathon Shlens. A tutorial on principal component analysis. *CoRR*, abs/1404.1100, 2014. URL http://arxiv.org/abs/1404.1100.

Connor Shorten and Taghi M. Khoshgoftaar. A survey on image data augmentation for deep learning. *J. Big Data*, 6:60, 2019. doi: 10.1186/s40537-019-0197-0. URL https://doi.org/10.1186/s40537-019-0197-0.

Amit Singhal. Introducing the knowledge graph: Things, not strings. *Official Google Blog*, 2012. URL https://blog.google/products/search/introducing-knowledge-graph-things-not/.

Alex J. Smola and Risi Imre Kondor. Kernels and regularization of graphs. In Suzanna Becker, Sebastian Thrun, and Klaus Obermayer, editors, *Annual Conference on Learning Theory, 2003*, pages 144–158. MIT Press, 2003.

Felipe Soares and Aitor gonzalez agirre. Spaccc-pos-tagger, April 2019. Funded by the Plan de Impulso de las Tecnologías del Lenguaje (Plan TL).

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. Matching the blanks: Distributional similarity for relation learning. In Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2895–2905. Association for Computational Linguistics, 2019. doi: 10.18653/v1/p19-1279. URL https://doi.org/10.18653/v1/p19-1279.

Mohammad Golam Sohrab, Pham Minh Thang, Makoto Miwa, and Hiroya Takamura. A neural pipeline approach for the pharmaconer shared task using contextual exhaustive models. In Jin-Dong Kim, Claire Nédellec, Robert Bossy, and Louise Deléger, editors, *Proceedings of The 5th Workshop on BioNLP Open Shared Tasks, BioNLP-OST@EMNLP-IJNCLP 2019, Hong Kong, China, November 4, 2019*, pages 47–55. Association for Computational Linguistics, 2019. URL https://doi.org/10.18653/v1/D19-5708.

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, 2014. doi: 10.5555/2627435.2670313. URL https://dl.acm.org/doi/10.5555/2627435.2670313.

Frans Stokman and Pieter Vries. *Structuring Knowledge in a Graph*, pages 186–206. 01 1988. ISBN 0-387-18901-7. doi: 10.1007/978-3-642-73402-1_12.

Víctor Suárez-Paniagua. VSP at pharmaconer 2019: Recognition of pharmacological substances, compounds and proteins with recurrent neural networks in spanish clinical cases. In Jin-Dong Kim, Claire Nédellec, Robert Bossy, and Louise Deléger, editors, *Proceedings of The 5th Workshop on BioNLP Open Shared Tasks, BioNLP-OST@EMNLP-IJNCLP 2019, Hong Kong, China, November 4, 2019*, pages 16–20. Association for Computational Linguistics, 2019. URL https://doi.org/10.18653/v1/D19-5703.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In Carey L. Williamson, Mary Ellen Zurko, Peter F. Patel-Schneider, and Prashant J. Shenoy, editors, *Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007*, pages 697–706. ACM, 2007. URL https://doi.org/10.1145/1242572.1242667.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D. Manning. Multi-instance multi-label learning for relation extraction. In Jun'ichi Tsujii, James Henderson, and Marius Pasca, editors, *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2012, July 12-14, 2012, Jeju Island, Korea*, pages 455–465. ACL, 2012. URL https://aclanthology.org/D12-1042/.

Shingo Takamatsu, Issei Sato, and Hiroshi Nakagawa. Reducing wrong labels in distant supervision for relation extraction. In *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, July 8-14, 2012, Jeju Island, Korea - Volume 1: Long Papers*, pages 721–729. The Association for Computer Linguistics, 2012. URL https://aclanthology.org/P12-1076/.

Thomas Pellissier Tanon, Denny Vrandecic, Sebastian Schaffert, Thomas Steiner, and Lydia Pintscher. From freebase to wikidata: The great migration. In Jacqueline Bourdeau, Jim Hendler, Roger Nkambou, Ian Horrocks, and Ben Y. Zhao, editors, *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016*, pages 1419–1428. ACM, 2016. URL https://doi.org/10.1145/2872427.2874809.

Michael Thelen and Ellen Riloff. A bootstrapping method for learning semantic lexicons using extraction pattern contexts. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 214–221. Association for Computational Linguistics, July 2002. doi: 10.3115/1118693.1118721. URL https://aclanthology.org/W02-1028.

Jörg Tiedemann and Santhosh Thottingal. OPUS-MT — Building open translation services for the World. In *Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT)*, Lisbon, Portugal, 2020.

Richard Tzong-Han Tsai, Shih-Hung Wu, Wen-Chi Chou, Yu-Chun Lin, Ding He, Jieh Hsiang, Ting-Yi Sung, and Wen-Lian Hsu. Various criteria in the evaluation of biomedical named entity recognition. *BMC Bioinformatics*, 7:92, 2006.

Lingraj S. Vannur, Balaji Ganesan, Lokesh Nagalapatti, Hima Patel, and M. N. Tippeswamy. Data augmentation for fairness in personal knowledge base population. In Manish Gupta and Ganesh Ramakrishnan, editors, *Trends and Applications in Knowledge Discovery and Data Mining - PAKDD 2021 Workshops, WSPA, MLMEIN, SDPRA, DARAI, and AI4EPT, Delhi, India, May 11, 2021 Proceedings*, volume 12705 of *Lecture Notes in Computer Science*, pages 143–152. Springer, 2021. URL `https://doi.org/10.1007/978-3-030-75015-2_15`.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017. URL `https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html`.

Amir Pouran Ben Veyseh, Franck Dernoncourt, Walter Chang, and Thien Huu Nguyen. Maddog: A web-based system for acronym identification and disambiguation. In Dimitra Gkatzia and Djamé Seddah, editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations, EACL 2021, Online, April 19-23, 2021*, pages 160–167. Association for Computational Linguistics, 2021. URL `https://doi.org/10.18653/v1/2021.eacl-demos.20`.

Amir Pouran Ben Veyseh, Nicole Meister, Seunghyun Yoon, Rajiv Jain, Franck Dernoncourt, and Thien Huu Nguyen. MACRONYM: A Large-Scale Dataset for Multilingual and Multi-Domain Acronym Extraction. In *arXiv*, 2022a.

Amir Pouran Ben Veyseh, Nicole Meister, Seunghyun Yoon, Rajiv Jain, Franck Dernoncourt, and Thien Huu Nguyen. Multilingual Acronym Extraction and Disambiguation Shared Tasks at SDU 2022. In *Proceedings of SDU@AAAI-22*, 2022b.

Oriol Vinyals and Quoc V. Le. A neural conversational model. *CoRR*, abs/1506.05869, 2015. URL `http://arxiv.org/abs/1506.05869`.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL `https://openreview.net/forum?id=rJ4km2R5t7`.

Hailin Wang, Ke Qin, Rufai Yusuf Zakari, Guoming Lu, and Jin Yin. Deep neural network-based relation extraction: an overview. *Neural Comput. Appl.*, 34(6):4781–4801, 2022. doi: 10.1007/s00521-021-06667-3. URL https://doi.org/10.1007/s00521-021-06667-3.

Linlin Wang, Zhu Cao, Gerard de Melo, and Zhiyuan Liu. Relation classification via multi-level attention CNNs. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1298–1307, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1123. URL https://aclanthology.org/P16-1123.

Xinyi Wang, Hieu Pham, Zihang Dai, and Graham Neubig. Switchout: an efficient data augmentation algorithm for neural machine translation. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 856–861. Association for Computational Linguistics, 2018a. doi: 10.18653/v1/d18-1100. URL https://doi.org/10.18653/v1/d18-1100.

Xinyi Wang, Hieu Pham, Zihang Dai, and Graham Neubig. SwitchOut: an efficient data augmentation algorithm for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 856–861, Brussels, Belgium, October-November 2018b. Association for Computational Linguistics. doi: 10.18653/v1/D18-1100. URL https://aclanthology.org/D18-1100.

Jason W. Wei and Kai Zou. EDA: easy data augmentation techniques for boosting performance on text classification tasks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 6381–6387. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1670. URL https://doi.org/10.18653/v1/D19-1670.

Robert West, Evgeniy Gabrilovich, Kevin Murphy, Shaohua Sun, Rahul Gupta, and Dekang Lin. Knowledge base completion via search-based question answering. In Chin-Wan Chung, Andrei Z. Broder, Kyuseok Shim, and Torsten Suel, editors, *23rd International World Wide Web Conference, WWW '14, Seoul, Republic of Korea, April 7-11, 2014*, pages 515–526. ACM, 2014. URL https://doi.org/10.1145/2566486.2568032.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In Qun Liu and David Schlangen, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing:*

System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020, pages 38–45. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://doi.org/10.18653/v1/2020.emnlp-demos.6.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144, 2016. URL http://arxiv.org/abs/1609.08144.

Qizhe Xie, Zihang Dai, Eduard H. Hovy, Minh-Thang Luong, and Quoc V. Le. Unsupervised data augmentation. *CoRR*, abs/1904.12848, 2019. URL http://arxiv.org/abs/1904.12848.

Ying Xiong, Yedan Shen, Yuanhang Huang, Shuai Chen, Buzhou Tang, Xiaolong Wang, Qingcai Chen, Jun Yan, and Yi Zhou. A deep learning-based system for pharmaconer. In Jin-Dong Kim, Claire Nédellec, Robert Bossy, and Louise Deléger, editors, *Proceedings of The 5th Workshop on BioNLP Open Shared Tasks, BioNLP-OST@EMNLP-IJNCLP 2019, Hong Kong, China, November 4, 2019*, pages 33–37. Association for Computational Linguistics, 2019. URL https://doi.org/10.18653/v1/D19-5706.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In Francis R. Bach and David M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 2048–2057. JMLR.org, 2015. URL http://proceedings.mlr.press/v37/xuc15.html.

Kun Xu, Siva Reddy, Yansong Feng, Songfang Huang, and Dongyan Zhao. Question answering on freebase via relation extraction and textual evidence. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics, 2016a. doi: 10.18653/v1/p16-1220. URL https://doi.org/10.18653/v1/p16-1220.

Wei Xu, Raphael Hoffmann, Le Zhao, and Ralph Grishman. Filling knowledge base gaps for distant supervision of relation extraction. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 2: Short Papers*, pages 665–670. The Association for Computer Linguistics, 2013. URL https://aclanthology.org/P13-2117/.

Yan Xu, Ran Jia, Lili Mou, Ge Li, Yunchuan Chen, Yangyang Lu, and Zhi Jin. Improved relation classification by deep recurrent neural networks with data augmentation. In Nicoletta Calzolari, Yuji Matsumoto, and Rashmi Prasad, editors, *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pages 1461–1470. ACL, 2016b. URL https://aclanthology.org/C16-1138/.

Vikas Yadav and Steven Bethard. A survey on recent advances in named entity recognition from deep learning models. In Emily M. Bender, Leon Derczynski, and Pierre Isabelle, editors, *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 2145–2158. Association for Computational Linguistics, 2018. URL https://aclanthology.org/C18-1182/.

Ikuya Yamada, Akari Asai, Jin Sakuma, Hiroyuki Shindo, Hideaki Takeda, Yoshiyasu Takefuji, and Yuji Matsumoto. Wikipedia2vec: An efficient toolkit for learning and visualizing the embeddings of words and entities from wikipedia. In Qun Liu and David Schlangen, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020*, pages 23–30. Association for Computational Linguistics, 2020. URL https://doi.org/10.18653/v1/2020.emnlp-demos.4.

Xi Yang, Zehao Yu, Yi Guo, Jiang Bian, and Yonghui Wu. Clinical relation extraction using transformer-based models. *CoRR*, abs/2107.08957, 2021. URL https://arxiv.org/abs/2107.08957.

Ze Yang, Wei Wu, Jian Yang, Can Xu, and Zhoujun Li. Low-resource response generation with template prior. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 1886–1897. Association for Computational Linguistics, 2019a. URL https://doi.org/10.18653/v1/D19-1197.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 5754–5764, 2019b.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy. Hierarchical attention networks for document classification. In Kevin Knight, Ani Nenkova, and Owen Rambow, editors, *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 1480–1489.

The Association for Computational Linguistics, 2016. doi: 10.18653/v1/n16-1174. URL https://doi.org/10.18653/v1/n16-1174.

Roman Yangarber, Winston Lin, and Ralph Grishman. Unsupervised learning of generalized names. In *COLING 2002: The 19th International Conference on Computational Linguistics*, 2002. URL https://aclanthology.org/C02-1154.

Usama Yaseen and Stefan Langer. Data augmentation for low-resource named entity recognition using backtranslation. 2021a. URL https://arxiv.org/abs/2108.11703.

Usama Yaseen and Stefan Langer. Neural text classification and stacked heterogeneous embeddings for named entity recognition in SMM4H 2021. In *Proceedings of the Sixth Social Media Mining for Health (#SMM4H) Workshop and Shared Task*, pages 83–87, Mexico City, Mexico, June 2021b. Association for Computational Linguistics. URL https://aclanthology.org/2021.smm4h-1.14.

Usama Yaseen and Stefan Langer. Domain adaptive pretraining for multilingual acronym extraction. *SDU@AAAI*, 2022.

Usama Yaseen, Pankaj Gupta, and Hinrich Schütze. Linguistically informed relation extraction and neural architectures for nested named entity recognition in bionlp-ost 2019. In Jin-Dong Kim, Claire Nédellec, Robert Bossy, and Louise Deléger, editors, *Proceedings of The 5th Workshop on BioNLP Open Shared Tasks, BioNLP-OST@EMNLP-IJNCLP 2019, Hong Kong, China, November 4, 2019*, pages 132–142. Association for Computational Linguistics, 2019. URL https://doi.org/10.18653/v1/D19-5720.

Kang Min Yoo, Youhyun Shin, and Sang-goo Lee. Data augmentation for spoken language understanding via joint variational generation. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 7402–7409. AAAI Press, 2019. doi: 10.1609/aaai.v33i01.33017402. URL https://doi.org/10.1609/aaai.v33i01.33017402.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le. Qanet: Combining local convolution with global self-attention for reading comprehension. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=B14TlG-RW.

Norshuhani Zamin and Alan Oxley. Building a corpus-derived gazetteer for named entity recognition. In Jasni Mohamad Zain, Wan Maseri Binti Wan Mohd, and Eyas El-Qawasmeh, editors, *Software Engineering and Computer Systems - Second International Conference, ICSECS 2011, Kuantan, Pahang, Malaysia, June 27-29, 2011, Proceedings, Part II*, volume 180 of *Communications in Computer and Information Science*, pages

73–80. Springer, 2011. doi: 10.1007/978-3-642-22191-0_6. URL https://doi.org/10.1007/978-3-642-22191-0_6.

Matthew D. Zeiler. ADADELTA: an adaptive learning rate method. *CoRR*, abs/1212.5701, 2012. URL http://arxiv.org/abs/1212.5701.

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. Distant supervision for relation extraction via piecewise convolutional neural networks. In Lluís Màrquez, Chris Callison-Burch, Jian Su, Daniele Pighin, and Yuval Marton, editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 1753–1762. The Association for Computational Linguistics, 2015. URL https://doi.org/10.18653/v1/d15-1203.

Hanchu Zhang, Leonhard Hennig, Christoph Alt, Changjian Hu, Yao Meng, and Chao Wang. Bootstrapping named entity recognition in E-commerce with positive unlabeled learning. In *Proceedings of The 3rd Workshop on e-Commerce and NLP*, pages 1–6, Seattle, WA, USA, July 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.ecnlp-1.1. URL https://aclanthology.org/2020.ecnlp-1.1.

Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=r1Ddp1-Rb.

Qi Zhang, Chao Liu, Ying Chi, Xuansong Xie, and Xian-Sheng Hua. A multi-task learning framework for extracting bacteria biotope information. In Jin-Dong Kim, Claire Nédellec, Robert Bossy, and Louise Deléger, editors, *Proceedings of The 5th Workshop on BioNLP Open Shared Tasks, BioNLP-OST@EMNLP-IJNCLP 2019, Hong Kong, China, November 4, 2019*, pages 105–109. Association for Computational Linguistics, 2019a. URL https://doi.org/10.18653/v1/D19-5716.

Rongzhi Zhang, Yue Yu, and Chao Zhang. Seqmix: Augmenting active sequence labeling via sequence mixup. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 8566–8579. Association for Computational Linguistics, 2020b. doi: 10.18653/v1/2020.emnlp-main.691. URL https://doi.org/10.18653/v1/2020.emnlp-main.691.

Shaodian Zhang and Noemie Elhadad. Unsupervised biomedical named entity recognition: Experiments with clinical and biological texts. *J. Biomed. Informatics*, 46(6):1088–1098, 2013. doi: 10.1016/j.jbi.2013.08.004. URL https://doi.org/10.1016/j.jbi.2013.08.004.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 28:*

Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada, pages 649–657, 2015. URL https://proceedings.neurips.cc/paper/2015/hash/250cf8b51c773f3f8dc8b4be867a9a02-Abstract.html.

Xingxing Zhang, Jianwen Zhang, Junyu Zeng, Jun Yan, Zheng Chen, and Zhifang Sui. Towards accurate distant supervision for relational facts extraction. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 2: Short Papers*, pages 810–815. The Association for Computer Linguistics, 2013. URL https://aclanthology.org/P13-2141/.

Yijia Zhang, Qingyu Chen, Zhihao Yang, Hongfei Lin, and Zhiyong Lu. Biowordvec, improving biomedical word embeddings with subword information and mesh. In *Scientific Data*, 2019b.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. Position-aware attention and supervised data improve slot filling. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 35–45. Association for Computational Linguistics, 2017a. URL https://doi.org/10.18653/v1/d17-1004.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45, Copenhagen, Denmark, September 2017b. Association for Computational Linguistics. doi: 10.18653/v1/D17-1004. URL https://aclanthology.org/D17-1004.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, Florence, Italy, July 2019c. Association for Computational Linguistics. doi: 10.18653/v1/P19-1139. URL https://aclanthology.org/P19-1139.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. Gender bias in coreference resolution: Evaluation and debiasing methods. In Marilyn A. Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 15–20. Association for Computational Linguistics, 2018. doi: 10.18653/v1/n18-2003. URL https://doi.org/10.18653/v1/n18-2003.

Guodong Zhou and Jian Su. Named entity recognition using an hmm-based chunk tagger. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 473–480. ACL, 2002. doi: 10.3115/1073083.1073163. URL https://aclanthology.org/P02-1060/.

Tong Zhou, Zhucong Li, Zhen Gan, Baoli Zhang, Yubo Chen, Kun Niu, Jing Wan, Kang Liu, Jun Zhao, Yafei Shi, Weifeng Chong, and Shengping Liu. Classification, extraction, and normalization : CASIA_Unisound team at the social media mining for health 2021 shared tasks. In *Proceedings of the Sixth Social Media Mining for Health (#SMM4H) Workshop and Shared Task*, pages 77–82, Mexico City, Mexico, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.smm4h-1.13. URL `https://aclanthology.org/2021.smm4h-1.13`.

Wenxuan Zhou, Hongtao Lin, Bill Yuchen Lin, Ziqi Wang, Junyi Du, Leonardo Neves, and Xiang Ren. NERO: A neural rule grounding framework for label-efficient relation extraction. In Yennun Huang, Irwin King, Tie-Yan Liu, and Maarten van Steen, editors, *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, pages 2166–2176. ACM / IW3C2, 2020. URL `https://doi.org/10.1145/3366423.3380282`.

Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 19–27. IEEE Computer Society, 2015. URL `https://doi.org/10.1109/ICCV.2015.11`.

Stefan Zwicklbauer, Christin Seifert, and Michael Granitzer. Robust and collective entity disambiguation through semantic embeddings. In Raffaele Perego, Fabrizio Sebastiani, Javed A. Aslam, Ian Ruthven, and Justin Zobel, editors, *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, SIGIR 2016, Pisa, Italy, July 17-21, 2016*, pages 425–434. ACM, 2016. doi: 10.1145/2911451.2911535. URL `https://doi.org/10.1145/2911451.2911535`.