ENTITY LINKING IN LOW-ANNOTATION DATA SETTINGS

by

Elliot Schumacher

A dissertation submitted to The Johns Hopkins University

in conformity with the requirements for the degree of Doctor of Philosophy.

Baltimore, Maryland

December, 2022

Abstract

Recent advances in natural language processing have focused on applying and adapting large pretrained language models to specific tasks. These models, such as BERT (Devlin et al., 2019) and BART (Lewis et al., 2020a), are pretrained on massive amounts of unlabeled text across a variety of domains. The impact of these pretrained models is visible in the task of entity linking, where a mention of an entity in unstructured text is matched to the relevant entry in a knowledge base. State-of-the-art linkers, such as Wu et al. (2020) and De Cao et al. (2021), leverage pretrained models as a foundation for their systems. However, these models are also trained on large amounts of annotated data, which is crucial to their performance. Often these large datasets consist of domains that are easily annotated, such as Wikipedia or newswire text. However, tailoring NLP tools to a narrow variety of textual domains severely restricts their use in the real world.

Many other domains, such as medicine or law, do not have large amounts of entity linking annotations available. Entity linking, which serves to bridge the gap between massive unstructured amounts of text and structured repositories of knowledge, is equally crucial in these domains. Yet tools trained on newswire or Wikipedia annotations are unlikely to be well-suited for identifying medical conditions mentioned in clinical notes. As most annotation efforts focus on English, similar challenges can be noted in building systems for non-English text. There is often a relatively small amount of annotated data in these domains. With this being the case, looking to other types of domain-specific data, such as unannotated text or highly-curated structured knowledge bases, is often required. In these settings, it is crucial to translate lessons taken from tools tailored for high-annotation domains into algorithms that are suited for low-annotation domains. This requires both leveraging broader types of data and understanding the unique challenges present in each domain.

Primary Reader and Advisor: Mark Dredze

Secondary Readers: James Mayfield & Tom Lippincott

Acknowledgments

This endeavor would not have been possible without Dr. Mark Dredze, who always urged me to explore new ideas, and focus on the fundamentals of research. In addition, Dr. James Mayfield also was a consistent source of timely advice. Also, many thanks to Dr. Tom Lippencott for his valuable feedback for this thesis. I am also grateful to many other research mentors, such as my masters advisors Dr. Maxine Eskenazi and Dr. Kevyn Collins-Thompson, and my Amazon internship mentor Dr. Lluís Marquez. Thanks should also go to the Human Language Technology Center of Excellence and Johns Hopkins University for funding my graduate experience. Perhaps most importantly, I very much appreciate the hard work of the administrative staff at Johns Hopkins – especially Ruth Scally, who works hard for all of the CLSP students.

I'd also like to thank my fellow students at the CLSP, who helped to create an enjoyable research experience. Many thanks to the students who made time for fun outside of research, whether climbing, practicing Spanish, or watching movies. I'd also like to thank my fellow lab mates who made Mark's lab into a fun workplace. I'd especially like to thank Huda for always being willing to listen and give advice, and for Adam, who always made our lab fun.

Finally, I'd like to thank my family, who always supported me throughout my graduate school experience. This includes my brother Grant, who gave great advice on good distractions, and my grandparents, who maybe didn't understand what I was studying but were proud nonetheless. The biggest thanks go to my parents, who have always supported me. Thank you all!

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In domains as disparate as College Football to Medicine, a great deal of effort has been undertaken to organize and structure domain knowledge. Such structured sources of data, known as knowledge bases (KB), are created to organize concepts and objects into structures that can be interpreted by human beings or automated processes. These typically consist of entries of domain-driven entities or concepts with supporting metadata, such as definitions, and relationships between entries. Knowledge bases allow for users to reference information about a topic – for example, that *heart attack* is formally known as *Myocardial Infarction*, caused by a disruption of the flow of blood to the heart, and is a generalization of more specific medical conditions such as *Acute myocardial infarction*. This can be useful for both domain experts, who might be looking for specific pieces of information, or for novices, who might be looking to understand complex pieces of information.

However, the structured nature of this information can leave it susceptible to being isolated from the vast amount of unstructured text available. This can include documents that discuss information that should be added to a knowledge base by the curators of the KB. More consequentially, consumers of unstructured text need to manually refer to information in the knowledge base. In the case of a single document, this might be straightforward. However, understanding what structured data is being discussed in a large corpus is significantly more challenging. For example, how often is *Myocardial Infarction* discussed in a set of medical documents?

Entity linking, also known as named entity normalization, is a task within natural language processing (NLP) that seeks to bridge this gap between unstructured text and structured data. For each mention of a concept or entity within an unstructured document, entity linking seeks to identify which, if any, knowledge base entry refers to the same entity as the mention. This allows for an end user to be provided with an automatic reference to information about an entity mentioned in the text (*e.g. 2010 Rose Bowl*). Further, entity linking can identify mentions of concepts or entities at a macro level, understanding trends within corpora. Finally, entity linking can provide useful signal to other NLP tasks, such as information retrieval (Dalton et al., 2014; P. et al., 2015; Tan et al., 2017; Cornolti et al., 2016; Blanco et al., 2015) and question answering (Khalid et al., 2008).

There has been a vast amount of work in entity linking, resulting in systems that achieve high levels of performance (Wu et al., 2020; De Cao et al., 2021) on a variety of datasets. However, this line of work tends to focus on datasets with similar characteristics. First, most entity linking systems use Wikipedia as a knowledge base, which restricts the entities studied and the type of information used to what is available in Wikipedia. Second, the unstructured text is often Wikipedia or Newswire, which tends to be written more formally than other forms of documents. Finally, both the knowledge bases and documents are usually English-language exclusively. While this serves as a useful common point for research, this leaves a large amount of linking tasks under-addressed.

Designing high-performance linking systems in low-annotation domains requires understanding where big data architectures can be best leveraged while still leveraging alternative domain-specific sources of data. Accomplishing this requires a multi-prong approach. First, in almost all cases, the amount of training data for linking in other domains is far less than in the standard setting (*e.g.* (Wu et al., 2020) uses 9 million entity linking annotations for training, compared to $1,964$ in a clinical linking dataset (Pradhan et al., 2013)). In some cases, related annotations can be used to provide additional training signal. In others, we must look to strategies that can be applied to settings with smaller amounts of annotations, such as leveraging unstructured text or training on structured data. Finally, understanding how linkers can be forced to learn generalized patterns that transfer to examples unseen in training is crucial in all linking settings but is critical in low-annotation settings.

This dissertation is structured as follows. In the background chapter, the task of entity linking is introduced in detail in Chapter 2.1, followed by background on standard entity linking models (Chapter 2.2 and 2.2.2). Applications beyond the standard entity linking settings are introduced in Chapter 2.3, and discussion of available datasets is detailed in Chapter 2.4.

Next, in Chapter 3, the task of cross-language entity linking is explored, with a specific focus on the zero-shot setting, where there is no in-language entity linking annotations available. This is an example of a setting where a linker needs to be trained on alternative annotations, and we must look to related annotations and other sources of data, such as popularity, to design a high-quality linker. Chapter 4 focuses on the related task of multi-language entity linking, where English-language entity linking annotations to English knowledge bases are used to train linkers deployed to sets of documents and knowledge bases in other languages. This builds upon the findings of the previous work, but also shows how unannotated text can also be used to improve linkers in low- or no-annotation settings. Both sections highlight the need to focus on more challenging entity matches. An approach to better handling challenging entity matches is discussed in Chapter 5. This linker augments an entity linker trained on massive amounts of data with information from the knowledge base, which helps correctly match challenging mentions.

While work that can be applied broadly within entity linking is useful, sometimes domain-specific methods are required. The importance of this is highlighted in Chapter 6, which first shows that entity linkers trained on Wikipedia and other common data sources do not transfer well to medical text. Therefore, a clinical linker, which can leverage the large number of synonyms available within a medical KB, is a more promising solution. Chapter 7 details a triage system that can support high linking performance in medical text for neural linking systems. Similar to the previous chapter, we find that the use of data sources unique to medicine improves performance over domain-general approaches. Finally, in Chapter 8, we explore how to improve finding synonyms within unstructured text as their inclusion is shown to improve performance in nearly all settings. In both cases, Finally, high-level conclusions are discussed, and future directions for this line of work are hypothesized.

Chapter 2

Background

2.1 Entity Linking

Structured sources of data, known as knowledge bases (KB), are created to organize concepts and objects into structures that can be interpreted by human beings. However, their utility is limited when not connected to the vast amount of unstructured text that exists in the world. Bridging the gap between unstructured and structured data allows for information to flow in each direction – consumers of text can refer to relevant structured data, and unstructured text can highlight important additions to the structured space. Within the field of natural language processing (NLP), the most common task in connecting the two sources is called entity linking.

The task of entity linking, also known as named entity disambiguation, automates the process of matching mentions of entities within the unstructured text to a relevant entry in a knowledge base. An entity is a real-world object that has formalized attributes, such as a name and description, present within a knowledge base. The various types of entities can be very broad and subject to constraints specified during the creation of a dataset or knowledge base. However, in the field of natural language processing, they fall commonly into select high-level types, such as Persons, Places, Geo-Political Entities, and Organizations. To understand the task, it is important to highlight the two most important resources for entity linking. First, how are knowledge bases commonly structured, and what data is available in them? Second, how are entities commonly discussed in free text?

Two of the party's ***European*** representatives voted against the motion backing current measures against the extremist group

name	*The European Union*
Alt. names	*E.U.*, *Europe* ...
desc.	*The European Union (EU) is a political and economic union of member states that are located primarily in Europe...*
types	Supranational unions, Trade Blocs, Political systems, ...
relations	*Capital*: *Brussels*, ...

Figure 2.1: An example entity linking annotation taken from the TAC 2015 Training set. The sentence includes a mention of *European*, which is a reference to the entity *The European Union*.

2.1.1 Knowledge Bases

A knowledge base collects entities into a single database and models relations between entities. Depending on the nature of the knowledge base, the breadth of entities present can be very narrow, such as a knowledge base about College Football, or very broad, such as DBPedia (Auer et al., 2007). Within a knowledge base, each entity often contains several different categories of information. In almost all cases, an entity within a KB will have a formalized, or preferred name. As highlighted in the example in Figure 2.1, in addition to the formalized name (*European Union*) there is often also alternative names, which can include acronyms (*E.U.*), or more informal shortened phrases (*Europe*). This synonym information is vital, as entities are frequently not referred to by their formalized names.

Descriptions or definitions, which are longer sections of free text related to the entity, are also often included. These longer sections of text can provide context explaining what the entity refers to, and helps to disambiguate between similar entities. These can range from a sentence to an entire article, depending on the knowledge base. Additionally, some knowledge bases define other attributes, such as *Foundation Date* for the entity *European Union*, which provide additional context.

Importantly, knowledge bases model how entities are related by providing structured information. At the entity level, this is usually in the form of type information. Types are a category that describes a coherent set of entities. The variety of types present within a knowledge base can also vary by domain, but they are often very granular. In many knowledge bases, entities are likely to have multiple types. For example, the entity *European Union* in DBPedia is labeled with types *Political System* and *Confederation*, while a broader set of types might only label it as a *Geopolitical entity*.

Finally, knowledge bases contain relational information between entities. A relation defines how two entities are connected. This often takes the form of a defined relationship between two entities, which can be unidirectional or bidirectional. For example, the relation of *Capital* between *European Union* and *Brussels* is unidirectional, while the relation of *Spouse* between *Barack Obama* and *Michelle Obama* is bidirectional. These relationships create a graph within the knowledge base, which illustrates which entities are similar to each other as defined by the knowledge graph creators.

2.1.2 Entities in Text

While knowledge bases attempt to reflect how entities are used in the real world, an author may use a variety of ways to refer to an entity. A reference to an entity within unstructured text is referred to as an entity mention (or, simply mention). In more formal text, it may be the case that they use the normalized entity name or an alternative name within the knowledge base. In a newswire article, an author is likely to refer to the entity *European Union* using its formalized name at least once, for example. However, it is often the case that the surface forms within text do not exactly match, and are not included in the knowledge base. This includes mentions that are partial matches (*the Union*), or complete rephrasing (*the Superstate*). While depending on the setting, generally pronouns are not considered named entities, as they would require resolving the mention to which the pronoun refers. For example, in the sentence *Its cornerstone is the Customs Union*,[1] the word *its* refers to *European Union*, but it is not named and therefore is outside the task scope.

Often, entity linking is a separate task from named entity recognition (NER, also known as mention detection), which locates spans of text within a document that are named entities. While a linker could consider whether all noun phrases in the document can link to the knowledge base, that would create a large number of entities to consider. Named entity recognition systems are trained to specifically identify named entities and often provide high-level type information. While often considered

[1] https://en.wikipedia.org/wiki/European_Union

a preceding step to an entity linking system, sometimes these two tasks are modeled jointly (Stern et al., 2012; Martins et al., 2019). In some cases, a mention identified by a NER system may not have a relevant entry in the knowledge base yet. These are often referred to as NIL mentions and may signify that information needs to be added to the knowledge base. The related task of NIL clustering (Li et al., 2011) seeks to cluster mentions that refer to the same entity, which serves as a useful basis for a creation of a new entity. Additionally, a closely related task is that of coreference resolution. This process attempts to identify mentions that refer to the same concept or entity within a single document. This can be useful in identifying which mentions should be linked to the same entity. Cross-document coreference extends this idea to identifying mentions across a set of documents. In both cases, however, the mentions are not resolved to the knowledge base as in entity linking.

2.1.3 Task setup

Entity linking is often modeled as a two-step process. As some knowledge bases can contain millions of entities, it can be expensive to apply accurate but computationally slower algorithms to all entities. Therefore, many linkers first focus on candidate selection (also known as candidate generation or triage step). This produces a manageable set of candidate entities to consider using a more complex reranking process. The candidate selection step is focused on speed and recall, while the reranking process tends to focus on accuracy, even if the resulting algorithm is slower.

The challenge of entity linking, therefore, is to use the available sources of information – the mention and surrounding sentences from the unstructured text, and the information present within the knowledge base, to accurately match a mention to an entity in the knowledge base. While this usually begins with some notion of similarity between the mention text and the entity name, resolving more challenging cases often requires leveraging a combination of the additional context in the unstructured text and the structured data available in the KB.

If done accurately, entity linking has the ability to unlock potential in both unstructured and structured data. For example, identifying unstructured text related to an entity might be a useful expansion within the knowledge base (Niu et al., 2012; Wang et al., 2012; Ré et al., 2014; Nguyen et al., 2017). Identifying links within unstructured text might be useful for an end user by simply providing visual references to KB information. Entity linking can also provide signal for other natural language processing tasks, such as information retrieval (Dalton et al., 2014; P. et al., 2015; Tan et al., 2017; Cornolti et al., 2016; Blanco et al., 2015) and question answering (Khalid et al., 2008).

2.1.4 Common Challenges

If a mention in the text uses the exact normalized entity name, and there is no other similarly named entity in the knowledge base, the linking task is very simple. For example, if instead of the mention *European*, the author had written out *European*

Union, we easily identify the relevant knowledge base entry via an exact match to the formal entity name *European Union*. However, given the variety of ways of referring to an entity, and the relatedness of entities within the knowledge base, more advanced algorithms are often required to find the correct link.

Some of these arise from the text itself – authors do not always refer to entities formally, and these paraphrasings are not always listed in a knowledge base. The example mention *European* is an example of a partial match. While this can be lexically matched to the entity name *European Union*, this could also be matched to other entities in the knowledge base, such as *European Parliament* or *European Space Agency*. Abbreviations are also common, such as *E.U.*, which if not listed in the knowledge base are challenging to resolve. Finally, nicknames or other paraphrasing can lead to mentions that have no clear lexical relation to the entity name, such as the mention *Supermax* and the entity *ADX Florence*. In some knowledge bases, alternative names include these more informal phrasings. If not, an entity linker must use other sources to resolve these links correctly.

Alternatively, ambiguity can arise from the knowledge base, such as when there are closely related normalized names. Even if a mention is written formally, such as *Michael Jordan*, it is a challenge to decide if the correct link is to the former basketball player *Michael Jordan*, or the computer scientist *Michael Jordan*. In these cases, additional information from the knowledge base, such as types or definitions, is often relied upon to help disambiguate. Alternatively, entities that are closely related, such as *European Union* and *European Parliament*, are often even more challenging to disambiguate, since they are likely to have similar types and relations.

Compounding these issues are cases where there are pieces of information that are not included in the knowledge base. This can include missing information. For example, if there are no descriptions or types present for two similar entities, linkers must look to other sources to learn which entities are most relevant for a mention. However, knowledge bases do not model some information. For example, the popularity of each entity (defined as how likely an entity is to be mentioned in a text) is an important element in selecting which entity is most appropriate. There are some heuristics to determine this within a knowledge base, such as the number of entities that have a relation with a given entity. However, this can vary heavily per corpus, and other sources are likely needed to calculate an accurate entity probability.

In addition to the above problems, issues can arise from annotation decisions made by curators of entity linking datasets (Ling et al., 2015). The curators of each dataset often create a guide that they ask annotators to follow. The design decisions made by the curators have a large impact on how a linker should be designed. First, what granularity of noun phrases should be linked? Some mentions, like *football*, are concepts, not named entities, but may have an appropriate link in the knowledge base. Therefore, it may vary from dataset to dataset if anything that can be linked to a KB is linked, or if there is a separate set of criteria, such as persons or places. An additional problem is that of specificity – for a mention like *the Super Bowl*, depending on the context, it could link to a general page *Super Bowl* or that year's iteration of the event *2021 Super Bowl*. Finally, many mentions are compounds of potential entities, like *Baltimore and Ohio Railroad*. While many datasets are designed to link the full compound, there are also other entities, such as *Baltimore*, which could be linked. In all cases, a set of annotation guidelines could reasonably be designed with different decisions for all of these. However, this adds a level of challenge when designing a linker for multiple datasets.

Most modern entity linking systems rely on some form of human-annotated data to resolve links. This reliance can lead to challenges when attempting to apply an entity linking system trained on one domain or knowledge base to another setting. Training data, that might have enabled a linker to correctly resolve challenging links for specific data, may not be expansive enough to capture new patterns. While many systems seek to learn general patterns from training data, these do not always hold in other domains. For example, while Wikipedia contains a wide variety of links, a system trained on Wikipedia data is unlikely to perform well on medical texts. This can arise because of lexical variations on the entity name that might not appear in the knowledge base, or it can arise due to the different structures of the knowledge base.

2.1.5 Early Work

Some of the earliest work in linking tasks took place in the clinical domain. Specifically, Metamap (Aronson, 2001) is a system built to resolve mentions of clinical concepts to a medical ontology. Metamap focuses on leveraging resources present in the knowledge base, such as synonyms, and uses a dictionary mapping approach to find the most appropriate concept for a medically-related mention. Beyond clinical-specific work, the earliest entity linking systems focused on Wikipedia. Entity linking in the settings where each Wikipedia page serves as an entity in the knowledge base is also commonly called Wikification. Systems such as Cucerzan (2007) and Bunescu and Paşca (2006), leveraged the fact that Wikipedia pages contain links to other pages in Wikipedia, and can serve as entity linking annotations.

However, only linking to pages in Wikipedia is inherently restricting, as there may not be a relevant Wikipedia page for an important entity. Work in knowledge bases, such as with DBpedia (Auer et al., 2007), sought to both expand the number of entities in the knowledge base and provide additional information about the entities, such as relations between entities, more coherently. Datasets such as those produced by shared tasks like the Knowledge Base Population track at the Text Analytics Conference (McNamee and Dang, 2009; Ji et al., 2010; Li et al., 2011) are linked to knowledge bases that have expanded information compared to Wikipedia. This early work (Rao et al., 2013; Zheng et al., 2010; Zhang et al., 2010b; Cucerzan, 2011) focused on linking English language documents to English language knowledge bases. Systems focused on handling challenging text matches between mentions and entity titles and disambiguating between similar entities. The algorithms behind these approaches are expanded on in Chapter 2.2.

2.1.6 Metrics

There are several common metrics to measure how well an entity linker performs on a corpus. Which metric is most important depends on the application. Some linkers, such as those used in a triage system, should be focused on including the correct entity in a larger set. By contrast, a final reranker needs the correct entity to be the highest-scored one. All metrics over a corpus C focus on measuring the alignment between a gold standard entity label e_c and a predicted entity label e_p. Another important metric can include the $0th$-indexed rank of the gold standard entity in the predicted list, r_c. This essentially meaures how well a system reproduces the ground truth annotations. Most modern entity linkers predict an ordered list of entities, and thus it is useful to quantify the ranking performance of a linker.

If the highest-scoring entity is of primary interest, such as with a final linker, then accuracy or f1 makes sense as the basic metric. In these cases, a metric that summarizes the performance of all of the examples in the corpora can be a useful starting point. The most common of these is accuracy (or recall at 1, top-1 accuracy), which is defined as the number of correctly predicted cases divided by the total examples;

$$\textbf{accuracy} = \frac{\sum_{e \in C} \mathbb{1}\{e_c == e_p\}}{|C|} \tag{2.1}$$

This formulation gives a straightforward picture of overall corpus performance. However, other metrics, such as precision and recall, have the benefit of focusing on entities that have a link in the knowledge base and focusing less on NIL entities.

$$\textbf{precision} = \frac{\sum_{e \in C}(\mathbb{1}\{e_c == e_p \ \& \ e_c \neq NIL\})}{\sum_{e \in C}(\mathbb{1}\{e_p \neq NIL\})} \tag{2.2}$$

$$\textbf{recall} = \frac{\sum_{e \in C}(\mathbb{1}\{e_c == e_p \ \& \ e_c \neq NIL\})}{\sum_{e \in C}(\mathbb{1}\{e_c \neq NIL\})} \tag{2.3}$$

F_1, the harmonic mean between precision and recall, is a commonly used combination of these two metrics.

However, in a situation like triage, the larger concern might be that the correct entity appears in the list. In that case, accuracy at n (or recall at n, coverage at n), where $n > 1$, might be more appropriate;

$$\textbf{recall at n} = \frac{\sum_{e \in C} \mathbb{1}\{r_c < n\}}{|C|} \tag{2.4}$$

Similarly, it might be useful to understand the ranking ability of the linker beyond the highest scoring prediction. This can be useful if the initial rank produced by a triage is input to a second-stage linker. In this case, mean reciprocal rank (or MRR) might be an appropriate metric;

$$\textbf{mean reciprocal rank} = \frac{\sum_{e \in C} \frac{1}{r_c + 1}}{|C|} \tag{2.5}$$

MRR can be thought of as giving full credit to examples where the correct label is at rank 0, and partial credit in all other cases.

For any of the above metrics, the corpus size can be restricted to examples of interest. For example, one common metric is non-NIL accuracy, which calculates the accuracy of examples that can be linked to the knowledge base. Alternatively, macro-level metrics can be useful, especially in datasets where there are large class imbalances. For example, macro precision can indicate performance for a specific entity e_t

$$\textbf{macro precision}(e_t) = \frac{\sum_{e \in C, e == e_t} \mathbb{1}\{e_c == e_p \ \& \ e_c \neq NIL\}}{\mathbb{1}\{e_c == e_p \ \& \ e_c \neq NIL\} + \mathbb{1}\{e_p == NIL \ \& \ e_c \neq NIL\}} \tag{2.6}$$

The resulting macro precision for each distinct entity e_t can then be averaged for a corpus-level metric;

$$\textbf{macro avg precision} = \frac{\sum_{e_t \in C} \textbf{macro precision}(e_t)}{|e_t \in C|} \tag{2.7}$$

2.2 Entity Linking Models

2.2.1 Non-Neural Approaches to Entity Linking

In the breadth of entity linking research prior to the advent of neural methods (Shen et al., 2014), various machine learning methods have been used to model entity linking. The earliest entity linking methods used heuristic approaches (Aronson, 2001), which consist of combinations of various measures of similarity. Some of the next generation of work focused on using binary classification approaches (Cucerzan, 2007; Zhang et al., 2010a; Chen and Ji, 2011; Pilz and Paaß, 2011). Such models predict if there is a link between a mention and each entity, and use heuristics to resolve cases where there are multiple positive predictions. More commonly used are learning-to-rank approaches (Bunescu and Paşca, 2006; Kulkarni et al., 2009; Zheng et al., 2010; Dredze et al., 2010a; Chen and Ji, 2011), which are trained to produce a ranking for a set of entities given a mention. This has the benefit of not requiring handling multiple positive predictions and allowing the use of negative examples in training. Most commonly, these approaches use Support Vector Machines (SVM, Graepel, Obermayer, et al. (2000)) to learn a ranking for mentions, although other learning-to-rank frameworks have been used.

In addition, other authors have explored probabilistic models (Han and Sun, 2011), integer linear programming (Hajishirzi et al., 2013), structured conditional random fields (Durrett and Klein, 2014), graph-based approaches (Pan et al., 2015), and unsupervised methods (Cucerzan, 2007). Lately, as discussed in Chapter 2.2.2, neural architectures are becoming more and more prevalent.

However, across the various machine learning approaches, there are important commonalities. First, one of the foundational elements revolves around capturing the similarity between the mention text and entity name. While this is simple if there is a single exact match between the mention name and an entity title, handling multiple or partial matches, or paraphrases, is much more challenging. Second, many methods look to additional information to augment the name similarity. This includes modeling similarity between the textual context from the mention's document and knowledge base or using type, relational information from the knowledge base, or entity-specific features (*e.g.* popularity).

2.2.1.1 Name Matching

Many early entity linking systems focus heavily on matching the mention string to one or more entity names in the knowledge base. One of the earliest, Aronson (2001), focuses on using downstream NLP tools, such as part-of-speech tagging and stemming, to preprocess the mention and the entity name. Following this, they use the large number of name variations included in the relevant knowledge base to generate potential rephrasings of the mention. Using a simple scoring function that combines four measures of similarity between the mention and the entity, the best candidate entity is selected. This approach works well if potential abbreviations or paraphrases are included in the knowledge base, but struggles in cases where that is not the case.

Many other linkers (Zheng et al., 2010; Zhang et al., 2010b; Cucerzan, 2011) include similar methods to try to resolve a mention via abbreviation expansion or identifying if a longer form of the mention occurs elsewhere in the document. Charton et al. (2014) builds a system that can match mentions to knowledge base titles using a set of rules, such as generating spelling corrections and shortened names. Other work (Cucerzan, 2007) uses large data sources, such as Wikipedia, to map potential surface forms of entities to their normalized component. This can be done by identifying internal Wikipedia links, and learning how an entity might be phrased in comparison to its normalized form. With a large amount of data, many of these surface form variations can be identified even if they are not included in the structured data. In addition to building a lookup table, this data can be used to learn probabilistic models. For example, Han and Sun (2011) models the probability of an entity given a surface form from Wikipedia data.

If a linker uses a two-step approach, the candidate selection stage almost always focuses heavily on mention - entity text similarity (Zhang et al., 2011; Han et al., 2011; Ratinov et al., 2011; Gottipati and Jiang, 2011; Shen et al., 2012), often built from a large data source such as Wikipedia. While at its simplest form, this can take the form of a dictionary, where for each mention, there is a list of potential entities, many opt for a probabilistic formulation. For example, in Ratinov et al. (2011), the authors use two features as probabilities. The first is the fraction of the times that a given entity was linked from a mention form, and the second is the fraction of the times any mention links to a given entity. While some systems rely on exact matches for the mention, others attempt to handle cases where the dictionary mention is only a partial match. Overall, this approach has the benefit of being computationally efficient yet achieving a reasonable level of recall. However, this candidate selection approach struggles to handle cases where the entity name and mention are not lexically related and not present in the training data.

2.2.1.2 Leveraging the Document

Relying on the similarity between the mention string and the entity name does not give a linker access to the variety of other information that may be helpful in selecting a link. For example, many linkers also model the similarity between the mention's surrounding context and the entity description (Ratinov et al., 2011; Hoffart et al., 2011; He et al., 2013). While this does not produce results that are as precise as the mention text and entity name similarity, it enables a linker to see if an entity is topically related to the document. For example, the words *representative* and *voted* in Figure 2.1 likely signal that an appropriate entity would be related to politics instead of sports. In early work (Bunescu and Paşca, 2006), this took the form of the cosine similarity between vectors for both the document and the entity description. The vectors were composed of term frequency-inverse document frequency (TF-IDF Jones (1972)) values for the words present in each text.

Other work (Guo et al., 2013; Hoffart et al., 2011; Han et al., 2011; Han and Sun, 2012; Shen et al., 2012; Stoyanov et al., 2012; Kulkarni et al., 2009; Pennacchiotti and Pantel, 2009; Han and Sun, 2011) uses the other mentions in a document to provide additional context. This provides a more narrow context than simply using all of the surrounding text, and can include mentions of similar entities. This process can include simple resolution steps, such as in Cucerzan (2007), where the authors use coreference resolution to try and identify if a mention that uses a formalized entity name (such as *George W. Bush*) can be resolved to other mentions that might use non-standard forms (such as *Bush*). Further, they propose a model that maximizes the agreement between the categories of the candidate entities in the document in addition to the mention - entity likelihood for all potential mentions in the document found by some NER system. For example, Ratinov et al. (2011) uses a local linking step for all mentions in a document, which is then used to construct a broader mention context for a document. The authors then apply a global step, which seeks to enforce coherence between all mentions, by leveraging measures of entity relatedness.

Later work focuses on collaboratively resolving entities more selectively. The set of all mentions in a document may be too broad of a context to disambiguate. For example, Cassidy et al. (2012) collaboratively disambiguates mentions which are topically related to each other. Alternatively, Cheng and Roth (2013) restricts mentions to those that are related within the knowledge base. Within the context of a social network, Huang et al. (2014) uses the social graph to find related mentions. Pan et al. (2015) seeks to formalize this approach by using abstract meaning representation. The author's system builds a contextual graph around both the mention and the entity, with connections including relations within the knowledge base. Entity links are selected by maximizing the similarity between the two graphs.

Many linkers (Stern et al., 2012; Clark and Manning, 2015; Le and Titov, 2018; Luo et al., 2015) leverage the relatedness of other information extraction work to jointly model two or more tasks. Commonly, this is true of entity linking and named entity recognition, given the interconnectedness of the tasks. As entity linking relies on a correct identification of mentions by named entity recognition, ensuring mentions are identified accurately has a large impact on linking performance. Sil and Yates (2013) proposes to perform a first step separately for each task which produces a larger amount of candidate predictions for both. The second step reranks the two tasks, selecting the predictions for each that maximize the constraints of both problems. Durrett and Klein (2014) jointly models coreference resolution, named entity recognition, and entity linking. In addition to a feature set specific to each task, they generate features that model the interaction between the tasks.

2.2.1.3 Beyond the Document

Other methods include approaches revolving around how to better model the information available within the knowledge base. One important factor is the relative popularity of entities within a knowledge base. While the context of a mention may provide sufficient information to disambiguate, understanding that *George W. Bush*, the former U.S. president, is more likely to occur than *George Bush*, the NASCAR driver, regardless of the mention form. In the earliest entity linking work, Cucerzan (2007) notes that the popularity of an entity is related to both the number of internal references in the knowledge base and the length of the description (in this case, a Wikipedia page). In some work (Han and Sun, 2011; Pennacchiotti and Pantel, 2009), popularity is measured by how often a mention of the entity occurs in a large dataset, looking to a broader resource to model this information. Other work (Rao et al., 2013) uses information from search engines to model popularity. However, if new entities emerge, or the linker is deployed the linker to new datasets that have different popularity characteristics, the addition of popularity might worsen performance.

Structured information from the knowledge base can also enable linkers to disambiguate between entities that have similar lexical forms. For example, if a mention was labeled as a Person by a NER system, it would follow that an entity with a person type would be a correct link. How these are used varies per system – for example, Cucerzan (2007) leverage Wikipedia categories as type identifiers. Relations can also be a useful feature, such as in Cheng and Roth (2013) and Pan et al. (2015), which use type and relational information to model how mentions relate within a document. Durrett and Klein (2014) jointly models type prediction and entity liking (in addition to coreference resolution), and their approach includes features that model the interactions between NER type predictions and the types available in the knowledge base. However. some work (Ling et al., 2015) reports that including NER type information as a feature can worsen performance, especially when the types are too fine-grained to generalize.

Many entity linking systems leverage information from search engines to attempt to resolve entities. This can include systems that use them as the triage step (Han and Zhao, 2009; Dredze et al., 2010a; Lehmann et al., 2010). A robust search engine, such as Google, can more easily map between mention text and entity names that are more complex given the larger amount of data it has available. Gottipati and Jiang (2011) adapted information retrieval techniques such as query expansion, which seeks to expand the context of a mention via a search engine, to the entity linking setting.

In addition to modeling what is present within a knowledge base, it is also important to model what is not present within a knowledge base. While some approaches (Cucerzan, 2007) do not include the ability to predict NIL labels for entities, it is an important component of a system designed to be deployed in a real-world setting. Early work (Bunescu and Paşca, 2006) applies a threshold approach to labeling NILs – if for a given mention no entity is scored higher than some selected threshold score, it is linked as NIL. Rao et al. (2013) proposed building specific feature sets for classifying NIL entities.

2.2.2 Neural Models

Most of the work in the previous Chapter relies on token matching approaches to enable matching the mention to entities in the knowledge bases. This approach, while efficient and interpretable, has several challenges. First, synonyms for entity titles must be either present in the knowledge base or in the training data for more complex matches to be completed. For example, the mention *America* likely refers to the entity *United States*, but the two strings are not lexically similar. In previous work, *America* would need to be included as a synonym in the knowledge base. However, structured data is by no means comprehensive, and this leads to cases where incorrect links are made.

In some deep learning approaches (Ganea and Hofmann, 2017; Francis-Landau et al., 2016a; Kolitsas et al., 2018), vector space models such as Word2Vec (Mikolov et al., 2013a) are used to bridge this gap. These pre-trained embeddings are learned via an unsupervised approach to appropriately identify that *America* and *United States* should have high similarity. In addition, while these embeddings are learned from an unannotated corpus, they can be updated during model training to learn patterns present in the target data. This approach enables more complex relationships between texts to be identified, in either names or longer forms of text. The use of vector-space representations was paired with a move away from SVM-based learning to rank architectures and towards neural architectures that can leverage these representations.

In Ganea and Hofmann (2017), the authors use self-attention (Vaswani et al., 2017) to create contextual representations for the mention in context and the entities from Word2Vec embeddings. The authors use a collective disambiguation approach that is similar to the work in the previous Chapter. Relatedly, in Francis-Landau et al. (2016a), the authors use Word2Vec embeddings to encode the mention, its surrounding sentence, its document, and the entity title and description. The respective sequence of embeddings is then fed through a convolutional neural network, and the resulting representations for the mention and entity are compared by cosine similarity. In both cases, beyond simply using vector space embeddings of text, the architectures allow representations to be learned of larger sections of text, such as the context and the entity description, that enable better linking performance.

Other neural work has focused on issues in entity linking beyond text representation. Specifically, there has been work in jointly modeling entity linking and named entity recognition Kolitsas et al. (2018). As discussed in the previous chapter, these two tasks are naturally linked, and a neural approach allows for error in either component to be backpropagated through the entire network. Additionally, there has been neural-based work that has focused on better modeling type information in entity linking (Raiman and Raiman, 2018; Onoe and Durrett, 2020). While the integration of type information into entity linking has been explored previously, as discussed in the previous Chapter, using neural architectures to learn type embeddings allows for better measures of similarity between types. Some work, such as Orr et al. (2020) and Bhargav et al. (2022), specifically focuses on using type information to resolve rarer entities.

2.2.2.1 Contextualized Representations

However, another challenge is present in architectures that use Word2Vec embeddings – how can these embeddings model the context of the mention? While some mentions can be unambiguously mapped to a single entity even if they do not share a lexical form, this is not always the case. Consider the mention *Pandora* – this could be reasonably linked to *Pandora (Greek Mythology)*, *Pandora (Avatar)*, *Pandora (Jewelry)*, *Pandora (Online Radio)*, or even *Pandora (Ohio)*. With a standard Word2Vec approach, all of these aspects of the string *Pandora* would be included in a single embedding. Methods described in the Chapter previous to this would likely rely on the similarity between the context to resolve this issue. However, given that the surrounding sentence informs us of the meaning of the mention (*e.g. The town of Pandora sits on the Riley Creek in Northwest Ohio.*), it is beneficial to enable the embedding for a specific token to be influenced by its context.

Broader work within NLP proposed methods that address this issue. One of the first models proposed was Context2Vec (Melamud et al., 2016), which built upon the standard Word2Vec architecture by stacking two LSTMs (Long Short Term Memory, Hochreiter and Schmidhuber (1997)) on the token embeddings. These two LSTMs – one right to left, one left to right, encoded the context of the token, and the resulting embeddings at the token step are combined to create a single embedding. While this model showed some improvement, the resulting embeddings were still built on token embeddings. This is an additional challenge, as tokens not present in the original training data cannot be modeled.

The later ELMo model (Peters et al., 2018) proposed using character-level embeddings in addition to modeling the sequence, which enables the model to encode any token sequence even if not seen in the training data. ELMo representations are trained using a neural Bidirectional Language Model (BiLM), which models the forward language model probability of a token t_k given its history (t_1, ..., t_{k-1}). The model computes a context-independent token representation x_k^{LM}, using a convolutional neural network over the token's characters. The token representation is passed through $L = 2$ layers of an LSTM – the final layer is used to predict the next token using a softmax layer. The backward language model is the same, except the probability of token t_k is trained given its future context (t_{k+1}, ... t_N), and the final layer predicts the previous token. The parameters for the token representation and the softmax layer are tied between the forward and backward models, while all other LSTM parameters are independent.

After training the BiLM model, representations for each word in a sentence are built by passing an entire sentence through the language model and recording the resulting layers at each time step. This results in sentence-specific representations for each word, as opposed to the general representations in Word2vec or Context2vec. For each word, there are three representations – the token representation (referred to as layer 0), the intermediate representation from the first layer of the LSTM (layer 1), and the final representation resulting from the top layer of the LSTM (layer 2). Both representations from the LSTM, layers 1 and 2, are the result of concatenating the respective representations from the forward and backward LSTMs. Each of these representations provides different types of information about the word. The authors note that the second layer is most effective for word sense disambiguation, a semantically-orientated task, whereas the token representation is more lexically-oriented.

Further evolution in this space led to BERT (Bidirectional Encoder Representations from Transformers, Devlin et al. (2019)). BERT includes several important changes, compared to ELMo, that leads it to be one of the foundational architectures in NLP. Instead of using character embeddings, BERT uses the Byte Pair Encoding (BPE) to split tokens into subword elements, which produces a balance between character embeddings and whole-token embeddings. These, along with positional embeddings, are fed through several layers of transformers. The transformer architecture (Vaswani et al., 2017) enables the embedding for a given position in the input to be jointly conditioned on the surrounding context. This is opposed to ELMo and Context2Vec, which model the left and right sides separately. In addition, the layers of transformers allow for interactions between higher-order representations.

BERT is most commonly trained by masking a word in the input and forcing the model to predict the missing word only using the context. This forces the model to rely on the context of the masked word, instead of only the input at that position. This pretraining procedure enables it to be trained on massive amounts of unannotated data, allowing task-specific architectures to use this general textual modeling ability.

Advances in contextualized language models, such as BERT, have fueled substantial performance gains across various tasks in natural language processing. This includes question answering (Devlin et al., 2019; Beltagy et al., 2020), named entity recognition (Devlin et al., 2019), document classification (Beltagy et al., 2020), coreference resolution (Joshi et al., 2019), information retrieval (Akkalyoncu Yilmaz et al., 2019), and sentence similarity (Reimers and Gurevych, 2019), among other tasks. In light of this general trend, combined with the previous challenges unique to entity linking, it is easy to expect that contextualized representations would lead to performance improvements in Entity Linking.

In addition to the BERT encoding approach, there have been alternative models proposed. The most relevant to entity linking is BART (Lewis et al., 2020a), which leverages the Transformer architecture to process sequence-to-sequence tasks. BART, in part inspired by the GPT architecture (Radford et al., 2018), alters the BERT-style encoder. Instead of the output being conditioned on the left and right context, GPT is trained to produce output with only the left context. This autoregressive approach allows GPT to be used for generation tasks, as each generated token is conditioned on both the previously generated token, and the input to the left of the current token. BART iterates on this by first encoding the entire input – as with BERT – and then decoding the output string again conditioned on the input string, as with GPT. This allows for the output to be conditioned on the entire document. Applying this type of model to classification tasks, such as entity linking, instead of text generation tasks, is rarer. However, as detailed in Chapter 5, this architecture is frequently used in information extraction tasks.

2.2.2.2 Contextual Representations and Entity Linking

Much recent work in entity linking relies on contextualized embeddings to represent text from the mention, surrounding sentence, entity title, and entity description. This includes the linkers proposed in Sections 3, 4, 5, and 6, which will also discuss work following those proposed systems. However, there are some important commonalities in systems that use BERT in an entity linking setting.

Logeswaran et al. (2019) was one of the first systems to leverage BERT in entity linking, specifically in a setting where a linker needs to adapt to different domains. The authors propose a simple architecture – the mention m and entity description e are encoded in the same embedding;

$$\text{[CLS] m [SEP] e [SEP]} \tag{2.8}$$

The mention m consists of the mention embedded in the surrounding sentence, with a special marker around the tokens of the mention. The authors note that the cross-encoding ability of a transformer allows for better performance than encoding the two separately. The highest-level embedding at the [CLS] token is then multiplied by a learned weight, which produces a score for each entity given the mention. The authors explore whether pretraining on unannotated data using the BERT objective helps entity linking performance, and they find that it does help.

In addition to their findings surrounding the utility of BERT pretraining, the authors also propose the setting of zero-shot entity linking. In essence, a zero-shot entity linker is designed to be trained on one set of domains but deployed to a distinct set of domains. For example, the authors train their model on American Football data but evaluate it on Ice Hockey data. This tests the generalizability of the linker, as domain-specific patterns in American Football may not be useful in other settings. This follows the trend of zero-shot approaches in other NLP tasks (Pelicon et al., 2021; Wu et al., 2021; Duan et al., 2019; Srivastava et al., 2018; Ma et al., 2021; Levy et al., 2017).

The BLINK model (Wu et al., 2020) is a natural extension of Logeswaran et al. (2019). While Logeswaran et al. (2019) uses a token-based approach for triage, BLINK uses BERT for this step as well. First, a bi-encoder architecture is used to rank entities given a mention. Independent representations are created for the mention and the entity. The mention text, marked by the special symbols [$\mathrm{M_S}$] and [$\mathrm{M_E}$], surrounded by the original sentence, noted as $cntxt_l$ and $cntxt_r$, is encoded for the mention.

$$[\mathrm{CLS}]\ cntxt_l\ [\mathrm{M_S}]\ mention\ [\mathrm{M_E}]\ cntxt_r\ [\mathrm{SEP}]$$

For the entity, the name and description are encoded separated by the special symbol

$[ENT]$.

$$\text{[CLS] } \textit{title} \text{ [ENT] } \textit{description} \text{ [SEP]}$$

The score of an entity given a mention is produced by the dot product of the two representations. The BERT models are fine-tuned by learning to maximize the correct mention and entity pair over negatively sampled entities. Given the list of n candidates, a separate reranker is trained that uses a similar approach to Logeswaran et al. (2019).

There are several benefits to this approach. The first triage step is very accurate – on the TAC KBP 2010 dataset, the triage step is only 1.6% worse than the cross-encoding step. The embeddings for the entities in the bi-encoder step can be precalculated, and when paired with an efficient dense search index such as FAISS (Karpukhin et al., 2020), the triage step can be performed with high computational efficiency. This finding was also discussed in Gillick et al. (2019). In addition to the architectural contributions, the authors train the model on entity linking annotations taken from Wikipedia, which provides a large training set. They find this combination results in state-of-the-art performance on a variety of datasets.

Linkers that use similar architecture, such as Vyas and Ballesteros (2021), encode additional information from the knowledge base beyond the entity name and description. Their setting focuses on the ability to use varying information from different knowledge base schemas but still focuses on text fields such as *date of birth* or *location*. However, it is important to note that in BLINK and Logeswaran et al. (2019), linkers achieve high levels of performance by only using the name and description information from the knowledge base. This results in linkers that can be applied to different knowledge bases, which is not the case when using knowledge base-specific type information. On the other hand, it does mean that it is not leveraging much of the structured data available with the KB.

Except Logeswaran et al. (2019), all of these linkers are trained on massive amounts of annotated data. In the case of BLINK, this consists of nearly 9 million training examples covering 5.9 million entities. As that dataset is generated from Wikipedia, there is no cost to annotate this data. However, this large amount of training data is not available for a variety of other domains, from clinical text to food science data. Therefore, how can these advances achieved in general entity linking be translated into advances in other domains?

2.3 Applications

Work in entity linking modeling, as discussed in Section 2.2, has generally focused on a narrow set of data. First, most entity linking focuses on English language documents and English language knowledge bases. Second, the majority of entity linking work focuses on entities that are present in very restrictive domains, such as Wikipedia or newswire text. While some knowledge bases, such as DBPedia, augment this information, this results in a focus on entities discussed in newswire or discussion forum posts. Given the vast amount of unstructured text available, this is inherently limiting.

Beyond the English-focused entity linking research, some work has looked at extending the task of entity linking to settings where multiple languages are present. The most common of these multilingual tasks looks at cross-language entity linking, where documents in multiple languages are linked to an English language knowledge base. Later work also considers zero-shot approaches to cross-language linking, which helps alleviate the imbalances in annotations present between languages. Enabling entity linking to be multilingual allows for unstructured data in a variety of languages to be linked to a knowledge base, thus massively expanding the amount of information available.

Beyond tackling the language barrier, there are a vast amount of other texts in fields that have curated structured knowledge bases that can benefit from linking approaches. As discussed in Chapter 2.3.2.1, linking tasks for medical texts is frequently studied, but other fields also have received attention. The most popular of these is Logeswaran et al. (2019), which proposed using Wikia as an entity linking dataset. These annotations, built similarly to those from Wikipedia, are separated into domain-specific sets, such as College Football and the Muppets. Other fields include Food Science (Popovski et al., 2019), Chemical (Dogan et al., 2021), and Creative Works Brasoveanu et al. (2020) data. Other work (Dai et al., 2018; Dredze et al., 2016; Fang and Chang, 2014) focus on linking in social media text. While social media tasks often link to commonly used knowledge bases such as Wikipedia, the length and nature of the text differs substantially from standard entity linking datasets.

Across all of these cases, the task of entity linking has distinct challenges that lead to general-domain entity linking approaches performing poorly, and suggest that domain-specific approaches are required. First is a question of data – substantial amounts of data has been annotated, manually or otherwise, for training standard entity linkers. This largely focuses on Wikipedia and TAC datasets, and these documents and knowledge bases have very divergent content as compared to medicine or food science. In many, such as medicine, a high level of expertise is required to build annotations. Second, the structure of the knowledge base is often unique to a domain. For example, medical knowledge bases focus more on building a hierarchical structure, whereas Wikipedia-based KBs contain more distributed relations. Beyond simply adapting the state-of-the-art entity linking techniques to a given domain, it is important to understand the underlying task and data involved in a specific setting.

2.3.1 Linking in Multiple Languages

... el jefe de la **Oficina de la Presidencia** *(m.01p1k, ORG)*, Aurelio Nuño y ...

name	*President of Mexico* (m.01p1k)
desc.	*The President of the United ...*
type	government_office

Figure 2.2: Cross-language entity linking: example Spanish mention *Oficina de la Presidencia*, which is a link to entity *President of Mexico*

There are a number of settings within the scope of entity linking for multiple languages. Within the field of natural language processing, the one that has historically received the most attention is cross-language (or cross-lingual) entity linking. This is the task of linking mentions of documents in a variety of different languages (*e.g.* Spanish and Chinese) to a knowledge base in a single language (almost always English). This contrasts with the setting of multi-language (or multi-lingual) entity linking, where the knowledge base contains information in a variety of languages. This reflects the fact that much early work in structured knowledge sources was English focused. A general trend towards increased research in multilingual settings can be seen in a variety of information extraction tasks (Johnson et al., 2019; Rahimi et al., 2019).

2.3.1.1 Cross-Language Entity Linking

There are several challenges present in this cross-language setting that do not arise in a monolingual setting. Consider the example in Figure 2.2, where a mention from a Spanish-language document, *Oficina de la Presidencia*, is linked to the entity

Presidency of Mexico in an English knowledge base. First, a linker must have the ability to recognize that text in two different languages are referring to the same entity. In some cases, such as with Person names written in the Latin alphabet, this is trivial. However, in the given example, a linker must be able to identify that *Presidencia* and *Presidency* are equivalent, in addition to handling the partial match between the mention (in English: *Office of the President*) and the entity title. Again, while this can be handled via lexical match in some closely related languages, this is not the case when linking between languages without a shared writing system, such as Chinese and English.

A second challenge in similar to one faced in monolingual entity linking models – adapting to entities unseen in the training data. The distribution of entities discussed in non-English texts often are different than those discussed in English-language documents. For example, English newsire text is unlikely to discuss the *Presidency of Mexico*, but is more likely to discuss superficially similar entities such as the *Presidency of the United States*. Popularity bias, as discussed in the previous Chapter, is a powerful tool for disambiguating entities, but a prior built off of English language text may not transfer to texts in other languages.

The first work to propose cross-language entity linking was McNamee et al. (2011), which built a set of documents across multiple languages using the knowledge base provided by the TAC English-language entity linking challenges. The authors proposed a now-common approach to cross-language entity linking: transliterating non-English mentions into English strings. Specifically, McNamee et al. (2011) uses a transliteration corpus to train a support vector machine ranker, which uses common entity linking features such as name and context matching, co-occurring entities, and an indicator for NIL (no matching candidate.) Later work, such as Pan et al. (2017) uses transliteration data for a set of 282 languages to generate all possible combinations of mentions, building off of a monolingual graph-based entity linking system (Pan et al., 2015).

The benefit of this approach is that transliteration does not require in-languages annotations. Additionally, for many languages with alphabetic writing systems, such as Russian, there are often easy mappings between the non-English source language and English. However, transliteration is far more challenging in languages without alphabetic writing system, such as Chinese or Japanese, or in settings where a mention may be phrased differently in the source language. A related approach is to use machine translation to translate a document into English, and then use an English entity linker. However, an machine translation (MT) system may not be available, and it further needs a specialized name module to properly translate entity names. Several systems from the TAC 2015 KBP Entity Discovery and Linking task (Ji et al., 2015) translate non-English documents into English, then use standard Entity Linking systems.

Later cross-language work began to leverage the multilingual nature of Wikipedia to build linking systems. For each page in Wikipedia, there are links to equivalent pages in other languages. This index, combined with internal links from within Wikipedia, allow for the construction of a cross-lingual dataset. This approach typically uses English Wikipedia as the KB, though it could use a KB in other languages. One of the first works to study this was Tsai and Roth (2016b), who use a two-step linking approach, first using an information retrieval-based triage system. Second, they use a candidate ranking step based on a linear ranking SVM model with several features, including contextual, document, and coreference.

Upadhyay et al. (2018) proposes a more advanced model in the same setting. They use FastText (Bojanowski et al., 2017; Smith et al., 2017) to align embeddings across languages, and a small dictionary to identify alignments. They pass these representations through a convolutional neural network to create a mention representation. They in turn use the other mention representations in the document to create a contextual representation, and also use a separate type vector. They train their network on hyperlinks from multiple languages in Wikipedia. Before the ranking step, they use a triage system similar to that of Tsai and Roth (2016b). They evaluate on several entity linking datasets, including TAC 2015 KBP Ji et al. (2015). Their results show that training on all languages, instead of monolingual or bilingual training, generally performs best. For zero-shot entity linking, they train on English language Wikipedia. They find that their performance is heavily dependent on a prior probability derived from the triage system – otherwise, there is a large drop in performance.

Most work in the cross-language entity linking space focuses on languages that are high resource, even if they have few entity linking annotations in-language. Other work, such as Rijhwani et al. (2019), investigate zero-shot entity linking on low-resource languages. They propose a model consisting of a similarity model using encoders separately trained on high-resource language mentions, related to the low-resource language, and English entities. They then use the high-resource language as a pivot language for low resource language mentions, allowing them to score mentions in an unseen language.

Relatively less research has explored other multi-language entity linking settings. Of those that do so, the most common approach is to use a system trained on English-language documents and knowledge bases to link non-English documents to a non-English knowledge base. For example, Raiman and Raiman (2018) seeks to transfer an English-trained system to French-language Wikipedia. They formulate a type system as a mixed integer problem, which they use to learn a type system from knowledge graph relations. Their training approach uses broad amounts of annotated data with type information (*e.g.* all of English Wikipedia). Since we do not train English Wikipedia models, and also do not use that magnitude of training data, we were not able to produce numbers using their system that are comparable to ours despite our best efforts to do so. Work using unsupervised graph methods, such as Wang et al. (2015b), are applied in non-English language pairs, such as Chinese. These models are not leveraging some text-based cross-language ability, but are rather relying on the inherently cross-language nature of graphs.

2.3.2 Linking in Different Fields

Some work in entity linking, such as Logeswaran et al. (2019) and Wu et al. (2020), explore designing entity linkers than can be applied to domains that were unseen in the training data. In some cases, such as Logeswaran et al. (2019), the authors constructed a dataset consisting of several different Wikia domains, such as College Football and Lord of the Rings. A linker was trained on four domains, but evaluated on other domains that remained unseen during training. Often however, work exploring domain adaption in entity linking is relegated to entities that can be linked to common knowledge bases, such as Wikipedia.

Yet in a variety of other fields, there are pairs of unstructured text and structured knowledge bases available. In many cases, both the unstructured text and the structured data have different characteristics that make deploying standard entity linking approaches challenging. First, the unstructured text in a field such as Medicine is vastly different than that present in Wikipedia. While Wikipedia does contain a vast amount of knowledge, a model such as Wu et al. (2020) trained only on Wikipedia may not identify that the medical terms *heart attack* and *myocardial infarction* are synonymous. Second, the structure of knowledge basses in different fields may be very different. For example, some may have more of a hierarchical structure, but contain fewer descriptions than are present in DBPedia.

Despite these challenges, the fundamental characteristics of the linking task remain unchanged. First, how can we build linkers that can identify the variety of lexical

The patient reports a history of **seizure disorder** ...

name	*Epilepsy (C0014544)*
desc.	*A disorder characterized by recurrent seizures*
parent concepts	Brain Diseases, ...
child concepts	Acute repetitive seizure, ...
synonyms	Seizure, E.P.,..

Figure 2.3: Medical concept linking: example mention *seizure disorder*, which is a link to concept *Epilepsy*

forms used to refer to entities in a knowledge base? Second, how can we leverage context from the document and the knowledge base to disambiguate between similar entities? And finally, are there opportunities to use data within the knowledge base to improve liking? In many fields, such as Medicine, solutions to these challenges have been proposed in a distinct tract from general entity linking. However, lessons can be taken from these disparate approaches to improve linking more broadly.

2.3.2.1 Medicine

Linking within the domain of Medical text has been frequently studied. Medical concept linking (aliases: "mention normalization", "medical concept parsing", "biomedical entity linking") produces structured topical content from clinical free text (Aronson and Lang, 2010). Healthcare providers often refer to medical concepts in clinical text notes that are absent from associated health record metadata despite their importance to understanding a patient's medical status. Following the example in Figure 2.3, the mention *seizure disorder* refers to the concept *epilepsy* contained within the Unified Medical Language System (UMLS) ontology (Bodenreider, 2004). However, this may be absent from metadata as it is not part of the current diagnosis. Concept mentions can use non-standard terms (e.g. *epilepsy*), thus concept linking requires non-lexical methods. Additionally, some terms (**cancer**) are ambiguous and could refer to multiple concepts (*breast cancer*, *colon cancer*, etc.)

In contrast to the dense KBs in entity linking, medical ontologies are sparser and contain only a unique identifier (CUI), title, and links to synonyms and related concepts. In more recent versions of the UMLS, there is an increasing amount of descriptive text, but it varies per area. Therefore, while the concept **epilepsy** has many synonyms in UMLS, it has no definition or other long description. Furthermore, UMLS concept names are more formal than clinical notes, making mention matching challenging. Additionally, Entity Linking systems are often able to leverage greater amounts of annotated data, which are not available in the clinical space. Text that does not have restrictive privacy protections can be annotated more easily through crowdsourcing, or other sources of non-gold standard data collected (e.g., Wikipedia cross-links). As the annotation of clinical notes is expensive due to the knowledge required of annotators and the protected status of clinical records, any effort in clinical concept linking must focus on leveraging a small number of annotations, and using larger amounts of related or unannotated data when possible.

However, there remains a large amount of overlap between entity linking and concept linking. First, the common paradigm of named entity recognition (or mention identification), triage (or candidate selection), and final reranking is widely used. In both cases, the challenge of NIL mentions (or CUI-less mention) persists. Both tasks also share the challenge of resolving ambiguities in both the document and knowledge base, even if the nature of the ambiguity is different in both tasks.

Previous work in Medical Concept Linking focused on building end-to-end systems, which combine candidate selection and final linking. This includes one of the earliest linking systems, Metamap (Aronson, 2001; Aronson and Lang, 2010), which consists of a pipeline to detect candidate spans and link concepts to the UMLS. The system consists of a pipeline that also performs pre-processing tasks, such as tokenization, negation detection, word sense disambiguation, and named entity recognition, to identify potential candidate mentions. The candidate generation approach used in the original version consists of generating a candidate list consisting of concepts that contain a variation of the mention phrase. These are then scored by an evaluation function that considers the type of variation – spelling variants are not penalized, while derivational variants are the most penalized. The CTakes medical natural language processing pipeline (Kipper-Schuler et al., 2008; Savova et al., 2010) consists of a similar set of natural language processing tools to process clinical notes, and includes concept linking. The original system used a dictionary matching algorithm to match mention spans to entries in the ontologies and their variant forms.

More recent work in medical concept linking systems uses a triage and final linker configuration - in Aggarwal and Barker (2015), they generate candidates from concepts containing variants of the tokens in the mention text, weighing them by inverse document frequency. The candidate list is then re-ranked by the similarity between the mention and candidate context, defined as a bag of words in the mention sentence and concept definition. Many medical concept linking systems do not include a distinct candidate generation phase. This includes a Sieve-Based method (D'Souza and Ng, 2015) which uses an ordered set of rules to identify a matching concept. This system does not include a separate candidate generation phrase but relies on a set of high-precision rules to match mentions to concepts from the entire set of candidates from the ontology. Rajani et al. (2017) combine the output of several systems, and they then train a system to learn the strengths and weaknesses (e.g. that a system is very precise, but has poor recall) of each by using auxiliary features, such as context-concept similarity. Finally, they ensemble the output of all systems by also considering which system is best suited for a specific mention.

Many systems have focused on the related task of Bio-medical literature concept linking (Doğan et al., 2014; Zheng et al., 2015; Tsai and Roth, 2016a), using a pairwise ranking approach, abstract meaning representation, and an indirectly supervised ranking approach, respectively. Biomedical literature is similar to clinical concept linking in that both are commonly linked to the UMLS. However, Biomedical literature has a very formalized text structure compared to clinical notes, which are often produced rapidly by medical experts. Additionally, biomedical literature does not have the same privacy concerns as clinical notes, so annotation can be more easily generated.

2.3.3 Task-specific Transformers

Advances in contextualized language models (LMs), as discussed in Chapter 2.2.2.1, also have been broadly adopted in task- and field-specific forms. While these LMs were originally trained on standard NLP domain texts such as Wikipedia and CommonCrawl, the unsupervised nature of their training means that models tailored to specific domains can be produced. This is especially useful in the settings discussed in this Chapter, in which annotations are challenging to produce, but unannotated data is more widely available. Contextualized language models that have been pretrained on specific domains also have the potential to alleviate some of the text-matching problems that are unique to their setting. For example, the original BERT model likely would not embed 翰·霍普金斯大 and *Johns Hopkins University* in the same space, even though they are direct translations. However, multi-language contextualized models, such as mBERT, can do so. Similarly, a linker can more easily identify that *stroke* and *myocardial infarction* are synonymous if using representations from a BERT model trained on clinical data, where those phrases likely have been seen in similar contexts

In the multilingual setting, contextualized language models have fueled advances in a variety of tasks (Wu and Dredze, 2019; Pires et al., 2019; Gonen et al., 2020; Chi et al., 2020; Choudhary and O'riordan, 2021; Roy et al., 2020). The earliest proposed model, mBERT (Devlin et al., 2019), uses the same architecture and training procedure as the monolingual version discussed in Chapter 2.2.2.1. However, instead of only training on English-language data, the authors train mBERT on the top 100 languages in Wikipedia, and those language-specific Wikipedias were used as the training data. Note that the training method for mBERT remains unsupervised, meaning that the challenge of transforming a monolingual model into a multilingual model is reduced. Further, mBERT can learn to learn cross-language relationships between text without any alignment, and without information about which specific languages it is embedding.

There have been several lines of research on how these models learn cross-language alignments without supervision (Wu and Dredze, 2019; Pires et al., 2019). The result means that NLP systems can work across languages with relatively simple adaptations. Other research has shown that BERT models trained on one or two languages often work better than mBERT (Xu et al., 2021), the performance differences are usually slight. Later multilingual language models, such as XLM-R (Conneau et al., 2020), build off of this foundation. The authors train their model on the CommonCrawl corpus, which provides substantially more text for the model to learn from.

Beyond multilingual models, there has also been a proliferation of domain-specific transformer models, from domains as distinct as Legal Text (Chalkidis et al., 2020) and Educational Text (Sung et al., 2019). This includes BioBERT (Lee et al., 2020) and ClinicalBERT (Alsentzer et al., 2019), which continue to train the original BERT model on Biomedical and Clinical text, respectively. Similar to the trends in monolingual and cross-lingual tasks, the use of these models has led to performance gains in a variety of medical and other domain-specific tasks (Yue and Zhou, 2020; Vassileva et al., 2021; Lewis et al., 2020b; He et al., 2020). Echoing the broader reasons for adoption, these domain-specific models can robustly represent text in their specific domains far better than either lexically-powered methods or vector space representations that are not contextualized such as Word2Vec.

2.4 Data

2.4.1 Entity Linking

2.4.1.1 Knowledge Bases

Wikipedia. Wikipedia is commonly used as a knowledge base. Usually, each page in Wikipedia serves as a knowledge base entry, with the title of the page serving as the standard entity name. The article body serves as the description – for longer articles, this results in a very large and perhaps imprecise description. Wikipedia categories are largely used for types. While sometimes these can be informative, such as the example in Figure 2.1, they can be overly specific or overly broad. For example, another category for *European Union* is *articles containing video clips*, which is essentially meaningless. Additionally, *Johns Hopkins University* has several categories, including *educational institutions established in 1876*, which is very granular. Finally, some pages do not have categories at all. Relational information between entities is less frequently used, although the number of internal links can serve as a related signal.

In addition to using Wikipedia as a single-language knowledge base, it can be used as a multi-language knowledge base. As each page in Wikipedia has a list of articles in other languages on the same topic, information can be collected on an entity in multiple languages. However, not all entities have entries in all languages.

The best practice for using Wikipedia as a knowledge base is to select a specific monthly Wikipedia dump, and only use the pages present in that dump. For example, BLINK (Wu et al., 2020) uses the 2019/08/01 Wikipedia dump. More recent work uses the KILT pre-processed version of Wikipedia.[2] This is simply a preprocessed version of that 2019/08/01 in JSON format, which is convenient. However, it does not add any additional information.

There are several benefits to using Wikipedia as a KB. The entities are frequently updated without assigning curators of KBs to do so. There are comparable pieces of information to a more structured knowledge base available. However, the types and granularity of entities in Wikipedia are restricted. Additionally, categories and articles are not exactly types and descriptions.

Wikidata. Wikidata[3] builds off of the foundation of Wikipedia but adds much more information. In addition to a link to Wikipedia information, this includes a shorter description, relationship information between other entities, and additional type information. Beyond that, each entity page can include a wide variety of structured data (*e.g.* such as the entity's Twitter handle). While all entities in Wikipedia have a Wikidata entry, the reverse is not true, as Wikidata collects entities from a broad variety of sources. For example, anyone with an ORCID ID (serving as a unique identifier for a researcher) has a Wikidata entry.[4] Therefore, there may be a lot more noise in Wikidata compared to other KBs.

TAC KBP Reference Knowledge Base. There are various knowledge bases

[2] https://github.com/facebookresearch/KILT
[3] https://www.wikidata.org/wiki/Wikidata:Main_Page
[4] https://www.wikidata.org/wiki/Q89561211

released with TAC entity linking annotations. For example, the 2014 TAC KBP Reference Knowledge Base[5] was used for the TAC 2015 KBP dataset.[6] This is similar to Wikidata, in that it adds to the information within Wikipedia. It does contain multilingual information, but it is similarly incomplete as Wikipedias. They do add type information (*e.g.* *Person*, *Organization*, *Geopolitical Entity*, *Unknown*) that matches the more high-level approach of some NER systems.

DBPedia. DBPedia is another knowledge base resource, again building upon Wikipedia.[7] It is similar to Wikidata and the TAC Reference knowledge base in that it augments Wikipedia with type information and relationships between entities. It is less likely to add datasets

2.4.1.2 Entity Linking Annotations

Wikipedia. In addition to being used as a knowledge base, Wikipedia can also be used as a source of entity linking annotations. Within any Wikipedia article, there are frequent references to other Wikipedia articles. The text and referral link of these mentions can serve as entity linking annotations. The major benefit of using Wikipedia as a source of annotations is that manual annotation is not required, and a large number of them can be created automatically. For example, the BLINK model is trained on 9 million annotations containing 5.9 million entities. This scale is nearly

[5] https://catalog.ldc.upenn.edu/LDC2014T16

[6] https://catalog.ldc.upenn.edu/LDC2019T02

[7] https://www.dbpedia.org/

impossible to recreate with a manual annotation effort. Additionally, multilingual annotations can be easily created. If a link to an article in one language contains a link to another page that also has versions in other languages, a cross-language annotation can be created. Similar to the Wikipedia KB setting, usually a specific Wikipedia snapshot is used as the source of annotations. Two common sources of Wikipedia annotations are from BLINK[8] and (Pan et al., 2017).

However, there are some downsides to this approach. First, while the annotations do not require human annotations, the variations in the ways entities are referred to are much more restricted than in other settings. For example, 82.9% of examples in the GENRE Wikipedia-based test set have a Jaro-Winkler score of 0.8 or higher. This means that when transferring to datasets where there is more variation in the ways mentions are referred to, challenging links may not be identified. Second, we are restricted to entities in the Wikipedia KB, with the drawbacks noted in the previous approach. True NIL annotations, which are present in other datasets, don't exist in Wikipedia. They can be artificially generated, however, but the effect may not be the same.

2015 TAC KBP Entity Linking dataset (Ji et al., 2015). This dataset consists of newswire and discussion form posts in English, Spanish, and Mandarin Chinese linked to the TAC KBP reference knowledge base. Some work, such as Upadhyay et al. (2018), also use this dataset but only for evaluation, instead

[8] https://github.com/facebookresearch/BLINK

training on Wikipedia and treating mentions that are linked to TAC entities without Wikipedia links as NIL. The training set consists of 30,834 mentions (6,857 NIL) across 447 documents. The evaluation set consists of 32,459 mentions (8,756 NIL) across 502 documents. For each annotation, there is also a type (very high-level). For NIL mentions, there is also cluster information, so this can also be used for NIL clustering. This is a higher quality dataset than Wikipedia, but far more limited in amount.

Wikia. The Wikia entity linking dataset (Logeswaran et al., 2019) was constructed from the Wikia website, which consists of individual community-written encyclopedias on a particular subject or theme. This was constructed in the same manner as the Wikipedia dataset – mentions taken from the text of in-page hyperlinks, and each document serves as an entity. The authors collect 16 Wikias, each with a different topic. Each topic has its knowledge base, thus serving as a challenging adaptation for our Wikipedia-trained models. The authors exclude all NIL entities and provide candidate sets for each mention of size 64, retrieved via BM25.

The topics are partitioned by training, validation, and test set so that each appears in only one set. Each mention is categorized by the amount of token overlap between the mention text and the normalized entity title by the dataset creators. The categories include *high overlap* (downsampled to 5% of mentions) , where the mention text and normalized entity title are exact matches, *multiple categories* (28% of mentions) , where the normalized entity title consists of the mention text plus a disambiguation phrase (*e.g.* mention *Batman*, entity title *Batman (Lego)*), and *ambiguous substring*

(8% of mentions), where the mention is a substring of the title. The category *low overlap* (59% of mentions) includes all remaining mentions. Note that in the original paper, the authors labeled this as *low overlap* – however, that is misleading, as many examples in that category have a high degree of lexical similarity. For example, of the *other* examples that have a candidate identified in the validation set, 28.96% of mention span - entity title pairs have a Jaro-Winkler (Winkler, 1990) of over 0.794, which is fairly high.

Below are other datasets relevant to the entity linking task;

- Other Datasets that use Wikipedia as the KB
    - WikilinksNED Unseen-Mentions (Onoe and Durrett, 2020). Partitions the annotations such that the test set contains entities unseen in the training set.
    - The TAC KBP 2010 Dataset[9] consists of English newswire documents linked to the TAC reference KB.
    - GENRE[10] also has preprocessed versions of a variety of entity linking datasets connected to the Wikipedia knowledge base, including documents from newswire (*e.g.* MSNBC) and web corpora (*e.g.* CWeb).
- Social media

[9] https://catalog.ldc.upenn.edu/LDC2018T16

[10] https://github.com/facebookresearch/GENRE

  - Twitter at the Grammys (Dredze et al., 2016) is a dataset of Tweets during the 2013 Grammys Award Show linked to Wikipedia.
  - Yelp (Dai et al., 2018) is a dataset of Yelp reviews, where mentions of businesses are linked to the corresponding Yelp business page.
  - Reddit entity linking dataset (Botzer et al., 2021) consists of Reddit posts with links to Wikipedia.
  - Twitter (Liu et al., 2013) is a dataset of Tweets linked to Wikipedia.

- Other
  - Creative Works (Brasoveanu et al., 2020) is a collection of documents about TV Shows and Movies linked to Wikipedia.

2.4.2 Linking In Other Domains

The Unified Medical Language System (Bodenreider, 2004). The Unified Medical Language System (UMLS) is the most frequent knowledge base used for linking tasks in medical or related fields. Unlike other KBs discussed here, the UMLS is actually a collection of knowledge bases (called *controlled vocabulary*). The UMLS provides links between entries within each vocabulary that refer to the same concept. However, attributions such as names and descriptions, and relationships between concepts, are at the vocabulary level. Commonly used vocabularies include SNOMED-CT and RxNorm. The UMLS is updated twice a year, and therefore it is best practice for annotations to be linked to a specific UMLS version. In many ways, the UMLS has similar information to a knowledge base such as BaseKB (Ellis et al., 2015), such as a preferred concept name, description, and relationships between concepts. However, the UMLS has a far more expansive collection of synonyms available for each concept. Additionally, the relationships between concepts are most commonly hierarchical – *e.g. Hypertensive disease* is a child concept of *heart*, but a parent concept of *Accelerated and malignant hypertension*. In early versions of the UMLS, there were far fewer descriptions available for concepts, but more have been added in later versions. The types are very high level – *e.g. Findings*.

2.4.2.1 Clinical Concept Linking

ShARe/CLEF eHealth Evaluation Lab 2013 Task 1b (Pradhan et al., 2013). This dataset consists of concept span annotations built on a subset of MIMIC 2.5 clinical notes (Saeed et al., 2011). The publicly available training set consists of 200 clinical notes, which we split into a training set consisting of 100 notes (1964 included mentions), and development and testing sets consisting of 50 notes each (957 and 1076 included mentions, respectively).

The annotations guidelines state that the concept candidates should be limited to the SNOMED-CT portion of the Disorder Semantic Group in the Unified Medical Language System version 2011AA (Campbell et al., 1998), and lists the semantic types included in the Disorder Semantic Group. In our experiments in Sections 6 and 7, we found several annotations linked to concepts not included in that list, including the *Finding*, *Body Substance*, and *Mental Process* semantic types, and therefore we expanded our ontology to include those concepts. Finally, we include all preferred entries, with the default settings of UMLS 2011AA, in the SNOMED-CT Disorder Semantic group (accounting for 116,436 unique concepts), but also include the first non-preferred entries that do not have a preferred entry (accounting for 8,926 unique concepts). We exclude any concept mentions that are not annotated with a SNOMED-CT Disorder concept, including non-concept annotations. We restricted synonyms to only include preferred entries, so only 22,769 out of 125,362 concepts have at least one synonym included.

MCN corpus (Luo et al., 2019). This corpus consists of medical notes linked to SNOMED and RXNorm, two other ontologies within the UMLS. They target a broad coverage of medical mentions, including problems, tests, treatments, and disorders. Compared to earlier clinical concept linking corpora, the authors annotate many concepts at a more granular level.

Bennerd Corpus (Sohrab et al., 2020) This corpus consists of Coronavirus-related documents (Sohrab et al., 2020). This dataset only has a small manually annotated test set paired with larger machine-annotated mentions. The machine-annotated mentions are created via a non-neural linking system and are not high-quality.

2.4.2.2 Other Linking Tasks

NLM-Chem corpus (Dogan et al., 2021). This is a corpus of 150 scientific articles split into test, train, and development sections. Mentions of chemical and drug names are linked to the MeSH ontology within UMLS, a collection of medical and scientific ontologies (Bodenreider, 2004). Although the corpus is annotated to the UMLS, the unstructured text is scientific articles, not medical text. While most mentions link to a single concept (or none, *i.e.* *CUI-less* or NIL), several examples are annotated with multiple entities (6% of the development corpus). This is an extremely challenging corpus since many of the chemical naming schemes are very complex.

The following datasets focus on linking tasks in other domains;

- Biomedical entity linking
    - Concept annotation in the CRAFT corpus (Bada et al., 2012) consists of 67 full text biomedical journal articles linked to UMLS.
    - The NCBI disease corpus (Doğan et al., 2014) consists of PubMed articles linked to MeSH.
    - BioCreative V CDR task corpus (Li et al., 2016) consists of Pubmed articles annotated with chemical and disease links to UMLS.
- Other domains
    - Food Science (Popovski et al., 2019) is a corpus of recipes linked to a subset of the UMLS.

Chapter 3

Cross-Lingual Transfer in Zero-Shot Cross-Language Entity Linking

3.1 Introduction

Entity linking work has primarily focused on English documents and knowledge bases (Chapter 2.3.1), but subsequent work expanded the task to consider multiple languages (McNamee et al., 2011).[1] For example, the TAC KBP shared task (Ji et al., 2015) links mentions in Chinese and Spanish documents with an English KB. Successfully linking a mention across languages requires adapting several common entity linking components to the cross-language setting. Consider the example in Figure 2.2, which contains the Spanish mention *Oficina de la Presidencia*, a reference to the entity *President of Mexico* in an English KB. To link the mention to the relevant entity we must compare the mention text and its surrounding textual context in Spanish to the English entity name and entity description, as well as compare the mention and entity type. Previous work has focused on transliteration or translation approaches for name and context (McNamee et al., 2011; Pan et al., 2015), or leveraging large amounts of cross-language information (Tsai and Roth, 2016b) and multilingual embeddings (Upadhyay et al., 2018).

Since this early cross-lingual work emerged, there have been major advances in multilingual NLP (Wu and Dredze, 2019; Pires et al., 2019). Mainstream approaches to multilingual learning now use multilingual encoders, trained on raw text from multiple languages (Devlin et al., 2019). These models, such as multilingual BERT

[1] Elliot Schumacher, James Mayfield, and Mark Dredze. 2021. Cross-Lingual Transfer in Zero-Shot Cross-Language Entity Linking. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 583–595, Online. Association for Computational Linguistics.

or XMLR (Conneau et al., 2020), have achieved impressive results on a range of multilingual NLP tasks, including part of speech tagging (Tsai et al., 2019), parsing (Wang et al., 2019; Kondratyuk and Straka, 2019), and semantic similarity (Lo and Simard, 2019; Reimers and Gurevych, 2019). However, even with these advances, one of the challenges of crosslingual entity linking lies in the amount of training data available. Due to the large focus on English-language entity linking data, there are far more English-language entity linking annotations compared to other languages. Further, some languages may not have in-language entity linking annotations, even if they have other resources available. Therefore, we propose to explore whether the multilingual abilities of BERT can help bridge the performance gap in languages with fewer annotations. The importance of a multilingual text encoder is higher in a cross-lingual setting compared to a multilingual, as the knowledge base only contains text in a single language. Therefore, cross-language understanding is required.

To construct a linker that can leverage annoations in multiple languages, we use text representations with multilingual BERT (Devlin et al., 2019) for cross-language entity linking to handle the mention text, entity name, mention context, and entity description.[2] We use a neural ranking objective and a deep learning model to combine these representations, along with a one-hot embedding for the entity and mention type, to produce a cross-lingual linker. We use this ranking architecture to highlight the ability of mBERT to perform this task without a more complex architecture.

[2] Our code is available at https://github.com/elliotschu/crosslingual-el

Although previous work tends to use multilingual encoders for one language at a time, *e.g.* train a Spanish NER system with mBERT, we ask: can our model effectively link English-language entities to documents in other languages? We find that, somewhat surprisingly, our approach does exceedingly well; scores are comparable to previously reported best results that are trained on data not available to our model (they have access to non-English names). Next, we consider a multilingual setting, in which a single system is simultaneously trained to link mentions in multiple languages to an English KB. Previous work (Upadhyay et al., 2018) has shown that multilingual models can perform robustly on cross-language entity linking. Again, we find that, surprisingly, a model trained on multiple languages at once does about as well, or in some cases better, than the same model trained separately on each language.

These encouraging results lead us to explore the challenging task of zero-shot transfer, in which we train a model to link single-language documents (*e.g.* English) to an English KB, but apply it to unseen language (*e.g.* Chinese) documents. This zero-shot ability will enable us to leverage annotations in languages such as English, which have more resources available, and apply them to languages with none. While the resulting model certainly does worse on an unobserved language, the reduction in performance is remarkably small. This result leads us to ask: 1) Why do zero-shot entity linking models do so well? 2) What information is needed to allow zero-shot models to perform as well as multilingually-trained models? Using a series of ablation experiments we find that correctly comparing the mention text and entity name is the most important component of an entity linking model. Therefore, we propose an auxiliary pre-training objective to improve zero-shot performance. However, we find that this text-focused approach does not improve performance substantially. Rather, we find that much of the remaining loss comes not from the language transfer, but from mismatches of entities mentioned across the datasets. This suggests that future work on the remaining challenges in zero-shot entity linking should focus on topic adaptation, rather than on improvements in cross-language representations. The ongoing challenge of topic adaptation can be found in other work, such as in Chapter 6.

In summary, we use a simple ranker to explore effective cross-language entity linking with multiple languages. We demonstrate its effectiveness at zero-shot linking, evaluate a pre-training objective to improve zero-shot transfer, and lay out guidelines to inform future research on zero-shot linking.

3.2 Entity Linking Model

We propose a cross-language entity linker based on a pointwise neural ranker that scores a mention m and entity e pair, adapting from an architecture discussed in Dehghani et al. (2017). Unlike a classification architecture, a ranking architecture is able to score previously unseen entities. As is standard, we use a two-stage system: triage followed by ranking; this reduces the number of entities that must be ranked

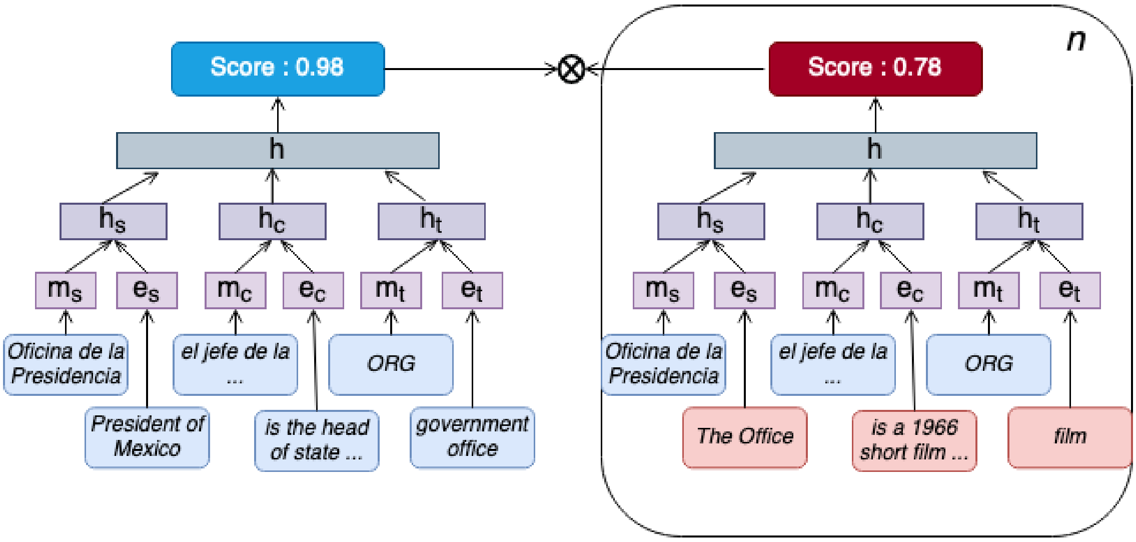

Figure 3.1: The architecture of our model, following the example in Figure 2.2 and a negatively-sampled entity *The Office*.

and results in better performance. Our system is shown in Figure 3.1. We select this architecture so as to focus on the ability of multilingual transformers to handle this task.

The ranker takes as input information about the mention and entity: 1) the mention string and entity name; 2) the context of the mention and entity description; and 3) the types of the mention and entity. We represent the mention string, entity name, mention context, and entity description using a pre-trained multilingual deep transformer encoder (Devlin et al., 2019), while the mention and entity types are represented as one-hot embeddings. We describe the multilingual representation, model architecture, and training procedure.

3.2.1 Multilingual Representations

We use multilingual BERT (mBERT) (Devlin et al., 2019),[3] which has been shown to create effective multilingual representations for downstream NLP tasks, as discussed in Chapter 2.3.1. Consider the Spanish example in Figure 2.2. First, we create a representation of the mention text m_s, *Oficina de la Presidencia*, by creating an mBERT representation of the entire sentence, selecting the lowest layer representations of each of the mention's sub-words,[4] and form a single representation using max pooling. We create a representation of the entity name e_s, *President of Mexico* in the same way, although there is no surrounding context as in a sentence.

For the mention context m_c we select the surrounding sentences up to BERT's 512 sub-word limit, positioning the mention in the middle, and pass the text to BERT, using the resulting top layer of the `[CLS]` token. We create a similar representation for the entity context e_c from the definition or other text in the KB, using the first 512 subword tokens from that description. For the mention type m_t and entity type e_t we create one-hot embeddings, omitting ones that do not occur more than 100 times in the training set.

[3] We found that XLM-R (Conneau et al., 2020) performed similarly and only report results on mBERT.

[4] We experimented with several BERT layers and found this to be the best performing on the **TAC** development set.

3.2.2 Architecture

We feed the representations of the name (m_s and e_s), context (m_c, e_c) and type (m_t, e_t) into a neural ranker. Each of these three pairs is passed into distinct multilayer perceptrons (MLPs), which each produce an embedding that captures the similarity between each type of information. For example, we input m_s and e_s into a text-specific hidden layer, which produces a combined representation r_s. The same is done for the context and type representations, producing representations r_c and r_t, respectively. These three representations are then fed into a final MLP, which produces a final score ($[-1, 1]$.) . We apply dropout at every layer, use ReLu as the intermediate activation function, and Tanh for the final layer. While additional features such as entity salience are likely useful for this task, we chose to restrict our model as much as possible to use only text features. This focuses on mBERT's multilingual ability and allows for easier adaptation to new KBs than with KB-specific features.

3.2.3 Model Training

We learn the parameters θ of our scoring function S using a pairwise approach; this allows us to train our model without annotated scores. Our ranker scores a mention m and positive entity e_+ pair, and separately scores the same mention paired with n sampled negative entities e_-. We apply the hinge loss between our correct entity and the highest scoring negative entity,

Parameter	**Values**
Context Layer(s)	[768], **[512]**, [256], [512,256]
Mention Layer(s)	[768], **[512]**, [256], [512,256]
Type Layer	[128], **[64]**, [32], [16]
Final Layer(s)	**[512,256]**, [256,128], [128,64], [1024,512], [512], [256]
Dropout probability	0.1, **0.2**, 0.5
Learning rate	1e-5, 5e-4, **1e-4**, 5e-3, 1e-3

Table 3.1: To select parameters for the ranker, we tried 10 random combinations of the above parameters and selected the configuration that performed best on the TAC development set. The selected parameter is in bold.

$$L(\theta) \;=\; \mathbf{max}\{0, \epsilon \,-\, (S(\{m, e_{+}\};\theta) \,-\, \mathbf{max}\{S(\{m, e_{0-}\};\theta)\ldots S(\{m, c_{n-}\};\theta)\}\}$$

We jointly train all components of the network, including the positive and negative portions of the network, with the ADAM optimizer. The major benefit of this pairwise approach is that it does not rely on annotated scores, but instead uses negative sampling to train the ranker. We tested random combinations of hidden layer sizes and dropout rates to find the best configuration. The specific parameters for our architecture are shown in Table 3.1. We report results after training for 500 epochs for TAC and 800 for Wiki. The full TAC multilingual model takes approximately 1 day to train on a single NVIDIA GeForce Titan RTX GPU, including candidate generation, representation caching, and prediction on the full evaluation dataset.

3.3 Datasets

We conduct our evaluation on two cross-language entity linking datasets. We predict NILs by applying a threshold; mentions, where all entities are below a given threshold, are marked as NIL. We evaluate all models using the evaluation script provided by Ji et al. (2015), which reports Precision, Recall, F_1, and Micro-averaged precision. The NIL threshold is selected based on the development **TAC** dataset. Unless noted, we use -0.8 for English and -1 otherwise.

We first consider the 2015 TAC KBP Entity Linking dataset (Ji et al., 2015), detailed in Chapter 2.4. We use their evaluation set, and provide a comparison to the numbers noted in Ji et al. (2015). The referenced systems had access to non-English language KB text which we exclude, and thus are a goal rather than a baseline. Later papers, such as Upadhyay et al. (2018), also use this dataset but only for evaluation, instead training on Wikipedia and treating mentions that are linked to TAC entities without Wikipedia links as NIL. Therefore, we cannot compare our evaluation to this work. We reserved a randomly selected 20% of these documents as our development set. The evaluation set consists of 32,459 mentions (8,756 NIL) across 502 documents.

We created a cross-language entity linking dataset from Wikipedia links (Pan et al., 2017) that includes Korean, Farsi, Arabic, and Russian. A preprocessed version of Wikipedia has links in non-English Wikipedia pages to other non-English pages annotated with that link and an English page link if a corresponding page was available. From these annotations, we created a dataset consisting of non-English mentions linked to English-language entities (Wikipedia page) using English Wikipedia as the KB. Some BaseKB entities used in the **TAC** dataset have Wikipedia links provided; we used those links as seed entities for retrieving mentions, retrieving mentions in proportion to their presence in the **TAC** dataset, and sampling a roughly equivalent number of non-TAC entities. We mark 20% of the remaining mentions as NIL. In total, we train and evaluate on 5,923 and 1,859 Arabic, 3,927 and 1,033 Farsi, 5,978 and 1,694 Korean, and 5,337 and 1,337 Russian mentions, respectively. We consider this to be silver-standard data because–unlike the **TAC** dataset–the annotations have not been reviewed by annotators. Since we do not have a separate development set for this dataset, we apply the hyperparameters selected on **TAC** development data to this dataset.

3.3.1 Triage

We assume gold-standard mention boundaries in our analysis. We use the triage system of Upadhyay et al. (2018), which is largely based on work in Tsai and Roth (2016b). This allows us to score a smaller set of entities for each mention as opposed to the entire KB. For a given mention m, a triage system will provide a set of k candidate entities $e_1 \dots e_k$. The system uses Wikipedia cross-links to generate a prior probability $\mathtt{P}_{\mathtt{prior}}(e_i|m)$ by estimating counts from those mentions. This prior is used to provide the top k English Wikipedia page titles for each mention ($k = 10$ for **TAC** and $k = 100$ for **Wiki**).

We use this system for both the **TAC** and **Wiki** datasets. However, while the triage system provides candidates in the same KB as the **Wiki** data, not all entities in the **TAC** KB have Wikipedia page titles. Therefore, the **TAC** triage step requires an intermediate step - using the Wikipedia titles generated by triage ($k = 10$), we query a Lucene database of BaseKB for relevant entities. For each title, we query BaseKB proportional to the prior provided by the triage system, meaning that we retrieve more BaseKB entities for titles that have a higher triage score, resulting in $l = 200$ entities. First, entities with Wikipedia titles are queried, followed by the entity name itself. If none are found, we query the mention string - this provides a small increase in triage recall. This necessary intermediate step results in a lower recall rate for the **TAC** dataset (85.1% for the evaluation set) than the **Wiki** dataset, which was 96.3% for the evaluation set.

3.4 Model Evaluation

We consider several different training and evaluation settings to explore the multilingual ability of transformers on this task. Recent studies suggest that multilingual models can achieve similar or even better performance on cross-language entity linking (Upadhyay et al., 2018). Another work (Mueller et al., 2020) has shown that this is not always the case. Therefore, we begin by asking: does our linker do better when trained on all languages (multilingual cross-language) or trained

Model		**micro**	**prec.**	**recall**	$\mathbf{F_1}$
en	NN	0.195	0.463	0.550	0.502
	Mono	0.586	**0.703**	0.619	**0.658**
	MultiDS	0.509	0.873	0.478	0.618
	Multi	**0.602**	0.691	**0.626**	0.655
	MultiOr	*0.654*	*0.773*	*0.641*	*0.703*
	Tri	—	*0.736*	*0.738*	*0.737*
zh	NN	0.207	0.889	0.449	0.597
	Mono	0.709	**0.867**	0.728	0.791
	MultiDS	**0.733**	**0.867**	**0.746**	**0.801**
	Multi	0.730	0.862	0.735	0.793
	MultiOr	*0.828*	*0.950*	*0.812*	*0.876*
	Tri	—	*0.854*	*0.809*	*0.831*
es	NN	0.214	0.508	0.552	0.529
	Mono	0.595	**0.921**	0.587	0.714
	MultiDS	0.604	0.918	0.590	0.718
	Multi	**0.652**	0.918	**0.625**	**0.744**
	MultiOr	*0.691*	*0.936*	*0.655*	*0.770*
	Tri	—	*0.804*	*0.804*	*0.804*

Table 3.2: Micro-avg. precision, precision, recall, and F_1 for **TAC** datasets.

Model		**micro**	**prec.**	**recall**	$\mathbf{F_1}$
ar	NN	0.171	0.414	0.602	0.491
	Mono	**0.660**	**0.683**	**0.816**	**0.743**
	Multi	0.637	0.661	0.778	0.715
fa	NN	0.330	0.694	0.734	0.714
	Mono	0.702	0.780	0.881	0.827
	Multi	**0.762**	**0.817**	**0.919**	**0.863**
ko	NN	0.269	0.816	0.597	0.690
	Mono	0.752	0.832	0.861	0.846
	Multi	**0.805**	**0.850**	**0.902**	**0.875**
ru	NN	0.358	0.841	0.529	0.649
	Mono	0.694	0.834	0.843	0.837
	Multi	**0.740**	**0.865**	**0.876**	**0.871**

Table 3.3: Micro-avg. precision, precision, recall, and F_1 for **Wiki** datasets.

separately on each individual language (monolingual cross-language)? Is the pattern of performance gain uniform, or specific to languages with less training data?

We train our model on each of the 7 individual languages in the two datasets (noted as **Mono**). Next, we train a single model for each dataset (3 languages in **TAC**, 4 in **Wiki**, each noted as **Multi**). **Mono** and **Multi** share the exact same architecture - there are no multilingual adjustments made, and the model contains no language-specific features. As **Multi** uses data available in all languages and thus has more training data than **Mono**, we include a model that is trained on a randomly-sampled subset of the multilingual training data that is set to match the training size of **Mono** (**MultiDS**). For **TAC Multi** models, we also report results using a candidate oracle instead of triage (**Multi+Or**), where the correct entity is always added to the candidate list. For all **Mono** and **Multi**-based models we report the average of three runs. The metric-specific standard deviations were all small, with all but one at or below 0.017. We note the best performing architecture from Ji et al. (2015) as **Tri**, again noting that those systems have access to the non-English text. We also evaluate a simple nearest neighbor model (noted as **NN**). This model scores each mention-entity pair using the cosine similarity between the mention name representation m_s and the entity representation e_s, and selects the highest-scoring pair.

Table 3.2 shows that for **TAC** there is a small difference between the **Mono** and **Multi** models. For **Wiki** in Table 3.3 the difference is often larger. **Multi** often does

		Evaluation Language							
		en	**zh**	**es**	**ar**	**fa**	**ko**	**ru**	
	Multi	0.66	0.79	0.74	0.72	0.86	0.88	0.87	Multi
Training	en	.00	−.03	−.02	+.03	−.08	−.08	−.05	ar
	zh	−.05	.00	−.03	−.14	−.04	−.16	−.10	fa
	es	−.06	−.06	−.03	−.20	−.13	−.03	−.09	ko
					−.20	−.08	−.13	−.03	ru

Table 3.4: ΔF_1 for each single-language trained model, compared to a multilingually-trained model, for each evaluation language. Each column is an evaluated language, and each row is a training setting.

better than **Mono**, suggesting that additional training data is helpful specifically for languages (*e.g.* Farsi) with smaller amounts of data. For languages with a substantial amount of in-language data, such as English, a monolingual model is better. Overall, these results are encouraging as they suggest that a single trained model for our system can be used for cross-language linking for multiple languages. This can reduce the complexity associated with developing, deploying, and maintaining multiple models in a multilingual environment. For some models, the **Multi** improvement may be due to additional data available, as shown in the difference in performance between **Multi** and **MultiDS** (*e.g.* Spanish F_1 **Multi** is +.026 over **MultiDS**). However, the small difference in performance shows that even by providing additional out-of-language training data, reasonable performance can be achieved even with reduced in-language training.

	en		**zh**		**es**	
	avg	**F$_1$**	**avg**	**F$_1$**	**avg**	**F$_1$**
name	0.59	0.70	0.45	0.71	0.42	0.73
+cont	+.12	+.05	+.22	+.05	+.14	+.05
+type	+.03	+.01	+.10	−.02	+.03	−.03
all	+.12	+.05	+.26	+.08	+.19	+.06

Table 3.5: English-only trained Δmicro-average and ΔF$_1$ when using a subset of linker features, compared to the name-only model for each language in the Development set.

BERT	**Lang**	**micro**	**prec.**	**recall**	**F$_1$**
en	en	−.07	+.17	−.13	−.03
en	es	−.01	.00	−.02	−.01
ar	ar	−.08	−.08	−.03	−.06
ar	fa	−.09	−.05	−.08	−.06

Table 3.6: Change in performance for monolingually-trained models using monolingually-trained BERT models, compared to monolingually-trained models using mBERT.

3.5 Zero-shot Language Transfer

Encouraged by the results of multilingual training, we explore performance in a zero-shot setting. How does a model trained on a single language perform when applied to an unseen language? We consider all pairs of languages, *i.e.* train on each language and evaluate all others in the same dataset.[5]

Table 3.4 shows the change in F$_1$ for monolingually-trained models compared to multilingual models. While zero-shot performance does worse than a model with access to within-language training data, the degradation is surprisingly small: often less than 0.1 F$_1$. For example, a model trained on all 3 **TAC** languages achieves an

[5] Work in Cross-language entity linking (Upadhyay et al., 2018; Tsai and Roth, 2016b) has done similar evaluations but focuses on using large external data sources (Wikipedia) to train their models.

F_1 of 0.79 on Chinese, but if only trained on English, it achieves an F_1 of 0.76. This pattern is consistent across both models trained on related languages (Arabic → Farsi, loss of 0.08 F_1), and on unrelated languages (Russian → Korean, loss of 0.13 F_1).

3.5.1 Analysis

Why does zero-shot language transfer do so well for cross-language entity linking? What challenges remain to eliminate the degradation in performance from zero-shot transfer?

We answer these questions by exploring the importance of each component of our cross-language ranking system: mention string, context, and type. We conduct ablation experiments investigating the performance loss from removing these information sources. We then evaluate each model in an English-trained zero-shot setting. First, we train a zero-shot model using only the mention text and entity name. We then compare the performance change that results from adding the context, the type, and both context and type (all features).

Table 3.5 shows that comparing the name and mention text alone accounts for most of the model's performance. This is a sensible result given that most of the task involves matching entity names. We find that context accounts for most of the remaining performance, with type information having a marginal effect. This highlights the importance of the multilingual encoder since both name and context rely on effective multilingual representations.

	en		**zh**		**es**	
Model	**avg**	**$\mathbf{F_1}$**	**avg**	**$\mathbf{F_1}$**	**avg**	**$\mathbf{F_1}$**
Baseline	0.64	0.75	0.51	0.69	0.53	0.75
w/ Name	.00	−.01	+.07	+.02	−.02	−.02
w/ Pop Train	.00	+.01	+.06	+.04	+.01	+.02
w/ Pop-All	+.04	+.03	+.12	+.06	+.10	+.06

Table 3.7: For each proposed Name matching or popularity re-ranking model, the change in performance (ΔF_1 and Δmicro-average) compared to the original **Rand** model.

Separately, how does using a multilingual transformer model, such as mBERT, affect the performance of our ranker? First, it is possible that using a monolingual linker with a BERT model trained only on the target language would improve performance since such a model does not need to represent several languages simultaneously. As shown in Table 3.6, model performance for these settings is largely worse for English-only and Arabic-only (Safaya et al., 2020) models when compared to using mBERT, with the exception that precision increases substantially for English. Second, perhaps a monolingual linker with a BERT model trained only on a related language – *e.g.* English BERT for Spanish, Arabic BERT for Farsi – would produce acceptable results. Again, as shown in Table 3.6, the performance is most often worse, illustrating that mBERT is an important aspect of the linker's performance.

	en		**zh**		**es**	
Model	**avg**	$\mathbf{F_1}$	**avg**	$\mathbf{F_1}$	**avg**	$\mathbf{F_1}$
Baseline	0.53	0.66	0.45	0.66	0.42	0.70
w/ Name	−.02	−.02	+.02	+.01	−.01	−.01
w/ Pop-Train	−.02	+.04	.00	+.07	−.01	+.06
w/ Pop-All	+.13	+.10	+.20	+.11	+.22	+.10

Table 3.8: For each proposed Name matching or popularity re-ranking model, the change in performance (ΔF_1 and Δmicro-average) compared to the original **Tail** models.

	en		**zh**		**es**	
	avg	$\mathbf{F_1}$	**avg**	$\mathbf{F_1}$	**avg**	$\mathbf{F_1}$
Multi	0.70	0.73	0.77	0.81	0.68	0.82
Rand	−.04	-.02	−.26	−.12	−.15	−.07
N-1	+.01	+.02	−.04	−.02	−.08	−.03
N-1U	−.24	-.14	−.49	−.22	−.38	−.19
Tail	−.16	-.08	−.31	−.15	−.26	−.12

Table 3.9: For each of the English-only training data subsets described in §3.6.2, ΔMicro-average and ΔF_1 compared to the full **Multi** model. Models that see even a single example of an entity (*e.g.* **N-1**) outperform models that see a portion (*e.g.* **Tail**) or none (*e.g.* **N-1U**).

3.6 Improving Zero-shot Transfer

3.6.1 Name Matching Objective

Given the importance of matching the mention string with the entity name, will improving this component enhance zero-shot transfer? While obtaining within-language entity linking data isn't possible in a zero-shot setting, we can use pairs of translated names, which are often more easily available (Irvine et al., 2010; Peng et al., 2015). Since Chinese performance suffers the most zero-shot performance reduction when compared to the multilingual setting, we use Chinese English name pair data (Huang, 2005) to support an auxiliary training objective. An example name pair: "巴尔的摩－俄亥俄铁路公司" and *Baltimore & Ohio Railroad.*

We augment model training as follows. For each update in a mini-batch, we first calculate the loss of the subset of the model that scores the mention string and entity name on a randomly selected pair $k = 25,000$ of the Chinese/English name pair corpus. We score the Chinese name z and the correctly matched English name e_+ pair, and separately score the same Chinese name paired with n negatively sampled English names e_-. We create representations for both z and e using the method described for names in §3.2.1 which are passed to the name-only hidden layer. We add a matching-specific hidden layer, which produces a score. We apply the hinge loss between positive and negative examples,

$$N(\theta) \quad = \quad \mathbf{max}\{0, \epsilon \;-\; (S(\{z, e_+\}; \theta) \;-\; \mathbf{max}\{S(\{z, e_{0-}\}; \theta) \ldots S(\{z, e_{n-}\}; \theta)\}\}$$

The name pair loss is then multiplied by a scalar $\lambda = 0.5$ and added to the loss described in §3.2.3. The resulting loss $L_{joint}(\theta) = (\lambda * N(\theta)) + L(\theta)$ is jointly minimized. After training, we discard the layer used to produce a score for name matches. This procedure still only uses source language entity linking training data, but makes use of auxiliary resources to improve the name matching component, the most important aspect of the model.

We analyze the resulting performance by considering modifications to our English-only training setting, which are designed to replicate scenarios where there is little training data available. To show the effect of a smaller training corpus, we select a random 50% of mentions, partitioned by document (**Rand**). To show the importance of training on frequently occurring entities, we select 50% of mentions that are linked to the least frequent entities in the English dataset (**Tail**).

Tables 3.7 (for the **Rand** setting) and 3.8 (for the **Tail** setting) shows the results on each of the three development **TAC** languages compared to the **Multi** model. For the **Rand** training set, we see a large improvement in Chinese micro-average and a small one in F$_1$, but otherwise see small reductions in performance. In the **Tail** training setting, a similar pattern occurs, with the exception that Chinese is less improved than in **Rand**. Overall, performance loss remains from zero-shot transfer which suggests that improvements need to be explored beyond just name matching.

3.6.2 Entities

Another possible source of zero-shot degradation is the lack of information on specific entities mentioned in the target language. For entity linking, knowledge of the distribution over the ontology can be very helpful in making linking decisions (as discussed in Chapter 2.2). While zero-shot models have access to general domain text, *i.e.* news, they often lack text discussing the same entities. For example, some entities that only occur in Chinese (231 unique entities in **Dev**), such as the frequently occurring entity *Hong Kong*, have a number of similar entities and thus are more challenging to disambiguate.

We measure this effect through several diagnostic experiments where we evaluate on the development set for all languages, but train on a reduced amount of English training data in the following ways: In addition to the **Rand** and **Tail** settings, we sample a single example mention for each entity (**N-1**), resulting in a much smaller training as compared to those datasets. We also take **N-1** and remove all evaluation set entities (**N-1U**), leaving all evaluation entities unseen at train time.

Table 3.9 reports results on these reduced training sets. All languages use a −1 NIL threshold. Compared to the multilingual baseline (**Multi**) trained on all languages, there is a decrease in performance in all settings. Several patterns emerge. First, the models trained on a subset of the English training data containing more example entities - *e.g.* **N-1** - have much higher performance than the models that do not. This is true even in non-English languages. Unobserved entities do poorly at test time, suggesting that observing entities in the training data is important.

However, a mention training example can improve the performance of a mention in another language if linked to the same entity, which suggests that this provides the model with data-specific entity information. Therefore, the remaining zero-shot performance degradation can be largely attributed not to a change in language, but to a change in topic, *i.e.* what entities are commonly linked to in the data. This may also explain why although the name matching component is so important in zero-shot transfer, our auxiliary training objective was unable to fully mitigate the problem. The model may be overfitting to observed entities, forcing the name component to memorize specific names of popular entities seen in the training data. This suggests we are faced with a topic adaptation rather than a language adaptation problem.

We validate this hypothesis by experimenting with information about entity popularity. Will including information about which entities are popular improve zero-shot transfer? We answer this question by re-ranking the entity linker's top ten predicted entities using popularity information and selecting the most popular entity from the list. Adding this feature into the model and re-training did not lead to a sizable performance gain. We define the popularity of an entity to be the number of times it occurred in the training data. We report results for two popularity measures–one using the popularity of the English subset of the data used for training, and one using all of the training data (including for Spanish and Chinese).

Tables 3.7 (**Rand**) and 3.8 (**Tail**) show that both strategies improve F_1, meaning that a missing component of zero-shot transfer is information about which entities are favored in a specific dataset. The gain from using popularity estimated from the training data only is smaller than using the popularity data drawn from all of **TAC**. With more accurate popularity information, we can better mitigate loss. This finding follows other work (see Chapter 2.2) which suggests that an accurate prior of entity probabilities enables accurate linking.

Several patterns emerge from most common corrections made with the Population reranking for **Tail**, included in Table 3.10. Many errors arise from selecting related entities that are closely related to the correct entity – for example, *United States Congress* instead of the *United States of America*. Additionally, people with similar names are often confused (e.g. *Edmund Hillary* instead of *Hillary Clinton*). Finally, many appear to be annotation decisions – often both the original prediction (e.g. *Islamic State*) and the corrected popular prediction (e.g. *Islamic State of Iraq and Syria*) appear reasonable choices. While most corrections were in Chinese (632), some occurred in both English (419) and Spanish (187). These errors – especially those in English – illustrate that much of the remaining error is in failing to adapt to unseen entities.

Original Prediction	**Popular Correction**	**Count**
United States Department of State	United States of America	146
united_states_congress	United States of America	121
Soviet Union	Russian	57
Central Intelligence Agency	United States of America	41
healthcare_of_cuba	Cuba	36
islamic_state	Islamic State of Iraq and Syria	33
edmund_hillary	First lady Hillary Rodham Clinton	32
United States Department of Defense	United States of America	32
Tamerlan Tsarnaev	Dzhokhar A. Tsarnaev	27
Carl Pistorius	Oscar Leonard Carl Pistorius	23
CUBA_Defending_Socialism_ ... documentary	Cuba	22
Barack Obama Sr.	Barack Hussein Obama II	18
Iraq War	Iraq	14
Dzhokhar Dudayev	Dzhokhar A. Tsarnaev	13
Sumter County / Cuba town	Cuba	13
United States Army	United States of America	13
military_of_the_united_states	United States of America	13
Republic of Somaliland	Somalian	13
ISIS	Islamic State of Iraq and Syria	13
Islamic_State_of_Iraq_and_Syria	Islamic State of Iraq and Syria	12
National Assembly of People's Power	Cuba	11
Sara Netanyahu	Benjamin Netanyahu	10

Table 3.10: All pairs of original prediction and popular prediction altered by the reranking procedure described in Chapter 3.6.2, for the **Tail** model

3.7 Conclusion

We demonstrate that a basic neural ranking architecture for cross-language entity linking can exploit the power of multilingual transformer representations to perform well on cross-language entity linking. Further, this enables a multilingual entity linker to achieve good performance, eliminating the need for language-specific models. Additionally, we find that this model does surprisingly well at zero-shot language transfer. We find that the zero-shot transfer loss can be partly mitigated by an auxiliary training objective to improve the name-matching components. However, we find that the remaining error is *not* due to language transfer, but to topic transfer. Future work that improves zero-shot transfer might focus on better ways to adapt to entity popularity in target datasets, instead of relying on further improvements in multilingual representations. Focusing on adapting to the topic and entities present in a given document is critical. This could be accomplished by adding a document-level representation or by leveraging other mentions in the document. English-focused work on rare entity performance (Orr et al., 2020; Jin et al., 2014) may provide additional direction.

Since the completion of this work, research has increasingly focused on entity linking in multilingual settings. Increasingly, however, research has focused on linking to knowledge bases that are inherently multilingual, instead of solely English. For example, Botha et al. (2020) released a dataset of entity linking annotations for 100 languages. The knowledge base contains descriptions in multiple langauges, and the authors select the cannonical description to be used for encoding by selecting the most frequently used link from the training data. The authors similarly use XLM-R and mBERT to build representations for the dataset. This approach has the benefit of using multilingual text resources if available but can only use a higher-resource language (*e.g.* English) if it is the only one available.

This approach has been extended in other research settings. For example, De Cao et al. (2022) uses multilingual autoregressive language models, such as BART, to predict entity links. The authors propose to use their model to independently score text from entities in multiple languages, and combine scores for an entity across all languages to achieve a final score for a mention-entity pair. More details of the model, as applied in a monolingual setting, are discussed in Chapter 5. While those two works focus on linking to Wikipedia, other work (Liu et al., 2021b; Galperin et al., 2022) applies similar techniques to other multi-languages settings, such as medical data. Overall, this approach has the benefit of leveraging text in multiple languages. However, it is also true that the vast majority of information in knowledge bases is in English, and thus methods that focus on this task remain useful.

Chapter 4

Improving Zero-Shot Multi-Lingual Entity Linking

4.1 Introduction

In Chapter 3, we discussed the challenges of entity linking in a cross-lingual setting.[1] In cross-lingual entity linking, all languages are linked to a single, typically English-language, knowledge base. While transferring a system to a new document language presents challenges, it does not consider issues that arise when transferring to a new KB language. KBs in different languages consider different topics, and matching text within the same language presents different challenges compared to building cross-language representations. People build KBs in many different languages, and we should explore how to link documents to these KBs. An additional challenge arises due to the smaller amounts of in-language annotations available for most non-English languages.

This project considers zero-shot cross-lingual adaptation of a trained entity linking system to a new monolingual setting: the same new language for both the query document and KB. We consider adaptation so as to utilize the extensive annotated data resources for English and other well-resourced languages, improving entity linking on languages that have little to no training data. Consider the example in Figure 4.1, which links the Spanish language mention *Senado* (English *Senate*) to the KB entry *Senado de la República* (English *Senate of the Republic of Mexico*). An entity linker uses the mention text and surrounding sentence paired with the KB entry

[1] Elliot Schumacher, James Mayfield, and Mark Dredze. 2022. Zero-shot Cross-Language Transfer of Monolingual Entity Linking Models. In *Multilingual Representation Learning Workshop at EMNLP 2022* . Association for Computational Linguistics.

(including information such as the name, and description) to score the likelihood of a match. Many approaches to entity linking learn these linkages by training on a set of hand-annotated links in the desired language. If there are no or few language-specific annotations, how can we train a model on an annotation-rich language to perform well on other languages?

We adopt our model from the one described in Chapter 3.2.2, while adapting the linker to a newer multilingual pretrained transformer model, XLM-Roberta (XLM-R) (Conneau et al., 2020). XLM-R is a multilingual model that yields robust representations of text in a wide variety of languages. However, we find that even with the cross-language ability of XLM-R, in-language annotation data is key to an accurate linker. We thus propose ways to improve the zero-shot cross-lingual transfer of a trained linker from one language to another.

We adapt a method from Chen and Cardie (2018) to add an adversarial objective to linker training which uses an intermediate layer in the linker to transform language-specific embeddings to language-agnostic embeddings via a language classification module. To train this language-agnostic layer, we force the language classifier alone to predict the incorrect language label for unannotated portions of the source (*e.g.* English) and target (*e.g.* Spanish) text. We jointly train the ranker and the language classifier using the correct source (*e.g.* English) language labels. which encourages the name and mention representation to be language-independent.

Second, we augment the entity linker with information from the target language

...lo acompañan el presidente del ***Senado*** ...

name	*Senado de la República*
desc.	*El Senado de los Estados Unidos de México...*

Figure 4.1: Example Spanish mention *Senado*, which is a link to the Spanish KB entity *Senado de la República* (the Senate of Mexico)

KB to capture the popularity of each entity, better handling entities that are common in the target language but rare in the source. We find that both model adjustments improve zero-shot performance on several language pairs and that the adversarial model specifically produces consistent improvement in recall. Overall, we demonstrate that entity linking models can be effectively adapted to a new language for both the query document and KB.

4.1.1 Architecture

We use a standard neural ranking architecture to focus on the mechanisms of transfer that have been applied successfully in cross-lingual entity linking (see Chapter 3.2.2). To score a mention m and candidate entity e, we leverage a pointwise neural ranker inspired by the architecture of Dehghani et al. (2017). This produces a score for each mention-entity pair, creating a ranking of entities specific to each mention. Additionally, this pointwise approach allows the scoring of previously unseen entities. We select a subset of entities to score using a triage system (§4.3.)

The only major architectural difference from the cross-language entity linker in the previous chapter is that this linker does not use type information. The use of

Parameter	**Values**
Context Layer(s)	[768], **[512]**, [256], [512,256]
Mention Layer(s)	[768], **[512]**, [256], [512,256]
Final Layer(s)	**[512,256]**, [256,128], [128,64], [1024,512], [512], [256]
Dropout probability	0.1, **0.2**, 0.5
Learning rate	1e-5, 5e-4, **1e-4**, 5e-3, 1e-3

Table 4.1: To select parameters for the ranker, we tried 10 random combinations of the above parameters and selected the configuration that performed best on the TAC development set. The selected parameter is in bold.

type information can be helpful but is also knowledge-base specific, which makes its inclusion challenging in a setting where there are multiple type systems. We use random combinations of parameters to select the best model configuration, which is shown in Table 4.1. The full TAC multilingual model takes approximately 1 day to train on a single NVIDIA GeForce Titan RTX GPU, including candidate generation, representation caching, and prediction on the full evaluation dataset – the Wiki model takes approximately 12 hours for the same set of steps.

4.1.2 Multilingual Representations

To create representations of the name and context for a mention-entity pair, we use XLM-Roberta (XLM-R, Conneau et al. (2020)), a multilingual transformer representation model. XLM-R outperforms other transformer models (such as mBERT (Devlin et al., 2019)) on multilingual tasks, and we confirmed this behavior in our initial experiments. Consider the Spanish example in Figure 4.1. We create a representation of the mention text m_s, *Senado*, by feeding the entire sentence through

XLM-R and form a single representation using max pooling on only the subwords of the mention. We create a representation of the entity name e_s, *Senado de la República* in the same way, except without any surrounding context.

To create m_c, we select the sentences surrounding the mention up to XLM-R's sub-word limit. We use max pooling over XLM-R to create a single representation. A similar method is used for the entity context e_c, but uses the definition or other text in the KB, using the first 512 subword tokens from that description.

4.2 Multilingual Transfer

The use of XLM-R makes our model inherently multilingual, allowing a single model to build representations in several languages. While this allows our models to do fairly well on previously unseen languages, we consider ways to further improve models during transfer: adaptation of the name matching model, and adaptation to the new knowledge base.

4.2.1 Language Adaptation

One source of error may arise from a linker learning language-specific patterns which do not generalize to other languages. Consider the example in Figure 4.1: would the model recognize that Spanish mention *Senado* is not linked to the *United States Senate*? While XLM-R provides a multilingual representation, the entity linking model

Algorithm 1 Pseudo-code of adversarial model training. In each epoch, a random set of text ($y = 5$) is used to adversarially train the language classifier. Then, the entity linker and the language classifier with the correct labels are jointly trained.

```
Require: Mentions $\mathbb{M}$, entity labels $\mathbb{E}$; English Text $\mathbb{A}$; L2 Text $\mathbb{B}$; Hyperparameter $\lambda > 0$, $y$, $z \in N$, $num\_epochs$
for $ep = 0$ to $num\_epochs$ do
    $l_{adv}$, $l = 0$
    for $i = 0$ to $y$ do                                          ▷ Adversarial Step
        $t_A$ = representation of $\mathbb{A}_i$
        $t_B$ = representation of $\mathbb{B}_i$
        $p_A = \mathcal{H}_{adv}(\mathcal{H}_{s0}(t_A))$
        $p_B = \mathcal{H}_{adv}(\mathcal{H}_{s0}(t_B))$                     ▷ Calculate Lang scores
        $l_{adv}$ += MSE($p_A$, **L2**) + MSE($p_B$, **ENG**) ▷ Calculate Loss using reversed labels
    end for
    Update $\mathcal{H}_{adv}$ using $l_{adv}$
    for $i = 0$ to $z$ do                                          ▷ Main Step
        $m$ = representation of $\mathbb{M}_i$
        $r_m = \mathcal{H}_{s0}(m)$
        $e$ = representation of $\mathbb{E}_i$
        $r_e = \mathcal{H}_{s0}(e)$
        $l$ = EL Loss (Eq. 1) with $r_m$ and $r_e$
        $p_M = \mathcal{H}_{adv}(r_m)$
        $p_E = \mathcal{H}_{adv}(r_e)$                                  ▷ Calculate Lang scores
        $l$ += $\lambda$ (MSE($p_M$, **ENG**) + MSE($p_E$, **ENG**))     ▷ Calculate Loss using correct labels
    end for
    Update all parameters except $\mathcal{H}_{adv}$ using $l$
end for
```

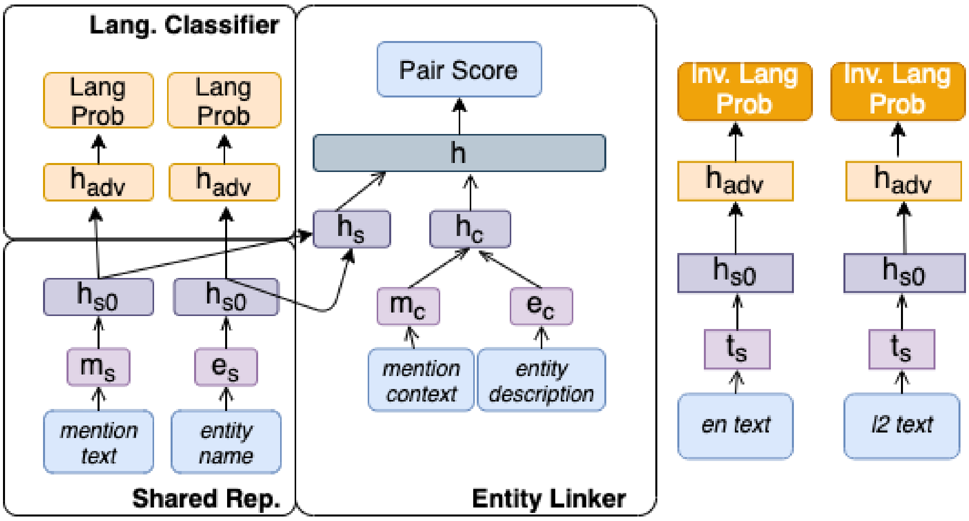

Figure 4.2: Our adversarial training approach consists of two steps – standard entity linking paired with training a language classifier (center) and adversarially training the language classifier (right). The hidden layer h_{s0} is shared.

has not been trained to learn this nuance in the Spanish knowledge base.

We add an adversarial objective to ensure that the model focuses on language-agnostic representations of the text, which will better transfer to other languages. The advantage of this approach is that it does not require annotated training data, but uses unannotated text to encourage desired model behavior. Chen and Cardie (2018) train a text classification system with an adversarial objective that forces the network to learn domain-invariant features. In addition to a standard text classifier that uses features from a shared and domain-specific feature extractor, they add a domain discriminator which uses the shared feature extractor as input. They run two training passes: 1) a training pass for the entire network that uses the correct classification and domain labels; 2) an adversarially trained domain discriminator and only the shared feature extractor, which uses the inverse of domain labels as the target. Prediction only uses the standard classification output. This objective improves performance when classifying text from previously unseen domains. We use this approach to learn language-invariant representations for our linking task, so they can be transferred to new languages using only source-language linking annotations.

Our proposed adversarial approach is described in Algorithm 1 and illustrated in Figure 4.2. For each epoch, we first adversarially train the language classifier. Using pairs of unannotated English $\mathbb{A}$ and L2 (second language) $\mathbb{B}$ text, we create representations in the same method as for m_s as described §3.2.1. Initially, we use randomly selected names from the ontology for $\mathbb{A}$ and $\mathbb{B}$ (see §4.4.3 for other approaches). Each of the two representations are fed into the shared invariant layer h_{s0}, the language classifier h_{adv}, and softmaxed to produce separate language likelihood scores for the English p_A and L2 p_B text. Importantly, we calculate the mean squared error (MSE) using the inverted language labels – for the English input, we calculate the error as if it was labelled as L2, and for the L2 input, we treat it as English. If we train with multiple L2 languages at the same time; all incorrect labels are applied with equal probability. We stop training the adversarial step after 50 epochs for one dataset (Wiki) based on development data performance.

We also run a standard entity linking training pass, in which we jointly train the linker and the language classifier using our set of training mentions $\mathbb{M}$ and corresponding entity labels $\mathbb{E}$. The entity linking loss is unchanged from §4.1.1, except that the m_s and e_s are first fed separately through the shared invariant layer h_{s0}. The loss for the language classifier is unchanged from the first step except that the correct labels are used. The effect of the language classifier loss is controlled by the parameter λ, which we set to be either 0.25 or 0.01 depending on the dataset. Models including this are referred to as **+A**. We experimented with adding the additional layers h_s0 and not applying the adversarial objective, and feeding both the language-invariant (*e.g.* m) and language-specific representations (*e.g.* r_m)) into the linker, but both performed worse in development experiments.

4.2.2 KB Adaptation

The second source of error comes from a change in the scope of the KB, not necessarily due to the change in language. Trained entity linkers tend to do well on popular, or previously seen entities. New entities, which are common when a linker changes to a new KB, do worse. Consider the example in Figure 4.1: a linker trained on English will favor the KB entry for the U.S. Senate, more common in English language documents, as opposed to the Mexican Senate, which is more common in Spanish documents. This is especially important since we consider models transferred from TAC to our Wiki data (§4.3), which cover different topics.

We adapt the model to a KB in a new language by supplying the entity linker with popularity measures drawn from the new KB. This information could normally be derived from some annotated entity linking data, but in the zero-shot cross-language transfer setting we instead leverage the cross-links among entities in the KB, a good indicator of entity popularity. For example, the entity *Senado de la República* might have a link to the lower legislature of Mexico, *Cámara de Diputados*, and the President of Senate, *Presidente de la Cámara de Senadores*. Others, such as *Senado de Arizona*, are likely to have fewer. We count unique cross-links between entities, divide by the median number of links, and feed the result into the final feed-forward neural network h (indicated as **+P**).

4.3 Datasets

We consider entity linking datasets in multiple languages from two sources. Both datasets were used in our cross-lingual work 3.3, but are preprocessed differently to emulate a setting with distinct knowledge bases. We treat each language as having a distinct KB, although entities may overlap in different languages. We predict NILs (mentions with no matching entity) as those where all candidate entities are below a given threshold (-1 unless otherwise noted). We evaluate using the script from Ji et al. (2015): Precision, Recall, F_1, and Micro-averaged precision. We use the triage system described in 3.3.1 for both datasets. Originally, the triage system was designed to produce links for non-English mentions to English titles. We tweak this approach by applying the same pipeline, but for in-language titles, which did not require any major algorithmic adaptations.

The 2015 TAC KBP Entity Discovery and Linking dataset (Ji et al., 2015) consists of newswire and discussion posts in English, Spanish, and Mandarin Chinese, and is detailed further in Chapter 2.4. The training set consists of mentions across 447 documents, and the evaluation set consists of mention annotations across 502 documents. This leaves us $14,793$ development mentions, of which $11,344$ are non-NIL.

We created a multi-language entity linking dataset from Wikipedia links (Pan et al., 2017) for Farsi and Russian. A preprocessed version of Wikipedia is annotated with links to in-language pages, which we treat as entities. Some BaseKB entities used in the TAC dataset have Wikipedia links provided; we used those links as seed entities for retrieving mentions, retrieving a sample mention of those, and adding the remaining links in the page. We mark 20% of the mentions as NIL. We consider this to be silver-standard data because–unlike TAC –the annotations are automatically derived. Thus the resulting distribution of mentions is different. Comparing the number of exact matches between the mention text and the entity name in Wikipedia (*e.g.*, in Farsi 54.5%) to TAC (*e.g.*, in Spanish 21.2%) underscores that TAC is a more illustrative dataset, thus we caution against treating Wikipedia as a replacement for a human-annotated entity linking dataset. This dataset is created in a similar fashion to the one in Chapter 3 but is filtered to find in-language links, instead of cross-language links.

4.4 Model Evaluation

We begin with a zero-shot evaluation: how well does a model trained on English (TAC) transfer to a new language without in-language training data? This baseline, which uses the same architecture as in Chapter 3.2.2, leverages only the crosslingual ability of XLM-R to apply English language annotations to the new languages. We evaluate the English-trained model on Spanish (es) and Chinese (zh) for TAC and Russian (ru) and Farsi (fa) for Wiki. We also train a separate model for each of these languages to establish an in-language performance baseline. We illustrate the difference in the performance of an English-only model as compared to an in-language trained one in Figure 4.3; the dashed line above each metric shows the increase in performance. To control for the effect of training set size we ensure that the training sets are of equivalent size for each language by randomly downsizing the larger training dataset (*e.g.* English) to match the smaller (*e.g.* Spanish). For comparison, we include a simple nearest neighbor baseline (noted as **nn**), which selects the highest scoring mention-entity pair using cosine similarity between the mention name m_s and the entity representation e_s.

We then apply our language (noted as **+A**) and KB (noted as **+AP**) adaptation strategies for each language, and measure the performance on both the target and English language. In all cases, reported metrics are averaged over three runs. We report results for each language in the form of micro-averaged precision (micro), recall (r), and F_1. See Table 4.4 for full results and additional metrics, and Tables 4.3 and

Train		**All**				**Non-NIL**			
/Test	**Model**	**micro**	**p**	**r**	**f1**	**micro**	**p**	**r**	**f1**
zh/zh	Baseline	0.795	0.890	0.830	0.859	0.801	0.884	0.884	0.884
en/zh	Baseline	0.202	0.905	0.697	0.788	0.077	0.899	0.721	0.800
en/zh	+A	0.439	0.897	0.732	0.806	0.367	0.892	0.764	0.823
en/zh	+A	0.381	0.911	0.756	0.827	0.296	0.907	0.794	0.847
en/zh	+PA	0.635	0.889	0.753	0.815	0.606	0.881	0.789	0.833
en/zh	+A (Desc)	0.266	0.908	0.718	0.802	0.156	0.903	0.747	0.818
en/zh	+PA (Desc)	0.645	0.885	0.774	0.826	0.618	0.877	0.815	0.845
en/zh	+P	0.544	0.894	0.685	0.776	0.494	0.888	0.707	0.787
es/es	Baseline	0.714	0.933	0.777	0.848	0.739	0.930	0.891	0.910
en/es	Baseline	0.488	0.942	0.643	0.764	0.444	0.944	0.716	0.815
en/es	+A	0.469	0.938	0.693	0.797	0.420	0.939	0.782	0.853
en/es	+A (multi)	0.548	0.952	0.753	0.841	0.523	0.956	0.860	0.906
en/es	+PA	0.654	0.931	0.695	0.796	0.660	0.931	0.784	0.851
en/es	+A (Desc)	0.496	0.943	0.737	0.828	0.455	0.949	0.839	0.891
en/es	+PA (Desc)	0.650	0.937	0.692	0.796	0.656	0.939	0.780	0.852
en/es	+P	0.664	0.928	0.698	0.797	0.674	0.930	0.788	0.853
zh/es	Baseline	0.378	0.942	0.661	0.777	0.301	0.943	0.739	0.829
zh/es	+A	0.514	0.939	0.785	0.855	0.479	0.945	0.902	0.923

Table 4.2: Single runs of Development TAC results for our reported models. Note that while we report results with the training sets equalized (zh and en training are set to be of equal size) for evaluation, the full development results do not have equalized training set sizes.

4.2 for development results.

4.4.1 Transfer Performance

Figure 4.3 and Table 4.4 show that zero-shot cross-language transfer from English gives worse performance compared to in-language models. For TAC languages (es and zh) there is a large decrease in micro-avg and F_1, and the same for Wiki languages (fa and ru), except that F_1 decreases more substantially than recall, illustrating a drop in precision. The overall drop in performance is not large - the largest drop in F_1 is only

Train/Test	**Model**	**micro**	**p**	**r**	**f1**	**Eval Epoch**
ru/ru	Baseline	0.650	0.823	0.888	0.854	800
en/ru	Baseline	0.484	0.762	0.855	0.806	550
en/ru	+A	0.451	0.712	0.893	0.792	50
en/ru	+A (multi)	0.419	0.652	0.865	0.743	200
en/ru	+P	0.473	0.685	0.860	0.762	50
fa/fa	Baseline	0.832	0.881	0.966	0.922	800
en/fa	Baseline	0.603	0.720	0.928	0.811	150
en/fa	+A	0.447	0.555	0.948	0.700	200
en/fa	+A (multi)	0.448	0.538	0.966	0.691	50

Table 4.3: Single runs of Development Wiki results for select reported models. Note that while we report results with the training sets equalized (ru and en training are set to be of equal size) for evaluation, the full development results do not have equalized training set sizes.

.1 less compared to the in-language baseline. This illustrates that the linker is able to transfer across language and knowledge bases effectively. Compared to the baseline nearest neighbor model, which one has the higher performance improvement depends on the language. For example, while Spanish F_1 is nearly the same, Chinese F_1 is slightly higher with the **nn**, but in Farsi, the English-trained model is an improvement for F_1. This finding is similar to that of Chapter 3.

We also evaluate other languages as sources of transfer. Table 4.4 shows results on training models on Chinese using the **+A** approach and testing on Spanish, demonstrating that our results are not specific to English. Note that the same pattern appears when transferring from a Chinese-trained model to a Spanish model. While the Spanish performance is understandably worse when transferring from Chinese instead of English, the reduction of F_1 performance is only $-.086$.

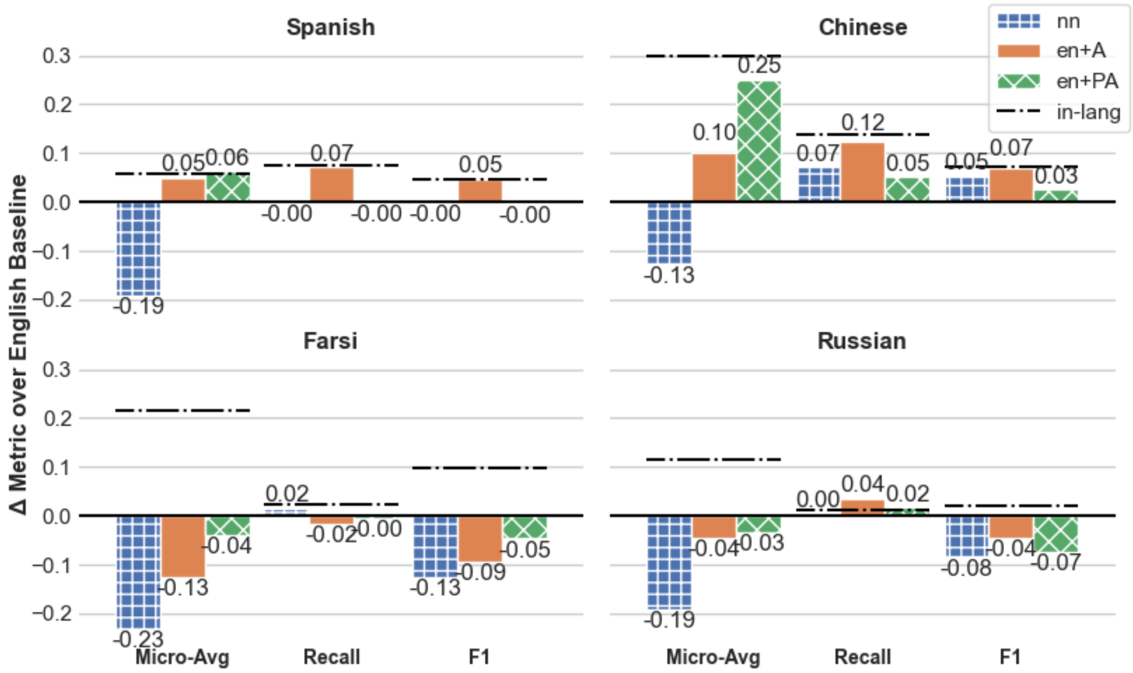

Figure 4.3: Compared to an English-only baseline (0.0 on y-axis), how do models with the adversarial objective (**+A**), the adversarial objective with popularity (**+PA**), and a nearest neighbor baseline (**nn**) perform?

4.4.2 Language and KB adaptation

We train the TAC and Wiki datasets with different configurations based on development results (see §4.4.3): TAC: $\lambda = 0.25$ and the adversarial step covers all of training; Wiki: $\lambda = 0.01$ and stop the adversarial step after 50 epochs. The difference in λ is large. This suggests that the TAC dataset benefits more from the cross-lingual training, perhaps due to the larger dataset size.

Applying the adversarial objective to English-trained models usually increases recall compared to the baseline English-trained models, and often even compared to the in-language trained models. For example, the English-trained, Chinese-tested model sees a large drop in recall which is almost completely eliminated when applying the adversarial objective. This increase in recall leads to nearly-equivalent F_1 performance in Spanish and Chinese in-language models and English-trained models with the adversarial objective. In short, adversarial training greatly improves the models' ability to locate the right KB entry, suggesting better name matching. This recall-focused improvement is useful for settings where high-recall is desired, such as in search. The exception to this is Farsi – this is likely because the high recall of 0.934 of the zero-shot model established a high starting point. Compared to the nearest neighbor baseline, the **+A** outperforms the baseline in all languages for F_1, nn F_1, micro-avg., and recall. The same pattern appears when transferring a Chinese model instead of English. The F_1 performance is only −.017 below the in-language trained model despite not sharing a writing system.

We also explored transferring a multilingual model: training on English with **+A** and testing on all target languages at once (see Table 4.4). In almost all cases, the multilingual adversarial approach performs worse than a single-language one, but only slightly; it may be preferable when targeting multiple languages. This is in contrast to the cross-language task 3.4, where multilingual training helped languages with less training data. This is likely due to the fact that in the cross-language task, the annotations are targeting the same knowledge base even if in a different language, unlike in this setting.

KB popularity (**+AP**) has the largest effect on micro-average precision by doing much better on rarer entities, specifically in the TAC dataset. While in Chinese the improvement in micro-average is larger in the **+AP** models than in **+A**, in all other cases the micro-average is close to the **+A** model.

We explored model behavior on different types of entities using the TAC evaluation dataset and provided mention types (see Table 4.5). For *Person* mentions, we see consistent performance between in-language, English, and English+**A** trained models. While this is not unexpected in Spanish (which has similar names to English), it is also true in Chinese, which uses a different orthography than English. The largest performance change occurred in *Geo-Political Entities*. For Chinese, F_1 drops 0.15 for an English trained model compared to an in-language trained model, but the deficit is erased in the English+**A** model. A similar pattern occurs in Spanish, suggesting that the adversarial model is able to improve the more challenging entity types.

4.4.3 Design of Adversarial Objective

How does the configuration of the **+A** model change its behavior? We vary three factors and measure results on TAC evaluation (full results shown in Table 4.6): 1) the size of the coefficient λ; 2) whether to train using the entity linking objective only for an additional 50 epochs instead of for all epochs (for lower λ and additional entity linking training, we found that both worked better on Wiki development data, while a higher λ and full training worked better for TAC); and 3) training **+A** using randomly selected names from English and the target language plausibly learns a better name

	Spanish (es) evaluation					**Chinese (zh) evaluation**				
Training	**micro**	**p**	**r**	**F_1**	**nn F_1**	**micro**	**p**	**r**	**F_1**	**nn F_1**
same	0.623	0.910	0.711	0.798	0.870	0.670	0.862	0.787	0.822	0.844
nn	0.375	0.924	0.633	0.751	0.809	0.244	0.910	0.719	0.803	0.826
en	0.565	0.925	0.635	0.753	0.810	0.371	0.893	0.647	0.750	0.757
en+A	0.615	0.923	0.706	0.800	0.876	0.472	0.877	0.770	0.820	0.839
en+P	0.632	0.919	0.616	0.738	0.790	0.462	0.869	0.636	0.734	0.734
en+PA	0.628	0.921	0.633	0.750	0.808	0.622	0.871	0.698	0.775	0.790
en+A (all)	0.562	0.917	0.694	0.790	0.862	0.466	0.882	0.722	0.794	0.813
zh	0.492	0.924	0.579	0.712	0.755	—	—	—	—	—
zh+A	0.523	0.901	0.690	0.781	0.852	—	—	—	—	—

	Farsi (fa) evaluation					**Russian (ru) evaluation**				
Training	**micro**	**p**	**r**	**F_1**	**nn F_1**	**micro**	**p**	**r**	**F_1**	**nn F_1**
same	0.838	0.902	0.958	0.929	0.908	0.526	0.729	0.827	0.775	0.721
nn	0.392	0.560	0.950	0.705	0.585	0.362	0.654	0.868	0.746	0.680
en	0.623	0.748	0.934	0.830	0.774	0.552	0.798	0.863	0.829	0.791
en+A	0.498	0.616	0.918	0.737	0.639	0.508	0.697	0.899	0.785	0.729
en+A (all)	0.525	0.631	0.955	0.759	0.668	0.516	0.758	0.852	0.802	0.755
en+P	0.627	0.700	0.958	0.809	0.741	0.565	0.700	0.889	0.783	0.728
en+PA	0.584	0.679	0.930	0.785	0.709	0.519	0.661	0.881	0.755	0.691

Table 4.4: Compared to an in-language trained model and a nearest-neighbor baseline (**nn**), how does a zero-shot model trained only on English transfer? For each setting, we report Micro-avg., precision, recall, F_1, and non-NIL F_1 on TAC and Wiki datasets.

			In-Language			**en**			**en+A**		
lang	**type**	**#**	**micro**	**r**	**f1**	**micro**	**r**	**f1**	**micro**	**r**	**f1**
zh	FAC	59	0.169	0.631	0.756	0.119	0.515	0.670	0.169	0.632	0.768
zh	GPE	3933	0.856	0.906	0.912	0.108	0.685	0.796	0.510	0.887	0.916
zh	LOC	461	0.729	0.947	0.886	0.488	0.810	0.840	0.547	0.933	0.892
zh	ORG	1441	0.160	0.726	0.774	0.299	0.629	0.722	0.127	0.799	0.821
zh	PER	3116	0.708	0.682	0.797	0.612	0.676	0.792	0.610	0.676	0.792
es	FAC	59	0.051	0.294	0.454	0.068	0.285	0.444	0.102	0.289	0.448
es	GPE	1570	0.664	0.891	0.927	0.338	0.674	0.791	0.532	0.830	0.888
es	LOC	174	0.144	0.824	0.874	0.672	0.717	0.810	0.787	0.863	0.892
es	ORG	799	0.451	0.681	0.782	0.444	0.678	0.779	0.444	0.691	0.788
es	PER	2022	0.715	0.624	0.755	0.693	0.602	0.741	0.723	0.624	0.755

Table 4.5: How do the results of in-language training compare to English-only trained models and models trained with the adversarial objective when looking at type-level performance?

model than it does language-invariant representations, so we instead train with the first 512 subwords of randomly selected descriptions.

Compared to a Chinese trained model, we considered versions with all non-baseline models trained on the joint entity linking and adversarial objective for 50 epochs, and the +EL models trained on EL data for an additional 50. Our reported setting for TAC, $\lambda = 0.25$ with name data, performs best on recall, F_1, and non-NIL F_1. However, when using the description data and $\lambda = 0.01$ with or without additional EL training, better micro-averaged precision is achieved. Generally, the models using name data perform slightly better than those using descriptions, but the overall difference is slight (*e.g.* +.009 F_1 for $\lambda = 0.25$ with name, $-.015$ F_1 with description), suggesting that the model is learning better multilingual representations. Finally, recall generally performs best with a higher λ and full adversarial training, and improves less with a lower λ and EL-only training.

4.4.4 Effect on English Performance

What effect does forcing an English-trained model to better orient to a target language have on English-language performance? Table 4.7 shows TAC English evaluation results in three settings: 1) a baseline linker with English training data matched to the size of the target language's training data; 2) the added **+A** objective; 3) the added **+AP** objective. These are the same models as in Table 4.7, except tested on English.

	Test	**micro**	**r**	$\mathbf{F_1}$	**nn** $\mathbf{F_1}$
	zh	0.674	0.789	0.824	0.846
	en base	−.341	−.123	−.060	−.071
+**A** name	.25	−.190	−.001	+..009	−.003
	.01	−.202	−.078	−.033	−.036
	.25+	−.205	−.123	−.062	−.073
	.01+	−.230	−.137	−.072	−.087
+**A** desc	.25	−.317	−.048	−.015	−.012
	.01	−.169	−.088	−.041	−.046
	.25+	−.287	−.188	−.108	−.133
	.01+	−.145	−.150	−.080	−.097

Table 4.6: How do adversarial settings affect performance? We consider the coefficient λ, type of text (names or descriptions), and entity-only training for 50 more epochs (*i.e.* we stop updating the language classifier, indicated by +).

Interestingly, the performance change is very small: a small increase for micro-average and a small decrease in F_1 and non-NIL F_1. The largest drop in performance is less than 0.05. This illustrates the capacity of the model: it can adapt to a new language while maintaining its performance on the source language.

4.4.5 Analysis

While our training methods are effective, they are inconsistent across our experiments. +**A** improves performance more on TAC data (Spanish and Chinese) than Wiki data (Farsi and Russian).

We postulate several explanations for this trend. First, the distribution of mentions is different between the two datasets. The lexical similarity between mentions and entity names – one measure of how easy the mentions are to link – is much higher in

Target	**micro**	$\mathbf{F_1}$	**nn** $\mathbf{F_1}$
en	0.484	0.672	0.797
zh+A	+.009	+.014	+.015
zh+P	+.030	−.025	−.031
en	0.472	0.678	0.802
es+A	+.004	−.014	−.017
es+P	+.011	−.036	−.043

Table 4.7: Compared to a baseline English TAC model (with training set size reduced to the noted language's training set size), we find that English performance is largely unchanged for both **+A** and **+P**.

Wiki. For Farsi development mentions, 54.5% were exact matches and also had an overall Jaro-Winkler (Winkler, 1990) lexical similarity of 94.1%. Compared to Spanish TAC (21.1% exact, 71.4% similarity) and Chinese (28% exact, 66.1% similarity), the Farsi data is relatively easy to link. While many entity linking studies rely on Wikipedia data due to its availability, it is not representative of other data types; we should build more human-annotated entity linking resources in non-English languages.

When comparing the drop in performance from an in-language trained model to an English trained model, recall drops in the TAC data, while precision drops in the Wiki data. The drop in precision may be because we use English TAC data to train the zero-shot Wiki models, and that recall is fairly easy given the high mention-entity similarity. Another factor is the possibility that Wikipedia text is less suited as adversarial training data, compared to that from TAC. Thus, while seeing an increase in recall in the Wiki models, this does not cancel out the reduction in precision.

4.5 Conclusion

We explored how to build a monolingually-trained entity linker that can be transferred to new languages that do not have annotated training data. With a neural ranker model using XLM-R, we see that while in-language-trained models perform better than English-trained models applied to second languages, the performance decrease is not large.

We have validated several ways to improve these zero-shot models and find that an adversarial language classifier improves recall and F_1 on many datasets. Furthermore, by adjusting the adversarial parameters, different performance objectives can be achieved, such as maximizing recall. We also present an analysis of our models, demonstrating which settings have the highest expectation of success. Overall, we find that training the model to learn language-invariant representations is effective in improving performance when transferring to both text and a KB in a new language.

As discussed in Chapter 3.7, recent work in multiple-language entity linking has focused on linking documents in multiple languages to language-agnostic knowledge bases. While this work only focuses on one-to-one pairings, the approaches taken in the work might be appropriate for settings where there is information available within the knowledge base that is specific to one language. Further, much of the recent multilingual work depends on the availability of large amounts of data, unannotated or otherwise. This assumption does not hold for a variety of domains. Beyond entity linking, adversarial approaches to multilingual tasks have been used in other settings, such as Question Answering (Rosenthal et al., 2021), Information Retrieval (Wang et al., 2021), and natural language inference (Dong et al., 2021).

Chapter 5

On the Surprising Effectiveness of Name Matching Alone in Autoregressive Entity Linking

5.1 Introduction

As detailed in Chapter 2.1, early work in entity linking in Wikipedia (Cucerzan, 2007; Bunescu and Paşca, 2006) followed by the formulation of the task at the TAC KBP shared task (McNamee and Dang, 2009; Ji et al., 2010; Li et al., 2011) has led to more than a decade of research into how to match textual mentions of entities to grounded entities in a knowledge base (KB). This large body of research has led to some clear findings (Dredze et al., 2010b; Durrett and Klein, 2014; Gupta et al., 2017; Lample et al., 2016; Francis-Landau et al., 2016b; Cao et al., 2018; Wang et al., 2015b; Witten and Milne, 2008; Piccinno and Ferragina, 2014). Entity linking is commonly modeled as a ranking task, in which a triaged set of KB entities is ranked by comparison to a textual entity mention. These ranking systems rely on different information sources. First, the entity mention is compared to the entity name in the KB (name matching), with allowances for aliases, acronyms, etc. Second, the context of the mention is compared to entity descriptions in the KB to select the correct entity among a set of similarly named candidates. Third, other relevant information from the KB (type information, links to related entities, popularity, etc.) can help disambiguate between candidates. This information is formulated as features (either engineered or learned) into the ranking system.

The recent emergence of autoregressive large language models as multi-task learners (Radford et al., 2019) has led to numerous new applications of these models. These models have been particularly effective in few-shot learning settings (Brown et al., 2020;

Chowdhery et al., 2022), but typically fall behind supervised training of traditional systems that can flexibly incorporate a range of features. Despite this trend, De Cao et al. (2021) presented GENRE, an autoregressive language model that uses supervised training to link textual mentions to entities in a KB. Given a sentence and a previously-identified mention span, the model generates an entity name selected from a set of (triaged) candidates, with the option to generate entities without any constraints (with worse performance). Surprisingly, aside from the entity name, GENRE uses no information from the KB, in contrast to other high-performing entity linking systems that rely on textual entity descriptions (Wu et al., 2020) or type information (Orr et al., 2020). We may expect an autoregressive LM to do well, but how can it beat the best available feature-based entity linking systems?

We explore the benefits and drawbacks of autoregressive entity linking. First, we ask – why GENRE performs so well? Our answer comes from an analysis of the behavior of GENRE across several different entity linking datasets. Specifically, we measure the generalization ability of the model by looking at performance on new datasets and knowledge bases. We find that GENRE relies heavily on memorization of name patterns, meaning that it struggles to generalize to new entities and KBs. KB information is often found to be useful in these cases, but its absence from GENRE means it struggles when name matching fails. The importance of this ability to match unseen entities can be seen in other work, including Chapter 3, where much of the performance reduction in the Zero-Shot setting arises from a lack of training

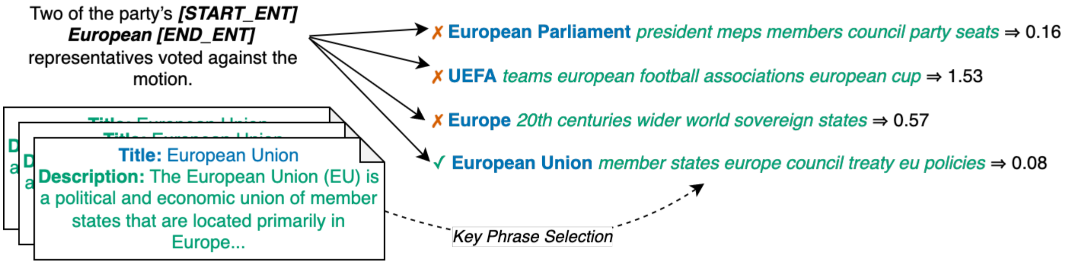
Two of the party's *[START_ENT]* *European [END_ENT]* representatives voted against the motion.
✗ **European Parliament** *president meps members council party seats* ⇒ 0.16
✗ **UEFA** *teams european football associations european cup* ⇒ 1.53
✗ **Europe** *20th centuries wider world sovereign states* ⇒ 0.57
✓ **European Union** *member states europe council treaty eu policies* ⇒ 0.08
Title: European Union
Description: The European Union (EU) is a political and economic union of member states that are located primarily in Europe...
Key Phrase Selection

Figure 5.1: An example mention taken from the TAC training set. In the original GENRE model, constrained decoding would be performed over only the **normalized entity names** (in blue, bolded) in the candidate list, given the mention and the sentence context. In our proposed GENRE-KP, we perform constrained decoding over the **normalized entity names** and *keywords* taken from entity descriptions in the knowledge base. These *keywords* help disambiguate between the correct entity (European Union) vs. similar but incorrect entities (European Parliament).

examples for specific entities. Therefore, our second question is: can GENRE make use of information from the KB when available? Specifically, we provide contextual information about an entity from the KB to GENRE and measure its resulting performance in various settings. We find that while it sometimes can make use of this information, it still struggles to learn generalizable patterns. Our analysis shows opportunities for incorporating KB information into an autoregressive entity linker, but also the challenges of doing so given current model architectures.

5.2 GENRE: An Autoregressive Entity Linker

GENRE (De Cao et al., 2021) is an autoregressive language model that links textual mentions to entities in Wikipedia through text generation. Autoregressive language models, such as BART (Lewis et al., 2020a), are trained to generate text, as opposed to other non-autoregressive based models (*e.g.* BERT (Devlin et al., 2019)), which are better suited for classification or scoring tasks. BART and similar models do very well at text generation tasks, including text summarization (Johner et al., 2021).

GENRE formulates entity linking as text generation as follows. Given the selected entity mention and its left context within the sentence, the model is trained to predict the next tokens as the normalized entity name. Consider the example in Figure 5.1. The model encodes the context *Two of the party's European*, and is trained to generate the correct normalized entity name *European Parliament* for this context. During training, the model is trained to minimize the smoothed cross-entropy loss between the generated entity name and the correct (normalized) entity name, where the normalized entity name matches the title of the associated node in the KB (Wikipedia page title). In this setup, negative sampling is not required. GENRE starts with a pretrained BART model and continues training on 9 million example entity mentions selected from Wikipedia, where the entity name is appended after each entity mention (see

Chapter 5.5).

Asking GENRE to freely generate a normalized name is both extremely challenging and unnecessary. In practice, a pre-filtering (triage) step can be used to automatically select the most likely entity candidates for a textual reference via a name matching algorithm.[1] De Cao et al. (2021) evaluated GENRE under several conditions. First, a free decoding step whereby the model could output any string; this did not do well. Second, constraining the model to generate a valid entity name from the KB. Third, constraining the model to generate an entity from the small set of triaged candidates. For the constrained generation case, the authors constructed a trie $\mathcal{T}$, where each node of the trie consists of a vocabulary entry, with a specialized token in the root. For each subword $t \in \mathcal{T}$, its children are allowed subword continuations.

In an evaluation on the several entity linking datasets, including Wikipedia and MSNBC (Derczynski et al., 2015), GENRE achieved state-of-the-art results compared to traditional entity linking systems. Yet the shocking thing about this result is what GENRE lacks. First, GENRE uses no information from the KB. Typical entity linking systems consider contextual overlap between the mention string and the KB entity description; GENRE does not. For example, when linking the textual mention *America*, a system would measure overlap with the KB description *The United States of America is a transcontinental country primarily located in North America* (United States) or *Americans are the citizens and nationals of the United States of America.*

[1] Previous work has noted that this task itself is a challenge, and relying on a candidate set that contains the correct entity is often unrealistic.

(American). Another popular feature is entity type, for example, *country* (United States) or *nationality* (American). Other features such as entity popularity, entity type, and related entities, are not available to GENRE. This information has long been used to disambiguate entities, and recent systems continue to show their ongoing effectiveness. Orr et al. (2020) use type information to help disambiguate entities that do not occur frequently. BLINK (Wu et al., 2020) build contexualized embeddings for each entity using entity descriptions. None of this information is available to GENRE.

Furthermore, due to the generation nature of BART, GENRE only uses the left context of the entity mention. In sentences such as that in Figure 5.1, a very limited left context is availble to provide any information. While GENRE can memorize associations between the limited left context and the entity name, it cannot generalize even this limited information to new settings.

Despite these limitations, GENRE represents a state-of-the-art entity linker.

5.3 GENRE and Generalization

How does GENRE achieve great entity linking results with such limited information? We explore this through the issue of generalization: how well does the model do on new unseen data?

Since the model does not have access to the KB, its predictions on new data are based entirely on what it can learn about entities from training data.

De Cao et al. (2021) suggested that GENRE predicts entities with contextualized name matching by leveraging large amounts of entity linking annotations during training. For example, while the original authors show that the model performs acceptably on rare entities (*e.g.* approximately 80% accuracy on Wikipedia entities seen once in the training data), the accuracy for entities unseen in the training data is only 50%. Bhargav et al. (2022) show that GENRE is very data-intensive to train; reducing training to 0.01% of the original size performs 11% worse than BLINK. Constrained decoding is also necessary for accurate predictions. Generating without triaged candidates drops the accuracy by 9.2%. However, the importance of training data is clearly central, as triage could be adapted to new settings separately.

What is GENRE learning from the massive training data? One possibility is that it learns how to normalize entity names (*Bill Clinton* to *Willian Clinton*) from annotated data. Pretraining on massive amounts of unannotated text followed by a large amount of entity linking annotations may also allow it to learn how to normalize certain informal names (*America*) to formal ones (*The United States*). Furthermore, pretraining may allow for robust modeling of the context before mentions. Finally, as in other NLP tasks, the effect of using the encoding of the context provided by the sentence is likely valuable.

If GENRE exhibits these behaviors, it can generalize certain abilities to new domains. However, if instead, it is memorizing the training data, e.g. learning specific entities that appear in training, it cannot generalize. For example, Wikipedia titles and mentions follow conventions, which may be learnable by the model, but will not generalize to settings that do not use Wikipedia data or KBs. Additionally, De Cao et al. (2021) report results on examples where the gold entity is found in the triage step, which biases toward lexical matches. Examples that can be lexically matched are likely to be solved by name matching. These links are far more common in Wikipedia than in other domains.

In short, while generalization is a challenge for any machine learning model, it may be especially challenging for the mechanisms used by GENRE to learn from the training data. Our first question is: Does GENRE learn generalizable patterns or does it memorize the entities in the training data? We answer by probing how GENRE leverages its training data to perform linking. We evaluate GENRE on new datasets (Chapter 5.5) more challenging than those reported in the original paper. We begin with datasets linked to Wikipedia KBs, then proceed to datasets with different KBs. These new KBs contain entities unobserved in training, especially difficult for GENRE because it cannot access the KB.

5.4 GENRE and the Knowledge Base

GENRE faces challenges in generalization from its lack of access to the KB, which contains information about unseen entities. If GENRE was able to access the KB, could it better generalize to new data? A long line of entity linking research suggests that the answer should be “yes”. In this Chapter, we modify the training data to provide this information to GENRE.

The key idea is to augment the training data with short descriptions of information in the KB. Specifically, we add several keywords that summarize an entity’s description in the KB to each training instance. GENRE is then asked (and trained) to generate the entity title followed by these keywords after each entity mention. This approach uses an unchanged GENRE model architecture to both learn to normalize names and bias the model towards entity descriptions (via keywords) that are most triggered by the (left) context of the mention.

We choose to use keywords instead of full-text descriptions for several reasons. First, in many KBs (especially Wikipedia) entity descriptions are quite long, often multiple paragraphs. This stretches the context beyond what GENRE can reasonably model. Even selecting a short snippet, e.g. the first sentence, also pushes the model beyond what is reasonable. Instead, selecting a few important phrases from the description allows us to easily control the length of the produced string. Furthermore, if selected correctly, these keywords can highlight topically related content, signaling a match with the left context of the entity.

Context enables GENRE to match the topic of the context with that of the candidate entity. For example, the entity *Washington, D.C.* is paired with the keywords *district city congress united states metropolitan area*, while the superficially similar entity *Washington (State)* is paired with *seattle united states british columbia* *cascade range*. Since topical relevance can more easily be learned from the pretraining, the model can better generalize to this (potentially) new entity. This idea is in the same spirit as Bevilacqua et al. (2022), which uses autoregressive language models for search, but decodes entire spans from a corpus, as opposed to keywords.

5.4.1 Keyword Selection

We use the PKE toolkit (Boudin, 2016) to select keywords from the entity description. After a careful examination of several of the unsupervised methods in the toolkit, we found that Topic Rank (Bougouin et al., 2013) produced the most descriptive keywords. We selected the top n keywords (phrases) and multiplied the Topic Rank score s by a frequency factor from the KB. For each keyword in the KB, we took a summation over their inverse rank ($\frac{1}{rank+1}$) within each entity-specific set. The final score for a keyword k for a given entity is

$$s_k * (1 + \log(\sum_{e \in \text{KB}, k \in e} \frac{1}{\text{rank}_k + 1})) \tag{5.1}$$

The addition of the frequency factor removed some highly-scored esoteric keywords (*e.g. Punic Wars* for *Spain*) that may not generalize well. We also experimented with the number of keywords to include, and found that adding at least five words was best. Many keywords are phrases with multiple words, which results in some sequences being just over five words. This selection procedure can be easily applied to other sources of information in KBs.

To avoid GENRE memorizing this training data, we use a different selection method during the training step. During training, we sample five words from the entire keyword list proportional to the Topic Rank score and resample for each training instance. Scores less than zero are set to a small value (0.0001), then normalized to form a probability distribution. At inference, we use the same top-scoring keywords for every instance of an entity. Examples of selected keywords are shown in Appendix Table 5.4.

5.4.2 Training and Inference

We closely follow the training procedure in De Cao et al. (2021). Beginning with the pretrained GENRE model, we train GENRE-KP to maximize the entity title and keyword sequence given the sentence context: maximize $logp_\theta(y|x)$ with respect to the model's parameters θ. We closely follow their choices of training methods and parameter selections, and use teacher forcing, dropout, and label smoothing. The authors originally add a special token to the beginning of each target sequence. In addition to using this token, we add special tokens before and after the keywords to indicate where keywords are present. We do not add these as tokens to the vocabulary due to Fairseq (Ott et al., 2019) constraints. We believe the performance difference is likely small.

Similarly, we use GENRE's candidate scoring with constrained beam search. For

Wikipedia-based datasets, we use the same beam size (10) as in their work. However, for other datasets, we found that a smaller beam size works better (5). Additionally, since we are scoring longer strings that likely vary much more in length than in the title-only model, we explored normalizing the likelihood of a candidate by its length (in number of byte pair encoding tokens). In some datasets, we found this provided a small improvement.

Training these models from scratch – 50 epochs on 9 million training examples – exceeded our computational resources, as would have multiple training runs. Therefore, we initialized training using the existing models. We trained each model on a single NVIDIA GeForce RTX 2080 for 32 hours, iterating over all the data.

5.5 Data

The authors of GENRE use the BLINK dataset, created by that method's authors (Wu et al., 2020) from Wikipedia. This was created from a May 2019 English Wikipedia dump, and includes 5.9 million entities. They use a 9 million sized-subset of Wikipedia-linked mentions (*e.g.* links within Wikipedia pages to other Wikipedia pages). The knowledge base consists of all pages within that snapshot of Wikipedia. We use this dataset to train our keyword model. While we also report evaluation results on the Wikipedia test set, we primarily target datasets that are in more challenging settings. For evaluation, we use the provided candidate sets.

Dataset		**w/ Retrieved Candidates**		**w/ Oracle**	
		GENRE	**GENRE-KP**	**GENRE**	**GENRE-KP**
Wikipedia	Acc.	92.11 ±.67	81.09 ±.97	90.85 ±.69	77.52 ±1.0
	MRR	0.952	0.874	0.943	0.845
TAC	Acc.	92.36 ±.56	91.84 ±.58	80.66 ±.75	80.87 ±.75
	MRR	0.950	0.950	0.856	0.862

Table 5.1: Results on datasets using Wikipedia as the KB, including evaluations on only examples where the correct entity is in the candidate set, and all examples with the correct candidate added if not present. Confidence Intervals (at 95%) are included for accuracy.

For evaluation, we consider two datasets. First, the English text within the 2015 TAC KBP Entity Linking dataset (Ji et al., 2015). While this dataset does not directly link to Wikipedia, almost all entities linked in the English dataset include a Wikipedia title in their metadata. Therefore, we convert all entities with Wikipedia links to their respective entry in the Wikipedia KB and convert all others to NIL. To generate a candidate set at inference time, we use the system of Upadhyay et al. (2018), which is largely based on work in Tsai and Roth (2016b). This approach uses Wikipedia cross-links to generate a prior probability $\mathtt{P}_{\mathtt{prior}}(e_i|m)$ by estimating counts from those mentions. This prior is used to provide the top k English Wikipedia page titles for each mention. Second, we use the Wikia entity linking dataset (Logeswaran et al., 2019) which was constructed from the Wikia website. The authors exclude all NIL entities and provide candidate sets for each mention of size 64, retrieved via BM25. More details on both datasets are in Chapter 2.4.

5.6 Experimental Setup

For GENRE-KP, we train all models on the Wikipedia dataset alone and select the best-performing model using the Wikipedia validation set's loss. In all cases, we do not use the Wikia or TAC training data for training but only as a validation set. For Wikia and TAC data, we provide the model with the sentence where the mention occurs. Sentence boundaries are identified with Spacy (Honnibal and Montani, 2017). We adopt the method of reporting results from Logeswaran et al. (2019), which reports normalized accuracy, which is calculated over the set of examples that are non-NIL and have the gold standard entity in their candidate set. As this restricts the types of examples to those that have mentions which are lexically similar to the entity name, we also report oracle results for some datasets, where we add the gold standard entity to all non-NIL examples if not already present.

5.7 Results

Our experiments address two questions. First, why does GENRE perform so well? We answer this by evaluating generalization to new datasets. Second, can GENRE utilize KB information to improve generalization (GENRE-KP)?

Method	**Validation**				**Test**			
	macro	**micro**	**mrr**	**top-K**	**macro**	**micro**	**mrr**	**top-K**
TF-IDF*	26.06							
Gupta et al*	27.03							
GENRE	29.09	26.89 ±1.0	.42	52.88	31.99	33.16 ±1.1	.44	43.01
GENRE-KP	29.53	29.63 ±1.0	.46	55.65	28.11	27.83 ±1.1	.42	44.64
Comb. (par)	35.54	35.14 ±1.1	.49	54.48	35.63	36.14 ±1.1	.47	43.89
Comb. (jw)	32.36	30.97 ±1.0	.46	58.82	34.48	35.00 ±1.1	.46	47.00

Table 5.2: Results on Wikia Datasets. Results for methods marked with an asterisk are taken from Logeswaran et al. (2019). The combination models are built off of the predictions of GENRE-KP and GENRE described in Chapter 5.7.2. Confidence Intervals (at 95%) are included for micro accuracy.

degree of sim.	**validation accuracy**			**test accuracy**		
	#	**GENRE**	**GENRE-KP**	#	**GENRE**	**GENRE-KP**
mult. categories	4106	11.93	26.04	2341	16.66	25.72
amb. substring	543	54.70	36.46	419	47.02	28.88
high overlap	501	89.22	71.66	825	91.03	62.30
other	2434	33.07	25.55	3227	28.54	20.42

Table 5.3: Results on Wikia Validation by the degree of similarity category. The count column indicates the number of examples that have the correct entity in the triage candidate set.

5.7.1 GENRE Generalization

To probe GENRE's reliance on the mention string matching the normalized entity name, we performed two experiments with the TAC training dataset using the original GENRE model. First, we remove the available context around the entity and replace it with a generic prompt: *This entity is called* ***mention***. In this setting, no context is available for linking decisions. Second, we keep the original context but remove the actual mention string. In this setting, GENRE relies on context alone.

How important to GENRE are each type of information: name matching and context? Compared to the normal model's performance of 49.1% on TAC data (unnormalized, *i.e.* including NIL entities), using only the mention string GENRE did nearly as well (41.6%). By comparison, using only context drops accuracy sizeably (26.8%). This suggests that GENRE largely relies on the training data to learn transformations between the mention and the entity name alone. The context adds a bit to the model's ability.

Despite this result, GENRE performs well on the more challenging datasets. Table 5.1 shows the performance of the GENRE model on the Wikipedia and TAC datasets. While it is unsurprising that GENRE performs well on Wikipedia, the performance on the TAC dataset is surprisingly high for the setting with only retrieved candidates. However, the performance on TAC in the oracle setting is substantially lower. As detailed in Chapter 5.6, we add the gold standard entity to the candidate set for any example where it isn't already present. Focusing only on the retrieved candidates restricts examples to those that can be lexically matched, as triage systems frequently rely on surface forms alone. The oracle setting highlights the fact that many of these more challenging matches cannot be linked by GENRE.

The results for Wikia are shown in Table 5.2. Previous work (Logeswaran et al., 2019) reports results on several baselines for the validation set. We include the best-performing baselines that also have not been trained on Wikia data.[2] We report macro accuracy (accuracy is calculated separately on each domain, and divided by the number of domains), and micro accuracy (accuracy is calculated on the corpus as a whole), in addition to mean reciprocal rank (MRR) and top-K accuracy ($k = 5$). In absolute terms, the performance on the Wikia dataset is worse, as it is not trained to link mentions to the Wikia knowledge bases.

However, it does outperform two previously reported baselines by a small margin, suggesting that even in this challenging setting GENRE is surprisingly effective. For linking mentions to the Wikipedia KB, the sheer amount of data GENRE is trained on enables it to recall which entity is likely best. Therefore, when the data allows for such a strategy, memorization can be effective when paired with a model that can also model the context.

[2] The authors of that paper also include several baselines that are trained on Wikia data but are an unfair comparison for this setting.

5.7.2 GENRE-KP

We evaluate GENRE-KP (GENRE augmented in training by keywords) on all of our datasets discussed in the previous chapter. For the Wikipedia dataset in Table 5.1, GENRE performs consistently better than GENRE-KP. This is unsurprising, given the model's ability to memorize training examples and that it has been trained on other Wikipedia data. As reported in the previous chapter, GENRE relies heavily on name matching, which is sufficient when the model stays within the same domain. In addition, 82.9% of examples in the test set have a Jaro-Winkler score of 0.8 or higher, indicating they are largely lexically similar.

However, performance on the TAC dataset is much closer. On the set of examples where the correct entity is present in the triage candidate set, GENRE performs slightly better on accuracy, while both models tie in MRR. However, in the oracle setting, GENRE-KP performs marginally better in both metrics. This suggests that when trying to link these more challenging examples, which a lexical triage system could not identify, GENRE-KP has an advantage. In short, when context matters, GENRE-KP is better. However, it is still challenging to overcome the memorization capacity of the original GENRE model, and GENRE-KP is still based on the same architecture.

As shown in Table 5.1, the confidence intervals for accuracy ($\alpha = 0.05$) suggest that the differences in top-predictions are not significant for TAC, but are for Wikipedia. However, to test whether GENRE and GENRE-KP produce rankings that are significantly different, we use a Wilcoxon signed-rank test. For the TAC dataset, the difference between the two models on the Retrieved Candidates setting ($p = 0.005$) and the Oracle setting ($p = 0.005$) are both significant. This suggests the two models produce different rankings despite their similar top-level predictions.

Table 5.2 shows results on the Wikia validation and test sets. Again, the differences between GENRE and GENRE-KP are small and depend on the dataset. In the validation set, GENRE-KP performs better in all metrics. In the test set, GENRE performs better with the exception of top-K accuracy, where GENRE-KP performs better. Comparing the rankings produced by the two models using a Wilcoxon signed-rank test, we find that the difference in the GENRE and GENRE-KP validation rankings is significant ($p = 2.1e{-}36$), but not significant for the test rankings ($p = 0.13$). In terms of micro accuracy, the confidence intervals show that the differences between GENRE and GENRE-KP are significant.

At first glance, this suggests that the validation data was overfitted. However, we believe this has more to do with the distribution of examples in each set. Table 5.3 breaks down accuracy by similarity categories (detailed in Chapter 5.5). In the validation set, the largest category is *multiple categories*, which are linked to entities that have a parenthetical in their name. In both sets, GENRE-KP performs consistently better than GENRE, but the portion of these examples is smaller in the test set. Conversely, it is unsurprising that in the cases of *high overlap* and *amb. substring* GENRE performs better since those are categories with high lexical similarity between mention and entity title. For the *other* category, GENRE performs well on examples with high lexical similarity. For example, in the validation set, while only 28.96% of *other* examples have a high lexical similarity, those examples consist of 52.9% of the examples that GENRE gets correct. GENRE performs better on test and GENRE-KP better on validation because the sets have a different distribution over example types.

GENRE and GENRE-KP are useful for different types of examples. GENRE is excellent when the name string alone is sufficient. GENRE-KP improves when context matters. Therefore, we explore combining the two systems. Table 5.2 shows two methods for model combination. First, we propose a model (labeled *par*) where we use the prediction from GENRE-KP if it predicts a parenthetical, and GENRE otherwise. Second, we combine scores of GENRE and GENRE-KP with the Jaro-Winkler lexical similarity between the GENRE model's top predicted entity and the mention serving as a scalar between the two scores (labeled *jw*).[3] This puts more weight on examples where GENRE thinks there is a lexically similar entity name to the mention, but more weight on GENRE-KP in dissimilar cases.

Neither model changes predictions based on the gold standard entity label – they only operate off of the top prediction of one of the two models. In both cases, across both data sets and metrics, both combination models outperform GENRE-KP and GENRE. The confidence intervals included in Table 5.2 suggest that while the

[3] We divide the GENRE score by the candidate's length, to match the length normalization procedure of GENRE-KP, as described in Chapter 5.4.1.

Entity Title	**Keywords**
Germany	german states country member berlin france
Church of England	local parishes christianity common people bishop
General officer	army air forces countries different systems
Flowering plant	plants families species pollen embryo
Civil liberties	religion european convention constitution personal freedoms
Julia Gillard	leader education australia university labor
1924 World Series	games washington ninth walter johnson giants
John Hodgman	radio episode death role appearance
Humoral immunity	function phagocytosis cellular components presence antibodies
Camino Real (play)	time tennessee williams esmeralda marguerite camille
Bumper Tormohlen	december known seasons nba draft record
Craig Wiseman	tim mcgraw blake shelton songs year
Carroll Gardens Historic District	brooklyn common new york city smith
Dallas	city southern united states universities texas
Phanagoria	town site augustus black sea auxiliary bishop
Pierre Berton	time books canada ontario canadian history
Military advisor	afghanistan capabilities marines infantry vietnam
Francesca Schiavone	fourth round italy semifinals french open
Show Boat (1951 film)	julie stage play characters song magnolia
Los Angeles County, California	pasadena arts san bernardino port cities
Metatheria	years earliest marsupials placentals north america
The New York Times	articles report publisher newspaper paper
Tamil Nadu	india coimbatore parts british chennai
Government of Hong Kong	chief secretary systems chief executive head
Roberto Matta	europe surrealist art life work le corbusier
DC Comics	series line picture stories second title
Marvel Comics	year american comic books titles series
Berkshire Hathaway	years share cash general decline stock
Portugal	lisbon portuguese government country territory spain
Methanosphaera	carbon dioxide taxonomy genus formate methanol

Table 5.4: Example keywords for the shuffled scoring selection method detailed in Chapter 5.4.1.

difference between the *jw* model and the best-performing individual model is not significant, the difference between the *par* model and the best-performing individual model is significant.

In summary, adding KB information to GENRE helps, but only where such information is informative to the correct prediction. A simple metric (Jaro Winkler) can successfully identify those cases.

5.8 Limitations

Our experiments focus solely on English-language entity linking. Similar models have been trained to perform entity linking in multiple languages (De Cao et al., 2022), but we do not consider performance beyond English. The issues faced in other languages are likely to be similar, but the multilingual element of other models might lead to different results. Further, how to select keywords in the multilingual setting is unclear.

In addition, we are limited by the available annotated entity linking datasets. Given that we need a large amount of data to train these models, they are inherently reliant on Wikipedia. These entity linking datasets are skewed towards specific types of matches, including ones that are frequently exact matches. The effectiveness of this model might change when trained on a dataset with different characteristics, even with a large amount of data.

Finally, the computational resources required to train these models are large, and our final results do not reflect numerous other preliminary experiments. This restricts our ability to run multiple experiments, train models from scratch easily, and potentially leads to underfitting of our final models.

5.9 Conclusion And Future Work

Autoregressive transformer-based sequence-to-sequence models, such as BART, have found increasing success in information extraction tasks. The GENRE model, which applies autoregressive sequence-to-sequence approaches to entity linking, has high performance on many datasets linked to the Wikipedia domain. However, its performance on other domains with different challenges produces mixed results.

We suggest that adding previously-explored entity linking features to GENRE can address some of these pitfalls. Specifically, descriptions are a commonly used source of text to make linking decisions. While we see performance decreases in the original Wikipedia datasets, we see some improvements in both newswire text and in applying GENRE-KP to previously unseen knowledge bases for more challenging matches. Yet, the ability of GENRE to work in even challenging settings suggests that it can memorize patterns useful for mention-entity pairs with high lexical similarity.

There are several unexplored directions for our model. Specifically, we used an off-the-shelf keyword selection method. Selecting keywords in a more targeted fashion – perhaps by selecting keywords for an entity that best separates it from another entity – may improve performance. Having the computational resources to train a model from scratch would also likely improve performance, as opposed to training from a GENRE checkpoint. Moreover, we focus on integrating descriptive information within the original GENRE framework. Future work may consider an autoregressive entity linker with a novel architecture that can integrate and learn representations of entities that would better utilize this information in learning.

Within information extraction more broadly, there have been other works that applied autoregressive models to multilingual entity linking (De Cao et al., 2022) and closed information extraction (Josifoski et al., 2022). Other autoregressive approaches to entity linking include De Cao et al. (2021), which seeks to alleviate some of the performance challenges with GENRE during inference.[4] More recently, CM3 (Aghajanyan et al., 2022) was proposed as a method that allows both the left and right context surrounding an entity mention to be modeled by producing the link at the end of the sequence. This alleviates one of the challenges of the GENRE model, which only can leverage the left context of the model. However, it remains the case that CM3 only uses entity name information. Therefore, the challenges that GENRE faces are likely true of CM3 as well.

[4] In early experiments, we found this performed substantially worse in domains for which the model did not have training data

Chapter 6

Challenges in Clinical Concept Linking

6.1 Introduction

Recent work (Chapter 2.2.2.1) has investigated transferring entity linking systems to new domains, such as sub-sections of Wikia fan-constructed Wikipedia-like sites (Logeswaran et al., 2019). However, a *domain* is an ambiguous term, and linking for some domains is a more challenging task than others. For example, Wikia is linked to smaller knowledge bases with less ambiguous distinctions present and similar text structure to the training data (often Wikipedia). Transferring a Wikipedia-trained linker to a college football dataset is likely easier than transferring to a Medical dataset, for example.

Linking concepts in the medical domain is a crucial task that has several unique characteristics. There are some standard challenges, such as partial matches (*e.g. balanced salt solution (BSS; pH 7.6 containing 5.5 mM anhydrous* ***d-glucose*** contains a mention of the concept *Glucose*). In addition, medical knowledge bases often contain concepts that are closely related in a hierarchical fashion. For example, the concept *Glucose* has a parent relationship with *Deoxyglucose*, and a child relationship with *Sugars*. Selecting which of these is the most appropriate concept within the hierarchy is an additional challenge that is not present within a Wikipedia-based knowledge base. Additionally, many concepts do not have definitions or other longer text snippets. Finally, medical and scientific documents contain tokens often not seen in general domain training data, such as chemical names like *1,2-dioleoyl-sn-glycero-3-phospho-l-serine*.

How does a state-of-the-art entity linker, such as BLINK (Wu et al., 2020), transfer to medical tasks? We evaluate the BLINK model on three medically-related datasets – clinical notes, Coronavirus-related documents, and Chemical documents. We propose a simple adaptation to handle the large number of synonyms present in the medical knowledge bases. BLINK performs competitively in situations where there is no training data available. However, BLINK performs far worse in situations where there is in-domain training data available, highlighting the importance of work that can be applied to domain-specific settings.

6.2 Adapting an Entity Linker to Medicine

We use the BLINK model (Chapter 2.2.2.1). In the original model, the authors propose a second step that jointly embeds the mention and candidate entity pairs into a single representation, which is paired with a learned weight layer to produce a final score. They use this approach for only the top candidates ($n = 10$), and this reranking step introduces marginal improvements. However, due to the high computation cost paired with a small performance improvement, we only report results using the bi-encoder model.

We adapted this model in several ways to better work with our knowledge base. First, most concepts within UMLS (United Medical Language System, Bodenreider

(2004), a set of medical knowledge bases) are assigned multiple names (*e.g. heart attack*, *myocardial infarction*, *cardiovascular stroke*) which can be lexically distinct. We create a distinct entity representation for each name paired with the entity's description and only consider the highest-ranking representation for each entity. We found this vastly improved our performance– for example, Recall@1 for the Chemical development dataset improved from 0.301 to 0.639, excluding NILs. Other methods of including the alternative names did not perform as well, such as including a small subset of the most lexically dissimilar in a single representation. Additionally, several concepts within the knowledge base do not have definitions. In this case, we include alternative names if available and otherwise include no definition.

6.2.1 Datasets

We use several scientific and medical datasets. First, we use the NLM-Chem corpus (Dogan et al., 2021), which is a corpus of 150 scientific articles split into test, train, and development sections. While most mentions link to a single concept (or none, *i.e. CUI-less* or NIL), there are several that are linked to multiple concepts. As BLINK does not have a mechanism for predicting multiple links and they are rare (6% of the development set), we always count these as incorrect. Additionally, we use two medical datasets consisting of clinical notes. We use the MCN corpus (Luo et al., 2019), which consists of medical notes linked to SNOMED and RXNorm, two other ontologies within the UMLS. Finally, we include a recently-collected dataset

Dataset	Test Set Size	Baseline Acc.	Acc.	R@20	R@50
MCN Corpus	6,925	85.26	38.9 (±0.01)	64.5	68.4
NLM-Chem	12,411	—	54.4	67.9	71.2
COVID	947	31.84	34.5 (±0.03)	70.3	73.3

Table 6.1: BLINK performance on several Medical datasets compared to the reported best baselines. The chemical dataset paper does not report accuracy. Considering the confidence intervals of BLINK's accuracy ($\alpha = 0.05$) compared to the baselines, the difference for the MCN Corpus is statistically significant, but the difference for the COVID corpus is not.

consisting of Coronavirus-related documents (Sohrab et al., 2020). While the first two datasets have manually annotated training sets, this dataset only has a small manually annotated test set paired with larger machine-annotated mentions. For additional information on all three datasets, refer to Chapter 2.4.

6.2.2 Results

BLINK's performance on the three medical datasets is shown in Table 6.1. For two datasets, we report the best baseline performance as reported in the respective papers. For the NLM-Chem corpus, the authors do not report accuracy metrics. In addition to accuracy, we report recall at 20 (R@20) and at 50 (R@50).

In cases where there are domain-specific training data available, such as in the MCN Corpus, BLINK performs very poorly in comparison. Common errors are shown in 6.2. Several patterns emerge when evaluating the errors produced in the dataset. First, BLINK often erroneously predicts a child concept instead of a parent one, or vice versa. For example, *A Chest X-Ray* should be linked to *Plain Chest X-ray*, but

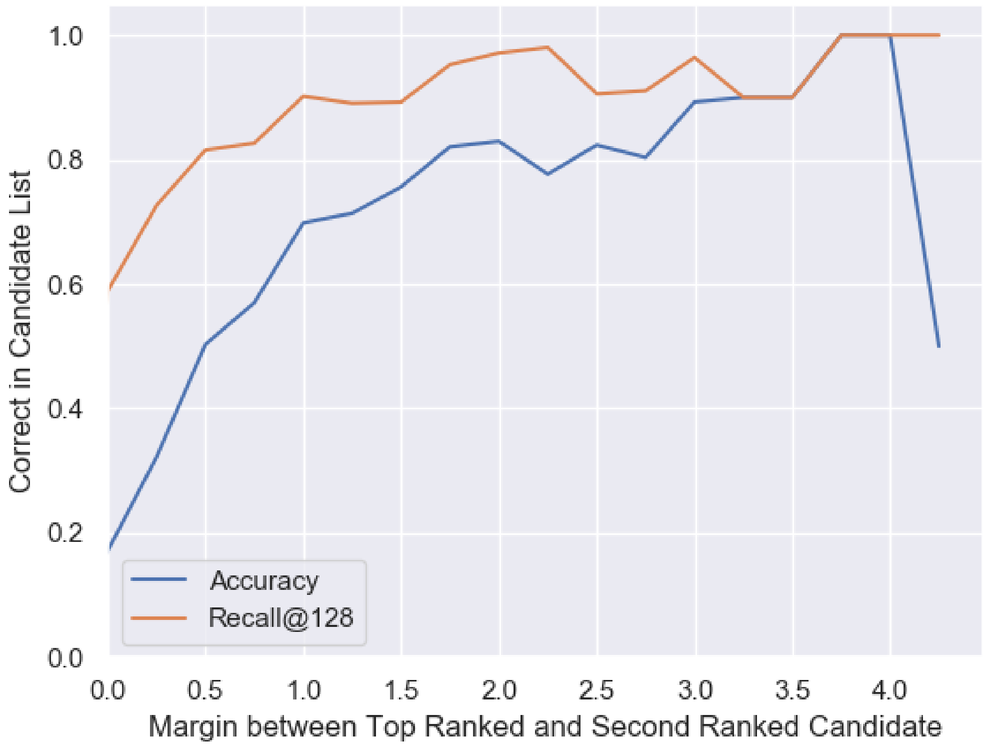

Figure 6.1: The margin between the Top Ranked Candidate and the Second Ranked Candidate, compared with Accuracy and Recall at 128.

BLINK predicts that it should be linked to the parent concept *X-Ray*. Alternatively, BLINK incorrectly predicts a child concept *Tylenol Cough Oral Liquid* in place of the correct parent concept *Tylenol*. Separately, there are also more nuanced differences in type that BLINK is not able to disambiguate between. For example, the mention *Heart Rate* should be linked to the act of measurement, *Pulse Taking*, but is incorrectly linked to the underlying concept *Heart Rate*.

For the COVID dataset, the authors only used automatically annotated data to train their linker. It is less surprising, therefore, that we see a small performance boost when using BLINK as compared to the baseline. In all other cases, BLINK performs poorly on these challenging domains. Overall, this highlight an important conclusion. As shown in the previous analysis of BLINK's performance, linking to medical documents has some task-specific characteristics that are challenging to model without domain-focused work. This includes the differing nature of what is available in the knowledge base, in terms of synonyms, and the hierarchical nature of the knowledge base. A model trained on Wikipedia data, which generally has clearly delineated entities, is not well suited for this task.

While the final linking performance is too poor on these datasets to be used with confidence, our experiments point to alternative ways that the BLINK linker might be used in practice. As seen in Figure 6.1, we find that the margin between the score of the top-ranked candidate, and the second-ranked score, correlates with how accurate that prediction is. In other words, the larger the margin between the predicted entity and the following candidate, the most likely it is to be accurate. While this does not help in end-linking performance, this could be useful when annotating a dataset. Annotators could focus on annotating the examples the linker is less able to disambiguate between, which would result in more informative annotations being produced. This can enable strategies like Thompson Sampling (Thompson, 1933) to be deployed.

Therefore, research that focuses squarely on building linking systems is critical to producing accurate predictions. In the next sections, we show three works that investigate three important tasks within medical concept linking. In the first, Chapter 6.3, we propose a final re-ranker method for medical linking that focuses on leveraging resources specific to the medical setting. In Chapter 7.1, we propose a method of candidate selection for clinical concept linking. As shown in the results in Table 6.1, non-medical systems can struggle to achieve a high level of recall even at a large candidate size, and so systems trained specifically for the task are required. Finally, in Chapter 8.1, we investigate how to best identify synonymous terms within an unlabelled corpus. As discussed in Chapter 6.2, leveraging synonyms is crucial for accurate linking performance, and automatically identifying alternative terms that refer to the same concept expands those available to a linker.

Mention	**Correct Concept**	**BLINK Prediction**
A Chest X-Ray	Plain chest X-ray	X-Ray
Medications	Pharmacotherapy	Medications
Tenderness	Sore to Touch	Tenderness, Muscle
Tylenol	Tylenol	Tylenol Cough Oral Liquid
Heart Rate	Pulse taking	Heart Rate
Right	Right	Left-to-right shunt

Table 6.2: Selected errors produced by BLINK on the MCN Corpus.

6.3 Designing Linkers for the Medical Domain

The challenges of adapting general domain entity linkers, such as BLINK, are highlighted in Section 6.2.[1] While such models have been trained on a vast amount of data, the domains of Wikipedia and Medicine are too distinct to result in high performance. Part of this is due to the different terms used – medical terms may be used in Wikipedia, but are likely far less frequent. Additionally, clinical concepts have fine-grained distinctions that are not present in Wikipedia. As highlighted in Table 6.2, there are often parent and child distinctions that confuse an entity linker trained on data that does not contain similar ones. Thus, we propose a clinical-specific entity linking method, which while leveraging clinical-specific resources, adopts methods proposed in general entity linking work.

We propose learning contextualized representations that leverage both free text and information from knowledge bases. We train a contextualized language model (Peters et al., 2018) on unannotated clinical text, leveraging sentence context to construct a mention. We explore several methods of building representations of the mention span and concept, including pooling and attention, and pre-training our linker with additional data from the ontology to augment the small amount of annotated data

[1] Elliot Schumacher, Andriy Mulyar, and Mark Dredze. 2020. Clinical Concept Linking with Contextualized Neural Representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8585–8592, Online. Association for Computational Linguistics.

present. The resulting ranker outperforms a non-contextualized version of our model and beats the previous best-performing system (Leaman et al., 2013) in most metrics.

6.4 Methods

Our concept linking system is based on a pairwise neural network ranker (§6.4.1) using contextualized representations (§6.4.2) for both the mention and concept. We leverage the context present in clinical notes for our representations and synonyms present within the UMLS to train our linker. This architecture is similar to the systems described in Chapter 3.2.2 and 4.1.1. However, type and description information are not included. The version of the UMLS that is used with this dataset does not contain descriptions for the majority of the concepts, so description features are not included. Additionally, type information is not included due to the more general nature of the types in UMLS.

6.4.1 Neural Ranker

For a given mention string m and document, the system ranks all possible candidates c in the KB. Figure 6.2 shows our ranking system, based on the *Rank model* of Dehghani et al. (2017). We learn the parameters θ of a scoring function $S(m, c; \theta)$, which consists of a feed-forward neural network with hidden layers d that takes input representations of m and c in addition to pairwise features. We train

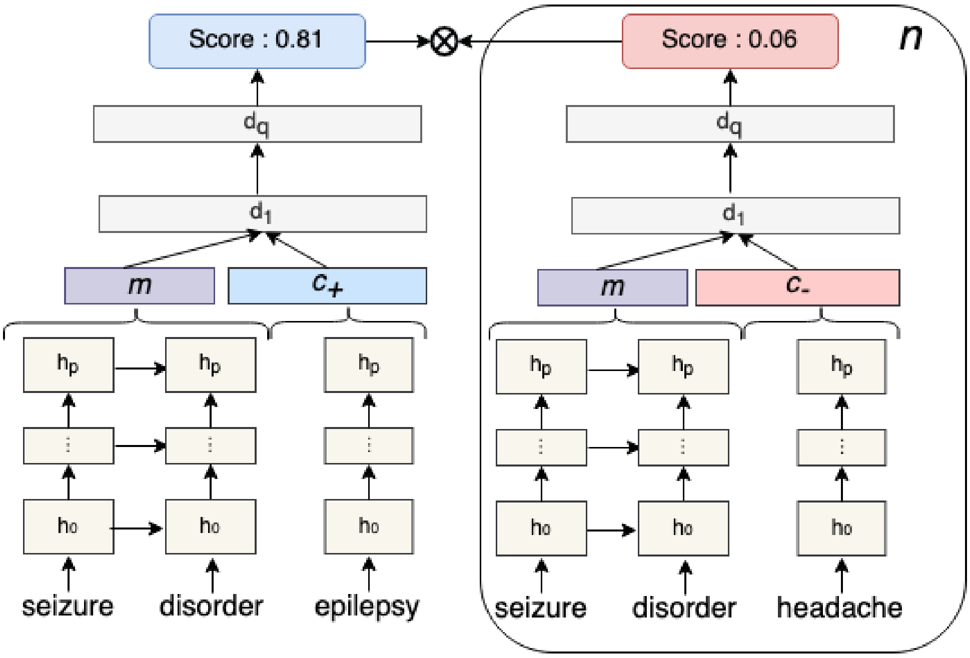

Figure 6.2: Architecture for our neural ranker. The input consists of gold standard mention string representation m (purple), gold standard concept representation c_+ (blue), and n randomly selected negative concept representation c_- pairings (red). The ELMo hidden states are noted as h, and the hidden states of our feed-forward neural network are noted as d. To build our ELMo representations for m, c_+ and c_-, we select the representation from the lowest layer of the model.

using pairwise loss, in which we have two point-wise networks – one which takes the mention m and correct concept c_+ as input, the other which takes the mention m and incorrect concept c_- – with shared parameters that are updated to minimize the loss function. Using a pairwise model allows us to learn a scoring function that does not rely on annotated scores.

Adapting the approach of Dehghani et al. (2017), we use adaptive hinge loss, which considers n negative concepts and selects the highest scoring concept as the negative sample. For mention m, correct concept c_+, and n negative samples c_{0-} to c_{n-}, our loss function is:

$$L(\theta) = \mathbf{max}\{0, \epsilon - (S(\{m, c_+\}; \theta) - \mathbf{max}\{S(\{m, c_{0-}\}; \theta) \ldots S(\{m, c_{n-}\}; \theta)\}\} \quad (6.1)$$

6.4.2 Contextualized Representations

As described in Chapter 2.2.2, contextualized representations of text have produced impressive performance gains in a variety of tasks, including clinical. For this work, we use ELMo, an early contextualized language model that leverages contextualized representations. These models are robust to out-of-vocabulary types, so they provide broad coverage to the diverse types present in clinical text. We train ELMo on clinical notes and create mention representations m by running the entire sentence through the model and selecting the resulting word representations for the mention (the lowest token representation) from the LSTM..[2] The concept representations c are created in the same manner as m except that only the name of the concept, as there is often no available context.[3]

For multi-word mentions and concept names, we explore two methods of creating a single embedding. First, we use max-pooling over the set of token embeddings (reported as **Max** in Table 6.3). Second, we run self-attention (Vaswani et al., 2017)[4] over the set of token embeddings, with a single head to attend over the tokens (noted as **Attention**).

6.4.3 Pre-training with Structured Data

Pre-training a model using an alternative data source has been frequently used in the field of machine learning (Erhan et al., 2010; Sharif Razavian et al., 2014). This includes work targeting entity linking (Tsujimura et al., 2019), presented at a recent shared task (Luo et al., 2019). A model is pre-trained on a large amount of data from a related dataset and then is trained on the target task, which allows a model to see more examples to achieve a better initialization for training on the final task.

As creation is expensive, most annotated clinical datasets are small, such as for our task. Therefore, we look to alternative data sources for pre-training our model. For a

[2] While there is now a multitude of deep transformer-based LMs (Devlin et al., 2019), the principle of contextualized representations is the same. Additionally, others have found ELMo trained on MIMIC does better than a similarly trained BERT model (Schumacher and Dredze, 2019)

[3] We ran experiments that padded the names with synonyms or other forms of available text within the knowledge base. However, we did not see consistent improvements.

[4] We use the implementation provided by `https://github.com/kaushalshetty/Structured-Self-Attention`.

given concept (e.g. **epilepsy**), the UMLS includes synonyms (e.g. **seizure disorder**, **epileptic fits**), which can be used to pre-train our linker. Unlike in the annotated clinical data, there is no surrounding context, and terms in the UMLS are more likely to be formal. However, training on synonyms will allow for a greater variety of terms to be seen by our model than otherwise possible.

Therefore, using all synonyms taken from the annotated subset of the UMLS, we pre-train our linker before training on the annotated clinical notes. We follow the previous training procedure by replacing the mention representation m with the synonym string representation only (without surrounding sentence), thus training the linker to assign a higher score to the synonym paired with the corresponding concept representation c_+ against negatively sampled concepts c_-. We use this pre-training initialization with the Attention model discussed in the previous chapter and note this as **Att. + Pre.** in Table 6.3.

6.5 Experimental Setup

We train and evaluate our system on the concept linking dataset released for ShARe/CLEF eHealth Evaluation Lab 2013 Task 1b (Pradhan et al., 2013). This dataset consists of concept span annotations built on a subset of MIMIC 2.5 clinical notes (Saeed et al., 2011). We do not report on the task-designated test set as it was unavailable. Each disorder mention in the clinical note is annotated with concept

	CUI		**All**	
	Acc	**MRR**	**Acc**	**MRR**
DNorm	**0.73**	0.75	0.55	0.57
Word2vec	0.26	0.33	0.21	0.30
Max	0.66	0.70	0.58	0.67
Attention	0.70	0.75	**0.62**	**0.71**
Att. + Pre.	0.70	**0.78**	0.59	**0.71**

Table 6.3: Accuracy (top-1) and MRR (mean reciprocal rank) for the test sets, for mentions with linked concepts (CUI) and all mentions (All).

information. This information either includes the relevant concept unique identifier (CUI), or annotations noting cases where the correct concept could not be identified – primarily with the CUI-less annotation. For more information on this data, refer to Chapter 2.4.

In Table 6.3, we report results on only mentions with links to the ontology (**CUI**) and mentions with links to the ontology and *CUI-less* mentions (**All**). We train ELMo on 199,987 clinical notes from MIMIC III (Johnson et al., 2016) as the source of our clinical text, pre-processing the data using the NLTK toolkit (Bird et al., 2009). For the Pre-training model, we augment the clinical text training data with synonyms, definitions, and names of related concepts from the selected subset of UMLS. Altogether, this resulted in 645,863 additional sentences of training data.

We compare our system to DNorm (Leaman et al., 2013) for the SHARE/Clef 2013 dataset, the best performing system in the SHARE/Clef 2013 shared task. Unlike many other concept linking systems, DNorm scores each mention against all concepts and does not use a triage system, allowing a fair comparison to our system.

DNorm builds term frequency-inverse document frequency (TF-IDF) representations of both the mention and concept and learns a weighted similarity to rank concepts for each mention. It is unable to return concept candidates for mentions that are out-of-vocabulary as it uses a word-level measure. The authors add a specific `CUI-less` representation, which is made of entries occurring more than four times in training. We report results on our recreated test set, as the evaluation set provided for the shared task was not available to us. We also compare using Word2vec (Mikolov et al., 2013b) representations instead of ELMo representations in the same linking architecture to test the effect of contextualized embeddings. We trained the Word2vec model on the MIMIC dataset. We created single embeddings ($d = 600$) for mentions and concepts by max pooling over all embeddings for words in the corresponding text, ignoring all out-of-vocabulary words.

We explored several parameter configurations for our model suggested in Dehghani et al. (2017), reporting the best performing models on development. These include hidden layers of size [256, 512, 1024] and number of layers in [1,2,3], with a Tanh activation function for final layer and ReLu (Glorot et al., 2011) for all others. We optimize using the ADAM optimizer (Kingma and Ba, 2014), and a dropout rate of 0.2. Parameter values and development metrics are available in Table 6.4.Note the pre-training model contains parameters for the pre-training stage only (and thus we do not note accuracy or mean reciprocal rank), while Pre + Att contains parameters for the final trained model. All GPU types have 12 GB of memory.

	Max	**Attention**	**Pretraining**	**Pre + Att**
Dev Acc (CUI)	0.685	0.730	-	0.704
Dev MRR (CUI)	0.719	0.766	-	0.776
Reported Epoch	2499	4000	1	750
Random Seed	3011457727	3027767026	589590319	3635932273
Learning Rate	1e-5	1e-5	1e-5	1e-5
Hidden Layers	[1024, 512]	[1024, 512]	[1024, 512]	[1024, 512]
Batch Size	12	12	32	16
Num. Negative Samples	10	10	10	10
Training Time per epoch (min.)	7.2	3.4	1860	4.6
GPU Type	Tesla K80	GTX 1080ti	Tesla K80	Tesla K80

Table 6.4: The above table contains replication information for the models trained on SHaRE data.

For the ELMo models, we trained for 10 epochs using the default configuration. For `CUI-less` mentions, we select a threshold score based on the development set, equal to the mean score of all `CUI-less` entries. If an entry does not have a scored concept above that threshold, we consider it `CUI-less`, adding `CUI-less` at that position in the list for MRR. We use the Pytorch framework and code from the Spotlight library (Kula, 2017).

6.6 Results

Table 6.3 reports accuracy and mean reciprocal rank (MRR) for all models. We compare our models (**Word2Vec**, **Max**, **Attention**, and **Att. + Pre.**) to DNorm for all mentions (All) and only those with links to concepts in the KB (CUI). While DNorm has higher accuracy on entries with CUIs, our models have higher MRR on entities with CUIs (**Att. + Pre.**) and perform best on all entities in both accuracy and MRR (**Attention** and **Att. + Pre.**). For each metric, we compare the best score (in bold) to the baseline using a two-tailed z-score test (for CUI ACC, we compare it to the next best score). We find that for all CUI models, the difference is not significant, while for All models, $p < 0.05$.

6.7 Discussion

Our neural ranking models with attention outperform all other models, except for CUI-only accuracy. In the case of entities with CUIs, we find that pre-training the model does provide a gain in ranking accuracy (MRR). In the case of all entities, we find that the attention models provide a sizable gain in both accuracy and MRR.

We conducted an error analysis of the best performing MRR model (**Att. + Pre.**) on the development data, looking at errors where the gold standard concept was not highly ranked (assigned a rank of 10 or above). Of those errors ($n = 110$), we find that 26% are mentions that contain only acronyms (e.g. *LBP* for *lower back pain*), and 14% are mentions containing some other abbreviation (a shorted word, e.g. *post nasal drip* for *Posterior rhinorrhoea*, or a partial acronym, *Seizure d / o* for *Epilepsy*). Compared to similar errors from **Attention** model ($n = 161$), we find that the number of acronym errors is nearly the same (24) as the better-performing model (26). In contrast, the number of non-abbreviation errors drops significantly. This suggests that pre-training provides a useful signal for mentions that consist of variations appearing in the ontology. However, it does not help with acronyms or other abbreviations that are less likely to appear in the ontology or are shorter and more ambiguous (e.g., ’R’ for Rhonchus).

While the linker often predicted unrelated concepts (40% of errors) for concepts where the correct concept was ranked above 10, many incorrect concept predictions were somewhat related to the gold concept (e.g., for mention *atherosclerotic plaque* with gold concept *Atherosclerotic fibrous plaque* our model predicted the concept *Atherosclerosis*). We further noticed that in 21% of cases the linker predicted a relevant concept (e.g., mention *thrombosed* and *Thrombosis*), but is not counted as correct due to annotation decisions. This could be due to multiple possible concepts in the ontology or the presence of closely-related concepts.

Deploying our system in a large-volume clinical setting would likely require several alterations. The main computational barrier to labeling a large amount of data, the speed of prediction, can be addressed by using an accurate candidate selection system to prune the number of concepts considered. Considering a smaller subset (e.g., 20) of concepts instead of all would significantly improve the speed. This highlights the importance of an accurate triage system, such as the one described in Chapter 7.1. Further, if using a consistent portion of the ontology, caching the concept embeddings c as opposed to building them in-model also enhances efficiency. Depending on the application, a less accurate but faster linker might be a better choice (e.g. for all clinical notes at a medical institution). In contrast, a more complex linker, such as ours, may be a better option for specific subsets of notes that require better accuracy (e.g., the results of specific clinical studies).

Our results demonstrate the advantages of using contextualized embeddings for ranking tasks, and that using information from the knowledge base for training is an essential direction for learning concept representations for sparse KB domains. Future work should consider additional methods for integrating ontology structure into representation learning.

This work has been cited in a recent work on concept linking (Kim et al., 2020; Xu and Miller, 2022). A large amount of this research has focused on biomedical concept linking data (Bo and Zhang, 2021), a task discussed in Chapter 2.3. This includes further exploration into leveraging synonyms and hypernyms for training a neural linker (Yan et al., 2021; Xu and Bethard, 2021). Liu et al. (2021a) proposes to learn representations of entities using the structure present within the UMLS. In nearly all of these methods, the authors use contextualized language models to build representations of text. While these mostly include more advanced models than ELMo (discussed in Chapter 2.3.3), this highlights the importance of that element in building NLP for clinical data.

Newer datasets for clinical concept linking have been released since this work was concluded (Luo et al., 2019). These are linked to versions of UMLS that include more descriptions for medical concepts, alleviating one challenging aspect of this problem faced in this work. However, challenges remain around clinical concept linking. In light of the large amount of data available for standard entity linking, linking annotations for medical data is still challenging to produce due to the expertise required and the privacy challenges to consider. This reality means that exploring alternatives to producing more training data is still required.

Chapter 7

Learning Efficient Entity Candidate Generation for Clinical Data

7.1 Introduction

The task of concept linking, detailed in Section 2.3.2.1, can be broken down into a two-step system.[1] The first is a candidate generation (or candidate selection, triage) step which produces a list of possible concepts from the ontology. A ranker then selects the most appropriate candidate from that list, based on machine learning and extracted features. This two-step approach has the benefit of pruning unlikely concept candidates prior to the final linking stage, a necessary step when dealing with knowledge bases with millions of concepts. This is in contrast to a single-step approach, which chooses a link from the entire ontology, and may require the computation of more fine-grained features over a larger set of concepts. The two-stage approach allows for a simpler feature set to be used in the first step, and a more fine-grained feature set to be used in the final step.

Previous work in concept linking has largely focused on developing rankers (or classifiers), assuming an existing method that produces a list of concepts that contain the correct answer. Candidate generation must be fast, so it often relies on basic lexical matching algorithms that produce a large list of candidates but do not incorporate features or machine learning. Such as system could consist of simple n-gram matching or other lexical similarity features comparing the mention string and the concept name. Others, such as in the first version of MetaMap (Aronson, 2001), generate

[1] Elliot Schumacher and Mark Dredze. 2018. Clinical Concept Linking with Contextualized Neural Representations. In *Automated Knowledge Base Construction (AKBC), 2018.*

possible variations of the mention drawn from the structured information within the ontology, and score each on the type of variation present. While more sophisticated than standard lexical matching, it requires all likely variations to be present in the ontology or annotated dictionaries. Similarly, Aggarwal and Barker (2015) has included a candidate over-generation phase, where possible variants are proposed and then re-ranked by inverse document frequency (IDF) of text from the document and knowledge base. While the work in Section 6 does not use a triage system, it only does so to compare to previous work. Deploying such a system in a real-world setting is likely too computationally inefficient.

We propose a candidate generation system that produces a candidate list that has both high coverage, and a ranking that is a useful starting point for a final classifier. We adapt DiscK (Chen and Van Durme, 2017), a framework that allows for feature template-level weighting, and efficient retrieval by feature projection. Using a feature-based system provides flexibility in selecting the criteria for candidate concepts. We consider several different feature templates useful for medical concept linking, and learn a retrieval function. We develop our system for linking disorder mentions, and evaluate using information retrieval metrics that measure the quality of the ranked candidate list. We find that our approach improves over several standard lexical matching baselines. Finally, we integrate our candidate generation system into an existing concept linking system (Leaman et al., 2013). Although restricting candidates to those generated by DiscK causes a small reduction in coverage and mean reciprocal rank, large gains in efficiency are made due to the reduced number of concepts considered in the final linking stage.

7.2 Triage for concept linking

We define the task of triage for medical concept linking as follows. For a mention m containing a concept reference, the candidate generation process for selecting N candidate concepts can be defined as follows. Given a set of concepts c from an ontology $C = (c_0, ..., c_Q)$ containing Q candidates, we can select N candidate concepts from the ontology by scoring each candidate with a candidate likelihood function f. Therefore, for each candidate i in C, a score can be calculated as

$$\text{score}_i = f(m, c_i) \tag{7.1}$$

The set of candidates C is then sorted by the candidate score, with the top N candidates from the sorted candidate list selected as the final candidate list.

The scoring function f can be formulated using several different methodologies. Many systems use non-feature-based approaches, only considering candidates that match a single criterion. These include string matching algorithms such as the Levenshtein or Jaro-Winkler distance. While this approach benefits from simplicity, it excludes candidates with low string similarity. As discussed in Chapter 2.1.4, there are often mentions of entities, or in this case, concepts, that do not share lexical forms.

In contrast, a feature-based approach can consider multiple attributes, such as string similarity, but can be designed to include correct candidates that may not be identified by a simple method by using a variety of features. In general, we would prefer to use a more flexible feature-based approach to triage but are limited in that we cannot efficiently compute features and scores for a large set of candidates.

7.3 Discriminative Information Retrieval for Knowledge Discovery

Chen and Van Durme (2017) introduced the framework *DiscK*, which formulates candidate generation as a feature-based classification retrieval problem. Using a simple feature set, it learns a weighted similarity score for a query and each concept, creating a ranked list of concepts for each query. Given a query q and a candidate set $D = \{p_1, ..., p_N\}$, the system scores the pair by a specified feature function $F(q, p)$ and retrieves the top-k candidates:

$$\underset{p \in D}{\operatorname{argmax}} \{F(q, p)\} \tag{7.2}$$

This normally requires scores to be calculated between every query and candidate, which is not efficient for larger sets. However, DiscK proposes a feature set formulation that allows for feature projection – for a given query, the expected feature values for the relevant candidate can be calculated. This allows for efficient retrieval by indexing, and therefore pairwise scoring is not required.

To allow for this efficient retrieval, they restrict features to two feature types.[2] The Cartesian Product, one of the feature types, for a query and a candidate is defined as

$$f_Q(q) \otimes f_P(p) = \{((k_i, k_j) = (v_i, v_j), w_i)\} \tag{7.3}$$

for a query $F_Q(q) = \{(k_i = v_i, w_i)\}$ and for a candidate $f_P(p) = \{(k_j = v_j, 1)\}$. The variables k_i and k_j refer to specific features type instances and v_i and v_j refer to feature values, and w_i refers to the weight of that feature instance. For example, if the feature type is the bigram word count of the query string *broken leg*, k_i would be the feature instance (e.g. *broken_leg*) and v_i is the feature value (e.g. *1*). The projection of the Cartesian Product is defined as

$$t_\theta^{\otimes}(f) = \{(k' = v', w\theta_{(k,k')=(v,v')} | k = v, w) \in f\} \tag{7.4}$$

for all k', v' such that $\theta_{(k,k')=(v,v')} \neq 0$. With this definition, they show that with model parameters θ,

$$t_\theta^{\otimes}(f) \cdot g = \theta \cdot (f \otimes g) \tag{7.5}$$

meaning that the projected features of the candidate multiplied by the features of the candidate are equivalent to the weighted pairwise score of the query and candidate.

[2] We do not consider one, the Join type, for features in our current system.

The term w is specified for each feature – this can be any real number in $[0, 1]$, representing a Boolean or a normalized count, for example. The feature parameters θ are selected to optimize the retrieval equation (noted as Equation 7.2) on the training data. These parameters θ are trained using a negative sampling procedure, with the goal of learning a set of weights that will correctly predict mention-concept pairs. For each training mention, it is paired with the correct ontology entry and 50 incorrect ontology entries. The resulting weights are used to project which ontology entry is most suited to the mention feature set. This is trained using a log-linear model. This formulation is computationally efficient because it only involves a sparse feature set, which allows for efficient retrieval. However, it also restricts the types of features.

DiscK allows us to efficiently retrieve candidates over a large ontology in sublinear time, and select a small subset containing likely links. A final linker can then use this subset to make linking decisions. As a ranker only needs to consider a subset of the entire ontology (e.g. 1% of candidates), computationally-intensive features can be used at a smaller total computation cost. In larger sets of clinical notes, this will reduce the total computational cost, making the entire concept linking pipeline more efficient.

7.4 DiscK for Clinical Concept Linking

Using the framework discussed in the previous Chapter, we developed a version of DiscK suited for clinical concept linking. The feature set developed for this task consists of feature templates that capture the relatedness of the mention text and properties of a UMLS concept. While features from additional mention properties, such as the surrounding sentence, were tested, none provided an improvement over features built from the mention span text alone. The features tested included a bag of word template using the entire sentence and a range of ngram sizes for words and char-grams. We imagine this is the case since the wider sentence context may often not be lexically similar to the concept name or definition.

We used several lexical features to adapt DiscK to concept linking candidate generation. These include the following feature templates.

- A full-string match between the mention and any of the concept names, which receive a feature value of 1 if they are identical.
- A bag of words feature template that matches overlapping individual words between the mention text and the concept name, where each overlapping word is individually weighted by its inverse document frequency.[3]
- A bag of words feature template that matches overlapping individual words between the mention text and the concept definition (if present), where each

[3] calculated in a separate corpus

overlapping word is individually weighted by its inverse document frequency. The inverse document frequency weights were calculated on a separate non-medical corpus.

- For some models, a bag of character-grams of length 6 are included (see below) – a range of lengths were tested, but this size resulted in the largest coverage increase on the development set. The resulting character-grams are also weighted by inverse document frequency.[4]

- An abbreviation dictionary built from the Wikipedia list of disease abbreviations[5] and matched to the mention text.

- For some models, an expanded abbreviation algorithm was included, which simply combines the first character in each word in the concept name to create an acronym.

- A lemmatized bag of word feature template, using the Stanford Toolkit (Manning et al., 2014), in order to capture any overlapping words that would be excluded due to differences in morphology.

While we found that many mentions could be matched to concepts by lexical features, a significant portion required non-lexical features (e.g. mention *joint pains*, concept *Arthralgia NOS*) Therefore, we added the mention text of any linked concept

[4] Character-gram IDF statistics were calculated on the MIMIC corpus.

[5] https://en.wikipedia.org/wiki/List_of_abbreviations_for_diseases_and_disorders

in the training data to the set of concept names. We then added these mentions to the concept name's bag of word feature template. This ontology augmentation step helped capture some non-lexical matches, but this only assists in non-lexical matches found in the training set. This highlights the importance of having a robust set of synonyms present in the ontology.

7.5 Evaluation and Results

We use the Share/CLEF 2013 Task dataset described in Chapter 2.4. While we are using (almost[6]) the same dataset as the Share/CLEF 2013 task (Pradhan et al., 2013), we are considering a different task. The systems in the shared task are end-to-end concept linking systems (Pradhan et al., 2013; Savova et al., 2010; Aggarwal and Barker, 2015; D'Souza and Ng, 2015), whereas we consider a candidate generation system. The concept linking systems that were evaluated in the shared task may have included a candidate generation stage, but evaluations of these stages are not provided and we were not able to locate the code of such a system.

Instead, we compare our weighted candidate generation (abbreviated as Wgt.) approach to several representative baselines.

- **Exact Match** – Selects concepts that are an exact string match to the mention text.

[6] We do not have access to the Test set, giving us less overall data and different evaluation sets.

- **Partial match** – Scores concepts by the number of overlapping words that occur between the concept name and the mention text.

- **Char 4-gram** – Scores concepts by the number of character 4-grams that overlap between the concept name and the mention text.

- **BM25** – Scores overlap between the concept name and the mention text using BM25 (Robertson et al., 1995), a common information retrieval method.

- **DiscK Binary** – Uses the same feature set as the weighted models (noted as **DiscK Weighted**), but uses binary weights instead of those learned in training. This evaluates the effectiveness of training a model based on these features compared to using un-weighted features.

- **DiscK Combined (abbreviated as Comb.)** – Combines the ranking of the best performing model with respect to mean reciprocal rank and at lower coverage levels (DiscK-1, R=0.4) with the best performing model at higher coverage levels (Char 4-gram). Specifically, we normalized the score of each to be between 0 and 1. For DiscK-1, we performed min-max normalization on each individual candidate list, and for Char 4-gram, we divided the number of overlapping character-grams by the number present in the mention. For each candidate, we selected the max score between the two models. If only one model assigned a score to a candidate, that score was used.

Two DiscK models are reported – one that excludes character-grams and only uses a dictionary to look up abbreviation expansion (DiscK-1), and one that includes 6 character-gram features and an expanded abbreviation algorithm (DiscK-2). All systems use synonym augmentation – for each mention (or text span) in the training data that is annotated with a link to a concept in the UMLS, the mention text is added as a synonym to the set of concept names already present in UMLS. This step, which is common in clinical concept linking, allows for additional synonyms to be identified, including those that are likely to only occur within clinical text. However, this is also limited by the size and diversity of the training data. All systems were trained on both train and development sets for the final tests.

For our implementations and baselines, we report both coverage (or recall, *i.e.* the percentage of instances that the relevant concept was generated in the candidate list), and mean reciprocal rank, to measure the effectiveness of the ranking. When calculating the mean reciprocal rank, if any concepts are tied, they are randomly ordered and assigned the corresponding rank. Several regularization parameters were tried on the DiscK model - regularization controls both the weights of the model and the feature selection of the model. The least regularized DiscK-2 model (R = 0.4) uses six feature templates, while the most regularized model (R = 0.25) contains five feature templates. While the weights resulting from regularization are not relevant to the Binary models, the feature template set from the non-Binary version is used in the Binary version.

As shown in Table 7.1, although DiscK-2 is competitive, the 4-character-gram

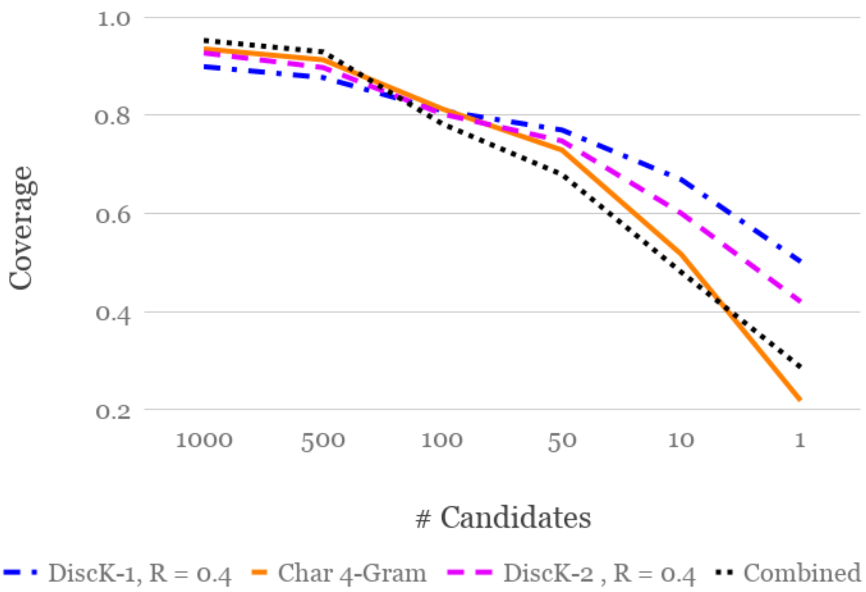

Figure 7.1: Coverage of DiscK models compared to baselines

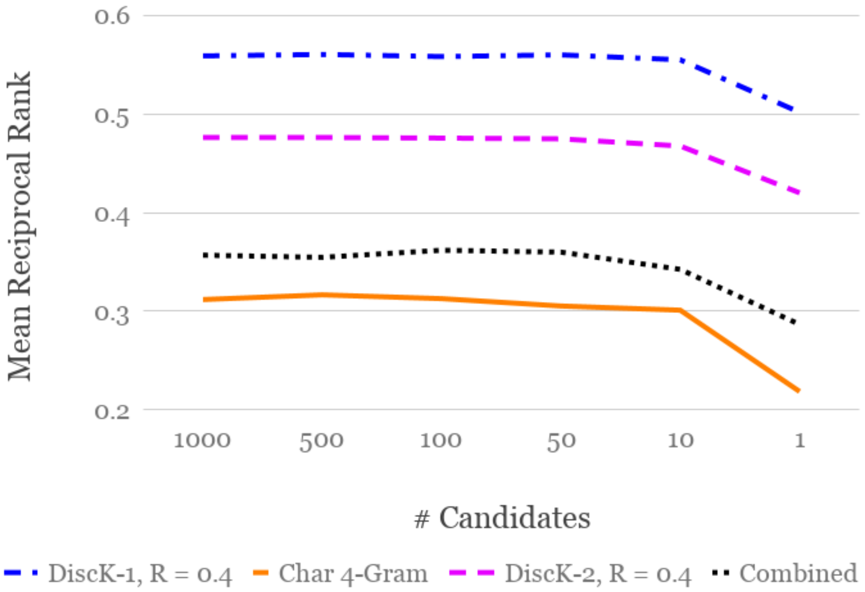

Figure 7.2: Mean Reciprocal Rank of DiscK models compared to baselines

baseline performs the best at $K = 100$, and the Combined model performs best at $K = 1000$. However, at smaller candidate sizes (e.g. 10 and 1), DiscK-1 provides the best coverage, representing a 13.9% improvement over the 4 character-gram baseline. This change is illustrated in comparing the coverage to the candidate size in Figure 7.1. For mean reciprocal rank, we find that the DiscK-1 model provides the best mean reciprocal ranking in all settings. Unlike with coverage, this improvement is clear at each candidate list size, with the best DiscK model at each size providing at least a 0.09 improvement in mean reciprocal rank compared to the binary models, and at least a 0.18 improvement over the non-DiscK baselines. This is illustrated in Figure 7.2, which compares mean reciprocal rank at different candidate sizes.

We ran feature ablation tests for the DiscK-2 model, with R = 0.25 and N = 1000, shown in Table 7.3. The most important feature templates are the Partial, Lemma, and 6 character-gram group – the coverage is reduced by half, and the mean reciprocal rank is also reduced. The ontology augmentation step (which is used in the partial matching feature template) is also an important component of the system, as its omission results in an 8.6% drop in coverage. The importance of adding terms to the ontology highlights the utility of methods described in Chapter 8.1, as adding synonyms not present in the knowledge base increases performance noticeably. While all feature templates contribute to increased coverage, the 6-character-gram omission results in increased MRR.

To determine whether using learned weights produces rankings that are distinct

	Model	$K = 1000$		$K = 100$		$K = 10$		$K = 1$	
		Cov.	MRR	Cov.	MRR	Cov.	MRR	Cov.	MRR
Baselines	Exact Match	25.2%	0.252	25.2%	0.252	25.2%	0.252	25.2%	0.252
	Partial Match	88.8%	0.301	78.2%	0.311	51.3%	0.297	21.1%	0.211
	Char 4-Gram	93.6%	0.312	**81.3%**	0.313	51.7%	0.301	21.8%	0.218
	BM25	78.9%	0.376	72.7%	0.371	53.0%	0.364	29.0%	0.290
DiscK-1	Wgt. (0.4)	90.0%	**0.559**	80.9%	0.558	**66.9%**	**0.555**	**50.2%**	**0.502**
	Wgt. (0.25)	90.0%	0.556	81.0%	**0.559**	66.4%	0.549	49.8%	0.498
	Binary (0.4)	89.2%	0.465	79.6%	0.467	59.8%	0.457	37.5%	0.375
	Binary (0.25)	89.2%	0.459	78.9%	0.464	61.4%	0.452	39.1%	0.391
	Comb. (0.4)	**95.3**%	0.357	78.3%	0.362	48.0%	0.342	28.6%	0.287
DiscK-2	Wgt. (0.4)	92.8%	0.476	80.3%	0.476	60.0%	0.468	42.0%	0.420
	Wgt. (0.25)	92.9%	0.469	80.3%	0.469	58.7%	0.459	41.3%	0.413
	Binary (0.4)	89.5%	0.448	79.0%	0.455	59.9%	0.438	37.3%	0.373
	Binary (0.25)	89.5%	0.448	79.5%	0.441	60.2%	0.450	36.4%	0.364

Table 7.1: Coverage (the percentage of instances that the relevant concept was generated) and Mean Reciprocal Rank (MRR) for DiscK and Baselines, for varying candidate list sizes K on the test data. The DiscK models are described in Chapter 7.3 and the Baseline models in Chapter 7.5.

from those produced by binary models, we used the Wilcoxon signed-rank test to compare each DiscK model with one using binary weights. We find that with the exception of DiscK-2 with R = 0.25, the p-value for the test is less than 0.01, allowing us to reject the hypothesis that the weighted versions produce the same ranking as non-weighted versions.

Model	p-value
DiscK-1, R = 0.25	2.20e-8
DiscK-1, R = 0.4	1.99e-8
DiscK-2, R = 0.25	0.077
DiscK-2, R = 0.4	0.002

Table 7.2: Wilcoxon signed-rank test comparing DiscK-1 weighted and binary rankings, using the models shown in Table 7.1. With the exception of *DiscK-2, R = 0.25*, all have a p-value < 0.01, which shows a significant difference between the weighted and binary rankings of DiscK.

Model	**Coverage**	**MRR**
Full model	92.9%	0.469
– Ontology aug.	84.3%	0.394
– Exact Match	92.1%	0.324
– Partial Match	92.5%	0.470
– Lemma	92.1%	0.468
– Partial & Lemma	91.7%	0.441
– Char 4-Gram	89.9%	0.497
– Partial, Lemma & Char 4-Gram	43.8%	0.298
– Definition	91.5%	0.432
– Abbreviation	92.2%	0.468

Table 7.3: Feature ablation results for one DiscK model (DiscK-2, R = 0.25, N = 1000) on the test set. The change in coverage and MRR is shown for the removal of each feature or set of features. The features are described in Chapter 7.3.

7.6 Discussion

While several baselines do an equivalent or better job on coverage (including the correct concept in the candidate list), DiscK consistently does a better job of assigning a higher rank to the right link (higher MRR). DiscK achieves at least a 0.09 improvement over binary DiscK in MRR, and at least a 0.18 improvement over all other baselines. Excluding the combined model, the best performing coverage at N = 1000, 4 char-gram, does exceed the best DiscK model by 0.7. However, the best DiscK model for mean reciprocal rank exceeds the 4 char-gram model by 0.18. While many baseline systems assign the relevant concept a high score, they also produce candidate lists with many ties, which reduces the usefulness of the ranking. The effect of this can also be seen in coverage with smaller candidate list sizes (e.g. 10). The discriminative ranking of the DiscK models results in relevant concepts receiving higher relative scores, resulting in better coverage for smaller list sizes.

The Combined model explores whether the benefits of the DiscK-1 model (high mean reciprocal rank) can be combined with the benefits of the Character 4-gram model (high coverage). This model provides a mean reciprocal rank that is higher than that in the Character 4-gram model, and the highest level of coverage for $K = 1000$ of any model. However, the addition of the Character 4-gram candidates increases the amount of noise in the candidate list, and thus while the coverage is competitive at higher levels, the MRR is consistently lower than the DiscK models alone.

The baseline and feature ablation results also show that lexical matching algorithms can provide a high level of coverage in generating candidate lists. With partial matching, for example, 88.8% of relevant concepts are retrieved, which is competitive with the best DiscK model. As seen in Table 7.3, the removal of partial, lemma, and character-gram matching (as they often provide similar information) reduces the coverage to only 43.8%. However, the performance of lexical matching algorithms is partially deceiving, as many lexical matches are made with augmented synonyms. Without the augmentation step, the effectiveness of DiscK model 2 drops to 84.3% coverage. Without this step, many concept links would require non-lexical transformation.

In reviewing the coverage errors for the DiscK-2 model (R = 0.25, N = 1000), which provides the highest coverage, several patterns emerged. First, 25 of the 76 errors would require non-lexical transformation to match the mention and concept. An additional 26 could achieve a partial match with some lexical transformation, but some tokens would require non-lexical transformation. The remaining 25 errors could achieve a match with the correct lexical transformation – most were not retrieved due to morphology or abbreviation. Additional non-lexical errors were avoided due to the synonym augmentation step. However, this is less useful when applying this solution to a larger dataset with a bigger vocabulary, as many non-lexical transformations may not have been seen in the training data.

7.7 Concept Linking Improvements

To demonstrate the effectiveness of our candidate generation system, we used it in conjunction with an end-to-end concept linking system. We selected DNorm (Leaman et al., 2013), a concept linking system that builds weighted TF-IDF representations of both the mention string and concepts and learns a weighted similarity measure to rank concepts. DNorm was the highest performing concept linking system in the Share/CLEF 2013 task (Pradhan et al., 2013). While DNorm is accurate, it must calculate the similarity between each mention and every concept in the knowledge base. For our dataset, the number of candidates is $n = 125,362$. To reduce the number of concepts considered, we used our candidate generation method to filter concepts evaluated by DNorm. As DiscK retrieval is a sublinear operation, generating candidates in this manner is more efficient. We now operate over candidate lists of (at most) size k,[7] instead of the full knowledge base, a significant gain in efficiency.

We use the DiscK-2 model (as it had the highest coverage at 1000),[8] and we retrained DNorm using the train, development, and tests splits noted in Chapter 6.5. In Table 7.4, we report the coverage for varying k and for varying sizes of DNorm ranked lists of size d. We do not report improvements in terms of time, as DiscK could not be directly integrated into DNorm. Table 7.5 reports mean reciprocal rank. Since our candidate generation method eliminates some correct concepts from consideration, the DNorm version with filtered concepts performs worse than if considering all candidate sizes. However, the difference in accuracy is small for larger candidate sizes – for $k = 5000$, the accuracy at $d = 1000$ is only 1.97% worse than when considering all candidates. Similarly, the mean reciprocal rank for $d = 1000$ is 0.023 points lower than when considering all candidates. For this small reduction in performance we see dramatic speedups; DNorm with a candidate list of size $k = 5000$ only considers 4% of the original candidates. For smaller levels of k, the performance in terms of coverage and mean reciprocal rank continue to decrease, but are paired with larger gains in efficiency. In more extreme cases, such as that of $k = 50$ and $d = 50$, there is a 15.7% decrease in accuracy, but only considers 0.04% of the original candidates. These large gains in efficiency are particularly attractive when performing concept linking over a large corpus, such as the electronic health records of a large hospital.

[7] The candidate lists generated by DiscK will contain at most k candidates, but may contain less if fewer matches are retrieved

[8] The character model requires pairwise comparisons, so it would not improve DNorm efficiency.

		DNorm Ranked List Size d					
		1	10	50	100	500	1000
Cand. List Size k	50	61.92%	67.78%	73.29%			
	100	64.07%	72.81%	77.84%	78.80%		
	250	69.09%	78.87%	84.13%	85.32%	86.28%	
	500	71.03%	80.21%	86.17%	87.49%	88.65%	
	1000	70.60%	81.31%	87.86%	89.17%	90.24%	90.83%
	2000	71.67%	82.26%	87.62%	90.36%	91.55%	92.38%
	5000	71.90%	81.90%	88.74%	90.95%	92.74%	93.57%
	All	**74.79%**	**83.23%**	**88.94%**	**92.03%**	**94.88%**	**95.48%**

Table 7.4: Coverage results for DNorm using varying Candidate List sizes k. Cells for DNorm's ranked list with a size larger than k are left empty.

		DNorm Ranked List Size d					
		1	10	50	100	500	1000
Cand. List Size k	50	0.619	0.643	0.643			
	100	0.641	0.674	0.676	0.676		
	250	0.690	0.724	0.726	0.727	0.726	
	500	0.710	0.740	0.740	0.744	0.743	
	1000	0.706	0.742	0.745	0.746	0.746	0.744
	2000	0.717	0.754	0.755	0.758	0.757	0.759
	5000	0.719	0.755	0.757	0.757	0.757	0.758
	All	**0.748**	**0.777**	**0.781**	**0.782**	**0.781**	**0.781**

Table 7.5: Mean Reciprocal Rank results for DNorm using varying Candidate List sizes k. Cells for DNorm's ranked list with a size larger than k are left empty.

7.8 Conclusion

For medical concept linking, using a weighted feature-based candidate generation step produces a more robust candidate list than standard triage steps. Compared to baselines, we find that DiscK produces a candidate list that has a high level of coverage but also ranks the relevant concept higher than standard methods. This approach provides improved input for a final linking method, as the DiscK candidate list better disambiguates between relevant concepts and non-relevant concepts. We find that the majority of concept links can be identified with lexical features, but identifying concepts that are not lexically similar requires additional investigation. Integration of our candidate generation step into an existing concept linking program (Leaman et al., 2013) shows that with a small reduction in accuracy, large efficiency gains can be made by replacing a complete pairwise search of the possible candidates with the sublinear DiscK candidate generation system.

For the task of triage outside of the medical domain, there has been an increased focus on using dense representations. This was first proposed by the authors of the BLINK (Wu et al., 2020) linker, described in detail in Chapter 2.2.2. BLINK creates separate representations of the mention within its sentence and each entity in the knowledge base. The triage step consists of performing a nearest neighbor search for the top n closest entities given the mention representation. This approach can also be made relatively computationally efficient by two additions. First, the authors propose to cache the representations of the entities – as they are static, they do not need to be reproduced at inference time. Second, they use an efficient nearest neighbor search algorithm, such as HNSW (Malkov and Yashunin, 2018), to achieve sub-linear performance.

While much later work in clinical concept linking does not focus on triage, some work (Sung et al., 2020) uses a similar dense representation method for candidate generation. In comparing this approach to DiscK, a contextualized language model will likely produce a more robust representation of the mention and entity than permitted by the restricted feature set. This suggests that a dense triage approach, such as in BLINK, is likely to be more accurate. However, it is also likely to be far less computationally efficient, as contextualized representations for each mention need to be created at inference. This computational inefficiency is compounded if a second contextualized representation needs to be created for the final reranker, as is the case with the BLINK reranker.

Chapter 8

Unsupervised Discovery of Synonyms for Clinical Concepts

8.1 Introduction

Given a word or phrase, the task of synonym discovery identifies other words or phrases that have the same or similar meaning to the original.[1] Constructing lists of synonyms can be helpful in a range of downstream applications, such as linking concepts to a knowledge base (Mihalcea and Csomai, 2007) or query expansion in information retrieval. Both tasks rely on an expanded list of synonyms to ensure that relevant concepts or documents are retrieved even if they do not contain the query term. Synonym discovery, and the related task of paraphrase identification (Bannard and Callison-Burch, 2005; Ganitkevitch et al., 2013; Sekine, 2005), have been explored using a variety of methods (Grefenstette, 2012; Hagiwara, 2008; Lindén and Piitulainen, 2004; Leeuwenberg et al., 2016). This task builds on work in measuring semantic similarity between words and phrases (Mihalcea et al., 2006; Resnik, 1995).

Synonym discovery is especially important within the clinical medical domain (Pedersen et al., 2007; McCrae and Collier, 2008; Wang et al., 2015a). Medical synonyms aid in a variety of clinical tasks, such as automatic phenotyping and cohort selection for comparative effectiveness research (Voorhees and Hersh, 2012). While ontologies contain synonyms for a specific term, these often do not contain the variety of synonyms that can occur in clinical notes across different authors and different domains. Alternatively, new or rare terms may not be present in a standardized

[1] Elliot Schumacher and Mark Dredze. 2019. Learning unsupervised contextual representations for medical synonym discovery. In *JAMIA Open, 2019.*

ontology. An additional challenge is that a fixed synonym list may not accurately reflect meaning, as abbreviations and shortened references have meanings that are contextually dependent. The impact of this is shown in both Sections 6 and 7, where the presence of synonymous terms in the knowledge base has a clear effect on the performance of the systems. For these reasons, we are interested in methods that can automatically identify new synonyms from clinical notes without supervision.

In this work, we consider the task of identifying whether two textual mentions of a disorder refer to the same underlying medical concept. While synonym discovery is critical within clinical NLP, the task is especially challenging for disorder mentions in the clinical domain. First, while typical synonyms tend to be lexically dissimilar, medical concepts are both lexically similar ("dilated RA" and "dilated RV") or dissimilar ("cerebrovascular accident" and "stroke"). A solution to this problem is to use representations of terms that move away from using the lexical items themselves. Along these lines, work by Wang et al. (2015a) proposed word embeddings for this task, and found that Word2vec representations improved over the previous best approach. However, type-level representations cannot address the second challenge: synonym determination is often contextual. For example, the two terms "diabetes type 2" and "diabetes" can be synonymous in that they refer to the same underlying concept, or they could refer to two different types of diabetes. The key distinguishing factor is the context of how the terms appear in the clinical note. Context can come from both the surrounding text, as well as information about the patient. Without this context, a method cannot distinguish when two disorder mentions are synonymous. Both of these challenges are similar to those faced by entity linking (see Chapter 2.1.4) when matching mentions of entities to standardized knowledge base terms.

We propose learning representations of clinical text for unsupervised synonym discovery of disorder mentions using contextualized representations. Rather than build type-level embeddings as in previous work (Wang et al., 2015a), we build on work in learning contextualized text representations (Melamud et al., 2016; Peters et al., 2018), and discussed in Chapter 2.2.2. These methods incorporate the mention context into a representation of the mention. Additionally, we augment the context from learned representations of the patient (Choi et al., 2016). Incorporating context from the patient record can indicate that certain concepts are more or less likely for a mention. Additionally, our methods are fully unsupervised, in contrast to the previous work that used supervision (Wang et al., 2015a). We greatly prefer unsupervised methods as they can scale to a large number of medical subdomains as medical annotations are particularly expensive.

We evaluate our proposed method on the task of finding disorder synonyms (Pradhan et al., 2014) from English clinical free text, where we define mentions as synonymous if they both refer to the same medical concept. We consider baselines of both unlearned representations (character ngrams), and learned non-contextualized word embeddings (Word2vec). We show improvements on the dataset released for ShARe/CLEF eHealth Evaluation Lab 2013 Task 1b (Pradhan et al., 2013), which includes span-level annotations for disorder concepts built on a subset of MIMIC 2.5 clinical notes. We find that both text context and patient context improve over previously established baselines, yielding significant improvements in the state of the art. Finally, we find that our methods identify synonyms that are more lexically dissimilar than Word2vec.

8.2 Synonym Discovery

We consider the task of finding disorder synonyms (Pradhan et al., 2014) in English clinical free text. We define mentions as synonymous if they both refer to the same medical concept. To obtain annotations of this task, we identify mentions that link to the same medical concept in the dataset released for ShARe/CLEF eHealth Evaluation Lab 2013 Task 1b (Pradhan et al., 2013), which includes span-level annotations for disorder concepts built on a subset of MIMIC 2.5 clinical notes. For training representations, we use MIMIC III (Johnson et al., 2016) which is a superset of MIMIC 2.5.

Consider the sentence “The patient showed signs of a *stroke*” and the sentence “The patient’s father previously had a *cerebrovascular accident* at the same age.” The annotated dataset links both mentions (italicized) to the same ontology concept *Cerebrovascular accident*. From this, we derive that these mentions are synonymous. Note that while this task is similar to concept or entity linking, the goal is to identify synonyms present in the unstructured corpus, not to link those to a knowledge base. Therefore, we only use the concept's unique identifier from the knowledge base to score whether a synonym is correct, and do not use additional information.

The contextual information of the mention is vital in correctly identifying the right synonym. For example, the mention *stroke* could also refer to the concept *Heat Stroke*, and further information is needed to identify the correct synonym. The surrounding text of the mention is one key source of information. Considering the previous example, the presence of the term *hemorrhage* would likely indicate the mention is synonymous with *Cerebrovascular accident* and not *Heat Stroke.* Additionally, patient information is often a relevant indicator. If the patient is an infant, *Heat Stroke* may be more likely than *Cerebrovascular accident* due to the higher likelihood of *Heat Stroke* in that population. Our goal is to learn representations that capture the synonymous relationship between the mention and concept by incorporating the context.

We formalize synonym discovery as follows. For a dataset of N medical records, each record d corresponds to patient p and contains zero or more highlighted textual mentions m linked to medical concept c. We then learn a representation of each m in the corpus. To construct the candidate synonym list, we consider each mention as a query m_q and rank all other mentions as candidate synonyms m_c based on cosine similarity, and consider the top $k = 50$ mentions to be candidate synonyms to m_q. We measure the effectiveness of our approach by ranked list quality, where a correct synonym is one where m_q and m_c share medical concept c.

Previous work (Wang et al., 2015a) investigated creating type-level representations for synonym discovery using Word2vec continuous bag of words model (CBOW) (Mikolov et al., 2013b). They utilized a semi-supervised variant of this method for synonym discovery and found it to be the best-performing model they considered. However, this method does not consider context and does not yield representations specific to individual tokens. We consider several methods that capture the context in which the word occurs, and that may be more likely to identify synonyms with divergent lexical forms. Additionally, we integrate learned representations of patient data, specifically diagnosis codes, that can provide additional context. To focus on synonym discovery we assume gold mention spans.

8.2.1 Patient Medical Context

As described in Chapter 2.2.2, Context2vec, ELMo, and BERT are all used to create contextualized representations of text. In addition to modeling the context, ELMo and BERT use character or subword embeddings, respectively, to alleviate the out-of-vocabulary issues that other representation methods may face. In this work, we focus on the ELMo language model to create contextualized representations of the mentions present in the corpus. For mentions containing multiple words, we explore using both the dimensional average and maximum to create a single mention representation.

Additionally, clinical medical records contain extensive structured data. ICD-9

(International Classification of Diseases) diagnostic codes are commonly used to represent a patient since they indicate symptoms and diagnosed conditions. Our corpus is a large de-identified medical records dataset. It contains one or more text records for a single hospital admission for a patient, and a set of ICD-9 codes that applies to the admission overall.[2] We utilize the Med2vec (Choi et al., 2016) toolkit in order to learn representations of patient codes. For each admission, the assigned codes are converted into a binary vector representing all present codes. The binary vector representation is fed into a second hidden layer, which is concatenated with a vector containing demographic information about the patient. This in turn is fed into a final output layer, which is trained to predict neighboring visits using a skip-gram architecture. We use the provided author's code to train it on patient data taken from MIMIC III.

We integrate Med2vec as patient-level context into both the Context2vec and ELMo models, in the hope that providing additional patient-level context will improve the representations of the words. We omit integration with BERT as it performs worse than ELMo in our tuning set (see Table 1). We integrate Med2vec with Context2vec by concatenating a single vector of averaged ICD-9 code embeddings for the linked hospital admission (noted as v) to the final states of both LSTMs, passing the combined vector as input into the final multi-layer perceptron (MLP) in the model. The input to the MLP becomes

[2] As noted on MIMIC's website, all ICD codes in MIMIC III are in ICD-9 format, and these codes will be switched to ICD-10 in later releases. Our model is not specific to the ICD-9 code format - a representation for ICD-10 or ICD-11 codes could be learned in a similar way.

$$\mathrm{lLS}(l_{1:i-1}) \oplus \mathrm{rLS}(r_{n:i+1}) \oplus \mathrm{ICD9}(v) \tag{8.1}$$

where lLS and rLS are left to right and right to left LSTM embeddings of the sentence up to the target word, respectively. By combining the ICD-9 context with the sentence context as input to a MLP, the model may learn patient-level context that informs the word-level representations created by Context2vec. The architecture is shown in Figure 1.

We explore two approaches to integrating Med2vec into ELMo: in the input layer and the output layer.[3] In both cases, we create a single ICD-9 representation by calculating the dimensional max over the Med2vec representations of each ICD-9 code, as there are multiple ICD-9 codes assigned to each note. First, we concatenate a matrix of ICD-9 codes to the token representation layer as input to the LSTM. Each position in the token representation layer $h^{LM}_{k,0}$ becomes $h^{LM}_{k,0} \oplus \mathrm{ICD9}(v)$.

Second, we add ICD-9 representations to the output layer as additional input to the softmax which predicts the preceding or future tokens. Previous work on neural language modeling (Hoang et al., 2016) has shown that integrating additional information into the language model can lower perplexity, specifically when added to the output layer. While our goal isn't to improve language model perplexity, added information may similarly inform our task. Instead of the softmax input being the

[3] We also experimented with multi-task training, but initial results did not show an improvement.

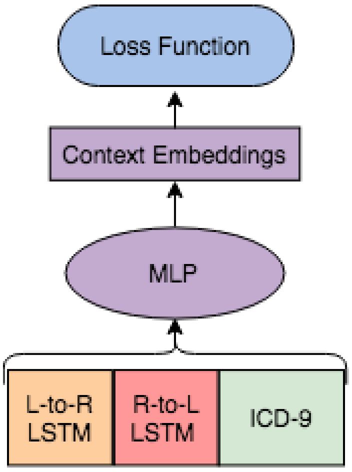

Figure 8.1: Context2vec with Med2vec

concatenated second layers from the forward and backward LSTMs, $h_{k,2}^{LM} = \overrightarrow{h_{k,2}^{LM}} \oplus \overleftarrow{h_{k,2}^{LM}}$, we add the ICD-9 codes to yield

$$h_{k,2}^{LM} = \overrightarrow{h_{k,2}^{LM}} \oplus \overleftarrow{h_{k,2}^{LM}} \oplus \mathrm{ICD9}(v).$$

The ICD-9 code matrix consists of a single ICD9 representation for each word (taken from the ICD9 codes linked to the clinical note). For the input version, the ICD9 representations are separately input into the respective LSTMs, while in the output only one matrix representation is included. In all cases, a single ICD-9 representation is created by calculating the dimensional max operation over the representations of the ICD-9 codes assigned to the admission. ELMo with Med2vec Input is illustrated in Figure 8.2 and ELMo with Med2vec Output is illustrated in Figure 8.3.

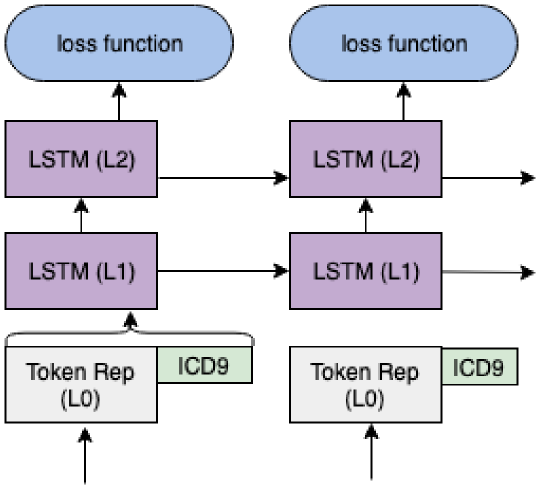

Figure 8.2: ELMo with Med2Vec Input

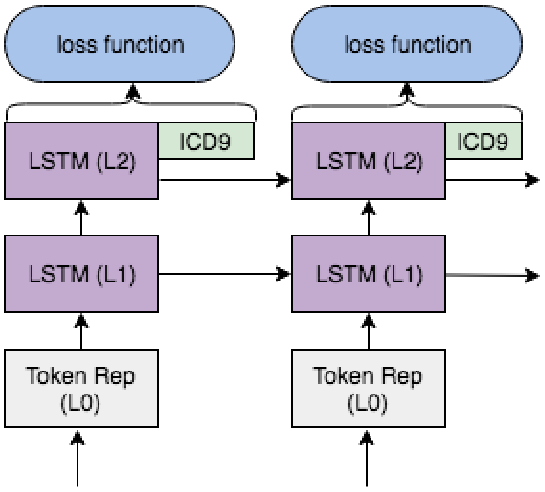

Figure 8.3: ELMo with Med2vec Output

8.3 Data

We use two datasets: MIMIC III (Johnson et al., 2016) to train our representations, and the concept linking dataset released for ShARe/CLEF eHealth Evaluation Lab 2013 Task 1b (Pradhan et al., 2013) to evaluate discovered synonyms. At the time of this research, this dataset was the only relevant English dataset publicly available, and further details are included in Chapter 2.4. The shared task dataset consists of span-level annotations for disorder concepts built on a subset of MIMIC 2.5 clinical notes (Saeed et al., 2011) (which is a subset of the current version, MIMIC III (Johnson et al., 2016)). Since we evaluate an unsupervised method, we use a “tuning” set as it is only used to tune model hyperparameters, while the test set is for evaluating synonyms. The “training” dataset is only used for training the representations and consists of unannotated clinical notes.

We train our representations on a subset of MIMIC III. As the annotated clinical notes are in the dataset, we excluded any patients that had an annotated clinical note from our representation tuning data. We used 213,466 clinical notes and associated admissions-level diagnoses data for training representations. For Med2vec, similarity is reported using the admission-level ICD-9 code representation and does not use any of the clinical note text. We use an embedding dimensionality of 600 for Context2vec, and 200 for Med2vec, which was chosen based on the dimensionality used in the Context2vec and Med2vec papers, respectively. The Context2vec + Med2vec reported model was produced after 6 epochs of training, and we used the default setting for all other model parameters as noted in their respective papers. For ELMo, we used the standard parameters noted in the original paper - we trained for 10 epochs, with a dimensionality of 512 for each LSTM. For BERT, we used the pretrained model provided by Alsentzer et al. (2019), trained on a variety of clinical notes in MIMIC (and discussed in Chapter 2.3).

8.4 Evaluation

8.4.1 Baselines

We include two baselines drawn from the work of Wang et al. (2015a) to serve as the state-of-the-art methods for learned and unlearned representations. We do not evaluate their supervised model as we consider the unsupervised setting.

8.4.1.0.1 Character ngrams

We calculate the number of n-length sequences of characters that appear in both the mention string and the candidate synonym. Each word is padded with unique start and end characters. Each mention-candidate pair is assigned a score equal to $(|\text{ngram}_c \cap \text{ngram}_q|)/(|\text{ngram}_q|)$, where ngram_q is the set of ngrams from the query mention and ngram_c is the set of ngrams from the candidate synonym. This score is used to create a ranking of synonyms, as with all other models.

8.4.1.0.2 Word2vec

Word2vec performed best in the unsupervised setting in previous work (Wang et al., 2015a). We use Word2vec (Mikolov et al., 2013b) to learn representations of mention strings. We used the gensim toolkit (Řehůřek and Sojka, 2010). We use negative sampling combined with the skip-gram training algorithm, which are the best performing parameters from previous work (Wang et al., 2015a). For mentions containing multiple words, the dimensional average of all words is used as a single mention representation. All out-of-vocabulary words are excluded from the final representation, although this situation was rare.

8.4.2 Evaluation Metrics

For a single mention, we calculate the cosine similarity between that mention's representation and all other mention representations, creating a ranking. From this we take the top 50 mentions as potential synonym candidates and calculate mean reciprocal rank (MRR) and coverage (*i.e.* recall). We exclude mention pairs that are equivalent strings as they are the same terms, though identical candidate mention strings that occur multiple times in the corpus may appear more than once in the ranking. We only evaluate query mentions that had a synonym present in the data, i.e. another (different) mention string linked to the same concept as those without synonymous mentions present cannot be matched in this dataset. We only include candidates from within the same data fold.

We scored a synonym as correct if both that mention and the query linked to the same concept in the gold annotation. We measured mean reciprocal rank (MRR), coverage (percentage of time a correct synonym appeared in the ranked list), and top-1 accuracy. For the top-1 results in each model, we calculate the Jaro-Winkler distance (Winkler, 1990), measuring lexical similarity, between the mention text and the synonym. The mean Jaro-Winkler distance between a mention and all of its gold label synonyms was 0.504 for the tuning and 0.476 for the test sets. Both the tuning and test sets were not used for training representations, but only the test set was held out until the end of experimentation.

8.5 Results

Table 8.1 shows the results for candidate models on the tuning data. We selected the model that produced the highest MRR for each model type (noted by separators in the Table) and evaluated it on the test set (Table 8.2). In all cases, ELMo models outperform the Word2vec, character ngram, Context2vec, and BERT models. For MRR, the ELMo models outperform others by 0.09 in tuning and 0.12 in testing. For Top-1, ELMo provides an 11.3% increase in tuning and a 13.9% increase in testing. For coverage, ELMo provides smaller but noticeable improvements – a 4.3% increase for tuning and a 4.1% increase for testing. Adding Med2vec information to the ELMo

			Tuning (n=1275)		
	Model		**MRR**	**Cov.**	**Top-1**
Baselines	***Word2vec***		0.373	71.9%	30.9%
	Char. Bigram		0.404	70.6%	33.7%
	Char. Trigram		0.417	69.8%	33.7%
	Char. Fourgram		0.414	68.8%	34.0%
Contextual Models	Context2vec		0.374	58.1%	31.8%
	C2v + m2v		0.385	63.9%	28.4%
	ELMo	L0, Avg	0.499	71.9%	43.7%
		L0, Max	0.503	67.8%	**45.3%**
		L1, Avg	0.370	71.5%	28.5%
		L1, Max	0.344	63.3%	27.4%
		L2, Avg	0.299	71.1%	21.9%
		L2, Max	0.291	63.7%	22.2%
	ELMo + M2v In	***L0, Avg***	**0.504**	**76.2%**	44.5%
		L0, Max	0.488	69.8%	43.5%
		L1, Avg	0.346	72.2%	26.5%
		L1,Max	0.336	65.5%	26.9%
		L2, Avg	0.279	70.0%	19.8%
		L2, Max	0.281	63.0%	21.1%
	ELMo + M2v Out	L0, Avg	0.484	72.2%	42.6%
		L0, Max	0.493	69.6%	44.9%
		L1, Avg	0.371	73.3%	29.1%
		L1,Max	0.351	66.8%	27.7%
		L2, Avg	0.309	71.1%	23.2%
		L2, Max	0.302	64.5%	23.1%
	BERT	***L1, Avg***	0.491	65.2%	43.9%
		L1, Max	0.489	65.0%	44.3%
		L4, Avg	0.438	62.0%	39.4%
		L4, Max	0.435	63.3%	38.4%
		L8, Avg	0.340	53.9%	30.9%
		L8, Max	0.395	51.7%	35.5%
		L12, Avg	0.324	47.2%	29.3%
		L12, Max	0.372	50.7%	33.3%

Table 8.1: Mean reciprocal rank, coverage, and top-1 accuracy, for pairwise identification of synonyms of the top 50 results run on the tuning data (n=1275). For ELMo and BERT models, L(0/1/2) indicates layer number, and Avg or Max indicates combination method for multiple word phrases. Bolded entries are the best-performing result for that measure. We report test results on the model names listed in bolded italics, selecting the best model for MRR in each category (noted by line separators)

	Test (n=599)			
Model	**MRR**	**Cov.**	**Top-1**	**JW T-1**
Word2vec	0.355	69.4%	29.2%	0.798
Char. Trigram	0.359	67.9%	28.0%	0.826
C2v + M2v	0.335	60.6%	28.6%	0.719
ELMo (L0,Max)	0.474	62.4%	43.1%	0.838
ELMo+M2v In (L0,Avg)	0.476	**73.5%**	40.7%	0.813
ELMo+M2v Out (L0,Max)	**0.487**	63.4%	**44.7%**	0.814
BERT (L1, Avg)	0.442	64.9%	39.1%	0.835

Table 8.2: Mean reciprocal rank, coverage, and top-1 accuracy, and Jaro-Winkler average for correct synonyms in the top-1 for pairwise identification of synonyms of the top 50 results run on the test data (n=599). Significance tests were performed using a two-sided Z-score test to compare the best-performing models (bolded) to the baseline models.

model does not provide consistent improvements to any metric, with the exception of coverage. The ELMo model with Med2vec integration provides a small increase in coverage over the standard ELMo model and the other model types. The Jaro-Winkler distance of the ELMo model varies by layer level – the lowest layer has the most lexically similar synonyms, while the higher layers have the least lexically similar synonyms of any model. Overall, we see clear benefits by moving from the type level embeddings to contextualized representations, with some benefits to incorporating patient context.

8.5.1 Synonym Analysis

In addition to the quantitative results, we perform a qualitative analysis of one model, ELMo (L0, Max). For 400 Top-1 errors (the mention was not matched with a synonym as the first result, but may be matched lower in the list), we categorized the

Ex. Mention	**Top-1 Synonym**
varicosities	varices
left atrial enlargement	LA enlargement
difficulty ... breathing	shortness of breath
hypokinesis	hypokinetic
decreased responsiveness	poorly responsive
uterine fibroid	fibroid
mitral regurgitation	mitral regurg
rib fx	fractures ... rib
septic	sepsis ... rib

Table 8.3: Correct Top-1 Examples from the ELMo (L0, Max) Model tuning set results.

Category	**Perc.**	**Example Mention**	**Top-1 Synonym**
Synonym Overlap	52%	left atrium ... dialated Myocardial infarction diabetes mellitus aortic valve disease dilated RA	right atrium ... dialated inferior myocardial infarction diabetes millitus type 2 valvular heart disease dilated RV
Abbreviation	19%	AR UTI	MR ptx
Morph. or Lexical Overlap	16%	hypokinesis bradycardic cyanosis	akinesis tachycardic stenosis
No Relation	9%	nausea clubbing	masses bleeding
Sim. Con.	5%	bleed	bleeding

Table 8.4: Incorrect Top-1 Examples from the ELMo (L0, Max) Model tuning set results.

error as one of five types. Correct and incorrect examples with error types are listed in Tables 8.3 and 8.4 respectively. The first and most common error type was word overlap - the mention and the incorrect synonym shared at least one word, but the remaining non-shared words contrasted the meaning of the mention and synonym. This may be due to the simple method we use to combine words into a single representation (in this case, the dimensional maximum operation). For example, for the mention "diabetes mellitus" the top synonym is "diabetes mellitus type 2" – the two share words and may be linked to related concepts, but the model does not put enough weight on the distinction provided by the words "type 2". Second, mentions with abbreviations were commonly mismatched with other abbreviations. Some abbreviations are linked to concepts that are related (e.g. "AR" is an abbreviation for Aortic Valve Insufficiency and "MR" is an abbreviation for Mitral Valve Insufficiency), while others share no relation (e.g. "UTI" is an abbreviation for Urinary Tract Infection and "ptx" is an abbreviation for Pneumothorax). The third class of error was morphological or partial lexical overlap - the mention and incorrect synonyms do not share a word, but often shared a prefix or suffix (e.g. "hypokinesis" and "akinesis" share the suffix kinesis). Fourth, some errors consisted of mention and incorrect synonym pairs that were not correct due to annotation decisions in the data - they often have the same lexical form but are different concepts in the ontology referenced in the annotations, or may have been assigned a non-concept annotation. Finally, we could not explain some errors as there was no clear relation between the mention and the incorrect synonym. In all,

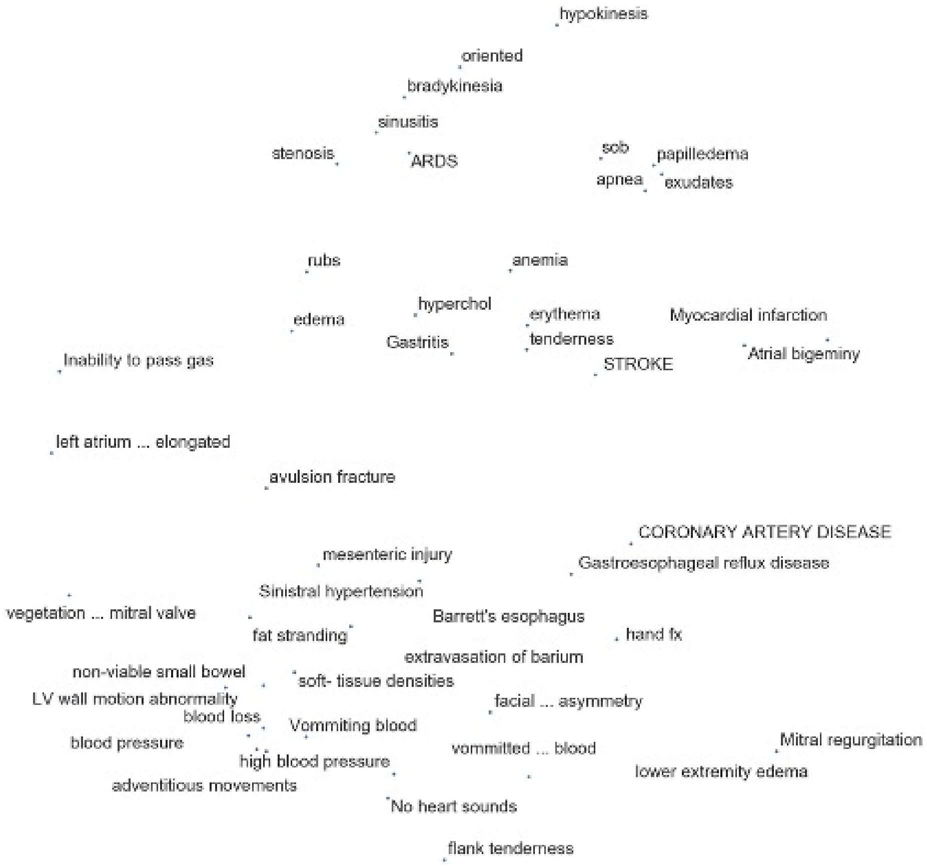

Figure 8.4: We performed dimensionality reduction using t-SNE on the tuning set mention representations from the ELMo (L0, Max) Model, randomly selected 5% of unique mention strings.

these errors are similar to those present in work in concept linking in Chapter 6. We visualize selected synonyms from the tuning set in Figure 8.4 using t-SNE (Maaten and Hinton, 2008).

8.5.2 Background

Previous research has studied identifying medical synonyms from within the UMLS ontology using unsupervised representations, such as Wang et. al. (Wang et al., 2015a) using a method centered on Word2vec's CBOW method. Other work (Henriksson et al., 2014) uses Random Indexing and Random permutation to identify synonyms in clinical notes and journal article data. Unlike Wang et. al., this is an unsupervised method that uses a ranking approach but is limited by its reliance on term statistics instead of character-based representations. Related work explored applying a similar method to Japanese patient blogs (Ahltorp et al., 2016). Earlier work (McCrae and Collier, 2008) explored retrieving synonyms for biomedical text in UMLS and other ontologies using a pattern generation algorithm. While benefiting from interpretability, this does not allow for the integration of character or contextual models that our work provides. Additional work has studied approaches to synonym expansion in non-medical domains (Leeuwenberg et al., 2016; Gupta et al., 2015), and the related tasks of addressed abbreviation and acronym resolution (Kirchhoff and Turner, 2016; Finley et al., 2016) in the clinical space.

8.6 Discussion and Conclusion

Models using ELMo consistently provide the best performance for using unsupervised representations for the pairwise mention synonym identification task – in MRR, Coverage, and Top-1 accuracy. We attribute this to two factors. First, the ELMo model allows the sentence surrounding the mention to influence the final representation, which better incorporates the context in which the mention occurs.

Mentions of concepts that do not share a similar lexical form may appear in similar contexts with similar words, and including the sentence allows for this to be reflected in the final representation.

Second, using a character model may better handle out of vocabulary words and morphology. Integration of Med2vec into ELMo provides an improvement in coverage, which indicates that integrating patient information can better inform representation learning for this task. To explore this further, we trained an ELMo model that used tokens instead of characters, and an ELMo model that didn't use the full sentence to build representations. In both cases, the performance was worse than the standard ELMo models, and further, it wasn't clear which is the more important factor.

While recent work has shown that BERT performs well on a variety of clinical tasks (Alsentzer et al., 2019), we find that it performs slightly worse than ELMo for this task. In the general task of synonym identification within the medical domain, other work has also shown that a BERT-based method performs worse than other proposed methods. In the case of Yang et al. (2021), the authors show that a knowledge graph-based method performs better than BERT, suggesting that BERT alone may not be sufficient for this task. Other work has shown that fine-tuning is vital for BERT performance (Peters et al., 2019). This is one potential factor in the lower performance of this model. Since we assume an unsupervised setting, we cannot conduct task specific fine-tuning, which might alleviate this performance gap.

Other research that cited this work explores how to use ICD codes with contextualized representations (Chen et al., 2021). However, more work has focused on using contextualized representations directly in a task, such as entity linking (see Chapter 2.2) or concept linking (see Chapter 6). With this approach, a model such as BERT should create related representations for synonymous terms, especially if trained on an in-domain corpus. However, for a model like DiscK (discussed in Chapter 7), this approach is still useful. In settings where using a contextualized model, such as triage, is too expensive, this synonym identification approach is still highly applicable.

Chapter 9

Conclusion

9.1 Summary

This dissertation contains several research findings concerning the task of entity linking. In Chapter 3, we show that the cross-language representation ability of mBERT extends the utility of single-language annotations to multiple languages, either in a linker trained to use multiple languages, or one trained on a single language and applied to several unseen. In this zero-shot setting, the remaining loss is due to the model's inability to learn granular distinctions within the knowledge base, as opposed to failures in cross-language capacity. This can be rectified with popularity information which would have been implicitly learned with training data but can be calculated separately. Similarly, in Chapter 4, the cross-language representation ability of a later contextualized language model, XLM-R, allows for a linker trained on English documents and knowledge bases to be applied to documents and knowledge bases in other languages. Unlike in the cross-language setting, the multi-language entity linker does improve performance when we improve the language representation ability of the model. However, this comes from forcing the linker to be less language-specific with unannotated in-language text, rather than making the linker more language-specific.

Chapter 5 highlights the benefits and challenges of mainstream large-data approaches to entity linking. These linkers work very well on mentions of entities that are present in the training data but struggle with more complex matches. This is especially true of GENRE, which relies solely on the entity name to disambiguate between entities. Adding information from the knowledge base helps in some settings for in-domain data. However, its effect is more helpful when applied to other knowledge bases, where it improves prediction in more lexically-complex matches.

Chapter 6 shows the difficulty in translating advances in standard entity linking to domains with different data. In applying another high-performing entity linker to a set of technical linking datasets, including chemical and clinical data, we see that the linking performance is relatively poor. This highlights the importance of research that looks at domain-specific issues within linking and NLP more broadly. While some advances in broader linking, such as adapting the use of contextualized representations to clinical text, can be translated, others cannot. Specifically, the lack of descriptions for many entries in the knowledge base makes the use of available synonyms all the more important.

Finally, Sections 7 and 8 build upon that finding, to show two supporting systems for clinical linking. Chapter 7 shows the usefulness of an efficient triage system that goes beyond simple lexical matching but also highlights the usefulness of synonyms within the knowledge base. Chapter 8 works to identify potential synonyms from unannotated corpora and also highlights that contextualized representations of text are central to clinical NLP. In all cases, it is critical to understand the different characteristics of each task. For example, fine-grained distinctions between parent and child concepts are more likely to occur in clinical data compared to other domains.

9.2 Conclusions

Across all of these research projects, there are several clear conclusions. First, a linker's ability to model the similarity between mentions of entities and entities' names is by far the most crucial component. Examples of this can be seen in all of the linkers discussed in this thesis. Further, much of the recent work in the field illustrates this. The GENRE linker (see Chapter 5) essentially is a mention-name matcher, going to the extreme of not using any other information from the knowledge base. With a high-capacity deep learning model, a linker can essentially memorize the various ways entities are mentioned within the text, removing the need for other sources of disambiguation. In addition, cross-language representations can enable this relationship to be modeled in multiple languages by only learning from annotations in one.

An important caveat, however, is the fact that the ability to model this relationship does not transfer to new domains as easily as to new languages, especially for cases where more challenging lexical matches are required. Leveraging synonyms, either in inference or training (Chapter 6) is one way to address this. But even with a system to automatically identify synonymous terms (Chapter 8), it is still most effective, in terms of system accuracy, to use human annotations to gather this information. Using pretrained models does partially solve this problem. But the lack of in-domain corpora is one hurdle, in addition to having to retrain models from scratch. This means that previously trained related models, such as GENRE or BLINK, cannot be simply reused.

Overall, it is a challenge to characterize the relationships between linking in various domains. Cross-language representations such as mBERT have bridged the gap between cross-language and mono-language linking to a great extent, but this is not true elsewhere. For instance, in both clinical concept linking and entity linking, the core of the task is the same. The mention string needs to be matched to the relevant entity, with context from the document and metadata from the knowledge base supporting this effort.

But the knowledge base metadata is different in each case, from type systems to what descriptive text is available. In the document, mentions are often compositional in the clinical setting (*e.g. 2019-nCoV Vaccine mRNA-1273*), whereas this is rarer in Wikipedia entity linking. Therefore, it is challenging to decide whether this is simply a difference in domains, or if it is so different as to represent a separate task. In an ideal world, it would be more efficient to have a reusable foundation for linking in any domain, but that remains a challenge. Current trends in the field illustrate the ongoing division in approaches. For example, work in the task of biomedical entity linking (see Chapter 2.3) continues concurrently with that in standard entity linking, despite their similarities. This results in siloed approaches and rebuilding systems from scratch for every new data set and knowledge base pair. Perhaps a model that allows for components to be trained modularly – *e.g.* with an in-domain name matcher – is a path forward to removing redundancies.

Finally, future work in entity linking should focus on truly challenging datasets and domains. A major trend within both entity linking and NLP more broadly is to train systems with complex architectures (*e.g.* transformers) with large amounts of data. This pushes the datasets considered to those with large amounts of data, such as Wikipedia. Building systems off of Wikipedia is perfectly reasonable, due to the availability and amount of data, and the presence of many real-world entities are present in the knowledge base. However, many of these datasets contain many exact matches and do not focus on either rarer entities not seen in the training data, or more challenging lexical matches. Further, design decisions, such as only evaluating mentions that can be lexically matched to the entity title due to a triage step, inherently bias the distribution of the examples. And while some work proposes to tackle domain-transfer issues, the datasets used are relatively easy (*e.g.* is newswire a new domain for a Wikipedia-trained linker?). There is some work (Orr et al., 2020; Logeswaran et al., 2019) that targets more challenging sets of examples, but many state of the art linkers focus on too easy datasets. At a minimum, this should be reported in discussions of datasets. At best, the movement towards tackling challenging aspects of the task will help complete the original goal – linking unstructured text to structured data – in more and more settings.

9.3 Code Releases

The following code repositories were released as a result of the work done in this thesis.

- Cross-language entity linking
  - https://github.com/elliotschu/crosslingual-el
- Clinical Concept Linking
  - https://github.com/elliotschu/clinical-concept-linking

Bibliography

Nitish Aggarwal and Ken Barker (2015). "Medical Concept Resolution." *International Semantic Web Conference (Posters & Demos)* (cited on pages 48, 163, 170).

Armen Aghajanyan, Bernie Huang, Candace Ross, Vladimir Karpukhin, Hu Xu, Naman Goyal, Dmytro Okhonko, Mandar Joshi, Gargi Ghosh, Mike Lewis, and Luke Zettlemoyer (2022). *CM3: A Causal Masked Multimodal Model of the Internet.* DOI: 10.48550/ARXIV.2201.07520. URL: https://arxiv.org/abs/2201.07520 (cited on page 138).

Magnus Ahltorp, Maria Skeppstedt, Shiho Kitajima, Aron Henriksson, Rafal Rzepka, and Kenji Araki (2016). "Expansion of medical vocabularies using distributional semantics on Japanese patient blogs". *Journal of biomedical semantics* 7.1, page 58 (cited on page 203).

Zeynep Akkalyoncu Yilmaz, Shengjin Wang, Wei Yang, Haotian Zhang, and Jimmy Lin (Nov. 2019). "Applying BERT to Document Retrieval with Birch". *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*

(EMNLP-IJCNLP): System Demonstrations. Hong Kong, China: Association for Computational Linguistics, pages 19–24. DOI: 10.18653/v1/D19-3004. URL: https://aclanthology.org/D19-3004 (cited on page 33).

Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott (June 2019). “Publicly Available Clinical BERT Embeddings”. *Proceedings of the 2nd Clinical Natural Language Processing Workshop*. Minneapolis, Minnesota, USA: Association for Computational Linguistics, pages 72–78. DOI: 10.18653/v1/W19-1909. URL: https://aclanthology.org/W19-1909 (cited on pages 51, 195, 204).

Alan R Aronson (2001). “Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program.” *Proceedings of the AMIA Symposium*. American Medical Informatics Association, page 17 (cited on pages 15, 20, 21, 48, 162).

Alan R Aronson and François-Michel Lang (2010). “An overview of MetaMap: historical perspective and recent advances”. *Journal of the American Medical Informatics Association* 17.3, pages 229–236 (cited on pages 46, 48).

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives (2007). “Dbpedia: A nucleus for a web of open data”. *The semantic web*. Springer, pages 722–735 (cited on pages 8, 16).

Michael Bada, Miriam Eckert, Donald Evans, Kristin Garcia, Krista Shipley, Dmitry Sitnikov, William A Baumgartner, K Bretonnel Cohen, Karin Verspoor, Judith

A Blake, et al. (2012). “Concept annotation in the CRAFT corpus”. *BMC bioinformatics* 13.1, pages 1–20 (cited on page 62).

Colin Bannard and Chris Callison-Burch (2005). “Paraphrasing with bilingual parallel corpora”. *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, pages 597–604 (cited on page 184).

Iz Beltagy, Matthew E Peters, and Arman Cohan (2020). “Longformer: The long-document transformer”. *arXiv preprint arXiv:2004.05150* (cited on page 33).

Michele Bevilacqua, Giuseppe Ottaviano, Patrick Lewis, Wen tau Yih, Sebastian Riedel, and Fabio Petroni (2022). “Autoregressive Search Engines: Generating Substrings as Document Identifiers”. *arXiv pre-print 2204.10628*. URL: https://arxiv.org/abs/2204.10628 (cited on page 124).

G P Shrivatsa Bhargav, Dinesh Khandelwal, Saswati Dana, Dinesh Garg, Pavan Kapanipathi, Salim Roukos, Alexander Gray, and L Venkata Subramaniam (July 2022). “Zero-shot Entity Linking with Less Data”. *Findings of the Association for Computational Linguistics: NAACL 2022*. Seattle, United States: Association for Computational Linguistics, pages 1681–1697. DOI: 10.18653/v1/2022.findings-naacl.127. URL: https://aclanthology.org/2022.findings-naacl.127 (cited on pages 29, 121).

Steven Bird, Ewan Klein, and Edward Loper (2009). *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc." (cited on page 154).

Roi Blanco, Giuseppe Ottaviano, and Edgar Meij (2015). "Fast and Space-Efficient Entity Linking for Queries". *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining.* WSDM '15. Shanghai, China: Association for Computing Machinery, 179–188. ISBN: 9781450333177. DOI: 10.1145/2684822.2685317. URL: https://doi.org/10.1145/2684822.2685317 (cited on pages 3, 12).

Mumeng Bo and Meihui Zhang (2021). "Learning Dynamic Coherence with Graph Attention Network for Biomedical Entity Linking". *2021 International Joint Conference on Neural Networks (IJCNN).* IEEE, pages 1–8 (cited on page 159).

Olivier Bodenreider (Feb. 2004). "The Unified Medical Language System (UMLS): Integrating Biomedical Terminology". *Nucleic acids research* 32, pages D267–70. DOI: 10.1093/nar/gkh061 (cited on pages 47, 59, 62, 141).

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov (2017). "Enriching word vectors with subword information". *Transactions of the Association for Computational Linguistics* 5, pages 135–146 (cited on page 43).

Jan A. Botha, Zifei Shan, and Daniel Gillick (Nov. 2020). "Entity Linking in 100 Languages". *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP).* Online: Association for Computational Linguistics, pages 7833–7845. DOI: 10.18653/v1/2020.emnlp-main.630. URL: https://aclanthology.org/2020.emnlp-main.630 (cited on page 88).

Nicholas Botzer, Yifan Ding, and Tim Weninger (2021). "Reddit entity linking dataset". *Information Processing & Management* 58.3, page 102479 (cited on page 59).

Florian Boudin (Dec. 2016). "pke: an open source python-based keyphrase extraction toolkit". *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*. Osaka, Japan, pages 69–73. URL: http://aclweb.org/anthology/C16-2015 (cited on page 124).

Adrien Bougouin, Florian Boudin, and Béatrice Daille (Oct. 2013). "TopicRank: Graph-Based Topic Ranking for Keyphrase Extraction". *Proceedings of the Sixth International Joint Conference on Natural Language Processing*. Nagoya, Japan: Asian Federation of Natural Language Processing, pages 543–551. URL: https://aclanthology.org/I13-1062 (cited on page 124).

Adrian M.P. Brasoveanu, Albert Weichselbraun, and Lyndon Nixon (Nov. 2020). "In Media Res: A Corpus for Evaluating Named Entity Linking with Creative Works". *Proceedings of the 24th Conference on Computational Natural Language Learning*. Online: Association for Computational Linguistics, pages 355–364. DOI: 10.18653/v1/2020.conll-1.28. URL: https://aclanthology.org/2020.conll-1.28 (cited on pages 39, 59).

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al.

(2020). "Language models are few-shot learners". *Advances in neural information processing systems* 33, pages 1877–1901 (cited on page 115).

Razvan Bunescu and Marius Paşca (Apr. 2006). "Using Encyclopedic Knowledge for Named entity Disambiguation". *11th Conference of the European Chapter of the Association for Computational Linguistics*. Trento, Italy: Association for Computational Linguistics, pages 9–16. URL: https://aclanthology.org/E06-1002 (cited on pages 16, 20, 23, 27, 115).

Keith E Campbell, Diane E Oliver, and Edward H Shortliffe (1998). "The Unified Medical Language System: toward a collaborative approach for solving terminologic problems". *Journal of the American Medical Informatics Association* 5.1, pages 12–16 (cited on page 60).

Yixin Cao, Lei Hou, Juanzi Li, and Zhiyuan Liu (Aug. 2018). "Neural Collective Entity Linking". *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, pages 675–686. URL: https://aclanthology.org/C18-1057 (cited on page 115).

Taylor Cassidy, Heng Ji, Lev-Arie Ratinov, Arkaitz Zubiaga, and Hongzhao Huang (Dec. 2012). "Analysis and Enhancement of Wikification for Microblogs with Context Expansion". *Proceedings of COLING 2012*. Mumbai, India: The COLING 2012 Organizing Committee, pages 441–456. URL: https://aclanthology.org/C12-1028 (cited on page 24).

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos (Nov. 2020). "LEGAL-BERT: The Muppets straight out of Law School". *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, pages 2898–2904. DOI: 10.18653/v1/2020.findings-emnlp.261. URL: https://aclanthology.org/2020.findings-emnlp.261 (cited on page 51).

Eric Charton, Marie-Jean Meurs, Ludovic Jean-Louis, and Michel Gagnon (May 2014). "Improving Entity Linking using Surface Form Refinement". *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. Reykjavik, Iceland: European Language Resources Association (ELRA), pages 4609–4615. URL: http://www.lrec-conf.org/proceedings/lrec2014/pdf/899_Paper.pdf (cited on page 22).

Pei-Fu Chen, Ssu-Ming Wang, Wei-Chih Liao, Lu-Cheng Kuo, Kuan-Chih Chen, Yu-Cheng Lin, Chi-Yu Yang, Chi-Hao Chiu, Shu-Chih Chang, Feipei Lai, et al. (2021). "Automatic ICD-10 coding and training system: deep neural network based on supervised learning". *JMIR Medical Informatics* 9.8, e23230 (cited on page 205).

Tongfei Chen and Benjamin Van Durme (2017). "Discriminative information retrieval for question answering sentence selection". *European Chapter of the Association for Computational Linguistics (EACL)*. Volume 2, pages 719–725 (cited on pages 163, 165).

Xilun Chen and Claire Cardie (June 2018). “Multinomial Adversarial Networks for Multi-Domain Text Classification”. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, pages 1226–1240. DOI: 10.18653/v1/N18-1111. URL: https://aclanthology.org/N18-1111 (cited on pages 92, 97).

Zheng Chen and Heng Ji (July 2011). “Collaborative Ranking: A Case Study on Entity Linking”. *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*. Edinburgh, Scotland, UK.: Association for Computational Linguistics, pages 771–781. URL: https://aclanthology.org/D11-1071 (cited on page 20).

Xiao Cheng and Dan Roth (Oct. 2013). “Relational Inference for Wikification”. *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Seattle, Washington, USA: Association for Computational Linguistics, pages 1787–1796. URL: https://aclanthology.org/D13-1184 (cited on pages 24, 26).

Ethan A. Chi, John Hewitt, and Christopher D. Manning (July 2020). “Finding Universal Grammatical Relations in Multilingual BERT”. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pages 5564–5577. DOI: 10.18653/v1/

2020.acl-main.493. URL: https://aclanthology.org/2020.acl-main.493 (cited on page 50).

Edward Choi, Mohammad Taha Bahadori, Elizabeth Searles, Catherine Coffey, Michael Thompson, James Bost, Javier Tejedor-Sojo, and Jimeng Sun (2016). "Multi-layer representation learning for medical concepts". *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, pages 1495–1504 (cited on pages 186, 190).

Chinmay Choudhary and Colm O'riordan (Aug. 2021). "End-to-end mBERT based Seq2seq Enhanced Dependency Parser with Linguistic Typology knowledge". *Proceedings of the 17th International Conference on Parsing Technologies and the IWPT 2021 Shared Task on Parsing into Enhanced Universal Dependencies (IWPT 2021)*. Online: Association for Computational Linguistics, pages 225–232. DOI: 10.18653/v1/2021.iwpt-1.24. URL: https://aclanthology.org/2021.iwpt-1.24 (cited on page 50).

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. (2022). "Palm: Scaling language modeling with pathways". *arXiv preprint arXiv:2204.02311* (cited on page 116).

Kevin Clark and Christopher D. Manning (July 2015). "Entity-Centric Coreference Resolution with Model Stacking". *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). Beijing, China: Association for Computational Linguistics, pages 1405–1415. DOI: 10.3115/v1/P15-1136. URL: https://aclanthology.org/P15-1136 (cited on page 25).

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov (July 2020). "Unsupervised Cross-lingual Representation Learning at Scale". *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pages 8440–8451. DOI: 10.18653/v1/2020.acl-main.747. URL: https://aclanthology.org/2020.acl-main.747 (cited on pages 51, 65, 69, 92, 94).

Marco Cornolti, Paolo Ferragina, Massimiliano Ciaramita, Stefan Rüd, and Hinrich Schütze (2016). "A Piggyback System for Joint Entity Mention Detection and Linking in Web Queries". *Proceedings of the 25th International Conference on World Wide Web*. WWW '16. Montréal, Québec, Canada: International World Wide Web Conferences Steering Committee, 567–578. ISBN: 9781450341431. DOI: 10.1145/2872427.2883061. URL: https://doi.org/10.1145/2872427.2883061 (cited on pages 3, 12).

Silviu Cucerzan (June 2007). "Large-Scale Named Entity Disambiguation Based on Wikipedia Data". *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. Prague, Czech Republic: Association for Computational

Linguistics, pages 708–716. URL: https://aclanthology.org/D07-1074 (cited on pages 16, 20–22, 24, 26, 27, 115).

Silviu Cucerzan (2011). “TAC Entity Linking by Performing Full-document Entity Extraction and Disambiguation”. *Theory and Applications of Categories* (cited on pages 16, 22).

Hongliang Dai, Yangqiu Song, Liwei Qiu, and Rijia Liu (Oct. 2018). “Entity Linking within a Social Media Platform: A Case Study on Yelp”. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pages 2023–2032. DOI: 10.18653/v1/D18-1227. URL: https://aclanthology.org/D18-1227 (cited on pages 39, 59).

Jeffrey Dalton, Laura Dietz, and James Allan (2014). “Entity Query Feature Expansion Using Knowledge Base Links”. *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*. SIGIR ’14. Gold Coast, Queensland, Australia: Association for Computing Machinery, 365–374. ISBN: 9781450322577. DOI: 10.1145/2600428.2609628. URL: https://doi.org/10.1145/2600428.2609628 (cited on pages 3, 12).

Nicola De Cao, Wilker Aziz, and Ivan Titov (Nov. 2021). “Highly Parallel Autoregressive Entity Linking with Discriminative Correction”. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pages 7662–7669. DOI: 10.18653/v1/2021.emnlp-main.604. URL: https://aclanthology.org/2021.emnlp-main.604 (cited on page 138).

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni (2021). "Autoregressive Entity Retrieval". *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=5k8F6UU39V (cited on pages ii, 3, 116, 118, 119, 121, 122, 125).

Nicola De Cao, Ledell Wu, Kashyap Popat, Mikel Artetxe, Naman Goyal, Mikhail Plekhanov, Luke Zettlemoyer, Nicola Cancedda, Sebastian Riedel, and Fabio Petroni (2022). "Multilingual Autoregressive Entity Linking". *Transactions of the Association for Computational Linguistics* 10, pages 274–290. DOI: 10.1162/tacl_a_00460. URL: https://aclanthology.org/2022.tacl-1.16 (cited on pages 89, 136, 138).

Mostafa Dehghani, Hamed Zamani, Aliaksei Severyn, Jaap Kamps, and W Bruce Croft (2017). "Neural ranking models with weak supervision". *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, pages 65–74 (cited on pages 67, 93, 149, 151, 155).

Leon Derczynski, Diana Maynard, Giuseppe Rizzo, Marieke Van Erp, Genevieve Gorrell, Raphaël Troncy, Johann Petrak, and Kalina Bontcheva (2015). "Analysis of named entity recognition and linking for tweets". *Information Processing & Management* 51.2, pages 32–49 (cited on page 119).

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (June 2019). “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, pages 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://aclanthology.org/N19-1423 (cited on pages ii, 32, 33, 51, 64, 65, 68, 69, 94, 118, 152).

Rezarta Dogan, Robert Leaman, Sun Kim, Dongseop Kwon, Chih-Hsuan Wei, Donald Comeau, Yifan Peng, David Cissel, Cathleen Coss, Carol Fisher, Rob Guzman, Preeti Kochar, Stella Koppel, Dorothy Trinh, Keiko Sekiya, Janice Ward, Deborah Whitman, Susan Schmidt, and Zhiyong lu (Mar. 2021). “NLM-Chem, a new resource for chemical entity recognition in PubMed full text literature”. *Scientific Data* 8. DOI: 10.1038/s41597-021-00875-1 (cited on pages 39, 62, 142).

Rezarta Islamaj Doğan, Robert Leaman, and Zhiyong Lu (2014). “NCBI disease corpus: a resource for disease name recognition and concept normalization”. *Journal of biomedical informatics* 47, pages 1–10 (cited on pages 49, 62).

Xin Dong, Yaxin Zhu, Zuohui Fu, Dongkuan Xu, and Gerard de Melo (Aug. 2021). “Data Augmentation with Adversarial Training for Cross-Lingual NLI”. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1:*

Long Papers). Online: Association for Computational Linguistics, pages 5158–5167. DOI: 10.18653/v1/2021.acl-long.401. URL: https://aclanthology.org/2021.acl-long.401 (cited on page 113).

Mark Dredze, Nicholas Andrews, and Jay DeYoung (Nov. 2016). “Twitter at the Grammys: A Social Media Corpus for Entity Linking and Disambiguation”. *Proceedings of The Fourth International Workshop on Natural Language Processing for Social Media*. Austin, TX, USA: Association for Computational Linguistics, pages 20–25. DOI: 10.18653/v1/W16-6204. URL: https://aclanthology.org/W16-6204 (cited on pages 39, 59).

Mark Dredze, Paul McNamee, Delip Rao, Adam Gerber, and Tim Finin (Aug. 2010a). “Entity Disambiguation for Knowledge Base Population”. *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*. Beijing, China: Coling 2010 Organizing Committee, pages 277–285. URL: https://aclanthology.org/C10-1032 (cited on pages 20, 27).

Mark Dredze, Paul McNamee, Delip Rao, Adam Gerber, and Tim Finin (2010b). “Entity disambiguation for knowledge base population”. *Conference on Computational Linguistics (COLING)*. Association for Computational Linguistics, pages 277–285 (cited on page 115).

Jennifer D’Souza and Vincent Ng (2015). “Sieve-based entity linking for the biomedical domain”. *Association for Computational Linguistics (ACL)*, pages 297–302 (cited on pages 49, 170).

Xiangyu Duan, Mingming Yin, Min Zhang, Boxing Chen, and Weihua Luo (July 2019). “Zero-Shot Cross-Lingual Abstractive Sentence Summarization through Teaching Generation and Attention”. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pages 3162–3172. DOI: 10.18653/v1/P19-1305. URL: https://aclanthology.org/P19-1305 (cited on page 35).

Greg Durrett and Dan Klein (2014). “A Joint Model for Entity Analysis: Coreference, Typing, and Linking”. *Transactions of the Association for Computational Linguistics* 2, pages 477–490. DOI: 10.1162/tacl_a_00197. URL: https://aclanthology.org/Q14-1037 (cited on pages 20, 25, 26, 115).

Joe Ellis, Jeremy Getman, Dana Fore, Neil Kuster, Zhiyi Song, Ann Bies, and Stephanie M Strassel (2015). “Overview of Linguistic Resources for the TAC KBP 2015 Evaluations: Methodologies and Results.” *TAC* (cited on page 60).

Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio (2010). “Why does unsupervised pre-training help deep learning?” *Journal of Machine Learning Research* 11.Feb, pages 625–660 (cited on page 152).

Yuan Fang and Ming-Wei Chang (2014). “Entity Linking on Microblogs with Spatial and Temporal Signals”. *Transactions of the Association for Computational Linguistics* 2, pages 259–272. DOI: 10.1162/tacl_a_00181. URL: https://aclanthology.org/Q14-1021 (cited on page 39).

Gregory P Finley, Serguei VS Pakhomov, Reed McEwan, and Genevieve B Melton (2016). “Towards Comprehensive Clinical Abbreviation Disambiguation Using Machine-Labeled Training Data”. *AMIA Annual Symposium Proceedings*. Volume 2016. American Medical Informatics Association, page 560 (cited on page 203).

Matthew Francis-Landau, Greg Durrett, and Dan Klein (June 2016a). “Capturing Semantic Similarity for Entity Linking with Convolutional Neural Networks”. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, pages 1256–1261. DOI: 10.18653/v1/N16-1150. URL: https://aclanthology.org/N16-1150 (cited on pages 28, 29).

Matthew Francis-Landau, Greg Durrett, and Dan Klein (June 2016b). “Capturing Semantic Similarity for Entity Linking with Convolutional Neural Networks”. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, pages 1256–1261. DOI: 10.18653/v1/N16-1150. URL: https://www.aclweb.org/anthology/N16-1150 (cited on page 115).

Rina Galperin, Shachar Schnapp, and Michael Elhadad (May 2022). “Cross-Lingual UMLS Named Entity Linking using UMLS Dictionary Fine-Tuning”. *Findings of the*

Association for Computational Linguistics: ACL 2022. Dublin, Ireland: Association for Computational Linguistics, pages 3380–3390. DOI: 10.18653/v1/2022.findings-acl.266. URL: https://aclanthology.org/2022.findings-acl.266 (cited on page 89).

Octavian-Eugen Ganea and Thomas Hofmann (Sept. 2017). "Deep Joint Entity Disambiguation with Local Neural Attention". *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, pages 2619–2629. DOI: 10.18653/v1/D17-1277. URL: https://aclanthology.org/D17-1277 (cited on page 28).

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch (2013). "PPDB: The paraphrase database". *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 758–764 (cited on page 184).

Daniel Gillick, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldridge, Eugene Ie, and Diego Garcia-Olano (Nov. 2019). "Learning Dense Representations for Entity Retrieval". *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*. Hong Kong, China: Association for Computational Linguistics, pages 528–537. DOI: 10.18653/v1/K19-1049. URL: https://aclanthology.org/K19-1049 (cited on page 36).

Xavier Glorot, Antoine Bordes, and Yoshua Bengio (2011). “Deep sparse rectifier neural networks”. *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323 (cited on page 155).

Hila Gonen, Shauli Ravfogel, Yanai Elazar, and Yoav Goldberg (Nov. 2020). “It’s not Greek to mBERT: Inducing Word-Level Translations from Multilingual BERT”. *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*. Online: Association for Computational Linguistics, pages 45–56. DOI: 10.18653/v1/2020.blackboxnlp-1.5. URL: https://aclanthology.org/2020.blackboxnlp-1.5 (cited on page 50).

Swapna Gottipati and Jing Jiang (July 2011). “Linking Entities to a Knowledge Base with Query Expansion”. *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*. Edinburgh, Scotland, UK.: Association for Computational Linguistics, pages 804–813. URL: https://aclanthology.org/D11-1074 (cited on pages 22, 27).

Thore Graepel, Klaus Obermayer, et al. (2000). “Large margin rank boundaries for ordinal regression”. *Advances in large margin classifiers*. the MIT Press, pages 115–132 (cited on page 20).

Gregory Grefenstette (2012). *Explorations in automatic thesaurus discovery*. Volume 278. Springer Science & Business Media (cited on page 184).

Stephen Guo, Ming-Wei Chang, and Emre Kiciman (June 2013). “To Link or Not to Link? A Study on End-to-End Tweet Entity Linking”. *Proceedings of the 2013*

Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Atlanta, Georgia: Association for Computational Linguistics, pages 1020–1030. URL: https://aclanthology.org/N13-1122 (cited on page 24).

Dishan Gupta, Jaime G Carbonell, Anatole Gershman, Steve Klein, and David Miller (2015). "Unsupervised Phrasal Near-Synonym Generation from Text Corpora." *AAAI*, pages 2253 –2259 (cited on page 203).

Nitish Gupta, Sameer Singh, and Dan Roth (Sept. 2017). "Entity Linking via Joint Encoding of Types, Descriptions, and Context". *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, pages 2681–2690. DOI: 10.18653/v1/D17-1284. URL: https://aclanthology.org/D17-1284 (cited on page 115).

Masato Hagiwara (2008). "A supervised learning approach to automatic synonym identification based on distributional features". *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Student Research Workshop*. Association for Computational Linguistics, pages 1–6 (cited on page 184).

Hannaneh Hajishirzi, Leila Zilles, Daniel S. Weld, and Luke Zettlemoyer (Oct. 2013). "Joint Coreference Resolution and Named-Entity Linking with Multi-Pass Sieves". *Proceedings of the 2013 Conference on Empirical Methods in Natural Language*

Processing. Seattle, Washington, USA: Association for Computational Linguistics, pages 289–299. URL: https://aclanthology.org/D13-1029 (cited on page 20).

Xianpei Han and Le Sun (June 2011). "A Generative Entity-Mention Model for Linking Entities with Knowledge Base". *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, pages 945–954. URL: https://aclanthology.org/P11-1095 (cited on pages 20, 22, 24, 26).

Xianpei Han and Le Sun (July 2012). "An Entity-Topic Model for Entity Linking". *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Jeju Island, Korea: Association for Computational Linguistics, pages 105–115. URL: https://aclanthology.org/D12-1010 (cited on page 24).

Xianpei Han, Le Sun, and Jun Zhao (2011). "Collective Entity Linking in Web Text: A Graph-Based Method". *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '11. Beijing, China: Association for Computing Machinery, 765–774. ISBN: 9781450307574. DOI: 10.1145/2009916.2010019. URL: https://doi.org/10.1145/2009916.2010019 (cited on pages 22, 24).

Xianpei Han and Jun Zhao (2009). "NLPR_KBP in TAC 2009 KBP Track: A Two-Stage Method to Entity Linking." *TAC*. Citeseer (cited on page 27).

Yun He, Ziwei Zhu, Yin Zhang, Qin Chen, and James Caverlee (Nov. 2020). “Infusing Disease Knowledge into BERT for Health Question Answering, Medical Inference and Disease Name Recognition”. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, pages 4604–4614. DOI: 10.18653/v1/2020.emnlp-main.372. URL: https://aclanthology.org/2020.emnlp-main.372 (cited on page 52).

Zhengyan He, Shujie Liu, Yang Song, Mu Li, Ming Zhou, and Houfeng Wang (Oct. 2013). “Efficient Collective Entity Linking with Stacking”. *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Seattle, Washington, USA: Association for Computational Linguistics, pages 426–435. URL: https://aclanthology.org/D13-1041 (cited on page 23).

Aron Henriksson, Hans Moen, Maria Skeppstedt, Vidas Daudaravičius, and Martin Duneld (2014). “Synonym extraction and abbreviation expansion with ensembles of semantic spaces”. *Journal of biomedical semantics* 5.1, page 6 (cited on page 203).

Cong Duy Vu Hoang, Trevor Cohn, and Gholamreza Haffari (2016). “Incorporating side information into recurrent neural network language models”. *North American Chapter of the Association for Computational Linguistics*, pages 1250–1255 (cited on page 191).

Sepp Hochreiter and Jürgen Schmidhuber (Nov. 1997). "Long Short-Term Memory". *Neural Comput.* 9.8, 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. URL: https://doi.org/10.1162/neco.1997.9.8.1735 (cited on page 30).

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum (July 2011). "Robust Disambiguation of Named Entities in Text". *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*. Edinburgh, Scotland, UK.: Association for Computational Linguistics, pages 782–792. URL: https://aclanthology.org/D11-1072 (cited on pages 23, 24).

Matthew Honnibal and Ines Montani (2017). "spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing". To appear (cited on page 128).

Hongzhao Huang, Yunbo Cao, Xiaojiang Huang, Heng Ji, and Chin-Yew Lin (June 2014). "Collective Tweet Wikification based on Semi-supervised Graph Regularization". *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Baltimore, Maryland: Association for Computational Linguistics, pages 380–390. DOI: 10.3115/v1/P14-1036. URL: https://aclanthology.org/P14-1036 (cited on page 24).

Shudong Huang (2005). *Chinese - English Name Entity Lists v 1.0 LDC2005T34*. Linguistic Data Consortium (cited on page 82).

Ann Irvine, Chris Callison-Burch, and Alexandre Klementiev (2010). “Transliterating from all languages”. *Proceedings of the Ninth Conference of the Association for Machine Translation in the Americas* (cited on page 82).

Heng Ji, Ralph Grishman, Hoa Trang Dang, Kira Griffitt, and Joe Ellis (2010). “Overview of the TAC 2010 knowledge base population track”. *Third text analysis conference (TAC 2010).* Volume 3. 2, pages 3–3 (cited on pages 16, 115).

Heng Ji, Joel Nothman, Ben Hachey, and Radu Florian (2015). “Overview of TAC-KBP2015 Tri-lingual Entity Discovery and Linking”. *TAC.* URL: https://www.semanticscholar.org/paper/Overview-of-TAC-KBP2015-Tri-lingual-Entity-and-Ji-Nothman/955a78a8a5e4e31d10ffc827f365bd4c4f30d563 (cited on pages 42, 43, 56, 64, 72, 76, 100, 127).

Yuzhe Jin, Emre Kıcıman, Kuansan Wang, and Ricky Loynd (2014). “Entity Linking at the Tail: Sparse Signals, Unknown Entities, and Phrase Models”. *Proceedings of the 7th ACM International Conference on Web Search and Data Mining.* WSDM ’14. New York, New York, USA: Association for Computing Machinery, 453–462. ISBN: 9781450323512. DOI: 10.1145/2556195.2556230. URL: https://doi.org/10.1145/2556195.2556230 (cited on page 88).

Timo Johner, Abhik Jana, and Chris Biemann (2021). “Error Analysis of using BART for Multi-Document Summarization: A Study for English and German Language”. *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa).* Reykjavik, Iceland (Online): Linköping University

Electronic Press, Sweden, pages 391–397. URL: https://aclanthology.org/2021.nodalida-main.43 (cited on page 118).

Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark (2016). "MIMIC-III, a freely accessible critical care database". *Scientific data* 3, page 160035 (cited on pages 154, 187, 194).

Andrew Johnson, Penny Karanasou, Judith Gaspers, and Dietrich Klakow (June 2019). "Cross-lingual Transfer Learning for Japanese Named Entity Recognition". *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Industry Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, pages 182–189. DOI: 10.18653/v1/N19-2023. URL: https://aclanthology.org/N19-2023 (cited on page 40).

Karen Sparck Jones (1972). "A statistical interpretation of term specificity and its application in retrieval". *Journal of documentation* (cited on page 23).

Mandar Joshi, Omer Levy, Luke Zettlemoyer, and Daniel Weld (Nov. 2019). "BERT for Coreference Resolution: Baselines and Analysis". *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pages 5803–5808.

DOI: 10.18653/v1/D19-1588. URL: https://aclanthology.org/D19-1588 (cited on page 33).

Martin Josifoski, Nicola De Cao, Maxime Peyrard, Fabio Petroni, and Robert West (July 2022). "GenIE: Generative Information Extraction". *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Seattle, United States: Association for Computational Linguistics, pages 4626–4643. DOI: 10.18653/v1/2022.naacl-main.342. URL: https://aclanthology.org/2022.naacl-main.342 (cited on page 138).

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih (Nov. 2020). "Dense Passage Retrieval for Open-Domain Question Answering". *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, pages 6769–6781. DOI: 10.18653/v1/2020.emnlp-main.550. URL: https://aclanthology.org/2020.emnlp-main.550 (cited on page 36).

Mahboob Alam Khalid, Valentin Jijkoun, and Maarten de Rijke (2008). "The Impact of Named Entity Normalization on Information Retrieval for Question Answering". *Advances in Information Retrieval*. Edited by Craig Macdonald, Iadh Ounis, Vassilis Plachouras, Ian Ruthven, and Ryen W. White. Berlin, Heidelberg: Springer Berlin Heidelberg, pages 705–710. ISBN: 978-3-540-78646-7 (cited on pages 3, 12).

Han Kyul Kim, Sae Won Choi, Ye Seul Bae, Jiin Choi, Hyein Kwon, Christine P Lee, Hae-Young Lee, and Taehoon Ko (2020). “MARIE: A Context-Aware Term Mapping with String Matching and Embedding Vectors”. *Applied Sciences* 10.21, page 7831 (cited on page 159).

Diederik Kingma and Jimmy Ba (Dec. 2014). “Adam: A Method for Stochastic Optimization”. *International Conference on Learning Representations* (cited on page 155).

Karin Kipper-Schuler, Vinod Kaggal, James Masanz, Philip Ogren, and Guergana Savova (2008). “System evaluation on a named entity corpus from clinical notes”. *Language resources and evaluation conference, LREC 2008* (cited on page 48).

Katrin Kirchhoff and Anne M Turner (2016). “Unsupervised Resolution of Acronyms and Abbreviations in Nursing Notes Using Document-Level Context Models”. *Proceedings of the Seventh International Workshop on Health Text Mining and Information Analysis*, pages 52–60 (cited on page 203).

Nikolaos Kolitsas, Octavian-Eugen Ganea, and Thomas Hofmann (Oct. 2018). “End-to-End Neural Entity Linking”. *Proceedings of the 22nd Conference on Computational Natural Language Learning*. Brussels, Belgium: Association for Computational Linguistics, pages 519–529. DOI: 10.18653/v1/K18-1050. URL: https://aclanthology.org/K18-1050 (cited on pages 28, 29).

Dan Kondratyuk and Milan Straka (Nov. 2019). “75 Languages, 1 Model: Parsing Universal Dependencies Universally”. *Proceedings of the 2019 Conference on*

Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Hong Kong, China: Association for Computational Linguistics, pages 2779–2795. DOI: 10.18653/v1/D19-1279. URL: https://aclanthology.org/D19-1279 (cited on page 65).

Maciej Kula (2017). *Spotlight*. https://github.com/maciejkula/spotlight (cited on page 156).

Sayali Kulkarni, Amit Singh, Ganesh Ramakrishnan, and Soumen Chakrabarti (2009). “Collective Annotation of Wikipedia Entities in Web Text”. *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD ’09. Paris, France: Association for Computing Machinery, 457–466. ISBN: 9781605584959. DOI: 10.1145/1557019.1557073. URL: https://doi.org/10.1145/1557019.1557073 (cited on pages 20, 24).

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer (June 2016). “Neural Architectures for Named Entity Recognition”. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, pages 260–270. DOI: 10.18653/v1/N16-1030. URL: https://aclanthology.org/N16-1030 (cited on page 115).

Phong Le and Ivan Titov (July 2018). “Improving Entity Linking by Modeling Latent Relations between Mentions”. *Proceedings of the 56th Annual Meeting of the*

Association for Computational Linguistics (Volume 1: Long Papers). Melbourne, Australia: Association for Computational Linguistics, pages 1595–1604. DOI: 10.18653/v1/P18-1148. URL: https://aclanthology.org/P18-1148 (cited on page 25).

Robert Leaman, Rezarta Islamaj Doğan, and Zhiyong Lu (2013). "DNorm: disease name normalization with pairwise learning to rank". *Bioinformatics* 29.22, pages 2909–2917 (cited on pages 149, 154, 163, 178, 181).

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang (2020). "BioBERT: a pre-trained biomedical language representation model for biomedical text mining". *Bioinformatics* 36.4, pages 1234–1240 (cited on page 51).

Artuur Leeuwenberg, Mihaela Vela, Jon Dehdari, and Josef van Genabith (2016). "A minimally supervised approach for synonym extraction with word embeddings". *The Prague Bulletin of Mathematical Linguistics* 105.1, pages 111–142 (cited on pages 184, 203).

John Lehmann, Sean Monahan, Luke Nezda, Arnold Jung, and Ying Shi (2010). "LCC Approaches to Knowledge Base Population at TAC 2010." *TAC*. Citeseer (cited on page 27).

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer (Aug. 2017). "Zero-Shot Relation Extraction via Reading Comprehension". *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*. Vancouver, Canada: Association for Computational Linguistics, pages 333–342.

DOI: 10.18653/v1/K17-1034. URL: https://aclanthology.org/K17-1034 (cited on page 35).

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer (July 2020a). "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension". *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pages 7871–7880. DOI: 10.18653/v1/2020.acl-main.703. URL: https://aclanthology.org/2020.acl-main.703 (cited on pages ii, 33, 118).

Patrick Lewis, Myle Ott, Jingfei Du, and Veselin Stoyanov (Nov. 2020b). "Pretrained Language Models for Biomedical and Clinical Tasks: Understanding and Extending the State-of-the-Art". *Proceedings of the 3rd Clinical Natural Language Processing Workshop*. Online: Association for Computational Linguistics, pages 146–157. DOI: 10.18653/v1/2020.clinicalnlp-1.17. URL: https://aclanthology.org/2020.clinicalnlp-1.17 (cited on page 52).

Jiao Li, Yueping Sun, Robin J Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J Mattingly, Thomas C Wiegers, and Zhiyong Lu (2016). "BioCreative V CDR task corpus: a resource for chemical disease relation extraction". *Database* 2016 (cited on page 62).

Xuansong Li, Joe Ellis, Kira Griffitt, Stephanie M Strassel, Robert Parker, and Jonathan Wright (2011). "Linguistic Resources for 2011 Knowledge Base Population Evaluation." *TAC* (cited on pages 11, 16, 115).

Krister Lindén and Jussi Piitulainen (2004). "Discovering synonyms and other related words". *Proceedings of CompuTerm 2004: 3rd International Workshop on Computational Terminology*, pages 63–70 (cited on page 184).

Xiao Ling, Sameer Singh, and Daniel S. Weld (2015). "Design Challenges for Entity Linking". *Transactions of the Association for Computational Linguistics* 3, pages 315–328. DOI: 10.1162/tacl_a_00141. URL: https://aclanthology.org/Q15-1023 (cited on pages 14, 27).

Fangyu Liu, Ehsan Shareghi, Zaiqiao Meng, Marco Basaldella, and Nigel Collier (June 2021a). "Self-Alignment Pretraining for Biomedical Entity Representations". *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, pages 4228–4238. DOI: 10.18653/v1/2021.naacl-main.334. URL: https://aclanthology.org/2021.naacl-main.334 (cited on page 159).

Fangyu Liu, Ivan Vulić, Anna Korhonen, and Nigel Collier (Aug. 2021b). "Learning Domain-Specialised Representations for Cross-Lingual Biomedical Entity Linking". *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language*

Processing (Volume 2: Short Papers). Online: Association for Computational Linguistics, pages 565–574. DOI: 10.18653/v1/2021.acl-short.72. URL: https://aclanthology.org/2021.acl-short.72 (cited on page 89).

Xiaohua Liu, Yitong Li, Haocheng Wu, Ming Zhou, Furu Wei, and Yi Lu (Aug. 2013). "Entity Linking for Tweets". *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Sofia, Bulgaria: Association for Computational Linguistics, pages 1304–1311. URL: https://aclanthology.org/P13-1128 (cited on page 59).

Chi-kiu Lo and Michel Simard (Nov. 2019). "Fully Unsupervised Crosslingual Semantic Textual Similarity Metric Based on BERT for Identifying Parallel Data". *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*. Hong Kong, China: Association for Computational Linguistics, pages 206–215. DOI: 10.18653/v1/K19-1020. URL: https://aclanthology.org/K19-1020 (cited on page 65).

Lajanugen Logeswaran, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, Jacob Devlin, and Honglak Lee (July 2019). "Zero-Shot Entity Linking by Reading Entity Descriptions". *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pages 3449–3460. DOI: 10.18653/v1/P19-1335. URL: https://aclanthology.org/P19-1335 (cited on pages xiv, 34–37, 39, 45, 57, 127–129, 131, 140, 211).

Gang Luo, Xiaojiang Huang, Chin-Yew Lin, and Zaiqing Nie (Sept. 2015). "Joint Entity Recognition and Disambiguation". *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, pages 879–888. DOI: 10.18653/v1/D15-1104. URL: https://aclanthology.org/D15-1104 (cited on page 25).

Yen-Fu Luo, Weiyi Sun, and Anna Rumshisky (2019). "MCN: A comprehensive corpus for medical concept normalization". *Journal of Biomedical Informatics* 92, page 103132. ISSN: 1532-0464. DOI: https://doi.org/10.1016/j.jbi.2019.103132. URL: https://www.sciencedirect.com/science/article/pii/S1532046419300504 (cited on pages 61, 142, 152, 159).

Ji Ma, Ivan Korotkov, Yinfei Yang, Keith Hall, and Ryan McDonald (Apr. 2021). "Zero-shot Neural Passage Retrieval via Domain-targeted Synthetic Question Generation". *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Online: Association for Computational Linguistics, pages 1075–1088. DOI: 10.18653/v1/2021.eacl-main.92. URL: https://aclanthology.org/2021.eacl-main.92 (cited on page 35).

Laurens Van der Maaten and Geoffrey Hinton (2008). "Visualizing data using t-SNE." *Journal of machine learning research* 9.11 (cited on page 202).

Yu A Malkov and Dmitry A Yashunin (2018). "Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs". *IEEE* *transactions on pattern analysis and machine intelligence* 42.4, pages 824–836 (cited on page 182).

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky (2014). “The Stanford CoreNLP natural language processing toolkit”. *Association for Computational Linguistics (ACL): System Demonstrations*, pages 55–60 (cited on page 169).

Pedro Henrique Martins, Zita Marinho, and André F. T. Martins (July 2019). “Joint Learning of Named Entity Recognition and Entity Linking”. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*. Florence, Italy: Association for Computational Linguistics, pages 190–196. DOI: 10.18653/v1/P19-2026. URL: https://aclanthology.org/P19-2026 (cited on page 11).

John McCrae and Nigel Collier (2008). “Synonym set extraction from the biomedical literature by lexical pattern discovery”. *BMC bioinformatics* 9.1, page 159 (cited on pages 184, 203).

Paul McNamee and Hoa Trang Dang (2009). “Overview of the TAC 2009 knowledge base population track”. *Text analysis conference (TAC)*. Volume 17, pages 111–113 (cited on pages 16, 115).

Paul McNamee, James Mayfield, Dawn Lawrie, Douglas Oard, and David Doermann (Nov. 2011). “Cross-Language Entity Linking”. *Proceedings of 5th International Joint Conference on Natural Language Processing*. Chiang Mai, Thailand: Asian

Federation of Natural Language Processing, pages 255–263. URL: https://www.aclweb.org/anthology/I11-1029 (cited on pages 41, 42, 64).

Oren Melamud, Jacob Goldberger, and Ido Dagan (Aug. 2016). "context2vec: Learning Generic Context Embedding with Bidirectional LSTM". *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*. Berlin, Germany: Association for Computational Linguistics, pages 51–61. DOI: 10.18653/v1/K16-1006. URL: https://aclanthology.org/K16-1006 (cited on pages 30, 186).

Rada Mihalcea, Courtney Corley, Carlo Strapparava, et al. (2006). "Corpus-based and knowledge-based measures of text semantic similarity". *AAAI*. Volume 6, pages 775–780 (cited on page 184).

Rada Mihalcea and Andras Csomai (2007). "Wikify!: linking documents to encyclopedic knowledge". *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*. ACM, pages 233–242 (cited on page 184).

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean (2013a). "Efficient estimation of word representations in vector space". *arXiv preprint arXiv:1301.3781* (cited on page 28).

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean (2013b). "Distributed representations of words and phrases and their compositionality". *Advances in neural information processing systems*, pages 3111–3119 (cited on pages 155, 189, 196).

David Mueller, Nicholas Andrews, and Mark Dredze (July 2020). “Sources of Transfer in Multilingual Named Entity Recognition”. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pages 8093–8104. DOI: 10.18653/v1/2020.acl-main.720. URL: https://aclanthology.org/2020.acl-main.720 (cited on page 74).

Dat Ba Nguyen, Abdalghani Abujabal, Nam Khanh Tran, Martin Theobald, and Gerhard Weikum (Sept. 2017). “Query-Driven on-the-Fly Knowledge Base Construction”. *Proc. VLDB Endow.* 11.1, 66–79. ISSN: 2150-8097. DOI: 10.14778/3151113.3151119. URL: https://doi.org/10.14778/3151113.3151119 (cited on page 12).

Feng Niu, Ce Zhang, Christopher Ré, and Jude Shavlik (2012). “Elementary: Large-scale knowledge-base construction via machine learning and statistical inference”. *International Journal on Semantic Web and Information Systems (IJSWIS)* 8.3, pages 42–73 (cited on page 12).

Yasumasa Onoe and Greg Durrett (2020). “Fine-grained entity typing for domain independent entity linking”. *Proceedings of the AAAI Conference on Artificial Intelligence*. Volume 34. 05, pages 8576–8583 (cited on pages 29, 58).

Laurel J. Orr, Megan Leszczynski, Simran Arora, Sen Wu, Neel Guha, Xiao Ling, and Christopher Ré (2020). “Bootleg: Chasing the Tail with Self-Supervised Named Entity Disambiguation”. *CoRR* abs/2010.10363. arXiv: 2010.10363. URL: https://arxiv.org/abs/2010.10363 (cited on pages 29, 88, 116, 120, 211).

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli (June 2019). "fairseq: A Fast, Extensible Toolkit for Sequence Modeling". *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*. Minneapolis, Minnesota: Association for Computational Linguistics, pages 48–53. DOI: 10.18653/v1/N19-4009. URL: https://aclanthology.org/N19-4009 (cited on page 125).

Deepak P., Sayan Ranu, Prithu Banerjee, and Sameep Mehta (2015). "Entity Linking for Web Search Queries". *Advances in Information Retrieval.* Edited by Allan Hanbury, Gabriella Kazai, Andreas Rauber, and Norbert Fuhr. Cham: Springer International Publishing, pages 394–399. ISBN: 978-3-319-16354-3 (cited on pages 3, 12).

Xiaoman Pan, Taylor Cassidy, Ulf Hermjakob, Heng Ji, and Kevin Knight (May 2015). "Unsupervised Entity Linking with Abstract Meaning Representation". *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Denver, Colorado: Association for Computational Linguistics, pages 1130–1139. DOI: 10.3115/v1/N15-1119. URL: https://aclanthology.org/N15-1119 (cited on pages 20, 24, 26, 42, 64).

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji (2017). "Cross-lingual name tagging and linking for 282 languages". *Proceedings* *of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Volume 1, pages 1946–1958 (cited on pages 42, 56, 72, 101).

Ted Pedersen, Serguei VS Pakhomov, Siddharth Patwardhan, and Christopher G Chute (2007). "Measures of semantic similarity and relatedness in the biomedical domain". *Journal of biomedical informatics* 40.3, pages 288–299 (cited on page 184).

Andraž Pelicon, Ravi Shekhar, Matej Martinc, Blaž Škrlj, Matthew Purver, and Senja Pollak (Apr. 2021). "Zero-shot Cross-lingual Content Filtering: Offensive Language and Hate Speech Detection". *Proceedings of the EACL Hackashop on News Media Content Analysis and Automated Report Generation*. Online: Association for Computational Linguistics, pages 30–34. URL: https://aclanthology.org/2021.hackashop-1.5 (cited on page 35).

Nanyun Peng, Mo Yu, and Mark Dredze (July 2015). "An Empirical Study of Chinese Name Matching and Applications". *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Beijing, China: Association for Computational Linguistics, pages 377–383. DOI: 10.3115/v1/P15-2062. URL: https://aclanthology.org/P15-2062 (cited on page 82).

Marco Pennacchiotti and Patrick Pantel (Aug. 2009). "Entity Extraction via Ensemble Semantics". *Proceedings of the 2009 Conference on Empirical Methods in*

Natural Language Processing. Singapore: Association for Computational Linguistics, pages 238–247. URL: https://aclanthology.org/D09-1025 (cited on pages 24, 26).

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer (June 2018). "Deep Contextualized Word Representations". *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, pages 2227–2237. DOI: 10.18653/v1/N18-1202. URL: https://www.aclweb.org/anthology/N18-1202 (cited on pages 31, 148, 186).

Matthew E. Peters, Sebastian Ruder, and Noah A. Smith (Aug. 2019). "To Tune or Not to Tune? Adapting Pretrained Representations to Diverse Tasks". *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*. Florence, Italy: Association for Computational Linguistics, pages 7–14. DOI: 10.18653/v1/W19-4302. URL: https://aclanthology.org/W19-4302 (cited on page 204).

Francesco Piccinno and P. Ferragina (2014). "From TagME to WAT: a new entity annotator". *ERD '14* (cited on page 115).

Anja Pilz and Gerhard Paaß (2011). "From Names to Entities Using Thematic Context Distance". *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*. CIKM '11. Glasgow, Scotland, UK: Association for Computing Machinery, 857–866. ISBN: 9781450307178. DOI: 10.1145/2063576.2063700. URL: https://doi.org/10.1145/2063576.2063700 (cited on page 20).

Telmo Pires, Eva Schlinger, and Dan Garrette (July 2019). "How Multilingual is Multilingual BERT?" *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pages 4996–5001. DOI: 10.18653/v1/P19-1493. URL: https://aclanthology.org/P19-1493 (cited on pages 50, 51, 64).

Gorjan Popovski, Barbara Koroušić Seljak, and Tome Eftimov (2019). "FoodBase corpus: a new resource of annotated food entities". *Database* 2019 (cited on pages 39, 62).

Sameer Pradhan, Noémie Elhadad, Brett R South, David Martinez, Lee Christensen, Amy Vogel, Hanna Suominen, Wendy W Chapman, and Guergana Savova (2014). "Evaluating the state of the art in disorder recognition and normalization of the clinical narrative". *Journal of the American Medical Informatics Association* 22.1, pages 143–154 (cited on pages 186, 187).

Sameer Pradhan, Noemie Elhadad, Brett R South, David Martinez, Lee M Christensen, Amy Vogel, Hanna Suominen, Wendy W Chapman, and Guergana K Savova (2013). "Task 1: ShARe/CLEF eHealth Evaluation Lab 2013." *CLEF (Working Notes)* (cited on pages 4, 60, 153, 170, 178, 186, 187, 194).

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. (2018). "Improving language understanding by generative pre-training" (cited on page 33).

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. (2019). "Language models are unsupervised multitask learners". *OpenAI blog* 1.8, page 9 (cited on page 115).

Afshin Rahimi, Yuan Li, and Trevor Cohn (July 2019). "Massively Multilingual Transfer for NER". *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pages 151–164. DOI: 10.18653/v1/P19-1015. URL: https://aclanthology.org/P19-1015 (cited on page 40).

Jonathan Raphael Raiman and Olivier Michel Raiman (2018). "DeepType: multilingual entity linking by neural type system evolution". *Thirty-Second AAAI Conference on Artificial Intelligence* (cited on pages 29, 44).

Nazneen Fatema Rajani, Mihaela Bornea, and Ken Barker (2017). "Stacking with Auxiliary Features for Entity Linking in the Medical Domain". *BioNLP 2017*, pages 39–47 (cited on page 49).

Delip Rao, Paul McNamee, and Mark Dredze (2013). "Entity linking: Finding extracted entities in a knowledge base". *Multi-source, multilingual information extraction and summarization*. Springer, pages 93–115 (cited on pages 16, 26, 27).

Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson (June 2011). "Local and Global Algorithms for Disambiguation to Wikipedia". *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational

Linguistics, pages 1375–1384. URL: https://aclanthology.org/P11-1138 (cited on pages 22–24).

Christopher Ré, Amir Sadeghian, Zifei Shan, Jaeho Shin, Feiran Wang, Sen Wu, and Ce Zhang (2014). "Feature Engineering for Knowledge Base Construction". *IEEE Data Eng. Bull.* 37, pages 26–40 (cited on page 12).

Radim Řehůřek and Petr Sojka (May 2010). "Software Framework for Topic Modelling with Large Corpora". English. *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. http://is.muni.cz/publication/884893/en. Valletta, Malta: ELRA, pages 45–50 (cited on page 196).

Nils Reimers and Iryna Gurevych (Nov. 2019). "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks". *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pages 3982–3992. DOI: 10.18653/v1/D19-1410. URL: https://aclanthology.org/D19-1410 (cited on pages 33, 65).

Philip Resnik (1995). "Using information content to evaluate semantic similarity in a taxonomy". *International Joint Conference on Artificial Intelligence (IJCAI)* (cited on page 184).

Shruti Rijhwani, Jiateng Xie, Graham Neubig, and Jaime Carbonell (2019). "Zero-shot neural transfer for cross-lingual entity linking". *Proceedings of the AAAI Conference on Artificial Intelligence*. Volume 33, pages 6924–6931 (cited on page 44).

Stephen E Robertson, Steve Walker, Susan Jones, et al. (1995). “Okapi at TREC-3” (cited on page 171).

Sara Rosenthal, Mihaela A. Bornea, and Avirup Sil (2021). “Are Multilingual BERT models robust? A Case Study on Adversarial Attacks for Multilingual Question Answering”. *ArXiv* abs/2104.07646 (cited on page 113).

Uma Roy, Noah Constant, Rami Al-Rfou, Aditya Barua, Aaron Phillips, and Yinfei Yang (Nov. 2020). “LAReQA: Language-Agnostic Answer Retrieval from a Multilingual Pool”. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, pages 5919–5930. DOI: 10.18653/v1/2020.emnlp-main.477. URL: https://aclanthology.org/2020.emnlp-main.477 (cited on page 50).

Mohammed Saeed, Mauricio Villarroel, Andrew T Reisner, Gari Clifford, Li-Wei Lehman, George Moody, Thomas Heldt, Tin H Kyaw, Benjamin Moody, and Roger G Mark (2011). “Multiparameter Intelligent Monitoring in Intensive Care II (MIMIC-II): a public-access intensive care unit database”. *Critical care medicine* 39.5, page 952 (cited on pages 60, 153, 194).

Ali Safaya, Moutasem Abdullatif, and Deniz Yuret (Dec. 2020). “KUISAIL at SemEval-2020 Task 12: BERT-CNN for Offensive Speech Identification in Social Media”. *Proceedings of the Fourteenth Workshop on Semantic Evaluation*. Barcelona (online): International Committee for Computational Linguistics, pages 2054–2059. DOI: 10.18653/v1/2020.semeval-1.271. URL: https://aclanthology.org/2020.semeval-1.271 (cited on page 80).

Guergana K Savova, James J Masanz, Philip V Ogren, Jiaping Zheng, Sunghwan Sohn, Karin C Kipper-Schuler, and Christopher G Chute (2010). "Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications". *Journal of the American Medical Informatics Association* 17.5, pages 507–513 (cited on pages 48, 170).

Elliot Schumacher and Mark Dredze (Nov. 2019). "Learning unsupervised contextual representations for medical synonym discovery". *JAMIA Open.* ooz057. ISSN: 2574-2531. DOI: 10.1093/jamiaopen/ooz057. eprint: http://oup.prod.sis.lan/jamiaopen/advance-article-pdf/doi/10.1093/jamiaopen/ooz057/30350969/ooz057.pdf. URL: https://doi.org/10.1093/jamiaopen/ooz057 (cited on page 152).

Satoshi Sekine (2005). "Automatic paraphrase discovery based on context and keywords between ne pairs". *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*, pages 80–87 (cited on page 184).

Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson (2014). "CNN features off-the-shelf: an astounding baseline for recognition". *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 806–813 (cited on page 152).

Wei Shen, Jianyong Wang, and Jiawei Han (2014). “Entity linking with a knowledge base: Issues, techniques, and solutions”. *IEEE Transactions on Knowledge and Data Engineering* 27.2, pages 443–460 (cited on page 20).

Wei Shen, Jianyong Wang, Ping Luo, and Min Wang (2012). “Linden: linking named entities with knowledge base via semantic knowledge”. *Proceedings of the 21st international conference on World Wide Web*, pages 449–458 (cited on pages 22, 24).

Avirup Sil and Alexander Yates (2013). “Re-Ranking for Joint Named-Entity Recognition and Linking”. *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*. CIKM ’13. San Francisco, California, USA: Association for Computing Machinery, 2369–2374. ISBN: 9781450322638. DOI: 10.1145/2505515.2505601. URL: https://doi.org/10.1145/2505515.2505601 (cited on page 25).

Samuel L. Smith, David H. P. Turban, Steven Hamblin, and Nils Y. Hammerla (2017). “Offline bilingual word vectors, orthogonal transformations and the inverted softmax”. *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=r1Aab85gg (cited on page 43).

Mohammad Golam Sohrab, Khoa Duong, Makoto Miwa, Goran Topić, Ikeda Masami, and Takamura Hiroya (Oct. 2020). “BENNERD: A Neural Named Entity Linking System for COVID-19”. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, pages 182–188. DOI: 10.18653/v1/2020.emnlp-demos.24. URL: https://aclanthology.org/2020.emnlp-demos.24 (cited on pages 61, 143).

Shashank Srivastava, Igor Labutov, and Tom Mitchell (July 2018). “Zero-shot Learning of Classifiers from Natural Language Quantification”. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, pages 306–316. DOI: 10.18653/v1/P18-1029. URL: https://aclanthology.org/P18-1029 (cited on page 35).

Rosa Stern, Benoît Sagot, and Frédéric Béchet (Apr. 2012). “A Joint Named Entity Recognition and Entity Linking System”. *Proceedings of the Workshop on Innovative Hybrid Approaches to the Processing of Textual Data*. Avignon, France: Association for Computational Linguistics, pages 52–60. URL: https://aclanthology.org/W12-0508 (cited on pages 11, 25).

Veselin Stoyanov, James Mayfield, Tan Xu, Douglas Oard, Dawn Lawrie, Tim Oates, and Tim Finin (June 2012). “A Context-Aware Approach to Entity Linking”. *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX)*. Montréal, Canada: Association for Computational Linguistics, pages 62–67. URL: https://aclanthology.org/W12-3012 (cited on page 24).

Chul Sung, Tejas Dhamecha, Swarnadeep Saha, Tengfei Ma, Vinay Reddy, and Rishi Arora (Nov. 2019). "Pre-Training BERT on Domain Resources for Short Answer Grading". *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pages 6071–6075. DOI: 10.18653/v1/D19-1628. URL: https://aclanthology.org/D19-1628 (cited on page 51).

Mujeen Sung, Hwisang Jeon, Jinhyuk Lee, and Jaewoo Kang (July 2020). "Biomedical Entity Representations with Synonym Marginalization". *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pages 3641–3650. DOI: 10.18653/v1/2020.acl-main.335. URL: https://aclanthology.org/2020.acl-main.335 (cited on page 182).

Chuanqi Tan, Furu Wei, Pengjie Ren, Weifeng Lv, and Ming Zhou (Sept. 2017). "Entity Linking for Queries by Searching Wikipedia Sentences". *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, pages 68–77. DOI: 10.18653/v1/D17-1007. URL: https://aclanthology.org/D17-1007 (cited on pages 3, 12).

William R Thompson (1933). "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples". *Biometrika* 25.3-4, pages 285–294 (cited on page 146).

Chen-Tse Tsai and Dan Roth (2016a). “Concept grounding to multiple knowledge bases via indirect supervision”. *Transactions of the Association of Computational Linguistics.* Volume 4, pages 141–154 (cited on page 49).

Chen-Tse Tsai and Dan Roth (2016b). “Cross-lingual Wikification Using Multilingual Embeddings”. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.* Stroudsburg, PA, USA: Association for Computational Linguistics, pages 589–598. DOI: 10.18653/v1/N16-1072. URL: http://aclweb.org/anthology/N16-1072 (cited on pages 43, 64, 73, 78, 127).

Henry Tsai, Jason Riesa, Melvin Johnson, Naveen Arivazhagan, Xin Li, and Amelia Archer (Nov. 2019). “Small and Practical BERT Models for Sequence Labeling”. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP).* Hong Kong, China: Association for Computational Linguistics, pages 3632–3636. DOI: 10.18653/v1/D19-1374. URL: https://aclanthology.org/D19-1374 (cited on page 65).

Tomoki Tsujimura, Noriyuki Mori, Masaki Asada, Makoto Miwa, and Yutaka Sasaki (2019). *Neural Medical Concept Normalization with Two-Step Training* (cited on page 152).

Shyam Upadhyay, Nitish Gupta, and Dan Roth (Oct. 2018). “Joint Multilingual Supervision for Cross-lingual Entity Linking”. *Proceedings of the 2018 Conference* *on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pages 2486–2495. DOI: 10.18653/v1/D18-1270. URL: https://aclanthology.org/D18-1270 (cited on pages 43, 56, 64, 66, 72–74, 78, 127).

Sylvia Vassileva, Gergana Todorova, Kristina Ivanova, Boris Velichkov, Ivan Koychev, Galia Angelova, and Svetla Boytcheva (Sept. 2021). “Automatic Transformation of Clinical Narratives into Structured Format”. *Proceedings of the Student Research Workshop Associated with RANLP 2021*. Online: INCOMA Ltd., pages 219–227. URL: https://aclanthology.org/2021.ranlp-srw.30 (cited on page 52).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin (2017). “Attention is all you need”. *Advances in neural information processing systems*, pages 5998–6008 (cited on pages 28, 32, 152).

Ellen M Voorhees and William R Hersh (2012). “Overview of the TREC 2012 Medical Records Track.” *TREC* (cited on page 184).

Yogarshi Vyas and Miguel Ballesteros (June 2021). “Linking Entities to Unseen Knowledge Bases with Arbitrary Schemas”. *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, pages 834–844. DOI: 10.18653/v1/2021.naacl-main.65. URL: https://aclanthology.org/2021.naacl-main.65 (cited on page 36).

Chang Wang, Liangliang Cao, and Bowen Zhou (2015a). “Medical synonym extraction with concept space models”. *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 989 –995 (cited on pages 184–186, 189, 195, 196, 202).

Daisy Zhe Wang, Yang Chen, Sean Goldberg, Christan Grant, and Kun Li (June 2012). “Automatic Knowledge Base Construction using Probabilistic Extraction, Deductive Reasoning, and Human Feedback”. *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX)*. Montréal, Canada: Association for Computational Linguistics, pages 106–110. URL: https://aclanthology.org/W12-3020 (cited on page 12).

Han Wang, Jin Guang Zheng, Xiaogang Ma, Peter Fox, and Heng Ji (Sept. 2015b). “Language and Domain Independent Entity Linking with Quantified Collective Validation”. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, pages 695–704. DOI: 10.18653/v1/D15-1081. URL: https://www.aclweb.org/anthology/D15-1081 (cited on pages 44, 115).

Runchuan Wang, Zhao Zhang, Fuzhen Zhuang, Dehong Gao, Yi Wei, and Qing He (2021). “Adversarial Domain Adaptation for Cross-Lingual Information Retrieval with Multilingual BERT”. *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. CIKM ’21. Virtual Event, Queensland, Australia: Association for Computing Machinery, 3498–3502. ISBN: 9781450384469.

DOI: 10.1145/3459637.3482050. URL: https://doi.org/10.1145/3459637.3482050 (cited on page 113).

Yuxuan Wang, Wanxiang Che, Jiang Guo, Yijia Liu, and Ting Liu (Nov. 2019). "Cross-Lingual BERT Transformation for Zero-Shot Dependency Parsing". *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pages 5721–5727. DOI: 10.18653/v1/D19-1575. URL: https://aclanthology.org/D19-1575 (cited on page 65).

William E Winkler (1990). "String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage." *ERIC* (cited on pages 58, 111, 197).

Ian H Witten and David N Milne (2008). "An effective, low-cost measure of semantic relatedness obtained from Wikipedia links" (cited on page 115).

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer (Nov. 2020). "Scalable Zero-shot Entity Linking with Dense Entity Retrieval". *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, pages 6397–6407. DOI: 10.18653/v1/2020.emnlp-main.519. URL: https://aclanthology.org/2020.emnlp-main.519 (cited on pages ii, 3, 4, 35, 45, 54, 116, 120, 126, 141, 181).

Liwei Wu, Shanbo Cheng, Mingxuan Wang, and Lei Li (Aug. 2021). "Language Tags Matter for Zero-Shot Neural Machine Translation". *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Online: Association for Computational Linguistics, pages 3001–3007. DOI: 10.18653/v1/2021.findings-acl.264. URL: https://aclanthology.org/2021.findings-acl.264 (cited on page 35).

Shijie Wu and Mark Dredze (Nov. 2019). "Beto, Bentz, Becas: The Surprising Cross-Lingual Effectiveness of BERT". *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pages 833–844. DOI: 10.18653/v1/D19-1077. URL: https://aclanthology.org/D19-1077 (cited on pages 50, 51, 64).

Dongfang Xu and Steven Bethard (June 2021). "Triplet-Trained Vector Space and Sieve-Based Search Improve Biomedical Concept Normalization". *Proceedings of the 20th Workshop on Biomedical Language Processing*. Online: Association for Computational Linguistics, pages 11–22. DOI: 10.18653/v1/2021.bionlp-1.2. URL: https://aclanthology.org/2021.bionlp-1.2 (cited on page 159).

Dongfang Xu and Timothy Miller (2022). "A simple neural vector space model for medical concept normalization using concept embeddings". *Journal of Biomedical Informatics* 130, page 104080 (cited on page 159).

Haoran Xu, Benjamin Van Durme, and Kenton Murray (Nov. 2021). "BERT, mBERT, or BiBERT? A Study on Contextualized Embeddings for Neural Machine Translation". *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pages 6663–6675. DOI: 10.18653/v1/2021.emnlp-main.534. URL: https://aclanthology.org/2021.emnlp-main.534 (cited on page 51).

Cheng Yan, Yuanzhe Zhang, Kang Liu, Jun Zhao, Yafei Shi, and Shengping Liu (Nov. 2021). "Biomedical Concept Normalization by Leveraging Hypernyms". *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pages 3512–3517. DOI: 10.18653/v1/2021.emnlp-main.284. URL: https://aclanthology.org/2021.emnlp-main.284 (cited on page 159).

Yiying Yang, Xi Yin, Haiqin Yang, Xingjian Fei, Hao Peng, Kaijie Zhou, Kunfeng Lai, and Jianping Shen (2021). "KGSynNet: A Novel Entity Synonyms Discovery Framework with Knowledge Graph". *Database Systems for Advanced Applications: 26th International Conference, DASFAA 2021, Taipei, Taiwan, April 11–14, 2021, Proceedings, Part I*. Taipei, Taiwan: Springer-Verlag, 174–190. ISBN: 978-3-030-73193-9. DOI: 10.1007/978-3-030-73194-6_13 (cited on page 204).

Xiang Yue and Shuang Zhou (Nov. 2020). "PHICON: Improving Generalization of Clinical Text De-identification Models via Data Augmentation". *Proceedings of* *the 3rd Clinical Natural Language Processing Workshop*. Online: Association for Computational Linguistics, pages 209–214. DOI: 10.18653/v1/2020.clinicalnlp-1.23. URL: https://aclanthology.org/2020.clinicalnlp-1.23 (cited on page 52).

Wei Zhang, Yan Chuan Sim, Jian Su, and Chew Lim Tan (2011). "Entity Linking with Effective Acronym Expansion, Instance Selection, and Topic Modeling". *IJCAI* (cited on page 22).

Wei Zhang, Jian Su, Chew Lim Tan, and Wen Ting Wang (Aug. 2010a). "Entity Linking Leveraging Automatically Generated Annotation". *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*. Beijing, China: Coling 2010 Organizing Committee, pages 1290–1298. URL: https://aclanthology.org/C10-1145 (cited on page 20).

Wei Zhang, Chew Lim Tan, Yan Chuan Sim, and Jian Su (2010b). "NUS-I2R: Learning a Combined System for Entity Linking". *Theory and Applications of Categories* (cited on pages 16, 22).

Jin G Zheng, Daniel Howsmon, Boliang Zhang, Juergen Hahn, Deborah McGuinness, James Hendler, and Heng Ji (2015). "Entity linking for biomedical literature". *BMC medical informatics and decision making* 15, S4 (cited on page 49).

Zhicheng Zheng, Fangtao Li, Minlie Huang, and Xiaoyan Zhu (June 2010). "Learning to Link Entities with Knowledge Base". *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Los Angeles, California: Association for Computational

Linguistics, pages 483–491. URL: https://aclanthology.org/N10-1072 (cited on pages 16, 20, 22).

Vita

Elliot Schumacher holds a Bachelor of Science in Computer and Information Science, and Linguistics, from the Ohio State University, and a Master of Science in Language Technologies from Carnegie Mellon University. He joined the Computer Science Department and the Center for Language and Speech Processing at Johns Hopkins in the Fall of 2017.