October 2019

Extracting and Representing Entities, Types, and Relations

Patrick Verga

EXTRACTING AND REPRESENTING ENTITIES, TYPES, AND RELATIONS

A Dissertation Presented

by

PATRICK VERGA

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

September 2019

College of Information and Computer Sciences

EXTRACTING AND REPRESENTING ENTITIES, TYPES, AND RELATIONS

A Dissertation Presented

by

PATRICK VERGA

Approved as to style and content by:

Andrew McCallum, Chair

Brian Dillon, Member

Mohit Iyyer, Member

Brendan O'Connor, Member

Luke Zettlemoyer, Member

James Allan, Chair of the Faculty
College of Information and Computer Sciences

ACKNOWLEDGMENTS

I want to acknowledge and thank everyone who helped me get to this point. To my continued surprise, I have finally finished my PhD. I honestly did not think this would ever happen but I am grateful and happy that it has. I would not have been able to do this without the many people that have helped me along over the years.

My many research mentors not only taught me invaluable skills but, most importantly, they gave me an opportunity and a chance in the first place. In particular Ed Pace-Schott, Becky Spencer, Hava Siegelmann, James Allan, Brian Levine, and my thesis advisor Andrew McCallum. I was fortunate to work with excellent colleagues, co-authors, and lab mates over the years including Ben, David, Laura, and Arvind to name just a handful. I also want to thank my thesis committee members Brian Dillon, Mohit Iyyer, Brendan O'Connor, and Luke Zettlemoyer for all of their time, feedback, and their acceptance of my dissertation without revisions.

Both before and during grad school, my friends and family helped shape who I am and how I view the world and were essential in me getting to where I am. My dad Greg, sisters Abbie and Julia, grandparents Billy, MJ, Voo and Vaw, aunts, uncles and cousins, my friends Steve, Tyler, Justin, Quinn, Pinar, Jesse, Keen, as well as everyone else that I got to spend time with in both Gloucester and Amherst were all invaluable people to have in my life.

I want to specifically and emphatically thank the three people that had the biggest impact on my life, and helped me most along the way. My mom Kellie, who continually sacrificed, supported, taught, and loved me and the rest of her family. Bran, who has been the most loyal of friends and at my side for the past ten years and

throughout this entire process. And lastly, Emma; the greatest partner, co-author, and friend I could ever be fortunate enough to have.

Finally, and in many ways most importantly, I want to acknowledge the random number generator that allowed this all to happen. No matter how hard I worked, luck was and is always, the most important factor in all of this. I was lucky enough to be lucky.

ABSTRACT

EXTRACTING AND REPRESENTING ENTITIES, TYPES, AND RELATIONS

SEPTEMBER 2019

PATRICK VERGA

B.S., UNIVERSITY OF MASSACHUSETTS AMHERST

B.A., UNIVERSITY OF MASSACHUSETTS AMHERST

M.S., UNIVERSITY OF MASSACHUSETTS AMHERST

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Andrew McCallum

Making complex decisions in areas like science, government policy, finance, and clinical treatments all require integrating and reasoning over disparate data sources. While some decisions can be made from a single source of information, others require considering multiple pieces of evidence and how they relate to one another. Knowledge graphs (KGs) provide a natural approach for addressing this type of problem: they can serve as long-term stores of abstracted knowledge organized around concepts and their relationships, and can be populated from heterogeneous sources including databases and text. KGs can facilitate higher level reasoning, influence the interpretation of new data, and serve as a scaffolding for knowledge that enhances the acquisition of new information. A symbolic graph over a fixed, human-defined

schema encoding facts about entities and their relations is the predominant method for representing knowledge, but this approach is brittle, lacks specificity, and is inevitably highly incomplete. On the other extreme, recent work on purely text-based knowledge models lack abstractions necessary for complex reasoning.

In this thesis I will present work incorporating neural models, rich structured ontologies, and unstructured raw text for representing knowledge. I will first discuss my work enhancing universal schema, a method for learning a latent schema over both existing structured resources and unstructured free text, embedding them jointly within a shared semantic space. Next, I inject additional hierarchical structure into the embedding space of concepts, resulting in more efficient statistical sharing among related concepts and improved accuracy in both fine-grained entity typing and linking. I then present initial work representing knowledge in context, including a single model for extracting all entities and long-range relations simultaneously over full paragraphs while jointly linking these entities to a KG. I will conclude by discussing possible future directions for representing knowledge in context.

TABLE OF CONTENTS

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

CHAPTER 1

INTRODUCTION

Over their lifetimes, humans acquire and build up representations about the world that help them to make decisions, act, and survive. Through direct personal experience and explicit instruction the world is partitioned into relevant concepts and the ways in which they relate and interact with one another. This allows humans to think, reason, and make complex decisions by integrating many various pieces of information derived from explicit personal experiences and abstract generalizations.

Throughout the past century, scientists have attempted to algorithmically define these mechanisms to build artificial agents. Since the beginning of the field, artificial intelligence (AI) researchers have made strives in the area of knowledge representation and reasoning – how computers should represent information about the world in a computable form that they can use to solve complex problems. The earliest models were meant to be general purpose systems capable of solving any problems in any domain (Newell and Simon, 1956; Newell et al., 1959). These systems operated on symbolic logic and hand-written rules. Over time, the field expanded leading to research in various related areas with vastly different high level motivations. For example, cognitive architectures (Anderson, 1983; Laird et al., 1987) attempted to define and implement a theory of human cognition where the emphasis was often on understanding abstract mechanisms of knowledge representation and reasoning rather than solving any particular problem. On the other hand, expert systems focused on building useful tools which were capable of operating intelligently within a narrowly

defined problem area and were less concerned with biological plausibility or general application (Feigenbaum, 1980; Buchanan, 1984).

These systems typically consist of two subcomponents: an inference engine that uses logical rules to create new facts and a knowledge base which represents and stores those facts and rules. These systems suffer from two primary interconnected limitations: 1) facts and rules have to be predefined by humans and 2) their rigid symbolic representations do not easily generalize. When working within a narrow domain this paradigm may be sufficient to solve the given task, but when considering more complex areas like medicine or a general problem solver, this approach is untenable. For example, the longest running AI project CYC (Lenat et al., 1990) is attempting to codify all of human commonsense and has employed a staff to hand engineer facts and rules since 1984.

To address the first issue of acquiring facts, researchers have more recently focused on automatic knowledge base construction (AKBC), methods for building extensive sources of facts with minimal human effort. AKBC consists of many interconnected pieces that we will go over in greater detail in Chapter 2. In brief, knowledge is structured around concepts or entities along with their types and relationships. These methods often leverage existing human defined resources as weak supervision to automatically gather new facts without extensive further human effort. New concepts and their properties are extracted from raw text or inferred based on co-occurrence statistics of existing facts.

The second issue with early knowledge representation systems is their reliance on purely symbolic representations. Symbols have many desirable properties, particularly that they can be manipulated based on rules to perform logical inference and steps of reasoning. The downside however, is that they are very brittle and do not readily generalize to new concepts without additional explicit annotation. An alternative to purely symbolic representations that evolved alongside symbolic AI was

based on subsymbolic neural representations (Rosenblatt, 1958; Rttmelhart et al., 1986). Rather than relying on human defined semantics, neural representations can be learned directly from data and capture regularities, similarities, and relationships between different concepts. The past decade has seen huge advances in the abilities of neural network architectures (Collobert et al., 2011; Hinton et al., 2012; Krizhevsky et al., 2012; Silver et al., 2017) and they have become a fundamental component in knowledge representation, reasoning, and extraction models.

In this thesis we will introduce new methods for representing knowledge that build on embedded knowledge graphs, grounding symbolic concepts in sub-symbolic neural representations. These flexible representations facilitate both higher level reasoning and the automatic acquisition of new knowledge from text. In Chapters 3 and 4, we expand the generalization of universal schema (Riedel et al., 2013a) which combines explicit structured ontological types with latent types derived from raw textual expressions in a unified embedding space. Next, in Chapter 5 we enhance this embedded space with hierarchical information between entities and types. Lastly, in Chapters 6 and 7 we develop methods for large context information extraction that jointly consider both entity and relation prediction decisions to more effectively discover knowledge from text.

Before presenting the new work contained in this thesis, I will first go over the preliminary background materials on AKBC.

CHAPTER 2

AUTOMATIC KNOWLEDGE BASE CONSTRUCTION (AKBC)

As we discussed above, one of the major drawbacks of early symbolic systems was the knowledge base - the store of facts and rules - was populated by humans. Today, knowledge bases are still widely used in real world applications such as organizing biomedical findings (Bodenreider, 2004) and aiding search at large tech companies (Google, 2012; Dong, 2017). Unfortunately, because these knowledge bases tend to be built by human curators, they are inevitably incomplete. Automatic knowledge base construction (AKBC) is the task of populating a structured knowledge base (KB) of facts using raw text evidence, and often an initial seed KB to be augmented (Carlson et al., 2010; Suchanek et al., 2007a; Bollacker et al., 2008a) (See Figure 2.1).

2.1 Knowledge Bases/Graphs

The exact definition and instantiation of knowledge bases has evolved over time, but the most prominent form today refers to binary valued (*s, p, o*) facts (ie *subject*, *predicate*, *object*) that exist within a fixed pre-defined schema. The same information can equivalently be represented as a graph (knowledge graph or KG), in which entities are nodes and relations are labeled edges. KBs generally contain entity-type facts such as (*Melinda Gates*, *IsA*, *Person*) and relation facts such as (*Melinda Gates*, *co-founded*, *Bill and Melinda Gates Foundation*).

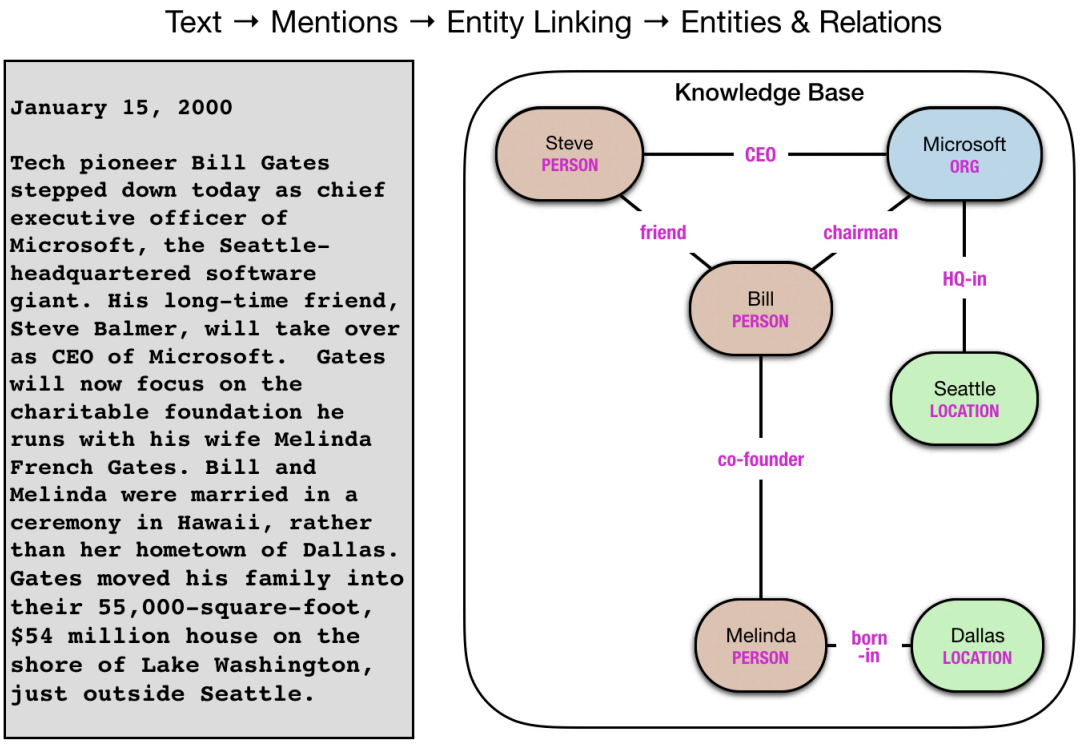


Figure 2.1: In general, the goal of automatic knowledge base construction is to go from a corpus of text documents to a knowledge graph of entities (nodes) and relations (edges). In addition to a an unstructured text corpus, methods often incorporate an initial incomplete structured knowledge graph as input.

2.1.1 Link Prediction

One method for discovering new facts in our KG is with *knowledge graph completion*. This task is akin to link prediction, assuming an initial set of (s, p, o) triples. See Nickel et al. (2015) for a review. No accompanying text data is necessary, since links can be predicted using properties of the graph, such as transitivity. In order to generalize well, prediction is often posed as low-rank matrix or tensor factorization. A variety of model variants have been suggested, where the probability of a given edge existing depends on a multi-linear form (Nickel et al., 2011b; García-Durán et al., 2016; Yang et al., 2015a; Bordes et al., 2013; Wang et al., 2014; Lin et al., 2015), or non-linear interactions between s, p, and o (Socher et al., 2013). Other approaches model the compositionality of multi-hop paths, typically for question answering (Bordes et al., 2014; Gu et al., 2015; Neelakantan et al., 2015a). While these methods are effective they are unable to discover new entities, types, or relations and are limited by to training on the existing structured knowledge source.

2.2 Information extraction

As the amount of available text data has exploded over the past several decades, researches have focused on developing methods for automatically extracting knowledge from text (Grishman and Sundheim, 1996). These information extraction approaches focused on mining large amounts of unstructured free text with the goal of converting it to a machine readable structured form. A common approach is to define a pipeline of mention finding (Ratinov and Roth, 2009), entity typing (Ling and Weld, 2012a; Shimaoka et al., 2017), entity linking and relation extraction. The resulting extracted facts can then be added to an existing knowledge base or used to create a new one.

January 15, 2000

Tech pioneer Bill Gates stepped down today as chief executive officer of Microsoft, the Seattle-headquartered software giant. His long-time friend, Steve Balmer, will take over as CEO of Microsoft. Gates will now focus on the charitable foundation he runs with his wife Melinda French Gates. Bill and Melinda were married in a ceremony in Hawaii, rather than her hometown of Dallas. Gates moved his family into their 55,000-square-foot, $54 million house on the shore of Lake Washington, just outside Seattle.

Figure 2.2: Example of named entity recognition. In this example, all instances of people (yellow), organizations (blue), and locations (green) have been identified and assigned their appropriate type.

2.2.1 Mention Detection

The goal of mention detection is to segment a sequence of text into a set of entity mentions. An input text string S is first split into n tokens, each denoted as s_i. Many early works relied on rules and lexicons to perform this task and these approaches are still used in low resource scenarios. However, most modern approaches utilize supervised machine learning methods.

This problem can be set up as a supervised sequence labeling task where the goal is to assign a label y_i to each token drawn from the label set Y. A common encoding for mention boundaries is to define a set of boundary labels Y_b to be BIO or BILOU. BIO represents the Beginning, Inside, and Outside of mentions where the first token of a mention would be labeled B, all subsequent tokens would be labeled I, and any tokens which are not part of any mention are labeled O. BILOU encoding adds two additional labels. L is the last token of any mention, and U is given to any mention which is a single token.

In addition to simply identifying the boundaries of entity mentions, it is common to simultaneously predict a small set of coarse grained types (ie named entity recognition (NER)) (See Figure 2.2). For example, in the CoNLL 2003 shared task dataset (Tjong Kim Sang and De Meulder, 2003), the type set Y_t are Person, Organization, and Location. To encode with BIO, the label set Y would be the cross product of Y_t and Y_b (except for O which is untyped). More recent work has attempted to predict a very large set of fine-grained entity types which we will discuss further in Chapter 5.

The basic model would map each token s_i to a feature vector x_i which would then be mapped through some other function producing per token logits $\hat{y}_i$. The entire model can be learned by stochastic gradient descent, for example, by minimizing the cross entropy between Y and $\hat{Y}$.

$$x_i = f_{ner}(s_i)$$

$$\hat{y}_i = g_{ner}(x_i)$$

Two primary approaches are to have f_{ner} be a mapping to hand engineered features and g_{ner} to be a linear model (Ratinov and Roth, 2009) or for f_{ner} to be a mapping to word embeddings and g to be a neural network model such as a recurrent neural network (Lample et al., 2016) or convolution neural network (Strubell et al., 2017). In both cases, it is also common to incorporate a linear chain conditional random field (Lafferty et al., 2001) to learn explicit dependencies over the outputs such as the fact that an Inside label can only follow a Begin label.

2.2.2 Coreference/Entity Linking

In our corpus, many of the entity mentions we discover will actually be individual instances of the same global entity. For example, there could be many sentences

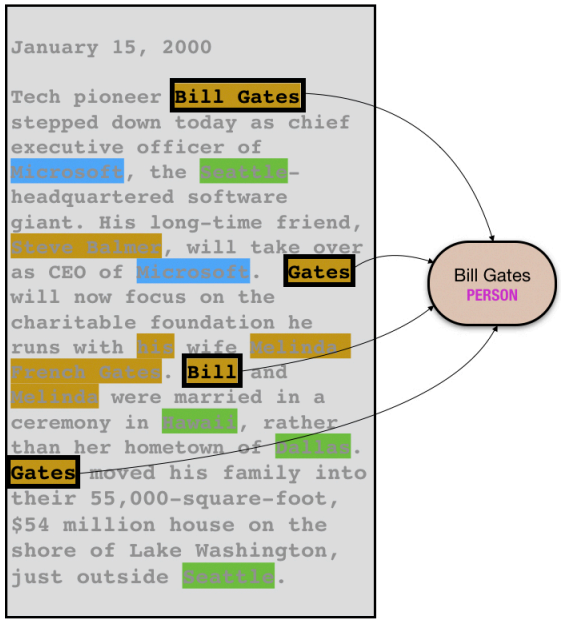


Figure 2.3: An example of coreference and entity linking. The four highlighted mentions in the document are all coreferent, referring to the same entity 'Bill Gates'. Entity linking typically refers to the act of connecting these mentions to an entity node in the knowledge graph.

talking about 'Bill Gates'. Each of those sentences would constitute a mention or instance and they are all referring to a single concept that is the entity 'Bill Gates' (See Figure 2.3). In order to aggregate these entity mentions together into our knowledge base, we need to cluster them such that each cluster contains all the mentions of a single entity. Broadly, this can take two different forms. The first is agglomerative where there are no predefined entities, and mentions determined to be referring to the same entity are merged into the same cluster. The second approach, typically referred to as entity linking, uses a predefined set of entities and each entity mention is assigned to one of these entity targets.

A common approach to entity linking is to cast the problem as classification. Given a mention m_e, the goal is to classify it as being an instance of exactly one entity e_i out of a set of known entities E. Because $|E|$ is typically very large, heuristics are often employed to prune the set of candidate entities to a reduced plausible set C based on surface form features of m_e. For example, given a mention 'Gates', one could restrict C to only contain entities whose canonical string name contains Gates such as 'Melinda Gates', 'Bill Gates', etc. We can then predict a link from m_e to $\hat{e}$ by

mapping m_e and each candidate entity to feature vectors and finding the maximum scoring candidate entity for m_e.

$$\bar{m_e} = f_{link}(m_e)$$
$$\bar{C} = g_{link}(C)$$
$$\hat{e} = max_{\bar{c_i} \in \bar{c}} h(\bar{m_e}, \bar{c_i})$$

2.2.3 Relation Extraction

A crucial component of understanding and representing knowledge is not simply identifying and categorizing entities but also discerning the relationships between the entities. Relation extraction is the task of automatically identifying these relationships from unstructured text. The input is an entity pair mention m_{ep}, typically a sentence containing a pre-identified pair of entities m_{e_1} and m_{e_2}. The goal is then to classify m_{ep} as expressing one of k predefined relation types R, which includes the null relation, ie no relation is being expressed between the two mentions m_{e_1} and m_{e_2} in m_{ep}.

$$\bar{e_{ep}} = f_{rel}(e_{ep})$$
$$\hat{R} = g_{rel}(\bar{e_{ep}})$$

2.2.3.1 Distant Supervision

Given labeled training data this model can be trained in the straight forward supervised learning setup, for example minimizing cross entropy between $\hat{R}$ and R. However, in many cases we do not have access to this type of mention level annotation but instead have access to a large number of entity level annotations. We can

leverage human annotated facts from existing structured knowledge bases as distant supervision (Craven and Kumlien, 1999) to train mention level relation extraction classifiers (Bunescu and Mooney, 2007; Mintz et al., 2009a).

For every fact in a knowledge base – for example, (*Melinda Gates*, *co-founder*, *Bill Gates*) – the simplest version of a distant supervision relation classifier would construct a training set by labeling every mention of the entity pair (*Melinda Gates* and *Bill Gates*) as expressing the relation *co-founder*. However, this naive assumption introduces noise into the training procedure because not every mention of (*Melinda Gates* and *Bill Gates*) is expressing the relation *co-founder*. For example, the sentence '*Melinda Gates* resides in Seattle with her husband *Bill*' expresses an entirely different relation *married*.

To address this, researchers have used versions of multi-instance learning (Craven and Kumlien, 1999; Riedel et al., 2010; Yao et al., 2010; Hoffmann et al., 2011; Surdeanu et al., 2012; Min et al., 2013; Zeng et al., 2015a). Instead of assigning all known labels of an entity pair to every mention of that entity pair, the mentions are pooled together into a single bag and the labels are instead applied to the bag. Intuitively this means that for each relation between an entity pair, *atleast one* of their mentions must express that relation which is a much softer assumption than the previous approach which stated that *every* mention must express *every* relation.

2.2.3.2 Open-Domain Relation Extraction

In the previous two approaches, prediction is carried out with respect to a fixed schema R of possible relations r. This may overlook salient relations that are expressed in the text but do not occur in the schema. In response, *open-domain* information extraction (OpenIE) lets the text speak for itself: R contains all possible patterns of text occurring between entities s and o (Banko et al., 2007; Etzioni et al., 2008; Yates and Etzioni, 2007). These are obtained by filtering and normalizing the

raw text. The approach offers impressive coverage, avoids issues of distant supervision, and provides a useful exploratory tool. On the other hand, OpenIE predictions are difficult to use in downstream tasks that expect information from a fixed schema.

Table 2.1 provides examples of OpenIE patterns. The examples in row two and three illustrate relational contexts for which similarity is difficult to be captured by an OpenIE approach because of their syntactically complex constructions. This concept is explored further in universal schema (Section 3.2.1)

Sentence (context tokens italicized)	OpenIE pattern
Khan *'s younger sister,* **Annapurna Devi**, who later married Shankar, developed into an equally accomplished master of the surbahar, but custom prevented her from performing in public.	*arg1*'s * sister *arg2*
A professor emeritus at Yale, **Mandelbrot** *was born in Poland but as a child moved with his family to* **Paris** where he was educated.	*arg1* * moved with * family to *arg2*
Kissel *was born in Provo, Utah, but her family also lived in* **Reno**.	*arg1* * lived in *arg2*

Table 2.1: Examples of sentences expressing relations. Context tokens (italicized) consist of the text occurring between entities (bold) in a sentence. OpenIE patterns are obtained by normalizing the context tokens using hand-coded rules. The top example expresses the per:siblings relation and the bottom two examples both express the per:cities_of_residence relation.

2.3 Universal Schema

Universal schema (Riedel et al., 2013a; Yao et al., 2013) in many ways combines aspects of all of the components above. The core idea of universal schema is to jointly model both fixed structure schema types and unstructured types expressed in raw text. Similar to the concept of open information extraction, this gives the model greater expressibility than one restricted to a small set of predefined schema types. This idea also extends to jointly modeling multiple sources of structured data which can naturally align partially disjoint schema.

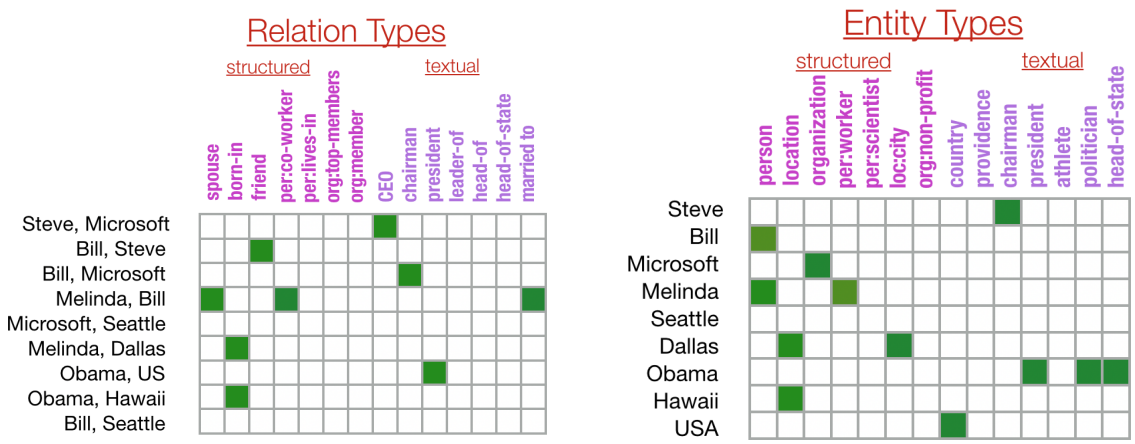


Figure 2.4: Universal schema matrix. a: Relation extraction. Relation types are represented as columns and entity pairs as rows of a matrix. Both KB relation types and textual patterns from raw text are jointly embedded into the same semantic space. b: Entity type prediction. Entity types are represented as columns and entities as rows of a matrix.

2.3.1 Modeling Universal Schema as Matrix Factorization

Universal schema relation extraction and entity type prediction is typically modeled as a matrix completion task[1]. In relation extraction, entity pairs and relations occupy the rows and columns of the matrix (Figure 2.4-a), while in entity type prediction, entities and types occupy the rows and columns of the matrix (Figure 2.4-b). During training, we observe some positive entries in the matrix and at test time, we predict the missing cells in the matrix. This is achieved by decomposing the observed matrix into two low-rank matrices resulting in embeddings for each column entry and each row entry. Test time prediction is performed using the learned low-rank column and row representations.

Modeling of a small set of of k predefined relation types reduces to a clustering over k centroids, mapping textual mentions to membership within one of these clusters. This bounds the representational power of the model to that directly encoded in the schema. On the other hand, the number of clusters represented within universal

[1]refereed to as Model F in Riedel et al. (2013a)

schema is equal to the number of entity pairs e used to fit the model. This leads to a level of specificity and expressiveness proportional to $|e|$ and the diversity and distribution of the training data.

Let T be the training set consisting of examples of the form (r, c), where row $r \in U$ and column $c \in V$, denote an entity pair and relation type in the relation extraction task, or entity and entity type in the entity type prediction task. Let $v(r) \in \mathbb{R}^d$ and $v(c) \in \mathbb{R}^d$ be the vector representations or embeddings of row $r \in U$ and column $c \in V$ that are learned during training. Given a positive example, $(r, c) \in T$ in training, the probability of observing the fact is given by,

$$P(y_{r,c} = 1) = \sigma(v(r).v(c))$$

where $y_{r,c}$ is a binary random variable that is equal to 1 when (r, c) is a fact and 0 otherwise, and σ is the sigmoid function. The embeddings are learned using Bayesian Personalized Ranking (BPR) (Rendle et al., 2009) in which the probability of the observed triples are ranked above unobserved triples.

Toutanova et al. (2015) extended USchema to not learn individual pattern embeddings v_r, but instead to embed text patterns using a deep architecture applied to word tokens. This shares statistical strength between OpenIE patterns with similar words. We employ a similar approach that we will discuss next in Chapter 3.

CHAPTER 3

COLUMNLESS UNIVERSAL SCHEMA

As previously discussed, *Universal schema* builds a knowledge base (KB) of entities and relations by jointly embedding all relation types from input KBs as well as textual patterns observed in raw text. In most previous applications of universal schema, each textual pattern was represented as a single embedding, preventing generalization to unseen patterns. More recently, extensive work in NLP has employed neural networks to capture patterns' compositional semantics, providing generalization to all possible input text. In this chapter, we put forth further improvements to the coverage and flexibility of universal schema relation extraction: predictions for entities unseen in training and multilingual transfer learning to domains with no annotation. We evaluate our model through extensive experiments on the English and Spanish TAC KBP benchmark, outperforming the top system from TAC 2013 slot-filling using no handwritten patterns or additional annotation. We also consider a multilingual setting in which English training data entities overlap with the seed KB, but Spanish text does not. Despite having no annotation for Spanish data, we train an accurate predictor, with additional improvements obtained by tying word embeddings across languages. Furthermore, we find that multilingual training improves English relation extraction accuracy.

3.1 Introduction

The goal of automatic knowledge base construction (AKBC) is to build a structured knowledge base (KB) of facts using a noisy corpus of raw text evidence, and

perhaps an initial seed KB to be augmented (Carlson et al., 2010; Suchanek et al., 2007a; Bollacker et al., 2008a). AKBC supports downstream reasoning at a high level about extracted entities and their relations, and thus has broad-reaching applications to a variety of domains.

One challenge in AKBC is aligning knowledge from a structured KB with a text corpus in order to perform supervised learning through *distant supervision*. *Universal schema* (Riedel et al., 2013a) along with its extensions (Yao et al., 2013; Gardner et al., 2014; Neelakantan et al., 2015a; Rocktaschel et al., 2015), avoids alignment by jointly embedding KB relations, entities, and surface text patterns. This propagates information between KB annotation and corresponding textual evidence.

The above applications of universal schema express each text relation as a distinct item to be embedded. This harms its ability to generalize to inputs not precisely seen at training time. Recently, Toutanova et al. (2015) addressed this issue by embedding text patterns using a deep sentence encoder, which captures the compositional semantics of textual relations and allows for prediction on inputs never seen before.

In this chapter, we further expand the coverage abilities of universal schema relation extraction by introducing techniques for forming predictions for new entities unseen in training and even for new domains with no associated annotation. In the extreme example of domain adaptation to a completely new language, we may have limited linguistic resources or labeled data such as treebanks, and only rarely a KB with adequate coverage. Our method performs multilingual transfer learning, providing a predictive model for a language with no coverage in an existing KB, by leveraging common representations for shared entities across text corpora. As depicted in Figure 3.1, we simply require that one language have an available KB of seed facts. We can further improve our models by tying a small set of word embeddings across languages using only simple knowledge about word-level translations,

learning to embed semantically similar textual patterns from different languages into the same latent space.

In extensive experiments on the TAC Knowledge Base Population (KBP) slot-filling benchmark we outperform the top 2013 system with an F1 score of 40.7 and perform relation extraction in Spanish with no labeled data or direct overlap between the Spanish training corpus and the training KB, demonstrating that our approach is well-suited for broad-coverage AKBC in low-resource languages and domains. Interestingly, joint training with Spanish improves English accuracy.

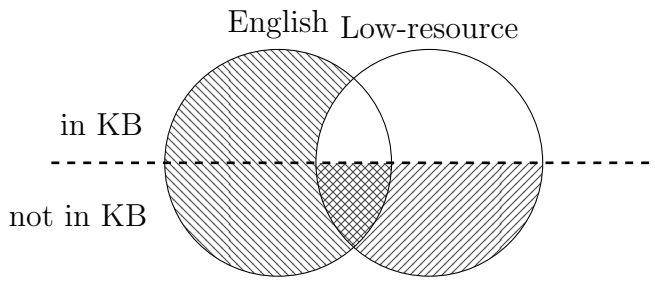
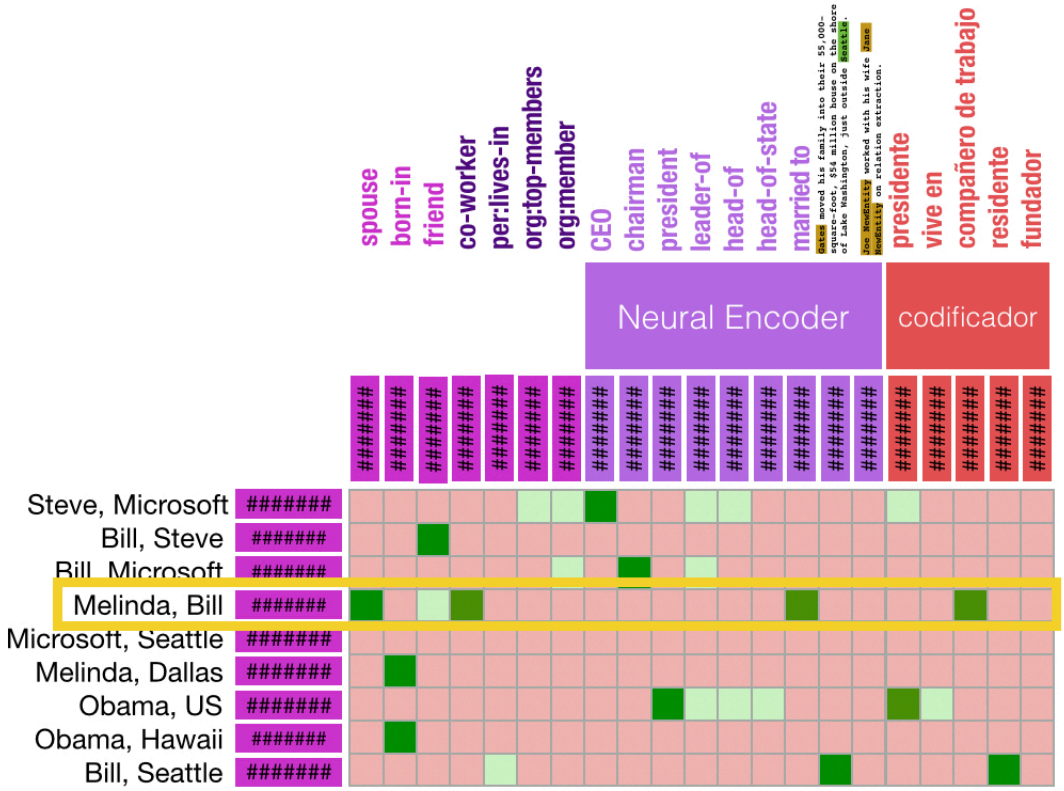


Figure 3.1: Splitting the entities in a multilingual AKBC training set into parts. We only require that entities in the two corpora overlap. Remarkably, we can train a model for the low-resource language even if entities in the low-resource language do not occur in the KB.

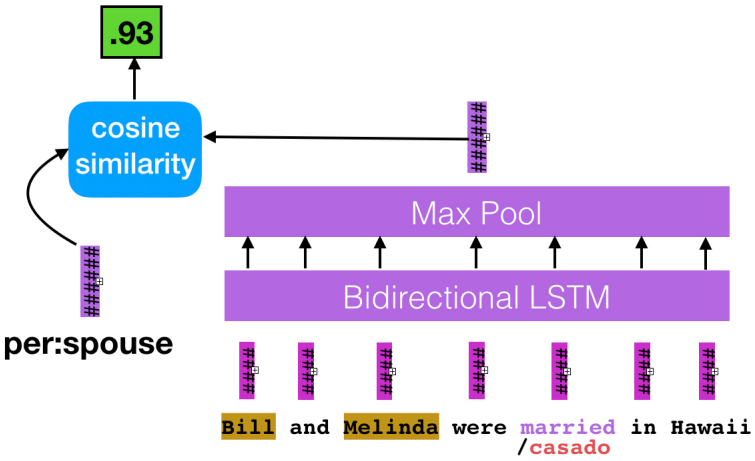
3.2 Model

3.2.1 Universal Schema as Sentence Classifier

As we discussed in Chapter 3.2.1, Riedel et al. (2013a) perform transductive learning, where a model is learned jointly over train and test data. Predictions are made by using the model to identify edges that were unobserved in the test data but likely to be true. The approach is vulnerable to the *cold start* problem in collaborative filtering (Schein et al., 2002): it is unclear how to form predictions for unseen entity pairs, without re-factorizing the entire matrix or applying heuristics.



(a)



(b)

Figure 3.2: Universal Schema jointly embeds KB and textual relations from Spanish and English, learning dense representations for entity pairs and relations using matrix factorization. a) Green cells indicate triples observed during training. Using transitivity through KB/English overlap and English/Spanish overlap (such as the entity pair 'Melinda, Bill' occurring in both languages), our model can predict that a text pattern in Spanish evidences a KB relation despite no overlap between Spanish/KB entity pairs. b) At test time we score compatibility between embedded KB relations and encoded textual patterns using cosine similarity. In our Spanish model we treat embeddings for a small set of English/Spanish translation pairs as a single word, e.g. casado and married.

In response, this work re-purposes USchema[1] as a means to train a sentence-level relation classifier. This allows us to avoid errors from aligning distant supervision to the corpus, but is more deployable for real world applications. It also provides opportunities in Section 3.2.4 to improve multilingual AKBC.

We produce predictions using a very simple approach: (1) scan the corpus and extract a large quantity of triplets (s, r_{text}, o), where r_{text} is an OpenIE pattern. For each triplet, if the similarity between the embedding of r_{text} and the embedding of a target relation r_{schema} is above some threshold, we predict the triplet $(s, r_{\text{schema}}, o)$, and its provenance is the input sentence containing (s, r_{text}, o). We refer to this technique as *pattern scoring*. In our experiments, we use the cosine distance between the vectors (Figure 7.2). In Section A.3, we discuss details for how to make this distance well-defined.

3.2.2 Using a Compositional Sentence Encoder to Predict Unseen Text Patterns

The pattern scoring approach is subject to an additional cold start problem: input data may contain patterns unseen in training. This section describes a method for using USchema to train a relation classifier that can take arbitrary context tokens (Section 2.2.3.2) as input.

Fortunately, the cold start problem for context tokens is more benign than that of entities since we can exploit statistical regularities of text: similar sequences of context tokens should be embedded similarly. Therefore, similar to Toutanova et al. (2015), we embed raw context tokens compositionally using a deep architecture. Unlike Riedel et al. (2013a), this requires no manual rules to map text to OpenIE patterns and can embed any possible input string. The modified USchema likelihood is:

[1]While universal schema is the general concept of jointly modeling text and structured data together, we will use USchema to refer to a particular concrete model which is equivalent to the matrix factorization model F from Riedel et al. (2013a)

$$\mathbb{P}\left((s, r, o)\right) = \sigma\left(u_{s,o}^{\top}\text{Encoder}(r)\right). \tag{3.1}$$

Here, if r is raw text, then Encoder(r) is parameterized by a deep architecture. If r is from the target schema, Encoder(r) is a produced by a lookup table (as in traditional USchema). Though such an encoder increases the computational cost of test-time prediction over straightforward pattern matching, evaluating a deep architecture can be done in large batches in parallel on a GPU.

Both convolutional networks (CNNs) and recurrent networks (RNNs) are reasonable encoder architectures, and we consider both in our experiments. CNNs have been useful in a variety of NLP applications (Collobert et al., 2011; Kalchbrenner et al., 2014; Kim, 2014). Unlike Toutanova et al. (2015), we also consider RNNs, specifically Long-Short Term Memory Networks (LSTMs) (Hochreiter and Schmidhuber, 1997a). LSTMs have proven successful in a variety of tasks requiring encoding sentences as vectors (Sutskever et al., 2014; Vinyals et al., 2014). In our experiments, LSTMs outperform CNNs.

There are two key differences between our sentence encoder and that of Toutanova et al. (2015). First, we use the encoder at test time, since we process the context tokens for held-out data. On the other hand, Toutanova et al. (2015) adopt the transductive approach where the encoder is only used to help train better representations for the relations in the target schema; it is ignored when forming predictions. Second, we apply the encoder to the raw text between entities, while Toutanova et al. (2015) first perform syntactic dependency parsing on the data and then apply an encoder to the path between the two entities in the parse tree. We avoid parsing, since we seek to perform multilingual AKBC, and many languages lack linguistic resources such as treebanks. Even parsing non-newswire English text, such as tweets, is extremely challenging.

3.2.3 Modeling Frequent Text Patterns

Despite the coverage advantages of using a deep sentence encoder, separately embedding each OpenIE pattern, as in Riedel et al. (2013a), has key advantages. In practice, we have found that many high-precision patterns occur quite frequently. For these, there is sufficient data to model them with independent embeddings per pattern, which imposes minimal inductive bias on the relationship between patterns. Furthermore, some discriminative phrases are idiomatic, i.e.. their meaning is not constructed compositionally from their constituents. For these, a sentence encoder may be inappropriate.

Therefore, pattern embeddings and deep token-based encoders have very different strengths and weaknesses. One values specificity, and models the head of the text distribution well, while the other has high coverage and captures the tail. In experimental results, we demonstrate that an ensemble of both models performs substantially better than either in isolation.

3.2.4 Multilingual Relation Extraction with Zero Annotation

The models described in previous two sections provide broad-coverage relation extraction that can generalize to all possible input entities and text patterns, while avoiding error-prone alignment of distant supervision to a corpus. Next, we describe techniques for an even more challenging generalization task: relation classification for input sentences in completely different languages.

Training a sentence-level relation classifier, either using the alignment-based techniques of Section 2.2.3.1, or the alignment-free method of Section 3.2.1, requires an available KB of seed facts that have supporting evidence in the corpus. Unfortunately, available KBs have low overlap with corpora in many languages, since KBs have cultural and geographical biases. In response, we perform multilingual relation extraction by jointly modeling a high-resource language, such as English, and an alter-

native language with no KB annotation. This approach provides transfer learning of a predictive model to the alternative language, and generalizes naturally to modeling more languages.

Extending the training technique of Section 3.2.1 to corpora in multiple languages can be achieved by factorizing a matrix that mixes data from a KB and from the two corpora. In Figure 3.1 we split the entities of a multilingual training corpus into sets depending on whether they have annotation in a KB and what corpora they appear in. We can perform transfer learning of a relation extractor to the low-resource language if there are entity pairs occurring in the two corpora, even if there is no KB annotation for these pairs. Note that we do not use the entity pair embeddings at test time: They are used only to bridge the languages during training. To form predictions in the low-resource language, we can simply apply the pattern scoring approach of Section 3.2.1.

In Section 4.3, we demonstrate that jointly learning models for English and Spanish, with no annotation for the Spanish data, provides fairly accurate Spanish AKBC, and even improves the performance of the English model. Note that we are not performing *zero-shot* learning of a Spanish model (Larochelle et al., 2008). The relations in the target schema are language-independent concepts, and we have supervision for these in English.

3.2.5 Tied Sentence Encoders

The sentence encoder approach of Section 5.3.2 is complementary to our multilingual modeling technique: we simply use a separate encoder for each language. This approach is sub-optimal, however, because each sentence encoder will have a separate matrix of word embeddings for its vocabulary, despite the fact that there may be considerable shared structure between the languages. In response, we propose

a straightforward method for tying the parameters of the sentence encoders across languages.

Drawing on the dictionary-based techniques described in Section 3.2.6, we first obtain a list of word-word translation pairs between the languages using a translation dictionary. The first layer of our deep text encoder consists of a word embedding lookup table. For the aligned word types, we use a single cross-lingual embedding. Details of our approach are described in Appendix A.5.

3.2.6 Multilingual Embeddings

Much work has been done on multilingual word embeddings. Most of this work uses aligned sentences from the Europarl dataset (Koehn, 2005) to align word embeddings across languages (Gouws et al., 2015; Luong et al., 2015; Hermann and Blunsom, 2014). Others (Mikolov et al., 2013b; Faruqui et al., 2014) align separate single-language embedding models using a word-level dictionary. Mikolov et al. (2013b) use translation pairs to learn a linear transform from one embedding space to another.

However, very little work exists on multilingual relation extraction. Faruqui and Kumar (2015) perform multilingual OpenIE relation extraction by projecting all languages to English using Google translate. However, as explained in Section 2.2.3.2 the OpenIE paradigm is not amenable to prediction within a fixed schema. Further, their approach does not generalize to low-resource languages where translation is unavailable – while we use translation dictionaries to improve our results, our experiments demonstrate that our method is effective even without this resource.

3.3 Task and System Description

We focus on the TAC KBP slot-filling task. Much related work on embedding knowledge bases evaluates on the FB15k dataset (Bordes et al., 2013; Wang et al.,

2014; Lin et al., 2015; Yang et al., 2015a; Toutanova et al., 2015). Here, relation extraction is posed as link prediction on a subset of Freebase. This task does not capture the particular difficulties we address: (1) evaluation on entities and text unseen during training, and (2) zero-annotation learning of a predictor for a low-resource language.

Also, note both Toutanova et al. (2015) and Riedel et al. (2013b) explore the pros and cons of learning embeddings for entity pairs vs. separate embeddings for each entity. As this is orthogonal to our contributions, we only consider entity pair embeddings, which performed best in both works when given sufficient data.

3.3.1 TAC Slot-Filling Benchmark

The aim of the TAC benchmark is to improve both coverage and quality of relation extraction evaluation compared to just checking the extracted facts against a knowledge base, which can be incomplete and where the provenances are not verified. In the slot-filling task, each system is given a set of paired query entities and relations or 'slots' to fill, and the goal is to correctly fill as many slots as possible along with provenance from the corpus. For example, given the query entity/relation pair (*Barack Obama, per:spouse*), the system should return the entity *Michelle Obama* along with sentence(s) whose text expresses that relation. The answers returned by all participating teams, along with a human search (with timeout), are judged manually for correctness, i.e. whether the provenance specified by the system indeed expresses the relation in question.

In addition to verifying our models on the 2013 and 2014 English slot-filling task, we evaluate our Spanish models on the 2012 TAC Spanish slot-filling evaluation. Because this TAC track was never officially run, the coverage of facts in the available annotation is very small, resulting in many correct predictions being marked incorrectly as precision errors. In response, we manually annotated all results returned by

the models considered in Table 3.3. Precision and recall are calculated with respect to the union of the TAC annotation and our new labeling[2].

3.3.2 Retrieval Pipeline

Our retrieval pipeline first generates all valid slot filler candidates for each query entity and slot, based on entities extracted from the corpus using FACTORIE (McCallum et al., 2009) to perform tokenization, segmentation, and entity extraction. We perform entity linking by heuristically linking all entity mentions from our text corpora to a Freebase entity using anchor text in Wikipedia. Making use of the fact that most Freebase entries contain a link to the corresponding Wikipedia page, we link all entity mentions from our text corpora to a Freebase entity by the following process: First, a set of candidate entities is obtained by following frequent link anchor text statistics. We then select that candidate entity for which the cosine similarity between the respective Wikipedia and the sentence context of the mention is highest, and link to that entity if a threshold is exceeded.

An entity pair qualifies as a candidate prediction if it meets the type criteria for the slot.[3] The TAC 2013 English and Spanish newswire corpora each contain about 1 million newswire documents from 2009–2012. The document retrieval and entity matching components of our relation extraction pipeline are based on RelationFactory (Roth et al., 2014a), the top-ranked system of the 2013 English slot-filling task. We also use the English distantly supervised training data from this system, which

[2]Following Surdeanu et al. (2012) we remove facts about undiscovered entities to correct for recall.

[3]Due to the difficulty of retrieval and entity detection, the maximum recall for predictions is limited. For this reason, Surdeanu et al. (2012) restrict the evaluation to answer candidates returned by their system and effectively rescaling recall. We do not perform such a re-scaling in our English results in order to compare to other reported results. Our Spanish numbers are rescaled. All scores reflect the 'anydoc' (relaxed) scoring to mitigate penalizing effects for systems not included in the evaluation pool.

aligns the TAC 2012 corpus to Freebase. More details on alignment are described in Appendix A.4.

As discussed in Section 3.2.3, models using a deep sentence encoder and using a pattern lookup table have complementary strengths and weaknesses. In response, we present results where we ensemble the outputs of the two models by simply taking the union of their individual outputs. Slightly higher results might be obtained through more sophisticated ensembling schemes.

3.3.3 Model Details

All models are implemented in Torch (code publicly available[4]). Models are tuned to maximize F1 on the 2012 TAC KBP slot-filling evaluation. We additionally tune the thresholds of our pattern scorer on a per-relation basis to maximize F1 using 2012 TAC slot-filling for English and the 2012 Spanish slot-filling development set for Spanish. As in Riedel et al. (2013b), we train using the BPR loss of Rendle et al. (2009). Our CNN is implemented as described in Toutanova et al. (2015), using width-3 convolutions, followed by tanh and max pool layers. The LSTM uses a bi-directional architecture where the forward and backward representations of each hidden state are averaged, followed by max pooling over time. See Section A.2

We also report results including an alternate names (AN) heuristic, which uses automatically-extracted rules to detect the TAC 'alternate name' relation. To achieve this, we collect frequent Wikipedia link anchor texts for each query entity. If a high probability anchor text co-occurs with the canonical name of the query in the same document, we return the anchor text as a slot filler.

[4] `https://github.com/patverga/torch-relation-extraction`

Model	Recall	Precision	F1
CNN	31.6	36.8	34.1
LSTM	32.2	39.6	**35.5**
USchema	29.4	42.6	34.8
USchema+LSTM	34.4	41.9	37.7
USchema+LSTM+Es	38.1	40.2	**39.2**
USchema+LSTM+AN	36.7	43.1	39.7
USchema+LSTM+Es+AN	40.2	41.2	**40.7**
Roth et al. (2014a)	35.8	45.7	40.2

Table 3.1: Precision, recall and F1 on the English TAC 2013 slot-filling task. AN refers to alternative names heuristic and Es refers to the addition of Spanish text at train time. LSTM+USchema ensemble outperforms any single model, including the highly-tuned top 2013 system of Roth et al. (2014a), despite using no handwritten patterns.

3.4 Experimental Results

In experiments on the English and Spanish TAC KBC slot-filling tasks, we find that both USchema and LSTM models outperform the CNN across languages, and that the LSTM tends to perform slightly better than USchema as the only model. Ensembling the LSTM and USchema models further increases final F1 scores in all experiments, suggesting that the two different types of model compliment each other well. Indeed, in Section 3.4.3 we present quantitative and qualitative analysis of our results which further confirms this hypothesis: the LSTM and USchema models each perform better on different pattern lengths and are characterized by different precision-recall trade-offs.

3.4.1 English TAC Slot-filling Results

Tables 3.1 and 3.2 present the performance of our models on the 2013 and 2014 English TAC slot-filling tasks. Ensembling the LSTM and USchema models improves F1 by 2.2 points for 2013 and 1.7 points for 2014 over the strongest single model on both evaluations, LSTM. Adding the alternative names (AN) heuristic described in Section 3.3.3 increases F1 by an additional 2 points on 2013, resulting in an F1

Model	Recall	Precision	F1
CNN	28.1	29.0	28.5
LSTM	27.3	32.9	**29.8**
USchema	24.3	35.5	28.8
USchema+LSTM	34.1	29.3	31.5
USchema+LSTM+Es	34.4	31.0	**32.6**

Table 3.2: Precision, recall and F1 on the English TAC 2014 slot-filling task. Es refers to the addition of Spanish text at train time. The AN heuristic is ineffective on 2014 adding only 0.2 to F1. Our system would rank 4/18 in the official TAC 2014 competition behind systems that use hand-written patterns and active learning despite our system using neither of these additional annotations (Surdeanu and Ji., 2014).

score that is competitive with the state-of-the-art. We also demonstrate the effect of jointly learning English and Spanish models on English slot-filling performance. Adding Spanish data improves our F1 scores by 1.5 points on 2013 and 1.1 on 2014 over using English alone. This places are system higher than the top performer at the 2013 TAC slot-filling task even though our system uses no hand-written rules.

The state of the art systems on this task all rely on matching handwritten patterns to find additional answers while our models use only automatically generated, indirect supervision; even our AN heuristics (Section 3.3.2) are automatically generated. The top two 2014 systems were Angeli et al. (2014) and RPI Blender (Surdeanu and Ji., 2014) who achieved F1 scores of 39.5 and 36.4 respectively. Both of these systems used additional active learning annotation. The third place team (Lin et al., 2014) relied on highly tuned patterns and rules and achieved an F1 score of 34.4.

Our model performs substantially better on 2013 than 2014 for two reasons. First, our RelationFactory (Roth et al., 2014a) retrieval pipeline was a top retrieval pipeline on the 2013 task, but was outperformed on the 2014 task which introduced new challenges such as confusable entities. Second, improved training using active learning gave the top 2014 systems a boost in performance. No 2013 systems, including ours, use active learning. Bentor et al. (2014), the 4th place team in the 2014 evaluation,

Model	Recall	Precision	F1
LSTM	9.3	12.5	10.7
LSTM+Dict	14.7	15.7	15.2
USchema	15.2	17.5	16.3
USchema+LSTM	21.7	14.5	17.3
USchema+LSTM+Dict	26.9	15.9	**20.0**

Table 3.3: Zero-annotation transfer learning F1 scores on 2012 Spanish TAC KBP slot-filling task. Adding a translation dictionary improves all encoder-based models. Ensembling LSTM and USchema models performs the best.

used the same retrieval pipeline (Roth et al., 2014a) as our model and achieved an F1 score of 32.1.

3.4.2 Spanish TAC Slot-filling Results

Table 3.3 presents 2012 Spanish TAC slot-filling results for our multilingual relation extractors trained using zero-annotation transfer learning. Tying word embeddings between the two languages results in substantial improvements for the LSTM. We see that ensembling the non-dictionary LSTM with USchema gives a slight boost over USchema alone, but ensembling the dictionary-tied LSTM with USchema provides a significant increase of nearly 4 F1 points over the highest-scoring single model, USchema. Clearly, grounding the Spanish data using a translation dictionary provides much better Spanish word representations. These improvements are complementary to the baseline USchema model, and yield impressive results when ensembled.

In addition to embedding semantically similar phrases from English and Spanish to have high similarity, our models also learn high-quality multilingual word embeddings. In Table 3.4 we compare Spanish nearest neighbors of English query words learned by the LSTM with dictionary ties versus the LSTM with no ties, using no unsupervised pre-training for the embeddings. Both approaches jointly embed Spanish and English word types, using shared entity embeddings, but the dictionary-tied model learns qualitatively better multilingual embeddings.

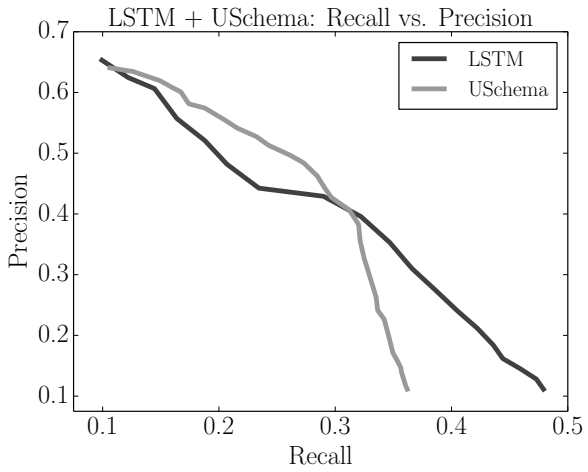


Figure 3.3: Precision-Recall curves for USchema and LSTM on 2013 TAC slot-filling. USchema achieves higher precision values whereas LSTM has higher recall.

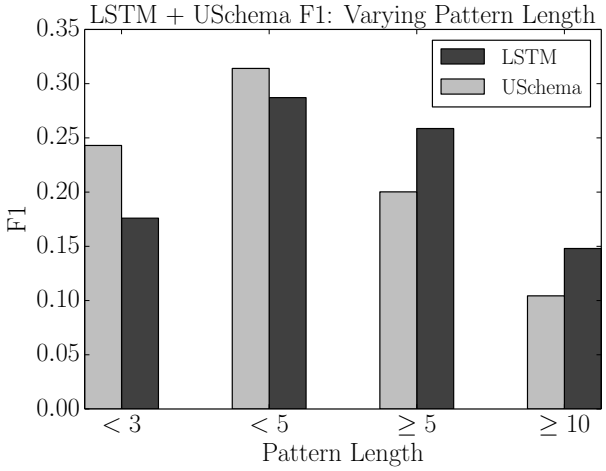


Figure 3.4: F1 achieved by USchema vs. LSTM models for varying pattern token lengths on 2013 TAC slot-filling. LSTM performs better on longer patterns whereas USchema performs better on shorter patterns.

3.4.3 USchema vs LSTM

We further analyze differences between USchema and LSTM in order to better understand why ensembling the models results in the best performing system. Figure 3.3 depicts precision-recall curves for the two models on the 2013 slot-filling task. As observed in earlier results, the LSTM achieves higher recall at the loss of some precision, whereas USchema can make more precise predictions at a lower threshold for recall. In Figure 3.4 we observe evidence for these different precision-recall trade-offs:

USchema scores higher in terms of F1 on shorter patterns whereas the LSTM scores higher on longer patterns. As one would expect, USchema successfully matches more short patterns than the LSTM, making more precise predictions at the cost of being unable to predict on patterns unseen during training. The LSTM can predict using any text between entities observed at test time, gaining recall at the loss of precision. Combining the two models makes the most of their strengths and weaknesses, leading to the highest overall F1.

Qualitative analysis of our English models also suggests that our encoder-based models (LSTM) extract relations based on a wide range of semantically similar patterns that the pattern-matching model (USchema) is unable to score due to a lack of exact string match in the test data. For example, Table 3.5 lists three examples of the *per:children* relation that the LSTM finds which USchema does not, as well as three patterns that USchema does find. Though the LSTM patterns are all semantically and syntactically similar, they each contain different specific noun phrases, e.g. *Lori*, *four children*, *toddler daughter*, *Lee and Albert*, etc. Because these specific nouns weren't seen during training, USchema fails to find these patterns whereas the LSTM learns to ignore the specific nouns in favor of the overall pattern, that of a parent-child relationship in an obituary. USchema is limited to finding the relations represented by patterns observed during training, which limits the patterns matched at test-time to short and common patterns; all the USchema patterns matched at test time were similar to those listed in Table 3.5: variants of *'s son, '*.

3.5 Conclusion

By jointly embedding English and Spanish corpora along with a KB, we can train an accurate Spanish relation extraction model using no direct annotation for relations in the Spanish data. This approach has the added benefit of providing significant accuracy improvements for the English model, outperforming the top system on the

2013 TAC KBC slot filling task, without using the hand-coded rules or additional annotations of alternative systems. By using deep sentence encoders, we can perform prediction for arbitrary input text and for entities unseen in training. Sentence encoders also provides opportunities to improve cross-lingual transfer learning by sharing word embeddings across languages. In future work we will apply this model to many more languages and domains besides newswire text. We would also like to avoid the entity detection problem by using a deep architecture to both identify entity mentions and identify relations between them.

CEO	
Dictionary	No Ties
jefe (chief)	CEO
CEO	director (principle)
ejecutivo (executive)	directora (director)
cofundador (co-founder)	firma (firm)
president (chairman)	magnate (tycoon)
headquartered	
Dictionary	No Ties
sede (headquarters)	Geológico (Geological)
situado (located)	Treki (Treki)
selectivo (selective)	Geofísico(geophysical)
profesional (vocational)	Normandía (Normandy)
basándose (based)	emplea (uses)
hubby	
Dictionary	No Ties
matrimonio (marriage)	esposa (wife)
casada (married)	esposo (husband)
esposa (wife)	casada(married)
casó (married)	embarazada (pregnant)
embarazada (pregnant)	embarazo (pregnancy)
alias	
Dictionary	No Ties
simplificado (simplified)	Weaver (Weaver)
sabido (known)	interrogación (question)
seudónimo (pseudonym)	alias
privatización (privatization)	reelecto (reelected)
nombre (name)	conocido (known)

Table 3.4: Example English query words (not in translation dictionary) in bold with their top nearest neighbors by cosine similarity listed for the dictionary and no ties LSTM variants. Dictionary-tied nearest neighbors are consistently more relevant to the query word than untied.

LSTM
McGregor *is survived by his wife, Lori, and four children, daughters Jordan,* **Taylor** and Landri, and a son, Logan.
In addition to his wife, **Mays** *is survived by a toddler daughter and a son,* **Billy Mays Jr.**, who is in his 20s.
Anderson *is survived by his wife Carol, sons Lee and Albert, daughter* **Shirley Englebrecht** and nine grandchildren.
USchema
Dio *'s son,* **Dan Padavona**, cautioned the memorial crowd to be screened regularly by a doctor and take care of themselves, something he said his father did not do.
But **Marshall** *'s son,* **Philip**, told a different story.
"I'd rather have Sully doing this than some stranger, or some hotshot trying to be the next Billy Mays," said the guy who actually is the next **Billy Mays***, his son* **Billy Mays III**.

Table 3.5: Examples of the *per:children* relation discovered by the LSTM and Universal Schema. Entities are bold and patterns italicized. The LSTM models a richer set of patterns

CHAPTER 4

ROWLESS UNIVERSAL SCHEMA

In its original form, universal schema can reason only about row entries and column entries explicitly seen during training. Unseen rows and columns observed at test time do not have a learned embedding. This problem is referred to as the *cold-start* problem in recommendation systems (Schein et al., 2002).

In Chapter 3, we discussed 'columnless' versions of universal schema models that generalize to unseen column entries. They learn compositional pattern encoders to parameterize the column matrix in place of individual column embeddings. However, these models still do not generalize to unseen row entries.

In this chapter, we present a 'row-less' extension of universal schema that generalizes to unseen entities and entity pairs. Rather than representing each row entry with an explicit dense vector, we encode each entity or entity pair as aggregate functions over their observed column entries. This is beneficial because when new entities are mentioned in text documents and subsequently added to the KB, we can directly reason on the observed text evidence to infer new binary relations and entity types for the new entities. This avoids the cumbersome effort of re-training the whole model from scratch to learn embeddings for the new entities.

To construct the row representation, we compare various aggregation functions in our experiments. We consider query independent and dependent aggregation functions. We find that query dependent attentional models that selectively focus on relevant evidence outperform the query independent alternatives. The query dependent attention mechanism also helps in providing a direct connection between the

prediction and its provenance. Additionally, our models have a much smaller memory footprint since they do not store explicit row representations.

It is important to note that our approach is different from sentence level classifiers that predict KB relations and entity types using a single sentence as evidence. First, we pool information from multiple pieces of evidence coming from both text and annotated KB facts, rather than considering a single sentence at test time. Second, our methods are not limited to a fixed schema but instead predict a richer set of labels (KB types and textual), enabling easier downstream processing closer to natural language interaction with the KB. Finally, our model gains additional training signal from multi-task learning of textual and KB types. Since universal schema leverages large amounts of unlabeled text we desire the benefits of entity pair modeling, and row-less universal schema facilitates learning entity pair representations without the drawbacks of the traditional one-embedding-per-pair approach.

The majority of current embedding methods for KB entity type prediction operate with explicit entity representations (Yao et al., 2013; Neelakantan and Chang, 2015a) and hence, cannot generalize to unseen entities. In relation extraction, entity-level models (Nickel et al., 2011b; García-Durán et al., 2016; Yang et al., 2015a; Bordes et al., 2013; Wang et al., 2014; Lin et al., 2015; Socher et al., 2013) can handle unseen entity pairs at test time. These models learn representations for the entities instead of entity pairs. Hence, these methods still cannot generalize to predict relations between an entity pair if even one of the entities is unseen. Moreover, Toutanova et al. (2015) and Riedel et al. (2013b) observe that the entity pair model outperforms entity models in cases where the entity pair was seen at training time.

Most similar to this work, Neelakantan et al. (2015a) classify KB relations by finding the maximum scoring path between two entities. This model is also 'row-less' and does not directly model entities or entity pairs. There are several important differences in this work. Neelakantan et al. (2015a) learn per-relation classifiers to

predict only a small set of KB relations, while we instead predict all relations, including textual relations. We also explore aggregation functions that pool evidence from multiple paths while Neelakantan et al. (2015a) only chose the maximum scoring path. Additionally, we demonstrate that our models can perform on par with those with explicit row representations while Neelakantan et al. (2015a) did not perform this comparison.

In this work we investigate universal schema models without explicit row representations on two tasks: entity type prediction and relation extraction. We use entity type and relation facts from Freebase (Bollacker et al., 2008a) augmented with textual relations and types from Clueweb text (Orr et al., 2013; Gabrilovich et al., 2013). We explore multiple aggregation functions and find that an attention-based aggregation function outperforms several simpler functions and matches a model using explicit row representations with an order of magnitude fewer parameters. More importantly, we then demonstrate that our 'row-less' models accurately predict relations on unseen entity pairs and types on unseen entities.

4.1 Model

In this section, we describe the model, discuss the different aggregation functions and give details on the training objective.

4.1.1 'Row-less' Universal Schema

While column-less universal schema addresses reasoning over arbitrary textual patterns, it is still limited to reasoning over row entries seen at training time. Verga et al. (2016a) use column-less universal schema for relation extraction. They address the problem of unseen row entries by using universal schema as a sentence classifier – directly comparing a textual relation to a KB relation to perform relation extraction. However, this approach is unsatisfactory for two reasons. The first is that this cre-

ates an inconsistency between training and testing. The model is trained to predict compatibility between rows and columns, but at test time it predicts compatibility between relations directly. Second, it considers only a single piece of evidence in making its prediction.

We address both of these concerns in our 'row-less' universal schema. Rather than explicitly encoding each row, we encode the row as a learned aggregation over their observed columns (Figure 4.1). A row contains an entity for type prediction and an entity pair for relation extraction while a column contains a relation type for relation extraction and an entity type for type prediction. A learned row embedding can be seen as a summarization of all columns observed with that particular row. Instead of modeling this summarization as a single embedding, we reconstruct a row representation from an aggregate of its column embeddings, essentially learning a mixture model rather than a single centroid.

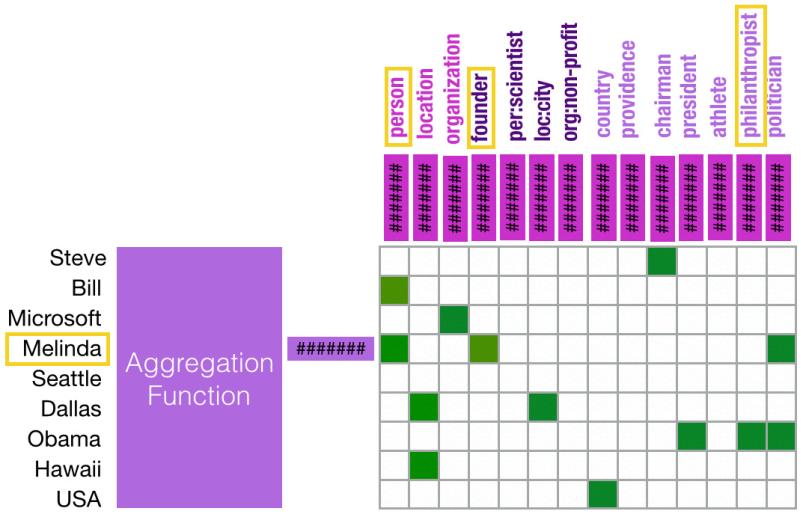


Figure 4.1: Row-less universal schema for relation extraction encodes an entity pair as an aggregation of its observed relation types.

4.1.2 Aggregation Functions

In this work we examine four aggregation functions to construct the representations for the row. Let $v(.)$ denote a function that returns the vector representation

for rows and columns. To model the probability between row r and column c, we consider the set $V\bar{(}r)$ which contains the set of column entries that are observed with row r at training time, i.e.,

$$\forall \bar{c} \in V\bar{(}r), (r, \bar{c}) \in T$$

The first two aggregation functions create a single representation for each row independent of the query. **Mean Pool** creates a single centroid for the row by averaging all of its column vectors,

$$v(r) = \sum_{\bar{c} \in V\bar{(}r)} v(\bar{c})$$

While this formulation intuitively makes sense as an approximation for the explicit row representation, averaging large numbers of embeddings can lead to a noisy representation.

Max Pool also creates a single representation for the row by taking a dimension-wise max over the observed column vectors:

$$v(r)_i = \max_{\bar{c} \in V\bar{(}r)} v(\bar{c})_i, \forall i \in 1, 2, \ldots, d$$

where a_i denotes the ith dimension of vector a. Both mean pool and max pool are query-independent and form the same representation for the row regardless of the query relation.

We also examine two query-specific aggregation functions. These models are more expressive than a single vector forced to to act as a centroid to all possible columns observed with that particular row. For example, the entity pair Bill and Melinda Gates could hold the relation 'per:spouse' or 'per:co-worker'. A query-specific aggregation mechanism can produce separate representations for this entity pair dependent on the query.

The **Max Relation** aggregation function represents the row as its most similar column to the query vector of interest. Given a query relation c,

$$c_{max} = argmax_{\bar{c} \in \bar{V(r)}} v(\bar{c}).v(c)$$
$$v(r) = v(c_{max})$$

A similar strategy has been successfully applied in previous work (Weston et al., 2013; Neelakantan et al., 2014, 2015a) for different tasks. This model has the advantage of creating a query-specific entity pair representation, but is more susceptible to noisy training data as a single incorrect piece of evidence could be used to form a prediction.

Finally, we look at an **Attention** aggregation function over columns (Figure 4.2) which is similar to a single-layer memory network Sukhbaatar et al. (2015). The *soft attention* mechanism has been used to selectively focus on relevant parts in many different models (Bahdanau et al., 2015; Graves et al., 2014; Neelakantan et al., 2016).

In this model the query is scored with an input representation of each column embedding followed by a softmax, giving a weighting over each relation type. This output is then used to get a weighted sum over a set of output representations for each column resulting in a query-specific vector representation of the row. Given a query relation c,

$$score_{\bar{c}} = v(c).v(\bar{c}), \forall \bar{c} \in \bar{V(r)}$$
$$p_{\bar{c}} = \frac{exp(score_{\bar{c}})}{\sum_{\hat{c} \in \bar{V(r)}} exp(score_{\hat{c}})}, \forall \bar{c} \in \bar{V(r)}$$
$$v(r) = \sum_{\bar{c} \in \bar{V(r)}} p_{\bar{c}} \times v(\bar{c})$$

The model pools relevant information over the entire set of observed columns and selects the most salient aspects to the query.

Model	Parameters
Entity Embeddings	3.7 e6
Attention	3.1 e5
Mean Pool/Max Pool/Max Relation	1.5 e5

Table 4.1: Number of parameters for the different models on the entity type dataset.

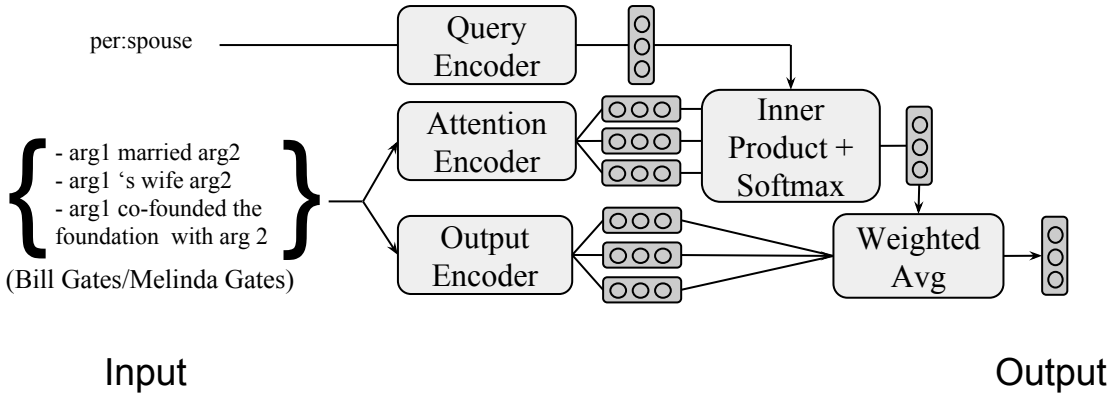


Figure 4.2: Example attention model in a row-less universal schema relation extractor. In the attention model, we compute the dot product between the representation of the query relation and the representation of an entity pair's observed relation type followed by a softmax, giving a weighting over the observed relation types. This output is then used to get a weighted sum over the set of representations of the observed relation types. The result is a query-specific vector representation of the entity pair. The Max Relation model takes the most similar observed relation's representation.

4.1.3 Training

The vector representation of the rows and the columns are the parameters of the model. Riedel et al. (2013b) use Bayesian Personalized Ranking (BPR) (Rendle et al., 2009) to train their universal schema models. BPR ranks the probability of observed triples above unobserved triples rather than explicitly modeling unobserved edges as negative. Each training example is an (entity pair, relation type) or (entity, entity type) pair observed in the training text corpora or KB.

Rather than BPR, Toutanova et al. (2015) use 200 negative samples to approximate the negative log likelihood[1]. In our experiments, we use the sampled approximate negative log likelihood which outperformed BPR in early experiments.

[1]Many past papers restrict negative samples to be of the same type as the positive example. We simply sample uniformly from the entire set of row entries

Each example in the training procedure consists of a row-column pair observed in the training set. For a positive example $(r, c) \in T$, we construct the set $V\bar{(}r)$ containing all the other column entries apart from c that are observed with row r.

To make training faster and more robust, we add 'pattern dropout' for entity pairs with many mentions. We set $V\bar{(}r)$ to be m randomly sampled mentions for entity pairs with greater than m total mentions. In our experiments we set $m = 10$ and at test time we use all mentions. We then use $V\bar{(}r)$ to obtain the aggregated row representation as discussed above.

We randomly sample 200 columns unobserved with row r to act as the negative samples. All models are implemented in Torch[2] and are trained using Adam Kingma and Ba (2014a) with default momentum related hyperparameters.

4.2 Related Work

Relation extraction for KB completion has a long history. Mintz et al. (2009a) train per relation linear classifiers using features derived from the sentences in which the entity pair is mentioned. Most of the embedding-based methods learn representations for entities (Nickel et al., 2011b; Socher et al., 2013; Bordes et al., 2013) whereas Riedel et al. (2013b) use entity pair representations.

'Column-less' versions of Universal Schema have been proposed (Toutanova et al., 2015; Verga et al., 2016a). These models can generalize to column entries unseen at training by learning compositional pattern encoders to parameterize the column matrix in place of embeddings. Most of these models do not generalize to unseen entity pairs and none of them generalize to unseen entities. Recently, Neelakantan et al. (2015a) introduced a multi-hop relation extraction model that is 'row-less' having no explicit parameters for entity pairs and entities.

[2]data and code available at `https://github.com/patverga/torch-relation-extraction/tree/rowless-updates`

Entity type prediction at the individual sentence level has been studied extensively (Pantel et al., 2012; Ling and Weld, 2012b; Shimaoka et al., 2016). More recently, embedding-based methods for knowledge base entity type prediction have been proposed (Yao et al., 2013; Neelakantan and Chang, 2015a). These methods have explicit entity representations, hence cannot generalize to unseen entities.

The task of generalizing to unseen row and column entries is referred to as the *cold-start* problem in recommendation systems. Methods proposed to tackle this problem commonly use user and item content and attributes (Schein et al., 2002; Park and Chu, 2009).

Multi-instance learning can be viewed as the relation classifier analogy of rowless universal schema. Riedel et al. (2010) used a relaxation of distant supervision training where all sentences for an entity pair (bag) are considered jointly and only the most relevant sentence is treated as the single training example for the bag's label. Surdeanu et al. (2012) extended this idea with multi-instance multi-label learning (MIML) where each entity pair / bag can hold multiple relations / labels. Recently Lin et al. (2016) used a selective attention over sentences in MIML.

Concurrent to our work, Weissenborn (2016) proposes a row-less method for relation extraction considering both a uniform and weighted average aggregation function over columns. However, Weissenborn (2016) did not experiment with max and max-pool aggregation functions or evaluate on entity-type prediction. They also did not combine the rowless model with an LSTM column-less parameterization and did not compare to a model with explicit entity-pair representations.

4.3 Experimental Results

In this section, we compare our models that have aggregate row representations with models that have explicit row representations on entity type prediction and relation extraction tasks. Finally, we perform experiments on a column-less universal

schema model. Table 4.1 shows that the row-less models require far fewer parameters since they do not explicitly store the row representations.

4.3.1 Entity Type Prediction

We first evaluate our models on an entity type prediction task. We collect all entities along with their types from a dump of Freebase[3]. We then filter all entities with less than five Freebase types leaving a set of 844780 (entity, type) pairs. Additionally, we collect 712072 textual (entity, type) pairs from Clueweb. The textual types are the 5000 most common appositives extracted from sentences mentioning entities. This results in 140513 unique entities, 1120 Freebase types, and 5000 free text types.

All embeddings are 25 dimensions, randomly initialized. We tune learning rates from {.01, .001}, ℓ_2 from {1e-8, 0}, batch size {512, 1024, 2048} and negative samples from {2, 200}.

For evaluation, we split the Freebase (entity, type) pairs into 60% train, 20% validation, and 20% test. We randomly generate 100 negative (entity, type) pairs for each positive pair in our test set by selecting random entity and type combinations. We filter out false negatives that were observed true (entity, type) pairs in our complete data set. Each model produces a score for each positive and negative (entity, type) pair where the type is the query. We then rank these predictions, calculate average precision for each of the types in our test set, and then use those scores to calculate mean average precision (MAP).

Table 4.2a shows the results of this experiment. We can see that the query dependent aggregation functions (Attention and Max Relation) performs better than the query independent functions (Mean Pool and Max Pool). The performance of models

[3]Downloaded March 1, 2015.

Model	MAP
Entity Embeddings	54.81
Mean Pool	39.47
Max Pool	32.59
Attention	**55.66**
Max Relation	**55.37**

(a)

Model	MAP
Entity Embeddings	3.14
Mean columns	34.77
Max column	43.20
Mean Pool	35.53
Max Pool	30.98
Attention	**54.52**
Max Relation	**54.72**

(b)

Table 4.2: Entity type prediction. Entity embeddings refers to the model with explicit row representations. Mean Columns and Max Column are equivalent to Mean Pool and Max Relation respectively (Section 4.1.2) but use the column embeddings learned during training of the Entity Embeddings model. b: Positive entities are unseen at train time.

with query dependent aggregation functions which have far fewer parameters match the performance of the model with explicit entity representations.

We additionally evaluate our model's ability to predict types for entities unseen during training. For this experiment, we randomly select 14000 entities and take all (entity, type) pairs containing those entities. We remove these pairs from our training set and use them as positive samples in our test set. We then select 100 negatives (entity, type) pairs per positive as above.

Table 4.2b shows the results of the experiment with unseen entities. There is very little performance drop for models trained with query dependent aggregation functions. The performance of the model with explicit entity representations is close to random.

4.3.1.1 Qualitative Results

A query specific aggregation function is able to pick out relevant columns to form a prediction. This is particularly important for rows that are not described easily by a single centroid such as an entity with several very different careers or an entity pair with multiple highly varied relations. For example, in the first row in Table 4.3,

Query	Observed Columns
/baseball/baseball_player	/**sports/pro_athlete**, /sports/sports_award_winner, /tv/tv_actor, /people/measured_person, /award/award_winner, /people/person
/architecture/engineer	**engineer**, /book/author, /projects/project_focus, /people/person, sir
/baseball/baseball_player	**baseman**, /sports/pro_athlete, /people/measured_person, /people/person, dodgers, coach
/computer/computer_scientist	/**education/academic**, /music/group_member, /music/artist, /people/person
/business/board_member	/**organization/organization_founder**, /award/award_winner, /computer/computer_scientist, /people/person, president, scientist
/education/academic	/**astronomy/astronomer**, /book/author

Table 4.3: Each row corresponds to a true query entity type (left column) and the observed entity types (right column) for a particular entity. The maximum scoring observed entity type for each query entity type is indicated in bold. The other types are in no particular order. It can be seen that the maximum scoring entity types are interpretable.

for the query */baseball/baseball_player* the model needs to correctly focus on aspects like */sports/pro_athlete* and ignore evidence information like */tv/tv_actor*. A model that creates a single query-independent centroid will be forced to try and merge these disparate pieces of information together.

4.3.2 Relation Extraction

We evaluate our models on a relation extraction task using the FB15k-237 dataset from Toutanova et al. (2015). The data is composed of a small set of 237 Freebase relations and approximately 4 million textual patterns from Clueweb with entities linked to Freebase Gabrilovich et al. (2013). In past studies, for each (subject, relation, object) test triple, negative examples are generated by replacing the object with all other entities, filtering out triples that are positive in the data set. The positive triple is then ranked among the negatives. In our experiments we limit the possible generated negatives to those entity pairs that have textual mentions in our training

set. This way we can evaluate how well the model classifies textual mentions as Freebase relations. We also filter textual patterns with length greater than 35. Our filtered data set contains 2740237 relation types, 2014429 entity pairs, and 176476 tokens. We report the percentage of positive triples ranked in the top 10 amongst their negatives as well as the MRR scaled by 100.

Models are tuned to maximize mean reciprocal rank (MRR) on the validation set with early stopping. The entity pair model used a batch size 1024, $\ell_2 = 1e\text{-}8$, $\epsilon = 1e\text{-}4$, and learning rate 0.01. The aggregation models all used batch size 4096, $\ell_2 = 0$, $\epsilon = 1e\text{-}8$, and learning rate 0.01. Each use 200 negative samples except for max pool which performed better with two negative samples. The column vectors are initialized with the columns learned by the entity pair model. Randomly initializing the query encoders and tying the output and attention encoders performed better and all results use this method. All models are trained with embedding dimension 25.

Our results are shown in Table 4.4a. We can see that the models with query specific aggregation functions give the same results as models with explicit entity pair representations. The Max Relation model performs competitively with the Attention model which is not entirely surprising as it is a simplified version of the Attention model. Further, the Attention model reduces to the Max Relation model for entity pairs with only a single observed relation type. In our data, 64.8% of entity pairs have only a single observed relation type and 80.9% have 1 or 2 observed relation types.

We also explore the models' abilities to predict on unseen entity pairs (Table 4.4b). We remove all training examples that contain a positive entity pair in either our validation or test set. We use the same validation and test set as in Table 4.4a. The entity pair model predicts random relations as it is unable to make predictions on unseen entity pairs. The query-independent aggregation functions, mean pool

Model	MRR	Hits@10
Entity-pair Embed	31.85	51.72
Mean Pool	25.89	45.94
Max Pool	29.61	49.93
Attention	31.92	51.67
Max Relation	31.71	51.94

(a)

Model	MRR	Hits@10
Entity-pair Embed	5.23	11.94
Mean Pool	18.10	35.76
Max Pool	20.80	40.25
Attention	**29.75**	49.69
Max Relation	28.46	48.15

(b)

Table 4.4: The percentage of positive triples ranked in the top 10 amongst their negatives as well as the mean reciprocal rank (MRR) scaled by 100 on a subset of the FB15K-237 dataset. All positive entity pairs in the evaluation set are unseen at train time. Entity-pair embeddings refers to the model with explicit row representations. b: Predicting entity pairs that are not seen at train time.

and max pool, perform better than models with explicit entity pair representations. Again, query specific aggregation functions get the best results, with the Attention model performing slightly better than the Max Relation model.

The two experiments indicate that we can train relation extraction models without explicit entity pair representations that perform as well as models with explicit representations. We also find that models with query specific aggregation functions accurately predict relations for unseen entity pairs.

4.3.3 'Column-less' universal schema

The original universal schema approach has two main drawbacks: similar textual patterns do not share statistics, and the model is unable to make predictions about entities and textual patterns not explicitly seen at train time.

Recently, 'column-less' versions of universal schema to address some of these issues (Toutanova et al., 2015; Verga et al., 2016a). These models learn compositional pattern encoders to parameterize the column matrix in place of direct embeddings. Compositional universal schema facilitates more compact sharing of statistics by composing similar patterns from the same sequence of word embeddings – the text patterns 'lives in the city' and 'lives in the city of' no longer exist as distinct atomic units. More importantly, compositional universal schema can thus generalize to all possible textual patterns, facilitating reasoning over arbitrary text at test time.

The column-less universal schema model generalizes to all possible input textual relations and the row-less model generalizes to all entities and entity pairs, whether seen at train time or not. We can combine these two approaches together to make an universal schema model that generalizes to unseen rows and columns.

The parse path between the two entities in the sentence is encoded with an LSTM model. We use a single layer model with 100 dimensional token embeddings initialized randomly. To prevent exploding gradients, we clip them to norm 10 while all the other hyperparameters are tuned the same way as before. We follow the same evaluation protocol from 4.3.2.

The results of this experiment with observed rows are shown in Table 4.5a. While both the MRR and Hits@10 metrics increase for models with explicit row representations, the row-less models show an improvement only on the Hits@10 metric. The MRR of the query dependent row-less models is still competitive with the model with explicit row representation even though they have far fewer parameters to fit the data.

4.4 Conclusion

In this chapter we explored a row-less extension of universal schema that forgoes explicit row representations for an aggregation function over its observed columns. This extension allows prediction between all rows in new textual mentions – whether

Model	MRR	Hits@10
Entity-pair Embed	31.85	51.72
Entity-pair Embed-LSTM	**33.37**	54.39
Attention	31.92	51.67
Attention-LSTM	30.00	53.35
Max Relation	31.71	51.94
Max Relation-LSTM	30.77	**54.80**

(a)

Model	MRR	Hits@10
Entity-pair Embed	5.23	11.94
Attention	**29.75**	49.69
Attention-LSTM	27.95	51.05
Max Relation	28.46	48.15
Max Relation-LSTM	29.61	**54.19**

(b)

Table 4.5: The percentage of positive triples ranked in the top 10 amongst their negatives as well as the mean reciprocal rank (MRR) scaled by 100 on a subset of the FB15K-237 dataset. Negative examples are restricted to entity pairs that occurred in the KB or text portion of the training set. Models with the suffix "-LSTM" are column-less. Entity-pair embeddings refers to the model with explicit row representations. b: Predicting entity pairs that are not seen at train time.

seen at train time or not – and also provides a natural connection to the provenance supporting the prediction. Our models also have a smaller memory footprint.

In this work we show that an aggregation function based on query-specific attention over relation types outperforms query independent aggregations. We show that aggregation models are able to predict on par with models with explicit row representations on seen row entries with far fewer parameters. More importantly, aggregation models predict on unseen row entries without much loss in accuracy. Finally, we show that in relation extraction, we can combine row-less and column-less models to train models that generalize to both unseen rows and columns.

CHAPTER 5

ENCODING HIERARCHIES

So far, we've discussed extraction from raw text to a knowledge base of entities and fine-grained types that has cast the problem as a prediction into a flat set of entity and type labels. However, this neglects the rich hierarchies that naturally exist over types and entities and are already present in many curated ontologies. Previous attempts to incorporate hierarchical structure have yielded little benefit and have been restricted to very shallow ontologies. In this chapter, we present new methods using real and complex bilinear mappings for integrating hierarchical information, yielding substantial improvement over flat predictions in entity linking and fine-grained entity typing, and achieving state-of-the-art results for end-to-end models on the benchmark FIGER dataset. We also present two new human-annotated datasets containing wide and deep hierarchies which we will release to the community to encourage further research in this direction: *MedMentions*, a collection of PubMed abstracts in which 246k mentions have been mapped to the massive UMLS ontology; and *TypeNet*, which aligns Freebase types with the WordNet hierarchy to obtain nearly 2k entity types. In experiments on all three datasets we show substantial gains from hierarchy-aware training.

5.1 Introduction

Identifying and understanding entities is a central component in knowledge base construction Roth et al. (2015) and essential for enhancing downstream tasks such as relation extraction Yaghoobzadeh et al. (2017b), question answering Das et al.

(2017c); Welbl et al. (2017) and search Dalton et al. (2014). This has led to considerable research in automatically identifying entities in text, predicting their types, and linking them to existing structured knowledge sources.

Current state-of-the-art models encode a textual mention with a neural network and classify the mention as being an instance of a fine grained type or entity in a knowledge base. Although in many cases the types and their entities are arranged in a hierarchical ontology, most approaches ignore this structure, and previous attempts to incorporate hierarchical information yielded little improvement in performance (Shimaoka et al., 2017). Additionally, existing benchmark entity typing datasets only consider small label sets arranged in very shallow hierarchies. For example, FIGER Ling and Weld (2012a), the *de facto* standard fine grained entity type dataset, contains only 113 types in a hierarchy only two levels deep.

In this chapter, we investigate models that explicitly integrate hierarchical information into the embedding space of entities and types, using a hierarchy-aware loss on top of a deep neural network classifier over textual mentions. By using this additional information, we learn a richer, more robust representation, gaining statistical efficiency when predicting similar concepts and aiding the classification of rarer types. We first validate our methods on the narrow, shallow type system of FIGER, out-performing state-of-the-art methods not incorporating hand-crafted features and matching those that do.

To evaluate on richer datasets and stimulate further research into hierarchical entity/typing prediction with larger and deeper ontologies, we introduce two new human annotated datasets. The first is *MedMentions*, a collection of PubMed abstracts in which 246k concept mentions have been annotated with links to the Unified Medical Language System (UMLS) ontology Bodenreider (2004), an order of magnitude more annotations than comparable datasets. UMLS contains over 3.5 million concepts in a hierarchy having average depth 14.4. Interestingly, UMLS does not distinguish be-

tween types and entities (an approach we heartily endorse), and the technical details of linking to such a massive ontology lead us to refer to our MedMentions experiments as entity linking. Second, we present *TypeNet*, a curated mapping from the Freebase type system into the WordNet hierarchy. TypeNet contains over 1900 types with an average depth of 7.8.

In experimental results, we show improvements with a hierarchically-aware training loss on each of the three datasets. In entity-linking MedMentions to UMLS, we observe a 6% relative increase in accuracy over the base model. In experiments on entity-typing from Wikipedia into TypeNet, we show that incorporating the hierarchy of types and including a hierarchical loss provides a dramatic 29% relative increase in MAP. Our models even provide benefits for shallow hierarchies allowing us to match the state-of-art results of Shimaoka et al. (2017) on the FIGER (GOLD) dataset without requiring hand-crafted features.

5.2 New Corpora and Ontologies

5.2.1 MedMentions

Over the years researchers have constructed many large knowledge bases in the biomedical domain Apweiler et al. (2004); Davis et al. (2008); Chatr-aryamontri et al. (2017). Many of these knowledge bases are specific to a particular sub-domain encompassing a few particular types such as genes and diseases Piñero et al. (2017).

UMLS Bodenreider (2004) is particularly comprehensive, containing over 3.5 million concepts (UMLS does not distinguish between entities and types) defining their relationships and a curated hierarchical ontology. For example *LETM1 Protein* IS-A *Calcium Binding Protein* IS-A *Binding Protein* IS-A *Protein* IS-A *Genome Encoded Entity*. This fact makes UMLS particularly well suited for methods explicitly exploiting hierarchical structure.

Accurately linking textual biological entity mentions to an existing knowledge base is extremely important but few richly annotated resources are available. Even when resources do exist, they often contain no more than a few thousand annotated entity mentions which is insufficient for training state-of-the-art neural network entity linkers. State-of-the-art methods must instead rely on string matching between entity mentions and canonical entity names Leaman et al. (2013); Wei et al. (2015a); Leaman and Lu (2016). To address this, we constructed MedMentions, a new, large dataset identifying and linking entity mentions in PubMed abstracts to specific UMLS concepts. Professional annotators exhaustively annotated UMLS entity mentions from 3704 PubMed abstracts, resulting in 246,000 linked mention spans. The average depth in the hierarchy of a concept from our annotated set is 14.4 and the maximum depth is 43.

MedMentions contains an order of magnitude more annotations than similar biological entity linking PubMed datasets (Doğan et al., 2014; Wei et al., 2015a; Li et al., 2016a). Additionally, these datasets contain annotations for only one or two entity types (genes or chemicals and disease etc.). MedMentions instead contains annotations for a wide diversity of entities linking to UMLS. Statistics for several other datasets are in Table 5.1 and further statistics are in Appendix-B.1.

Dataset	mentions	unique entities
MedMentions	246,144	25,507
BCV-CDR	28,797	2,356
NCBI Disease	6,892	753
BCII-GN Train	6,252	1,411
NLM Citation GIA	1,205	310

Table 5.1: Statistics from various biological entity linking data sets from scientific articles. NCBI Disease (Doğan et al., 2014) focuses exclusively on disease entities. BCV-CDR (Li et al., 2016a) contains both chemicals and diseases. BCII-GN and NLM (Wei et al., 2015a) both contain genes.

5.2.2 TypeNet

TypeNet is a new dataset of hierarchical entity types for extremely fine-grained entity typing. TypeNet was created by manually aligning Freebase types Bollacker et al. (2008b) to noun synsets from the WordNet hierarchy (Fellbaum, 1998), naturally producing a hierarchical type set.

To construct TypeNet, we first consider all Freebase types that were linked to more than 20 entities. This is done to eliminate types that are either very specific or very rare. We also remove all Freebase API types, e.g. the [/freebase, /dataworld, /schema, /atom, /scheme, and /topics] domains.

For each remaining Freebase type, we generate a list of candidate WordNet synsets through a substring match. An expert annotator then attempted to map the Freebase type to one or more synsets in the candidate list with a *parent-of*, *child-of* or *equivalence* link by comparing the definitions of each synset with example entities of the Freebase type. If no match was found, the annotator manually formulated queries for the online WordNet API until an appropriate synset was found. See Table B.2 for an example annotation.

Two expert annotators independently aligned each Freebase type before meeting to resolve any conflicts. The annotators were conservative with assigning equivalence links resulting in a greater number of *child-of* links. The final dataset contained 13 *parent-of*, 727 *child-of*, and 380 *equivalence* links. Note that some Freebase types have multiple *child-of* links to WordNet, making TypeNet, like WordNet, a directed acyclic graph. We then took the union of each of our annotated Freebase types, the synset that they linked to, and any ancestors of that synset.

We also added an additional set of 614 $FB \rightarrow FB$ links. This was done by computing conditional probabilities of Freebase types given other Freebase types from a collection of 5 million randomly chosen Freebase entities. The conditional probability $\mathrm{P}(t_2 \mid t_1)$ of a Freebase type t_2 given another Freebase type t_1 was calculated as

$\frac{\#(t_1,t_2)}{\#t_1}$. Links with a conditional probability less than or equal to 0.7 were discarded. The remaining links were manually verified by an expert annotator and valid links were added to the final dataset, preserving acyclicity.

5.3 Model

5.3.1 Background: Entity Typing and Linking

We define a textual mention m as a sentence with an identified entity. The goal is then to classify m with one or more labels. For example, we could take the sentence $m =$ "*Barack Obama is the President of the United States.*" with the identified entity string **Barack Obama**. In the task of *entity linking*, we want to map m to a specific entity in a knowledge base such as "m/02mjmr" in Freebase. In *mention-level typing*, we label m with one or more types from our type system T such as $t^m = \{$president, leader, politician$\}$ Ling and Weld (2012a); Gillick et al. (2014); Shimaoka et al. (2017). In *entity-level typing*, we instead consider a bag of mentions B_e which are all linked to the same entity. We label B_e with t^e, the set of all types expressed in all $m \in B_e$ Yao et al. (2013); Neelakantan and Chang (2015b); Verga et al. (2017a); Yaghoobzadeh et al. (2017a).

5.3.2 Mention Encoder

Our model converts each mention m to a d dimensional vector. This vector is used to classify the type or entity of the mention. The basic model depicted in Figure 5.1 concatenates the averaged word embeddings of the mention string with the output of a convolutional neural network (CNN). The word embeddings of the mention string capture global, context independent semantics while the CNN encodes a context dependent representation.

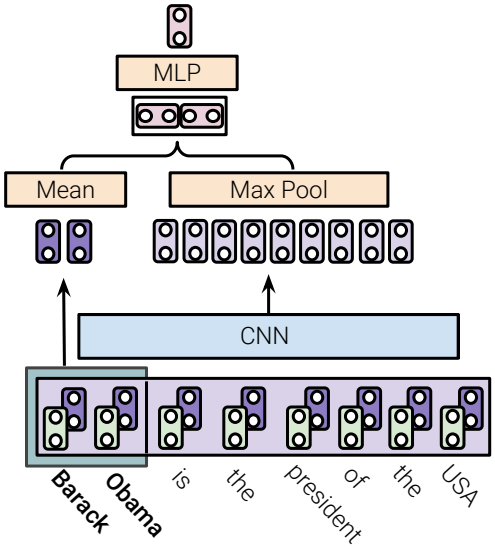


Figure 5.1: Sentence encoder for all our models. The input to the CNN consists of the concatenation of position embeddings with word embeddings. The output of the CNN is concatenated with the mean of mention surface form embeddings, and then passed through a 2 layer MLP.

5.3.2.1 Token Representation

Each sentence is made up of s tokens which are mapped to d_w dimensional word embeddings. Because sentences may contain mentions of more than one entity, we explicitly encode a distinguished mention in the text using position embeddings which have been shown to be useful in state of the art relation extraction models (dos Santos et al., 2015b; Lin et al., 2016) and machine translation (Vaswani et al., 2017b). Each word embedding is concatenated with a d_p dimensional learned position embedding encoding the token's relative distance to the target entity. Each token within the distinguished mention span has position 0, tokens to the left have a negative distance from $[-s, 0)$, and tokens to the right of the mention span have a positive distance from $(0, s]$. We denote the final sequence of token representations as M.

5.3.2.2 Sentence Representation

The embedded sequence M is then fed into our context encoder. Our context encoder is a single layer CNN followed by a tanh non-linearity to produce C. The

outputs are max pooled across time to get a final context embedding, m_{CNN}.

$$c_i = \tanh(b + \sum_{j=0}^{w} W[j]M[i - \lfloor \frac{w}{2} \rfloor + j])$$

$$m_{\text{CNN}} = \max_{0 \leq i \leq n-w+1} c_i$$

Each $W[j] \in \mathbb{R}^{d \times d}$ is a CNN filter, the bias $b \in \mathbb{R}^d$, $M[i] \in \mathbb{R}^d$ is a token representation, and the max is taken pointwise. In all of our experiments we set $w = 5$.

In addition to the contextually encoded mention, we create a global mention encoding, m_{G}, by averaging the word embeddings of the tokens within the mention span.

The final mention representation m_{F} is constructed by concatenating m_{CNN} and m_{G} and applying a two layer feed-forward network with tanh non-linearity (see Figure 5.1):

$$m_{\text{F}} = W_2 \tanh(W_1 \begin{bmatrix} m_{\text{SFM}} \\ m_{\text{CNN}} \end{bmatrix} + b_1) + b_2$$

5.4 Training

5.4.1 Mention-Level Typing

Mention level entity typing is treated as multi-label prediction. Given the sentence vector m_{F}, we compute a score for each type in typeset T as:

$$y_j = \mathbf{t_j}^\top m_{\text{F}}$$

where $\mathbf{t_j}$ is the embedding for the jth type in T and y_j is its corresponding score. The mention is labeled with t^m, a binary vector of all types where $t_j^m = 1$ if the jth type

is in the set of gold types for m and 0 otherwise. We optimize a multi-label binary cross entropy objective:

$$\mathcal{L}_{\text{type}}(m) = -\sum_j t_j^m \log y_j + (1 - t_j^m) \log(1 - y_j)$$

5.4.2 Entity-Level Typing

In the absence of mention-level annotations, we instead must rely on distant supervision Mintz et al. (2009a) to noisily label all mentions of entity e with all types belonging to e. This procedure inevitably leads to noise as not all mentions of an entity express each of its known types. To alleviate this noise, we use multi-instance multi-label learning (MIML) Surdeanu et al. (2012) which operates over *bags* rather than mentions. A bag of mentions $B_e = \{m^1, m^2, \ldots, m^n\}$ is the set of all mentions belonging to entity e. The bag is labeled with t^e, a binary vector of all types where $t_j^e = 1$ if the jth type is in the set of gold types for e and 0 otherwise.

For every entity, we subsample k mentions from its bag of mentions. Each mention is then encoded independently using the model described in Section 5.3.2 resulting in a bag of vectors. Each of the k sentence vectors m_{F}^i is used to compute a score for each type in t^e:

$$y_j^i = \mathbf{t_j}^\top m_{\text{F}}^i$$

where $\mathbf{t_j}$ is the embedding for the jth type in t^e and y^i is a vector of logits corresponding to the ith mention. The final bag predictions are obtained using element-wise LogSumExp pooling across the k logit vectors in the bag to produce entity level logits y:

$$y = \log \sum_i \exp(y^i)$$

We use these final bag level predictions to optimize a multi-label binary cross entropy objective:

$$\mathcal{L}_{\text{type}}(B_e) = -\sum_j t_j^e \log y_j + (1 - t_j^e) \log(1 - y_j)$$

5.4.3 Entity Linking

Entity linking is similar to mention-level entity typing with a single correct class per mention. Because the set of possible entities is in the millions, linking models typically integrate an alias table mapping entity mentions to a set of possible candidate entities. Given a large corpus of entity linked data, one can compute conditional probabilities from mention strings to entities Spitkovsky and Chang (2012). In many scenarios this data is unavailable. However, knowledge bases such as UMLS contain a canonical string name for each of its curated entities. State-of-the-art biological entity linking systems tend to operate on various string edit metrics between the entity mention string and the set of canonical entity strings in the existing structured knowledge base Leaman et al. (2013); Wei et al. (2015a).

For each mention in our dataset, we generate 100 candidate entities $e_c = (e_1, e_2, \ldots, e_{100})$ each with an associated string similarity score csim. See Appendix B.6.1 for more details on candidate generation. We generate the sentence representation m_{F} using our encoder and compute a similarity score between m_{F} and the learned embedding e of each of the candidate entities. This score and string cosine similarity csim are combined via a learned linear combination to generate our final score. The final prediction at test time $\hat{e}$ is the maximally similar entity to the mention.

$$\phi(m, e) = \alpha\ e^\top m_{\text{F}} + \beta\, \text{csim}(m, e)$$

$$\hat{e} = \underset{e \in e_c}{\text{argmax}}\ \phi(m, e)$$

We optimize this model by multinomial cross entropy over the set of candidate entities and correct entity e.

$$\mathcal{L}_{\text{link}}(m, e_c) = -\ \phi(m, e) + \log \sum_{e' \in e_c} \exp \phi(m, e')$$

5.5 Encoding Hierarchies

Both entity typing and entity linking treat the label space as prediction into a flat set. To explicitly incorporate the structure between types/entities into our training, we add an additional loss. We consider two methods for modeling the hierarchy of the embedding space: real and complex bilinear maps, which are two of the state-of-the-art knowledge graph embedding models.

5.5.1 Hierarchical Structure Models

Bilinear: Our standard bilinear model scores a hypernym link between (c_1, c_2) as:

$$s(c_1, c_2) = {c_1}^\top A c_2$$

where A $\in \mathbb{R}^{d \times d}$ is a learned real-valued non-diagonal matrix and c_1 is the child of c_2 in the hierarchy. This model is equivalent to RESCAL Nickel et al. (2011a) with a single IS-A relation type. The type embeddings are the same whether used on the left or right side of the relation. We merge this with the base model by using the parameter A as an additional map before type/entity scoring.

Complex Bilinear: We also experiment with a complex bilinear map based on the ComplEx model Trouillon et al. (2016), which was shown to have strong performance predicting the hypernym relation in WordNet, suggesting suitability for asymmetric, transitive relations such as those in our type hierarchy. ComplEx uses complex valued vectors for types, and diagonal complex matrices for relations, using Hermitian inner products (taking the complex conjugate of the second argument, equivalent to treating

the right-hand-side type embedding to be the complex conjugate of the left hand side), and finally taking the real part of the score[1]. The score of a hypernym link between (c_1, c_2) in the ComplEx model is defined as:

$$
\begin{aligned}
s(c_1, c_2) &= \mathrm{Re}(< c_1, r_{\text{Is-A}}, c_2 >) \\
&= \mathrm{Re}(\sum_k c_{1k} r_k \bar{c}_{2k}) \\
&= \langle \mathrm{Re}(c_1), \mathrm{Re}(r_{\text{Is-A}}), \mathrm{Re}(c_2) \rangle \\
&+ \langle \mathrm{Re}(c_1), \mathrm{Im}(r_{\text{Is-A}}), \mathrm{Im}(c_2) \rangle \\
&+ \langle \mathrm{Im}(c_1), \mathrm{Re}(r_{\text{Is-A}}), \mathrm{Im}(c_2) \rangle \\
&- \langle \mathrm{Im}(c_1), \mathrm{Im}(r_{\text{Is-A}}), \mathrm{Re}(c_2) \rangle
\end{aligned}
$$

where c_1, c_2 and $r_{\text{Is-A}}$ are complex valued vectors representing c_1, c_2 and the IS-A relation respectively. $\mathrm{Re}(z)$ represents the real component of z and $\mathrm{Im}(z)$ is the imaginary component. As noted in Trouillon et al. (2016), the above function is antisymmetric when $r_{\text{IS-A}}$ is purely imaginary.

Since entity/type embeddings are complex vectors, in order to combine it with our base model, we also need to represent mentions with complex vectors for scoring. To do this, we pass the output of the mention encoder through two different affine transformations to generate a real and imaginary component:

$$
\begin{aligned}
\mathrm{Re}(m_{\mathrm{F}}) &= W_{\text{real}} m_{\mathrm{F}} + b_{\text{real}} \\
\mathrm{Im}(m_{\mathrm{F}}) &= W_{\text{img}} m_{\mathrm{F}} + b_{\text{img}}
\end{aligned}
$$

[1]This step makes the scoring function technically not bilinear, as it commutes with addition but not complex multiplication, but we term it *bilinear* for ease of exposition.

where m_{F} is the output of the mention encoder, and W_{real}, $W_{\text{img}} \in \mathbb{R}^{d \times d}$ and b_{real}, $b_{\text{img}} \in \mathbb{R}^{d}$.

5.5.2 Training with Hierarchies

Learning a hierarchy is analogous to learning embeddings for nodes of a knowledge graph with a single hypernym/IS-A relation. To train these embeddings, we sample (c_1, c_2) pairs, where each pair is a positive link in our hierarchy. For each positive link, we sample a set N of n negative links. We encourage the model to output high scores for positive links, and low scores for negative links via a binary cross entropy (BCE) loss:

$$\begin{aligned}\mathcal{L}_{\text{struct}} &= -\log \sigma(s(c_{1i}, c_{2i})) \\ &\quad + \sum_{N} \log(1 - \sigma(s(c_{1i}, c'_{2i}))) \\ \mathcal{L} &= \mathcal{L}_{\text{type/link}} + \gamma \mathcal{L}_{\text{struct}}\end{aligned}$$

where $s(c_1, c_2)$ is the score of a link (c_1, c_2), and $\sigma(\cdot)$ is the logistic sigmoid. The weighting parameter γ is $\in \{0.1, 0.5, 0.8, 1, 2.0, 4.0\}$. The final loss function that we optimize is $\mathcal{L}$.

5.6 Experiments

We perform three sets of experiments: mention-level entity typing on the benchmark dataset FIGER, entity-level typing using Wikipedia and TypeNet, and entity linking using MedMentions.

5.6.1 Models

CNN: Each mention is encoded using the model described in Section 5.3.2. The resulting embedding is used for classification into a flat set labels. Specific implementation details can be found in Appendix B.3.

CNN+Complex: The CNN+Complex model is equivalent to the CNN model but uses complex embeddings and Hermitian dot products.

Transitive: This model does not add an additional hierarchical loss to the training objective (unless otherwise stated). We add additional labels to each entity corresponding to the transitive closure, or the union of all ancestors of its known types. This provides a rich additional learning signal that greatly improves classification of specific types.

Hierarchy: These models add an explicit hierarchical loss to the training objective, as described in Section 5.5, using either complex or real-valued bilinear mappings, and the associated parameter sharing.

5.6.2 Mention-Level Typing in FIGER

To evaluate the efficacy of our methods we first compare against the current state-of-art models of Shimaoka et al. (2017). The most widely used type system for fine-grained entity typing is FIGER which consists of 113 types organized in a 2 level hierarchy. For training, we use the publicly available W2M data (Ren et al., 2016) and optimize the mention typing loss function defined in Section-5.4.1 with the additional hierarchical loss where specified. For evaluation, we use the manually annotated FIGER (GOLD) data by Ling and Weld (2012a). See Appendix B.3 and B.4 for specific implementation details.

5.6.2.1 Results

In Table 5.2 we see that our base CNN models (CNN and CNN+Complex) match LSTM models of Shimaoka et al. (2017) and Gupta et al. (2017a), the previous state-of-the-art for models without hand-crafted features. When incorporating structure into our models, we gain 2.5 points of accuracy in our CNN+Complex model, matching the overall state of the art attentive LSTM that relied on handcrafted features from syntactic parses, topic models, and character n-grams. The structure can help

Model	Acc	Macro F1	Micro F1
Ling and Weld (2012a)	47.4	69.2	65.5
Shimaoka et al. (2017) †	55.6	75.1	71.7
Gupta et al. (2017a)†	57.7	72.8	72.1
Shimaoka et al. (2017)‡	**59.6**	**78.9**	**75.3**
CNN	57.0	75.0	72.2
+ hierarchy	58.4	76.3	73.6
CNN+Complex	57.2	75.3	72.9
+ hierarchy	**59.7**	**78.3**	**75.4**

Table 5.2: Accuracy and Macro/Micro F1 on FIGER (GOLD). † is an LSTM model. ‡ is an attentive LSTM along with additional hand crafted features.

our model predict lower frequency types which is a similar role played by hand-crafted features.

5.6.3 Entity-Level Typing in TypeNet

Next we evaluate our models on entity-level typing in TypeNet using Wikipedia. For each entity, we follow the procedure outlined in Section 5.4.2. We predict labels for each instance in the entity's bag and aggregate them into entity-level predictions using LogSumExp pooling. Each type is assigned a predicted score by the model. We then rank these scores and calculate average precision for each of the types in the test set, and use these scores to calculate mean average precision (MAP). We evaluate using MAP instead of accuracy which is standard in large knowledge base link prediction tasks Verga et al. (2017a); Trouillon et al. (2016). These scores are calculated only over Freebase types, which tend to be lower in the hierarchy. This is to avoid artificial score inflation caused by trivial predictions such as 'entity.' See Appendix B.5 for more implementation details.

5.6.3.1 Results

Table 5.3 shows the results for entity level typing on our Wikipedia TypeNet dataset. We see that both the basic CNN and the CNN+Complex models perform similarly with the CNN+Complex model doing slightly better on the full data regime.

Model	Low Data	Full Data
CNN	51.72	68.15
+ hierarchy	54.82	75.56
+ transitive	57.68	77.21
+ hierarchy + transitive	58.74	**78.59**
CNN+Complex	50.51	69.83
+ hierarchy	55.30	72.86
+ transitive	53.71	72.18
+ hierarchy + transitive	**58.81**	77.21

Table 5.3: MAP of entity-level typing in Wikipedia data using TypeNet. The second column shows results using 5% of the total data. The last column shows results using the full set of 344,246 entities.

We also see that both models get an improvement when adding an explicit hierarchy loss, even before adding in the transitive closure. The transitive closure itself gives an additional increase in performance to both models. In both of these cases, the basic CNN model improves by a greater amount than CNN+Complex. This could be a result of the complex embeddings being more difficult to optimize and therefore more susceptible to variations in hyperparameters. When adding in both the transitive closure and the explicit hierarchy loss, the performance improves further. We observe similar trends when training our models in a lower data regime with ∼150,000 examples, or about 5% of the total data.

In all cases, we note that the baseline models that do not incorporate any hierarchical information (neither the transitive closure nor the hierarchy loss) perform ∼9 MAP worse, demonstrating the benefits of incorporating structure information.

5.6.4 MedMentions Entity Linking with UMLS

In addition to entity typing, we evaluate our model's performance on an entity linking task using MedMentions, our new PubMed / UMLS dataset described in Section 5.2.1.

Model	original	normalized
mention tfidf	61.09	74.66
CNN	67.42	82.40
+ hierarchy	67.73	82.77
CNN+Complex	67.23	82.17
+ hierarchy	**68.34**	**83.52**

Table 5.4: Accuracy on entity linking in MedMentions. Maximum recall is 81.82% because we use an imperfect alias table to generate candidates. Normalized scores consider only mentions which contain the gold entity in the candidate set. Mention tfidf is *csim* from Section 7.2.2.

Tips and Pitfalls in **Direct Ligation** of Large Spontaneous Splenorenal Shunt during Liver Transplantation Patients with large spontaneous splenorenal shunt ... **baseline:** Direct [Direct → General Modifier → Qualifier → Property or Attribute] **+hierarchy:** Ligature (correct) [Ligature → Surgical Procedures → medical treatment approach]
A novel approach for selective chemical functionalization and localized assembly of one-dimensional **nanostructures**. **baseline:** Structure [Structure → order or structure → general epistemology] **+hierarchy:** Nanomaterials (correct) [Nanomaterials → Nanoparticle Complex → Drug or Chemical by Structure]
Gcn5 is recruited onto the **il-2** promoter by interacting with the NFAT in T cells upon TCR stimulation . **baseline:** Interleukin-27 [Interleukin-27 → IL2 → Interleukin Gene] **+hierarchy:** IL2 Gene (correct) [IL2 Gene → Interleukin Gene]

Table 5.5: Example predictions from MedMentions. Each example shows the sentence with entity mention span in bold. **Baseline**, shows the predicted entity and its ancestors of a model not incorporating structure. Finally, **+hierarchy** shows the prediction and ancestors for a model which explicitly incorporates the hierarchical structure information.

5.6.4.1 Results

Table 5.4 shows results for baselines and our proposed variant with additional hierarchical loss. None of these models incorporate transitive closure information, due to difficulty incorporating it in our candidate generation, which we leave to future work. The *Normalized* metric considers performance only on mentions with an alias table hit; all models have 0 accuracy for mentions otherwise. We also report the overall score for comparison in future work with improved candidate generation. We

see that incorporating structure information results in a 1.1% reduction in absolute error, corresponding to a ~6% reduction in relative error on this large-scale dataset.

Table 5.5 shows qualitative predictions for models with and without hierarchy information incorporated. Each example contains the sentence (with target entity in bold), predictions for the baseline and hierarchy aware models, and the ancestors of the predicted entity. In the first and second example, the baseline model becomes extremely dependent on TFIDF string similarities when the gold candidate is rare ($\leq$ 10 occurrences). This shows that modeling the structure of the entity hierarchy helps the model disambiguate rare entities. In the third example, structure helps the model understand the hierarchical nature of the labels and prevents it from predicting an entity that is overly specific (e.g predicting Interleukin-27 rather than the correct and more general entity IL2 Gene).

Note that, in contrast with the previous tasks, the complex hierarchical loss provides a significant boost, while the real-valued bilinear model does not. A possible explanation is that UMLS is a far larger/deeper ontology than even TypeNet, and the additional ability of complex embeddings to model intricate graph structure is key to realizing gains from hierarchical modeling.

5.7 Related Work

By directly linking a large set of mentions and typing a large set of entities with respect to a new ontology and corpus, and our incorporation of structural learning between the many entities and types in our ontologies of interest, our work draws on many different but complementary threads of research in information extraction, knowledge base population, and completion.

Our structural, hierarchy-aware loss between types and entities draws on research in knowledge completion such as complexTrouillon et al. (2016) and RESCAL Nickel et al. (2011a). Combining KB completion with hierarchical structure in knowledge

bases has been explored in Dalvi et al. (2015); Xie et al. (2016). Recently, Wu et al. (2017) proposed a hierarchical loss for text classification.

Linking mentions to a flat set of entities, often in Freebase or Wikipedia, is a long-standing task in NLP Bunescu and Pasca (2006); Cucerzan (2007); Durrett and Klein (2014); Francis-Landau et al. (2016). Typing of mentions at varying levels of granularity, from CoNLL-style named entity recognition Tjong Kim Sang and De Meulder (2003), to the more fine-grained recent approaches Ling and Weld (2012a); Gillick et al. (2014); Shimaoka et al. (2017), is also related to our task. A few prior attempts to incorporate a very shallow hierarchy into fine-grained entity typing have not lead to significant or consistent improvements Gillick et al. (2014); Shimaoka et al. (2017).

The knowledge base Yago (Suchanek et al., 2007b) includes integration with WordNet and type hierarchies have been derived from its type system (Yosef et al., 2012). Del Corro et al. (2015) use manually crafted rules and patterns (Hearst patterns Hearst (1992), appositives, etc) to automatically match entity types to Wordnet synsets.

Recent work has moved towards unifying these two highly related tasks by improving entity linking by simultaneously learning a fine grained entity type predictor Gupta et al. (2017a). Learning hierarchical structures or transitive relations between concepts has been the subject of much recent work Vilnis and McCallum (2015); Vendrov et al. (2016); Nickel and Kiela (2017)

We draw inspiration from all of this prior work, and contribute datasets and models to address previous challenges in jointly modeling the structure of large-scale hierarchical ontologies and mapping textual mentions into an extremely fine-grained space of entities and types.

5.8 Conclusion

We demonstrate that explicitly incorporating and modeling hierarchical information leads to increased performance in experiments on entity typing and linking across three challenging datasets. Additionally, we introduce two new human-annotated datasets: MedMentions, a corpus of 246k mentions from PubMed abstracts linked to the UMLS knowledge base, and TypeNet, a new hierarchical fine-grained entity typeset an order of magnitude larger and deeper than previous datasets.

While this work already demonstrates considerable improvement over non-hierarchical modeling, future work will explore techniques such as order embeddings Vendrov et al. (2016) and Poincaré embeddings Nickel and Kiela (2017) to represent the hierarchical embedding space, as well as methods to improve recall in the candidate generation process for entity linking. Most of all, we are excited to see new techniques from the NLP community using the resources we have presented.

CHAPTER 6

FULL ABSTRACT RELATION EXTRACTION

Most work in relation extraction, including all that we have discussed up to this point, form a prediction by looking at a short span of text within a single sentence containing a single entity pair mention. This approach often does not consider interactions across mentions, requires redundant computation for each mention pair, and ignores relationships expressed across sentence boundaries. These problems are exacerbated by the document- (rather than sentence-) level annotation common in biological text. In response, we propose a model which simultaneously predicts relationships between all mention pairs in a document. We form pairwise predictions over entire paper abstracts using an efficient self-attention encoder. All-pairs mention scores allow us to perform multi-instance learning by aggregating over mentions to form entity pair representations. We further adapt to settings without mention-level annotation by jointly training to predict named entities and adding a corpus of weakly labeled data. In experiments on two Biocreative benchmark datasets, we achieve state of the art performance on the Biocreative V Chemical Disease Relation dataset for models without external KB resources. We also introduce a new dataset an order of magnitude larger than existing human-annotated biological information extraction datasets and more accurate than distantly supervised alternatives.

6.1 Introduction

With few exceptions (Swampillai and Stevenson, 2011; Quirk and Poon, 2017; Peng et al., 2017), nearly all work in relation extraction focuses on classifying a short

span of text within a single sentence containing a single entity pair mention. However, relationships between entities are often expressed across sentence boundaries or otherwise require a larger context to disambiguate. For example, 30% of relations in the Biocreative V CDR dataset (§6.3.1) are expressed across sentence boundaries, such as in the following excerpt expressing a relationship between the chemical **azathioprine** and the disease **fibrosis**:

> *Treatment of psoriasis with **azathioprine**.* **Azathioprine** treatment benefited 19 (66%) out of 29 patients suffering from severe psoriasis. Haematological complications were not troublesome and results of biochemical liver function tests remained normal. Minimal cholestasis was seen in two cases and portal **fibrosis** of a reversible degree in eight. Liver biopsies should be undertaken at regular intervals if **azathioprine** therapy is continued so that structural liver damage may be detected at an early and reversible stage.

Though the entities' mentions never occur in the same sentence, the above example expresses that the chemical entity *azathioprine* can cause the side effect *fibrosis*. Relation extraction models which consider only within-sentence relation pairs cannot extract this fact without knowledge of the complicated coreference relationship between *eight* and *azathioprine treatment*, which, without features from a complicated pre-processing pipeline, cannot be learned by a model which considers entity pairs in isolation. Making separate predictions for each mention pair also obstructs multi-instance learning (Riedel et al., 2010; Surdeanu et al., 2012), a technique which aggregates entity representations from mentions in order to improve robustness to noise in the data. Like the majority of relation extraction data, most annotation for biological relations is distantly supervised, and so we could benefit from a model which is amenable to multi-instance learning.

In addition to this loss of cross-sentence and cross-mention reasoning capability, traditional mention pair relation extraction models typically introduce computational inefficiencies by independently extracting features for and scoring every pair of mentions, even when those mentions occur in the same sentence and thus could share

representations. In the CDR training set, this requires separately encoding and classifying each of the 5,318 candidate mention pairs independently, versus encoding each of the 500 abstracts once. Though abstracts are longer than e.g. the text between mentions, many sentences contain multiple mentions, leading to redundant computation.

However, encoding long sequences in a way which effectively incorporates long-distance context can be prohibitively expensive. Long Short Term Memory (LSTM) networks (Hochreiter and Schmidhuber, 1997b) are among the most popular token encoders due to their capacity to learn high-quality representations of text, but their ability to leverage the fastest computing hardware is thwarted due to their computational dependence on the length of the sequence — each token's representation requires as input the representation of the previous token, limiting the extent to which computation can be parallelized. Convolutional neural networks (CNNs), in contrast, can be executed entirely in parallel across the sequence, but the amount of context incorporated into a single token's representation is limited by the depth of the network, and very deep networks can be difficult to learn (Hochreiter, 1998). These problems are exacerbated by longer sequences, limiting the extent to which previous work explored full-abstract relation extraction.

To facilitate efficient full-abstract relation extraction from biological text, we propose Bi-affine Relation Attention Networks (BRANs), a combination of network architecture, multi-instance and multi-task learning designed to extract relations between entities in biological text without requiring explicit mention-level annotation. We synthesize convolutions and self-attention, a modification of the Transformer encoder introduced by Vaswani et al. (2017a), over sub-word tokens to efficiently incorporate into token representations rich context between distant mention pairs across the entire abstract. We score all pairs of mentions in parallel using a bi-affine operator, and aggregate over mention pairs using a soft approximation of the max function in order

to perform multi-instance learning. We jointly train the model to predict relations and entities, further improving robustness to noise and lack of gold annotation at the mention level.

In extensive experiments on two benchmark biological relation extraction datasets, we achieve state of the art performance for a model using no external knowledge base resources in experiments on the Biocreative V CDR dataset, and outperform comparable baselines on the Biocreative VI ChemProt dataset. We also introduce a new dataset which is an order of magnitude larger than existing gold-annotated biological relation extraction datasets while covering a wider range of entity and relation types and with higher accuracy than distantly supervised datasets of the same size. We provide a strong baseline on this new dataset, and encourage its use as a benchmark for future biological relation extraction systems.[1]

6.2 Model

We designed our model to efficiently encode long contexts spanning multiple sentences while forming pairwise predictions without the need for mention pair-specific features. To do this, our model first encodes input token embeddings using self-attention. These embeddings are used to predict both entities and relations. The relation extraction module converts each token to a *head* and *tail* representation. These representations are used to form mention pair predictions using a bi-affine operation with respect to learned relation embeddings. Finally, these mention pair predictions are pooled to form entity pair predictions, expressing whether each relation type is expressed by each relation pair.

[1]Our code and data are publicly available at: `https://github.com/patverga/bran`.

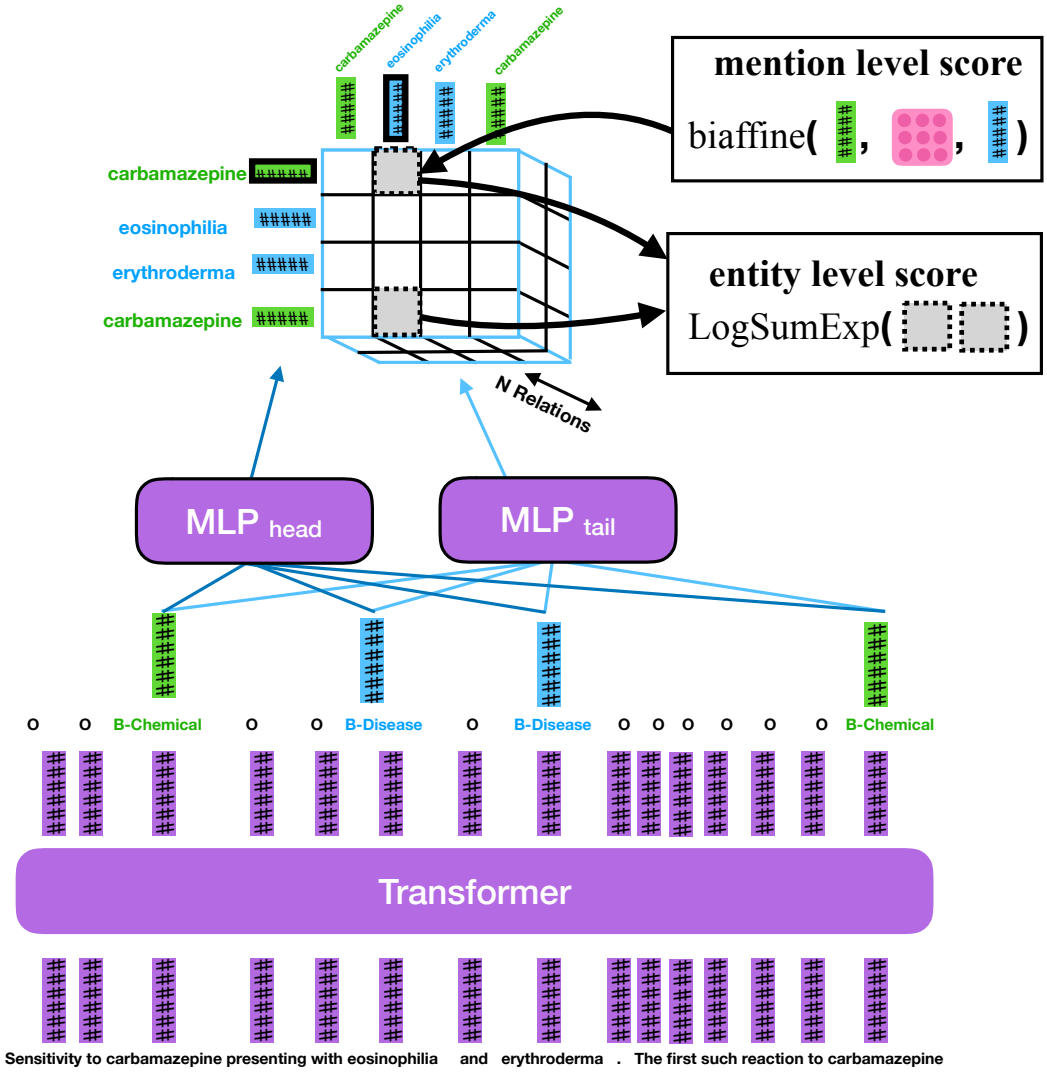


Figure 6.1: The relation extraction architecture. Inputs are contextually encoded using the Transformer(Vaswani et al., 2017a), made up of B layers of multi-head attention and convolution subcomponents. Each transformed token is then passed through a *head* and *tail* MLP to produce two position-specific representations. A bi-affine operation is performed between each *head* and *tail* representation with respect to each relation's embedding matrix, producing a pair-wise relation affinity tensor. Finally, the scores for cells corresponding to the same entity pair are pooled with a separate LogSumExp operation for each relation to get a final score.

6.2.1 Inputs

Our model takes in a sequence of N token embeddings in $\mathbb{R}^d$. Because the Transformer has no innate notion of token position, the model relies on positional embeddings which are added to the input token embeddings.[2] We learn the position embedding matrix $P^{m \times d}$ which contains a separate d dimensional embedding for each position, limited to m possible positions. Our final input representation for token x_i is:

$$x_i = s_i + p_i$$

where s_i is the token embedding for x_i and p_i is the positional embedding for the ith position. If i exceeds m, we use a randomly initialized vector in place of p_i.

We tokenize the text using byte pair encoding (BPE) (Gage, 1994; Sennrich et al., 2015). The BPE algorithm constructs a vocabulary of sub-word pieces, beginning with single characters. Then, the algorithm iteratively merges the most frequent co-occurring tokens into a new token, which is added to the vocabulary. This procedure continues until a pre-defined vocabulary size is met.

BPE is well suited for biological data for the following reasons. First, biological entities often have unique mentions made up of meaningful subcomponents, such as *1,2-dimethylhydrazine*. Additionally, tokenization of chemical entities is challenging, lacking a universally agreed upon algorithm (Krallinger et al., 2015). As we demonstrate in §6.3.3.2, the sub-word representations produced by BPE allow the model to formulate better predictions, likely due to better modeling of rare and unknown words.

[2]Though our final model incorporates some convolutions, we retain the position embeddings.

6.2.2 Transformer

We base our token encoder on the Transformer self-attention model (Vaswani et al., 2017a). The Transformer is made up of B blocks. Each Transformer block, which we denote Transformer_k, has its own set of parameters and is made up of two subcomponents: multi-head attention and a series of convolutions[3]. The output for token i of block k, $b_i^{(k)}$, is connected to its input $b_i^{(k-1)}$ with a residual connection (He et al., 2016). Starting with $b_i^{(0)} = x_i$:

$$b_i^{(k)} = b_i^{(k-1)} + \text{Transformer}_k(b_i^{(k-1)})$$

6.2.2.1 Multi-head Attention

Multi-head attention applies self-attention multiple times over the same inputs using separately normalized parameters (attention heads) and combines the results, as an alternative to applying one pass of attention with more parameters. The intuition behind this modeling decision is that dividing the attention into multiple heads make it easier for the model to learn to attend to different types of relevant information with each head. The self-attention updates input $b_i^{(k-1)}$ by performing a weighted sum over all tokens in the sequence, weighted by their importance for modeling token i.

Each input is projected to a key k, value v, and query q, using separate affine transformations with ReLU activations (Glorot et al., 2011). Here, k, v, and q are each in $\mathbb{R}^{\frac{d}{H}}$ where H is the number of heads. The attention weights a_{ijh} for head h between tokens i and j are computed using scaled dot-product attention:

[3]The original Transformer uses feed-forward connections, i.e. width-1 convolutions, whereas we use convolutions with width > 1.

$$a_{ijh} = \sigma\left(\frac{q_{ih}^T k_{jh}}{\sqrt{d}}\right)$$
$$o_{ih} = \sum_j v_{jh} \odot a_{ijh}$$

with $\odot$ denoting element-wise multiplication and σ indicating a softmax along the jth dimension. The scaled attention is meant to aid optimization by flattening the softmax and better distributing the gradients (Vaswani et al., 2017a).

The outputs of the individual attention heads are concatenated, denoted $[\cdot;\cdot]$, into o_i. All layers in the network use residual connections between the output of the multi-headed attention and its input. Layer normalization (Ba et al., 2016), denoted $\mathrm{LN}(\cdot)$, is then applied to the output.

$$o_i = [o_1; ...; o_h]$$
$$m_i = \mathrm{LN}(b_i^{(k-1)} + o_i)$$

6.2.2.2 Convolutions

The second part of our Transformer block is a stack of convolutional layers. The sub-network used in Vaswani et al. (2017a) uses two width-1 convolutions. We add a third middle layer with kernel width 5, which we found to perform better. Many relations are expressed concisely by the immediate local context, e.g. *Michele's husband Barack*, or *labetalol-induced hypotension*. Adding this explicit n-gram modeling is meant to ease the burden on the model to learn to attend to local features. We use $C_w(\cdot)$ to denote a convolutional operator with kernel width w. Then the convolutional portion of the transformer block is given by:

$$t_i^{(0)} = \text{ReLU}(C_1(m_i))$$
$$t_i^{(1)} = \text{ReLU}(C_5(t_i^{(0)}))$$
$$t_i^{(2)} = C_1(t_i^{(1)})$$

Where the dimensions of $t_i^{(0)}$ and $t_i^{(1)}$ are in $\mathbb{R}^{4d}$ and that of $t_i^{(2)}$ is in $\mathbb{R}^d$.

6.2.3 Bi-affine Pairwise Scores

We project each contextually encoded token $b_i^{(B)}$ through two separate MLPs to generate two new versions of each token corresponding to whether it will serve as the first (head) or second (tail) argument of a relation:

$$e_i^{head} = W_{head}^{(1)}(\text{ReLU}(W_{head}^{(0)} b_i^{(B)}))$$
$$e_i^{tail} = W_{tail}^{(1)}(\text{ReLU}(W_{tail}^{(0)} b_i^{(B)}))$$

We use a bi-affine operator to calculate an $N \times L \times N$ tensor A of pairwise affinity scores, scoring each (head, relation, tail) triple:

$$A_{ilj} = (e_i^{head} L) e_j^{tail}$$

where L is a $d \times L \times d$ tensor, a learned embedding matrix for each of the L relations. In subsequent sections we will assume we have transposed the dimensions of A as $d \times d \times L$ for ease of indexing.

6.2.4 Entity Level Prediction

Our data is weakly labeled in that there are labels at the entity level but not the mention level, making the problem a form of strong-distant supervision (Mintz et al., 2009a). In distant supervision, edges in a knowledge graph are heuristically applied to sentences in an auxiliary unstructured text corpus — often applying the edge label

to all sentences containing the subject and object of the relation. Because this process is imprecise and introduces noise into the training data, methods like multi-instance learning were introduced (Riedel et al., 2010; Surdeanu et al., 2012). In multi-instance learning, rather than looking at each distantly labeled mention pair in isolation, the model is trained over the aggregate of these mentions and a single update is made. More recently, the weighting function of the instances has been expressed as neural network attention (Verga and McCallum, 2016; Lin et al., 2016; Yaghoobzadeh et al., 2017b).

We aggregate over all representations for each mention pair in order to produce per-relation scores for each entity pair. For each entity pair (p^{head}, p^{tail}), let P^{head} denote the set of indices of mentions of the entity p^{head}, and let P^{tail} denote the indices of mentions of the entity p^{tail}. Then we use the LogSumExp function to aggregate the relation scores from A across all pairs of mentions of p^{head} and p^{tail}:

$$scores(p^{head}, p^{tail}) = \log \sum_{\substack{i \in P^{head} \\ j \in P^{tail}}} \exp(A_{ij})$$

The LogSumExp scoring function is a smooth approximation to the max function and has the benefits of aggregating information from multiple predictions and propagating dense gradients as opposed to the sparse gradient updates of the max (Das et al., 2017a).

6.2.5 Named Entity Recognition

In addition to pairwise relation predictions, we use the Transformer output $b_i^{(B)}$ to make entity type predictions. We feed $b_i^{(B)}$ as input to a linear classifier which predicts the entity label for each token with per-class scores c_i:

$$c_i = W^{(3)} b_i^{(B)}$$

We augment the entity type labels with the BIO encoding to denote entity spans. We apply tags to the byte-pair tokenization by treating each sub-word within a mention span as an additional token with a corresponding B- or I- label.

6.2.6 Training

We train both the NER and relation extraction components of our network to perform multi-class classification using maximum likelihood, where NER classes y_i or relation classes r_i are conditionally independent given deep features produced by our model with probabilities given by the softmax function. In the case of NER, features are given by the per-token output of the transformer:

$$\frac{1}{N}\sum_{i=1}^{N}\log P(y_i \mid b_i^{(B)})$$

In the case of relation extraction, the features for each entity pair are given by the LogSumExp over pairwise scores described in § 6.2.4. For E entity pairs, the relation r_i is given by:

$$\frac{1}{E}\sum_{i=1}^{E}\log P(r_i \mid scores(p^{head}, p^{tail}))$$

We train the NER and relation objectives jointly, sharing all embeddings and Transformer parameters. To trade off the two objectives, we penalize the named entity updates with a hyperparameter λ.

6.3 Results

We evaluate our model on three datasets: The Biocreative V Chemical Disease Relation benchmark (CDR), which models relations between chemicals and diseases (§6.3.1); the Biocreative VI ChemProt benchmark (CPR), which models relations between chemicals and proteins (§6.3.2); and a new, large and accurate dataset we

describe in §7.3 based on the human curation in the Chemical Toxicology Database (CTD), which models relationships between chemicals, proteins and genes.

The CDR dataset is annotated at the level of paper abstracts, requiring consideration of long-range, cross sentence relationships, thus evaluation on this dataset demonstrates that our model is capable of such reasoning. We also evaluate our model's performance in the more traditional setting which does not require cross-sentence modeling by performing experiments on the CPR dataset, for which all annotations are between two entity mentions in a single sentence. Finally, we present a new dataset constructed using strong-distant supervision (§6.2.4), with annotations at the document level. This dataset is significantly larger than the others, contains more relation types, and requires reasoning across sentences.

6.3.1 Chemical Disease Relations Dataset

The Biocreative V chemical disease relation extraction (CDR) dataset[4] (Li et al., 2016a; Wei et al., 2016) was derived from the Comparative Toxicogenomics Database (CTD), which curates interactions between genes, chemicals, and diseases (Davis et al., 2008). CTD annotations are only at the document level and do not contain mention annotations. The CDR dataset is a subset of these original annotations, supplemented with human annotated, entity linked mention annotations. The relation annotations in this dataset are also at the document level only.

6.3.1.1 Data Preprocessing

The CDR dataset is concerned with extracting only chemically-induced disease relationships (drug-related side effects and adverse reactions) concerning the most specific entity in the document. For example *tobacco causes cancer* could be marked as false if the document contained the more specific *lung cancer*. This can cause true

[4] `http://www.biocreative.org/`

relations to be labeled as false, harming evaluation performance. To address this we follow Gu et al. (2016, 2017) and filter hypernyms according to the hierarchy in the MESH controlled vocabulary[5]. All entity pairs within the same abstract that do not have an annotated relation are assigned the NULL label.

In addition to the gold CDR data, Peng et al. (2016) add 15,448 PubMed abstracts annotated in the CTD dataset. We consider this same set of abstracts as additional training data (which we subsequently denote +Data). Since this data does not contain entity annotations, we take the annotations from Pubtator (Wei et al., 2013a), a state of the art biological named entity tagger and entity linker. See §C.2 for additional data processing details. In our experiments we only evaluate our relation extraction performance and all models (including baselines) use gold entity annotations for predictions.

The byte pair vocabulary is generated over the training dataset — we use a budget of 2500 tokens when training on the gold CDR data, and a larger budget of 10,000 tokens when including extra data described above Additional implementation details are included in Appendix C.

Data split	Docs	Pos	Neg
Train	500	1,038	4,280
Development	500	1,012	4,136
Test	500	1,066	4,270
CTD	15,448	26,657	146,057

Table 6.1: Data statistics for the CDR Dataset and additional data from CTD. Shows the total number of abstracts, positive examples, and negative examples for each of the data set splits.

[5] https://www.nlm.nih.gov/mesh/download/2017MeshTree.txt

Model	P	R	F1
Gu et al. (2016)	62.0	55.1	58.3
Zhou et al. (2016a)	55.6	68.4	61.3
Gu et al. (2017)	55.7	68.1	61.3
BRAN	55.6	70.8	**62.1** $\pm$ 0.8
+ Data	64.0	69.2	**66.2** $\pm$ 0.8
BRAN(ensemble)	63.3	67.1	65.1
+ Data	65.4	71.8	**68.4**

Table 6.2: Precision, recall, and F1 results on the Biocreative V CDR Dataset.

6.3.1.2 Baselines

We compare against the previous best reported results on this dataset not using knowledge base features.[6] Each of the baselines are ensemble methods for within- and cross-sentence relations that make use of additional linguistic features (syntactic parse and part-of-speech). Gu et al. (2017) encode mention pairs using a CNN while Zhou et al. (2016a) use an LSTM. Both make cross-sentence predictions with featurized classifiers.

6.3.1.3 Results

In Table 6.2 we show results outperforming the baselines despite using no linguistic features. We show performance averaged over 20 runs with 20 random seeds as well as an ensemble of their averaged predictions. We see a further boost in performance by adding weakly labeled data. Table 6.3 shows the effects of ablating pieces of our model. 'CNN only' removes the multi-head attention component from the transformer block, 'no width-5' replaces the width-5 convolution of the feed-forward component of the transformer with a width-1 convolution and 'no NER' removes the named entity recognition multi-task objective (§6.2.5).

[6]The highest reported score is from Peng et al. (2016), but they use explicit lookups into the CTD knowledge base for the existence of the test entity pair.

Model	P	R	F1
BRAN (Full)	55.6	70.8	**62.1** ± 0.8
– CNN only	43.9	65.5	52.4 ± 1.3
– no width-5	48.2	67.2	55.7 ± 0.9
– no NER	49.9	63.8	55.5 ± 1.8

Table 6.3: Results on the Biocreative V CDR Dataset showing precision, recall, and F1 for various model ablations.

6.3.2 Chemical Protein Relations Dataset

To assess our model's performance in settings where cross-sentence relationships are not explicitly evaluated, we perform experiments on the Biocreative VI ChemProt dataset (CDR) Krallinger et al. (2017b). This dataset is concerned with classifying into six relation types between chemicals and proteins, with nearly all annotated relationships occurring within the same sentence.

6.3.2.1 Baselines

We compare our models against those competing in the official Biocreative VI competition (Liu et al., 2017). We compare to the top performing team whose model is directly comparable with ours — i.e. used a single (non-ensemble) model trained only on the training data (many teams use the development set as additional training data). The baseline models are standard state of the art relation extraction models: CNNs and Gated RNNs with attention. Each of these baselines uses mention-specific features encoding relative position of each token to the two target entities being classified, whereas our model aggregates over all mention pairs in each sentence. It is also worth noting that these models use a large vocabulary of pre-trained word embeddings, giving their models the advantage of far more model parameters, as well as additional information from unsupervised pre-training.

Model	P	R	F1
CNN†	50.7	43.0	46.5
GRU+Attention†	53.0	46.3	49.5
BRAN	48.0	54.1	**50.8** ± .01

Table 6.4: Precision, recall, and F1 results on the Biocreative VI Chem-Prot Dataset. † denotes results from Liu et al. (2017)

6.3.2.2 Results

In Table 6.4 we see that even though our model forms all predictions simultaneously between all pairs of entities within the sentence, we are able to outperform state of the art models classifying each mention pair independently. The scores shown are averaged across 10 runs with 10 random seeds. Interestingly, our model appears to have higher recall and lower precision, while the baseline models are both precision-biased, with lower recall. This suggests that combining these styles of model could lead to further gains on this task.

6.3.3 New CTD Dataset

6.3.3.1 Data

Existing biological relation extraction datasets including both CDR (§6.3.1) and CPR (§6.3.2) are relatively small, typically consisting of hundreds or a few thousand annotated examples. Distant supervision datasets apply document-independent, entity-level annotations to all sentences leading to a large proportion of incorrect labels. Evaluations on this data involve either very small (a few hundred) gold annotated examples or cross validation to predict the noisy, distantly applied labels Mallory et al. (2015); Quirk and Poon (2017); Peng et al. (2017).

We address these issues by constructing a new dataset using strong-distant supervision containing document-level annotations. The Comparative Toxicogenomics Database (CTD) curates interactions between genes, chemicals, and diseases. Each

relation in the CTD is associated with a disambiguated entity pair and a PubMed article where the relation was observed.

To construct this dataset, we collect the abstracts for each of the PubMed articles with at least one curated relation in the CTD database. As in §6.3.1, we use PubTator to automatically tag and disambiguate the entities in each of these abstracts. If both entities in the relation are found in the abstract, we take the (abstract, relation) pair as a positive example. The evidence for the curated relation could occur anywhere in the full text article, not just the abstract. Abstracts with no recovered relations are discarded. All other entity pairs with valid types and without an annotated relation that occur in the remaining abstracts are considered negative examples and assigned the NULL label. We additionally remove abstracts containing greater than 500 tokens[7]. This limit removed about 10% of the total data including numerous extremely long abstracts. The average token length of the remaining data was $\tilde{2}30$ tokens. With this procedure, we are able to collect 166,474 positive examples over 13 relation types, with more detailed statistics of the dataset listed in Table 6.5.

We consider relations between chemical-disease, chemical-gene, and gene-disease entity pairs downloaded from CTD[8]. We remove inferred relations (those without an associated PubMed ID) and consider only human curated relationships. Some chemical-gene entity pairs were associated with multiple relation types in the same document. We consider each of these relation types as a separate positive example.

The chemical-gene relation data contains over 100 types organized in a shallow hierarchy. Many of these types are extremely infrequent, so we map all relations to the highest parent in the hierarchy, resulting in 13 relation types. Most of these chemical-gene relations have an increase and decrease version such as increase_expression and decrease_expression. In some cases, there is also an affects relation (affects_expression)

[7]We include scripts to generate the unfiltered set of data as well to encourage future research

[8]`http://ctdbase.org/downloads/`

Types	Docs	Pos	Neg
Total	68,400	166,474	1,198,493
Chemical/Disease	64,139	93,940	571,932
Chemical/Gene	34,883	63,463	360,100
Gene/Disease	32,286	9,071	266,461

Table 6.5: Data statistics for the new CTD dataset.

	Train	Dev	Test
Total	120k	15k	15k
Chemical / Disease			
marker/mechanism	41,562	5,126	5,167
therapeutic	24,151	2,929	3,059
Gene / Disease			
marker/mechanism	5,930	825	819
therapeutic	560	77	75
Chemical / Gene			
increase_expression	15,851	1,958	2,137
increase_metabolic_proc	5,986	740	638
decrease_expression	5,870	698	783
increase_activity	4,154	467	497
affects_response	3,834	475	508
decrease_activity	3,124	396	434
affects_transport	3,009	333	361
increase_reaction	2,881	367	353
decrease_reaction	2,221	247	269
decrease_metabolic_proc	798	100	120

Table 6.6: Data statistics for the new CTD dataset broken down by relation type. The first column lists relation types separated by the types of the entities. Columns 2–4 show the number of positive examples of that relation type.

which is used when the directionality is unknown. If the affects version is more common, we map decrease and increase to affects. If affects is less common, we drop the affects examples and keep the increase and decrease examples as distinct relations, resulting in the final set of 10 chemical-gene relation types.

	P	R	F1
Total			
Micro F1	44.8	50.2	47.3
Macro F1	34.0	29.8	31.7
Chemical / Disease			
marker/mechanism	46.2	57.9	51.3
therapeutic	55.7	67.1	60.8
Gene / Disease			
marker/mechanism	42.2	44.4	43.0
therapeutic	52.6	10.1	15.8
Chemical / Gene			
increases_expression	39.7	48.0	43.3
increases_metabolic_proc	26.3	35.5	29.9
decreases_expression	34.4	32.9	33.4
increases_activity	24.5	24.7	24.4
affects_response	40.9	35.5	37.4
decreases_activity	30.8	19.4	23.5
affects_transport	28.7	23.8	25.8
increases_reaction	12.8	5.6	7.4
decreases_reaction	12.3	5.7	7.4
decreases_metabolic_proc	28.9	7.0	11.0

Table 6.7: BRAN precision, recall and F1 results for the full CTD dataset by relation type. The model is optimized for micro F1 score across all types.

6.3.3.2 Results

In Table 6.7 we list precision, recall and F1 achieved by our model on the CTD dataset, both overall and by relation type. Our model predicts each of the relation types effectively, with higher performance on relations with more support.

In Table 6.8 we see that our sub-word BPE model out-performs the model using the Genia tokenizer Kulick et al. (2012) even though our vocabulary size is one-fifth as large. We see a 1.7 F1 point boost in predicting Pubtator NER labels for BPE. This could be explained by the increased out-of-vocabulary (OOV) rate for named entities. Word training data has 3.01 percent OOV rate for tokens with an entity. The byte pair-encoded data has an OOV rate of 2.48 percent. Note that in both the word-tokenized and byte pair-tokenized data, we replace tokens that occur less than five times with a learned UNK token.

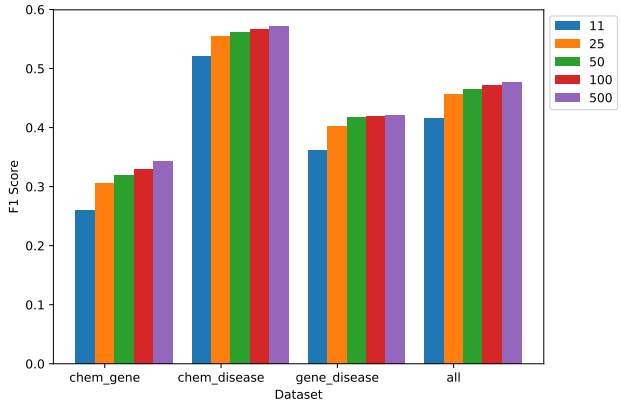


Figure 6.2: Performance on the CTD dataset when restricting candidate entity pairs by distance. The x-axis shows the coarse-grained relation type. The y-axis shows F1 score. Different colors denote maximum distance cutoffs.

Model	**P**	**R**	**F1**
Relation extraction			
Words	44.9	48.8	46.7 ± 0.39
BPE	44.8	50.2	**47.3** ± 0.19
NER			
Words	91.0	90.7	90.9 ± 0.13
BPE	91.5	93.6	**92.6** ± 0.12

Table 6.8: Precision, recall, and F1 results for CTD named entity recognition and relation extraction, comparing BPE to word-level tokenization.

Figure 6.2 depicts the model's performance on relation extraction as a function of distance between entities. For example, the blue bar depicts performance when removing all entity pair candidates (positive and negative) whose closest mentions are more than 11 tokens apart. We consider removing entity pair candidates with distances of 11, 25, 50, 100 and 500 (the maximum document length). The average sentence length is 22 tokens. We see that the model is not simply relying on short range relationships, but is leveraging information about distant entity pairs, with accuracy increasing as the maximum distance considered increases. Note that all results are taken from the same model trained on the full unfiltered training set.

6.4 Related work

Relation extraction is a heavily studied area in the NLP community. Most work focuses on news and web data (Doddington et al., 2004; Riedel et al., 2010; Hendrickx et al., 2009).[9] Recent neural network approaches to relation extraction have focused on CNNs (dos Santos et al., 2015a; Zeng et al., 2015b) or LSTMs (Miwa and Bansal, 2016a; Verga et al., 2016b; Zhou et al., 2016b) and replacing stage-wise information extraction pipelines with a single end-to-end model (Miwa and Bansal, 2016a; Ammar et al., 2017b; Li et al., 2017). These models all consider mention pairs separately.

There is also a considerable body of work specifically geared towards supervised biological relation extraction including protein-protein (Pyysalo et al., 2007; Poon et al., 2014; Mallory et al., 2015), drug-drug (Segura-Bedmar et al., 2013), and chemical-disease (Gurulingappa et al., 2012; Li et al., 2016a) interactions, and more complex events (Kim et al., 2008; Riedel et al., 2011). Our work focuses on modeling relations between chemicals, diseases, genes and proteins, where available annotation is often at the document- or abstract-level, rather than the sentence level.

Some previous work exists on cross-sentence relation extraction. Swampillai and Stevenson (2011) and Quirk and Poon (2017) consider featurized classifiers over cross-sentence syntactic parses. Most similar to our work is that of Peng et al. (2017), which uses a variant of an LSTM to encode document-level syntactic parse trees. Our work differs in three key ways. First, we operate over raw tokens negating the need for part-of-speech or syntactic parse features which can lead to cascading errors. We also use a feed-forward neural architecture which encodes long sequences far more efficiently compared to the graph LSTM network of Peng et al. (2017). Finally, our model considers all mention pairs simultaneously rather than a single mention pair at a time.

---

[9] And TAC KBP: `https://tac.nist.gov`

We employ a bi-affine function to form pairwise predictions between mentions. Such models have also been used for knowledge graph link prediction (Nickel et al., 2011b; Li et al., 2016b), with variations such as restricting the bilinear relation matrix to be diagonal (Yang et al., 2015b) or diagonal and complex (Trouillon et al., 2016). Our model is similar to recent approaches to graph-based dependency parsing, where bilinear parameters are used to score head-dependent compatibility (Kiperwasser and Goldberg, 2016; Dozat and Manning, 2017).

6.5 Conclusion

We present a bi-affine relation attention network that simultaneously scores all mention pairs within a document. Our model performs well on three datasets, including two standard benchmark biological relation extraction datasets and a new, large and high-quality dataset introduced in this work. Our model out-performs the previous state of the art on the Biocreative V CDR dataset despite using no additional linguistic resources or mention pair-specific features.

Our current model predicts only into a fixed schema of relations given by the data. However, this could be ameliorated by integrating our model into open relation extraction architectures such as Universal Schema Riedel et al. (2013a); Verga et al. (2016c). Our model also lends itself to other pairwise scoring tasks such as hypernym prediction, co-reference resolution, and entity resolution. We will investigate these directions in future work.

CHAPTER 7

JOINTLY MODELING ENTITIES AND RELATIONS

As we've discussed so far, understanding the meaning of language often involves reasoning about entities and their relationships. In the context of text, this requires identifying textual mentions of entities, linking them to a canonical concept, and discerning their relationships. These tasks are nearly always viewed as separate components within a pipeline, each requiring a distinct model and training data. The same holds true for the work from the previous chapters.

While relation extraction can often be trained with readily available weak or distant supervision, entity linkers typically require expensive mention-level supervision – which is not available in many domains. Instead, we propose a model which is trained to simultaneously produce entity linking and relation decisions while requiring no mention-level annotations. This approach avoids cascading errors that arise from pipelined methods and more accurately predicts entity relationships from text. We show that our model outperforms a state-of-the art entity linking and relation extraction pipeline on two biomedical datasets and can drastically improve the overall recall of the system.

7.1 Introduction

Making complex decisions in domains like biomedicine and clinical treatments requires access to information and facts in a form that can be easily viewed by experts and is computable by reasoning algorithms. The predominant paradigm for storing this type of data is in a knowledge graph. Much of these facts are populated from

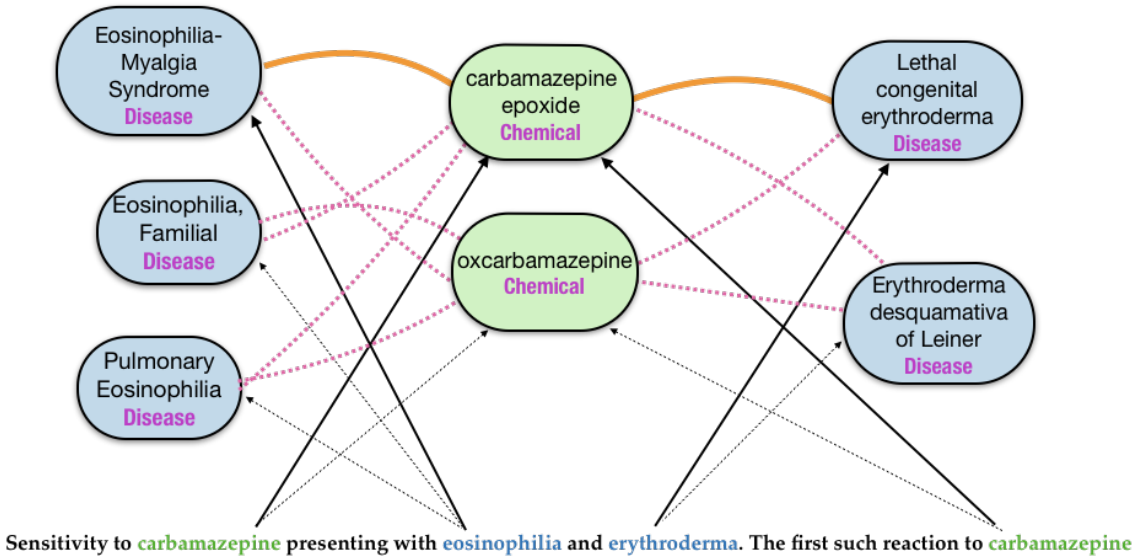


Figure 7.1: Overview of the graph extraction task. Given a document represented as a title and abstract. Text mentions are denoted with color and each can link to one of several possible entities. The model considers the full set of entity linking and relation edges (all lines) and predicts the graph of true entities and relations represented in the text. Dashed lines show possible (incorrect) edges and solid lines show the true edges.

hand curation by human experts, inevitably leading to high levels of incompleteness (Bodenreider, 2004; Suchanek et al., 2007c; Bollacker et al., 2008b). To address this, researchers have focused on automatically constructing knowledge bases by directly extracting information from text (Ji et al., 2010; Roth et al., 2014b).

This procedure can be broken down into three major components; identifying mentions of entities in text (Ratinov and Roth, 2009; Lample et al., 2016; Strubell et al., 2017), linking mentions of the same entity together into a single canonical concept (Cucerzan, 2007; Gupta et al., 2017b; Raiman and Raiman, 2018), and identifying relationships occurring between those entities (Bunescu and Mooney, 2007; Wang et al., 2016; Verga et al., 2018).

These three stages are nearly always treated as separate serial components in an extraction pipeline and current state-of-the-art approaches train separate machine learning models each with their own distinct training data. More precisely, this

data consists of mention-level supervision, that is individual instances of entities and relations which are identified and demarcated in text. This type of data can be prohibitively expensive to acquire, particularly in domains like biomedicine where expert knowledge is required to understand and annotate relevant information.

In contrast, forms of distant supervision are readily available as database entries in existing knowledge bases. This type of information encodes global properties about entities and their relationships without identifying specific textual instances of those facts. This form of distant supervision has been successfully applied to relation extraction models (Mintz et al., 2009b; Surdeanu et al., 2012; Riedel et al., 2013a; Verga et al., 2016c). However, all of these methods consume entity linking decisions as a preprocessing step, and unfortunately, accurate entity linkers and the mention-level supervision required to train them do not exist in many domains.

In this work, we instead develop a method to jointly consider and extract entities and their relationships together. We train our models leveraging readily available resources from existing knowledge bases and do not utilize any mention-level supervision. In experiments performed on two different biomedical datasets, we show that our model is able to substantially outperform a state-of-the-art pipeline of entity linking and relation extraction by jointly training and testing the two tasks together.

7.2 Model

The input to our model is the full title and abstract of an article and the output is the predicted graph of entities and relations represented in the text (see Fig. 7.1). This is done by first encoding the text using self-attention Vaswani et al. (2017a) to obtain a contextualized representation of each entity mention in the input. These contextualized representations are then used to predict both the distribution over entities at the mention-level and the distribution over relations at the mention-pair-level. These predicted probabilities are then combined for each mention-pair and pooled at

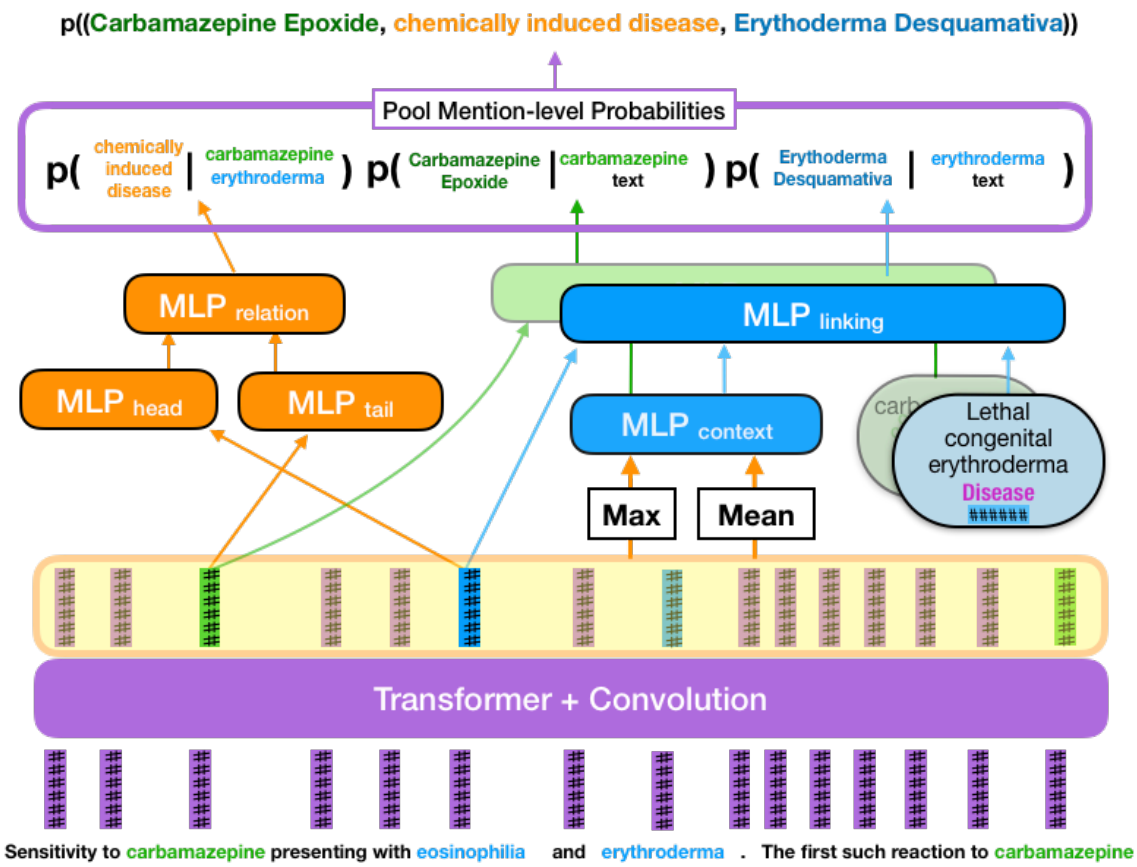


Figure 7.2: Architecture of the model. The text of the title and abstract are mapped to context independent token embeddings before being contextually encoded using a transformer architecture. The left side of the figure shows the procedure for scoring an individual relation mention using a separate head and tail MLP fed to a $\text{MLP}_{\text{relation}}$. The right side shows the entity linking component. The $\text{MLP}_{\text{linking}}$ model takes as input, an entity mention, a context representation derived from the mean and max over all contextualized token embeddings, and a candidate entity representation. These three probabilities (relation prediction and the two entity linking predictions) make up a single mention-level prediction. All mention-level predictions corresponding to the same entities are then pooled together to make a final entity-level prediction.

the document-level to get a final probability for predicting the tuple (e_1, r, e_2) for the text (see Fig. 7.2).

Notations: Let $[N]$ denote the set of natural numbers $\{1, \ldots, N\}$. Each document consists of a set of words $\{x_i\}$ indexed by $i \in [V]$ where V is the vocabulary size. Entity mentions in the document are found using a named entity recognition (NER) system Wei et al. (2013b). Let $\{m_j\}$ for $j \in [M]$ be the set of mention start

indices for the document, where M is the number of mentions in the document. For each mention string x_{m_i} we generate up to C candidate entities (Section 7.2.2). Let E be the set of all entities. Each document is annotated with the graph of entities and relations, given as a set of tuples $G_d = \{(e_k, r, e_l)\}$, where $e_k, e_l \in E$ and $r \in [R]$. This is obtained from a knowledge base under the strong distant supervision assumption Mintz et al. (2009b) (See 7.3.2). Let $E_d \subset E$ be the set of entities in the annotations for the document d. $[a; b]$ denotes concatenation of vectors a and b.

7.2.1 Text Encoder

The initial input to our model is the full title and abstract of a biomedical article from PubMed[1]. The sequence is tokenized and each token is mapped to a n-dimensional word embedding. The sequence of word embeddings are the input to our text encoder. The text encoder is based on the Transformer architecture of Vaswani et al. (2017a). The transformer applies multiple blocks of multi-head self attention followed by width 1 convolutions. We follow Verga et al. (2018) and add additional width 5 convolutions. The reader is referred to Verga et al. (2018) for the specific details. The output of the text encoder is an n-dimensional contextualized embedding h_i for each token x_i:

$$h_1, \ldots, h_N = \text{transformer}(x_1, \ldots, x_N)$$

From an efficiency perspective, we only encode the document once and use the contextualized token representations to predict both the entities and the relations.

7.2.2 Predicting entities

From the contextualized token representations $\{h_i\}$, we first obtain a document representation by concatenating the mean-pooled and max-pooled token representa-

[1] `https://www.ncbi.nlm.nih.gov/pubmed/`

tions and projecting it through a multi-layer perceptron (MLP).

$$\tilde{h} = W^2_{\text{doc}}(\text{ReLU}(W^1_{\text{doc}}[\text{mean}(\{h_i\}); \max(\{h_i\})]))$$

where mean$(\cdot)$ denotes an element-wise mean of a set of vectors and max$(\cdot)$ denotes an element-wise max of a set of vectors. Now, for each mention, we generate candidates entities for the mention. Such a candidate generation step is often used in entity-linking models Shen et al. (2015) and in many domains, such as for Wikipedia entities, high quality candidates can be generated by using prior linking counts of mention surface forms to entities obtained from Wikipedia anchor texts Ganea and Hofmann (2017); Raiman and Raiman (2018). However, such high quality candidate generation is not available in the biomedical domain and so we resort to an approximate string matching approach for generating candidate entities.

Candidate Generation: We followed procedures from previous work (Leaman and Lu, 2016; Murty et al., 2018). Each mention was first normalized by removing all punctuation, lower-casing, and then stemming (Porter, 1980). Next, these strings were converted to tfidf vectors consisting of both word and character ngrams. We considered character ngrams of lengths two to five, and for words we considered unigrams and bigrams. The same procedure was also applied to convert all canonical string names and synonyms for entities in our knowledge base. Finally, candidates for each mention were generated according to their cosine similarity amongst all entities in the knowledge base.

For each candidate entity e_i with type t_i, we generate a n-dimensional entity embedding as $\tilde{e}_i = \hat{e}_i + t_i$, by adding an entity-specific embedding $\hat{e}_i$ and a n-dimensional entity type embedding t_i. The entity-specific embedding can be learned or it can be a pre-trained embedding obtained from another source such as entity descriptions Ganea and Hofmann (2017); Xie et al. (2016) or by a graph embedding method Yang et al. (2014); Dettmers et al. (2018). Now, for the i-th mention in the document,

with starting index m_i, we consider h_{m_i} as a contextualized mention representation and define a score for predicting the candidate entity e for this mention using the candidate representation $\tilde{e}$, document representation $\tilde{h}$, and mention representation h_{m_i}. This is passed through a softmax function, normalizing over the set of candidates C_{m_i} for the mention to get a probability $p(e|m_i, \text{text})$ for linking the mention m_i to entity e.

$$
\begin{aligned}
l(e, m_i, \text{text}) &= W_l^2(\text{ReLU}(W_l^1[\tilde{e}; \tilde{h}; h_{m_i}]) \\
p(e|m_i, \text{text}) &= \underset{e \in C_{m_i}}{\text{softmax}}\left(l(e, m_i, \text{text})\right)
\end{aligned} \tag{7.1}
$$

We thus obtain a (M $\times$ C) matrix of linking probabilities for the document, where M is the maximum number of entity mentions in the document and C is the maximum number of candidates per mention. Note that there is no direct mention-level supervision available to train these probabilities.

7.2.3 Predicting relations

Given the contextualized mention representation, we obtain a head and tail representation for each mention to serve as the head or tail entity of a relation tuple (e_i, r, e_j). This is done by using two MLP to project each mention representation.

$$
\begin{aligned}
e_{m_i}^{\text{head}} &= W_{\text{head}}^2(\text{ReLU}(W_{\text{head}}^1 h_{m_i})) \\
e_{m_j}^{\text{tail}} &= W_{\text{tail}}^2(\text{ReLU}(W_{\text{tail}}^1 h_{m_j}))
\end{aligned}
$$

The head and tail representations are then passed through an MLP to predict a score for every relation r for a pair of mentions m_i and m_j. We pass this score vector through a sigmoid function to get a probability of predicting the relation from the mention-pair.

$$s(r, m_i, m_j) = W_r^2(\text{ReLU}(W_r^1[e_{m_i}^{\text{head}}; e_{m_i}^{\text{tail}}]))$$
$$p(r|m_i, m_j) = \sigma(s(r, m_i, m_j)) \tag{7.2}$$

We thus obtain a (M × M × R) matrix of probabilities for predicting all relations, where R is the maximum number of relations, from all pairs of entity mentions.

7.2.4 Combining entity and relation predictions

To predict the graph of entities and relations from the document, we need to assign a probability to every possible relation tuple (e_k, r, e_l). We first obtain the probability of predicting a tuple (e_k, r, e_l) from a mention-pair (m_i, m_j) by combining the probability for predicting the candidates for each of the mentions (7.1) and the relation prediction probability (7.2). If an entity is not a candidate for a mention then it's entity prediction probability is zero for that mention.

$$p\left((e_k, r, e_l)|m_i, m_j, \text{text}\right) =$$
$$p(e_k|m_i, \text{text})p(r|m_i, m_j)p(e_l|m_j, \text{text}) \tag{7.3}$$

Then, the probability of extracting the tuple (e_k, r, e_l) from the entire document can be obtained by pooling over all mention pairs (m_i, m_j). For example, we can use max-pooling, which corresponds to the inductive bias that in order to extract a tuple we must find at least one mention pair for the corresponding entities in the document that is evidence for the tuple.

$$p\left((e_k, r, e_l)|\text{text}\right) = \max_{i,j} p\left((e_k, r, e_l)|m_i, m_j, \text{text}\right) \tag{7.4}$$

Soft maximum pooling: It has been observed previously that the hard max operation is not ideal for pooling evidence as it leads to very sparse gradients Verga et al. (2017b); Das et al. (2017b). Recent methods Verga et al. (2018) thus use

the logsumexp function for pooling over *logits*, which allows for more dense gradient updates. However, we cannot use the logsumexp function in our case to pool over the probabilities (7.3) as the result of logsumexp over independent probabilities is not guaranteed to be a probability (in $[0, 1]$). Thus, we use a different operator that is considered a smooth relaxation of the maximum Bansal et al. (2015). Given a set of elements $\{a_i\}$, the smooth-maximum (smax) with temperature τ is defined as:

$$w_i = \underset{i}{\text{softmax}}\left(\frac{a_i}{\tau}\right); \quad smax(\{a_i\}) = \sum_i w_i a_i$$

Note that for $\tau \to 0$ the result of *smax* tends to the maximum of the set and for $\tau \to \infty$ the result is the average of the set. Thus, *smax* can smoothly interpolate between these extremes. We use this *smax* pooling over probabilities in (7.4) with a learned temperature τ.

7.2.5 Training

We are given ground-truth annotation for the set of tuples in the document, $G_d = \{(e_k, r, e_l)\}$. We train based on the cross-entropy loss from predicted tuple probabilities (7.4). Since we only have a subset of positive annotations, there is uncertainty in the set of negatives, and we deal with this by weighting the positive annotations by a weight w_t in the cross-entropy loss. Let $y_{krl} = 1$ if document is annotated with the relation tuple (e_k, r, e_l) and 0 otherwise, and p_{krl} be its predicted probability in (7.4), then we maximize $\log p(G_d|text)$:

$$\frac{1}{|G_d|} \sum_{k,r,l} w_t y_{krl} \log p_{krl} + (1 - y_{krl}) \log(1 - p_{krl})$$

In addition, since we can obtain *document-level* entity annotations from the set of annotated relation tuples, we can provide an additional document-level entity supervision to better train our entity linking probabilities. To do this, we perform max-pooling over all mentions for each candidate entity for the document in (7.1), to obtain

a document-level entity prediction score $p(e|\text{text}) = \max_m p(e|m, \text{text})$. We compute a weighted cross-entropy for these document-level predictions, again up-weighting the positive entities with a weight w_e. In summary, we combine graph prediction and document-level entity prediction objectives similar to multi-task learning (Caruana, 1993), so if E_d is the set of entities in annotation, we maximize:

$$\log p(G_d|text) + \alpha \log p(E_d|text) \tag{7.5}$$

Note that since we only have some positive annotations, there could be many mentions in the document for which the correct entity is not annotated. Thus, we down-weight the document-entity prediction term by α in the objective.

Technical Details: Since the size of G_d can be very large, in order to improve training efficiency we subsample the set of unannotated entities as the negative entities to a maximum of n^- per document. Pooling over the joint mention-level probability (7.4) requires an intermediate (L×L×M×M×R) tensor, where L is the total number of *candidate* entities for the document. Since this can be computationally prohibitive, we compute the top-k mentions per candidate entity based on the predicted probabilities (7.1) and only backpropagate the gradients through the top-k. We consider k as a hyperparameter and tune it on the validation set.

7.3 Experiments

Our experimental setting is that, for each test document (title and abstract), the model should produce the full graph of known entity-relationships expressed in that document (a single example is depicted in Fig. 7.1). Thus, we evaluate on micro-averaged precision, recall and F1 for predicting the entire set of annotated relation tuples across documents. Our experimental results show significant improvement in

F1 over a pipelined approach Verga et al. (2018). Implementation details can be found in the Supplementary [2].

7.3.1 Baselines

All of our models use the same basic architecture described in Section 7.2, consume the same predicted entity mentions from an external NER model (Wei et al., 2013b), and differ only in how they produce entity linking decisions (see Section 7.2.2). The first two baselines take hard entity linking decisions as inputs and do not do any internal entity linking inference. This is analogous to the typical pipelined approach.

Top Candidate produces entity linking decisions based on the highest scoring candidate entity (See 'Candidate Generation' in Section 7.2.2).

Linker produces entity linking decisions from a trained state-of-the-art entity linker. In this work we took annotations from a recent data dump from Wei et al. (2013b). This method is roughly equivalent to the BRAN model from Verga et al. (2018). The only difference is that our relation scoring function uses an MLP (Section 7.2.3), rather than the bi-affine scorer of the original work.[3]

End-to-End is our proposed model that does not take in any hard entity linking decisions as input and instead jointly predicts the full set of entities and relations within the text. For this model we considered 25 candidates per mention.

7.3.2 CTD Dataset

Our first set of experiments are on the CTD dataset first introduced in Verga et al. (2018)[4]. The data is derived from annotations in the Chemical Toxicology Database

[2]Code and data will be made publicly available

[3]In our experiments we found the MLP scoring function to perform slightly better than the bi-affine scorer.

[4]We slightly modified the data splits from the original dataset in order for the train, dev, and test sections to be consistent with those in the CDR dataset, allowing us to accurately evaluate entity linking (Section 7.3.3). Though the vast majority of document split assignments remain unchanged

Model	*Linker*	*Candidates*
Top Candidate	91.8%	67.0%
Linker	100%	60.4%
End-to-End	99.0%	80.0%

Table 7.1: Maximum recall on development set for each of the models on the two CTD dataset splits. *Linker* column refers to the data where relations were kept only if the external entity linker identified both entities in the title or abstract. *Candidates* column refers to the data filtered to relations where both entities were in top 250 candidates for mentions in the title or abstract.

(Davis et al., 2018), a curated knowledge base containing relationships between chemicals, diseases, and genes. Each fact additionally contains a reference to the document (a scientific publication) where the annotator identified the relationship[5]. This allows us to treat these annotations as a form of strong distant supervision (Mintz et al., 2009b). Here annotations are at the document-level rather than the mention-level (as in typical supervised learning) or corpus-level (as in standard distant supervision).

An aspect of the document-level supervision is that the original facts were annotated over complete documents. However, due to paywalls we often only have access to titles and abstracts of papers. Therefore, there is no guarantee that the relationship is actually expressed in the title or abstract, and further, even if it is there is no guarantee our model will be able to correctly identify the pair of entities in the text. Because of this we considered two different scenarios for filtering which set of relationships and documents to consider (maximum recall for different filtering techniques can be seen in Table 7.1).

---

and overall result scores and trends will be consistent, our numbers are not directly comparable to the original paper.

[5]This type of document annotation is fairly common in biomedical knowledge bases, further motivating this work.

Model	Precision	Recall	F1
Top Candidate	30.5	29.5	30.0
Linker	33.2	28.1	30.5
End-to-End	**41.1**	**43.4**	**42.2**

Table 7.2: Precision, Recall, and F1 for the CTD evaluation data filtered by recall of the 25 entity linking candidates. Top values in each column are in boldface.

7.3.2.1 Candidate-Based Filtering

The first method we evaluate is considering all relationships where both entities appear as candidates in the title and abstract of the document. That is, for each annotated tuple between entities e_1 and e_2 in document D, we consider that tuple if both e_1 and e_2 are candidates for at least one entity mention each in D. For creating the data split, we consider up to 250 candidates entities per mention[6]. The number of documents in dev and test set are 8177 and 8284.

In table 7.2, we can see that the End-to-End model that jointly considers both entity and relations together drastically outperforms the models that take hard linking decisions from an external model. This is primarily due to the huge drop in recall caused by cascading errors (See Table 7.1).

7.3.2.2 Linker-Based Filtering

The second data filtering approach we evaluated, is only considering the relationships where the external entity linking model was able to identify at least one mention of each of the two entities in the title or abstract of the document. This leads to a higher precision subset of data at the cost of recall. Importantly, this approach gives a substantial advantage to the external entity linker baseline as the data is filtered to only consider the relationships for which it could potentially make

[6]In our CTD experiments the end-to-end model uses top 25 candidates for every mention as we found that it performs better due to lesser training noise.

a prediction. The number of documents in dev and test set in this setting are 5857 and 5804 (significantly less than before).

In table 7.3, we can see that even under this disadvantage, the end-to-end model is able to perform comparably to the Linker baseline, even slightly outperforming it[7].

Model	Precision	Recall	F1
Top Candidate	43.5	47.9	45.6
Linker	46.1	**52.5**	49.1
End-to-End	**47.0**	52.0	**49.4**

Table 7.3: Precision, Recall, and F1 for the CTD evaluation data filtered by entity linker recall. Top values in each column appear in bold.

7.3.3 CDR Entity Linking Performance

In order to evaluate how much of the success of the End-to-End model can be attributed to the entity linking component (7.1), we evaluated its performance on the BioCreative V Chemical Disease Relation dataset (CDR) introduced in Wei et al. (2015b). Similar to the CTD dataset, CDR was also originally derived from the Chemical Toxicology Database. Expert annotators chose 1,500 of those documents and exhaustively annotated all *mentions* of chemicals and diseases in the text. Additionally, each mention was assigned its appropriate entity linking decision. We use this dataset as a gold standard to *validate* our entity linking models. Note that we do not use this data for training, but only for evaluation purposes.

We use the model that was trained on the CTD data and make it predict entities for every mention on the test set of CDR. For this evaluation we used the gold mention boundaries in the data. In order to analyze the effect of jointly predicting entities and relations on the entity linking performance, we also trained a model which learns to only predict entities (and ignores relations) from document-level entity supervision.

[7]Note that the numbers in Tables 7.2 and 7.3 are not comparable as the evaluation sets are significantly different.

We do this by only maximizing $p(E_d|text)$ in (7.5). Note that this model is also trained from document-level supervision on the CDT dataset and does not use any mention-level training data from CDR.

In Table 7.4, we see that our End-to-End model does learn to link entities better than the top candidate. Interestingly, on this particular data, the top candidate does perform quite well. As is common when evaluating on this data, we consider document-level rather than mention-level entity linking evaluation (Leaman and Lu, 2016), that is, how does the set of predicted entities compare to the gold set annotated in the document. Note that the model trained to jointly predict entities and relations performs slightly better than the model which predicts only entities. Breakdown of the results into Chemical and Disease prediction performance can be found in Supplementary.

Model	P	R	F1
Top Candidate	79.0	86.8	82.7
End-to-End			
– Entities only	82.9	90.0	86.3
– Entities & Relations	**83.3**	**90.2**	**86.6**

Table 7.4: Results for entity linking on the CDR dataset.

7.3.4 Disease-Phenotype Relations

To further probe the performance of our model we created a dataset of disease / phenotype (aka symptom) relations. The goal here is to identify specific symptoms caused by a disease. This type of information is particularly important in clinical treatments as it can lead to earlier diagnosis of rare diseases, faster application of appropriate interventions, and better overall outcomes for patients. This task also serves to further motivate our methods as accurate entity linking models for phenotypes are not readily available, nor is sufficient mention-level training data to build a supervised classifier.

Relation Annotations: We created this dataset with a similar technique to the construction of the CTD dataset. We started from the relations in the Human Phenotype Ontology (Köhler et al., 2018) that were annotated with a document containing that relationship.

Mention Detection: For disease mention detection we followed the same procedure from section 7.3 and used the annotated mentions from Wei et al. (2013b). Because there is not a readily available phenotype tagger, we trained our own model to identify mentions of phenotypes in text. We trained an iterated dilated convolution model Strubell et al. (2017).[8] Our training data came from Groza et al. (2015), which we split into train, dev, and test sets (see Supplementary). Our final NER model achieved a micro F1 score of 72.57.

We observed that disease and phenotype entity spans are often overlapping and nested. We thus over-generate the set of mentions by taking the predictions from both the taggers and adding them to the set of all mentions for the document, since our model is able to pool over all theses mentions even if they overlap.

Entity Linking: We followed a similar procedure as described in section 7.2.2 to generate phenotype entity linking candidates. Using the small set of gold entity linked text mentions from Groza et al. (2015) we were able to estimate our candidate's entity linking accuracy. In Figure 7.3 we show the recall of our candidate sets given different values of K. Our top candidate achieved an accuracy of 46.8 while the recall for 100 candidates was 76.5. This demonstrates the additional difficulty of the disease-phenotype dataset as these candidate accuracies are much lower than the results from Section 7.3.3.

Köhler et al. (2018) annotations make use of several disease vocabularies from OMIM (Hamosh et al., 2005), ORHPANET (Pavan et al., 2017) and DECIPHER

[8] `https://github.com/iesl/dilated-cnn-ner`

(Bragin et al., 2013) databases. For generating disease candidates, we use disease name strings from all of these.

The external entity linker that we used from Wei et al. (2013b) links diseases to the MeSH disease vocabulary. To align these with our disease-phenotype relation annotations, we use the MEDIC database (Davis et al., 2012) for mapping OMIM disease terms into the MeSH vocabulary.

The final dataset annotations were selected by filtering based on entities that can be found in document when considering up to 250 candidates per mention. See Supplementary for dataset statistics.

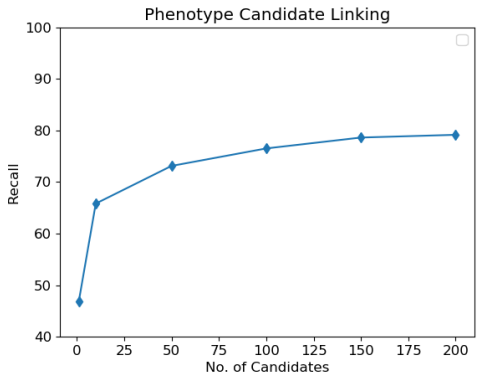


Figure 7.3: Recall for different numbers of candidates for phenotype entity linking

7.3.4.1 Pre-training Entity Embedding

Since the dataset has many unseen entities at test time, we need a method to address these unseen entities as generating the linking probabilities in (7.1) requires an entity embedding. For this, we obtained entity descriptions for the phenotypes and encoded them using pre-trained sentence embedding from BioSentVec Chen et al. (2018). However, not all test entities have descriptions. So, in addition to the descriptions we trained a graph embedding model, DistMult Yang et al. (2014), on the graph obtained from the set of all annotations in Human Phenotype Ontology excluding the dev/test annotations. We project both these pre-trained embeddings using a learned

linear transformation and sum the description and graph embedding to obtain the entity-specific embedding $\hat{e}$.

7.3.4.2 Baselines

The **Top Candidate** baseline uses the highest scoring candidate generated from the procedure described above for both diseases and phenotypes.

Linker uses the disease entity links from Wei et al. (2013b). Since we don't have access to an accurate pretrained phenotype entity linking model, this model also uses the top phenotype candidate as a hard phenotype entity linking decision.

End-to-End does not take any hard entity linking decisions and jointly reasons over entity linking and relation extraction decisions.

7.3.4.3 Results

Our disease-phenotype results show a similar trend to those from the CTD experiments (Section 7.3). Overall, the Top Candidate model performs the worst and the End-to-End model outperforms both models that use hard entity linking decisions.

Overall, our results indicate that this particular task is extremely challenging. This is likely the combination of several difficulties. The first is that the candidate set itself is not as accurate as the ones from the CTD experiment which we can see from comparing Figure 7.3 with the Top Candidate results in Table 7.4. Since we rely on the candidate set to filter the annotations for the documents, we might end up with significant annotations that are not present in the title and abstract. Secondly, the amount of training data is significantly less (see Supplementary) than in the CTD experiments, requiring research into unsupervised approaches Devlin et al. (2018) for this data. Lastly, dealing with out-of-vocabulary entities at test time required additional pre-training (described in Section 7.3.4.1), and our analysis indicated that these are not highly predictive for mention-level disambiguation due to the sparsity of the graph training data. Looking into more sophisticated methods

Model	Precision	Recall	F1
Top Candidate	8.9	5.3	6.6
Linker	11.3	6.6	8.3
End-to-End	**12.8**	**10.9**	**11.8**

Table 7.5: Results on the disease phenotype dataset

Xie et al. (2016); Gupta et al. (2017b) for dealing with unseen entities on this data would be an important problem for future work.

7.4 Related Work

Extracting entities and relations from text has been widely studied over the past few decades. In the biomedical domain specifically, there has been substantial progress on entity mention detection Greenberg et al. (2018); Wei et al. (2015a) and entity linking (often referred to as normalization in the bio NLP community) (Leaman and Gonzalez, 2008; Leaman et al., 2013, 2015; Leaman and Lu, 2016), and relation extraction (Wei et al., 2016; Krallinger et al., 2017a).

There have also been numerous works that have identified both entity mentions and relationships from text in both the general domain (Miwa and Bansal, 2016b) and in the biomedical domain (Li et al., 2017; Ammar et al., 2017a; Verga et al., 2018). Leaman and Lu (2016) showed that jointly considering named entity recognition (NER) and linking led to improved performance.

A few works have shown that jointly modeling relations and entity linking can improve performance. Le and Titov (2018) improved entity linking performance by modeling latent relations between entities. This is similar to coherence models Ganea and Hofmann (2017) in entity linking which consider the joint assignment of all linking decisions, but is more tractable as it focuses on only pairs of entities in a short context rather than complete sets within a document.

Luan et al. (2018) created a multi-task learning model for predicting entities, relations, and coreference in scientific documents. This model required supervision for

all three tasks and predictions amongst the different tasks were made independently rather than jointly.

7.5 Conclusion

In this chapter, we presented a model to simultaneously predict entity linking and entity relation decisions. This model can be trained without any mention-level supervision for entities or relations, and instead relies solely on weak and distant supervision at the document-level, readily available in many knowledge bases. To the best of our knowledge this is the first such model to consider this particular approach. The proposed model performs favorably as compared to a state-of-the-art pipeline approach to relation extraction by avoiding cascading error, while requiring less expensive annotation, opening possibilities for knowledge extraction in low-resource and expensive to annotate domains. Future work will look into a fully end-to-end model for document graph extraction, which does not rely on a trained NER system, as well as methods to simultaneously extract entity and relations from even weaker corpus-level distant supervision Mintz et al. (2009b).

CHAPTER 8

CONCLUSIONS

In this thesis we have presented methods for knowledge representation and extraction that combine neural architectures with structured ontologies. In Chapters 3 and 4 we built upon the work of universal schema, expanding its capacity for generalization to consider arbitrary text and newly encountered entities. In Chapter 5, we leveraged existing hierarchical ontologies to explicitly model hypernym relationships amongst concepts and types and enforcing these constraints over our learned embedding space. Finally, in Chapters 6 and 7 we developed methods that extract knowledge by considering a greater textual context while jointly reasoning about entity and relation decisions.

8.1 Future Directions

While we've presented new methods for enhancing knowledge representation, there is much left to be done. Some of this work involves general improvements to existing methods. For example, these tools are at the level where they can be quite useful in domains with adequate amounts of annotated training data. But in the more general case when we are considering low resource domains and very fine grained open semantic analysis, the accuracy of the models still needs to improve substantially before they can be fully exploited. Other areas are more open ended and I will next briefly describe a few of them.

8.1.1 Contextual Knowledge Graphs

One of the biggest areas lacking in current knowledge representation research is the notion of encoding knowledge in context. In general, KGs encode knowledge at an abstract level disentangled from concrete instances. While this can be very useful in many situations, such as reasoning about generalities of a concept, in other cases this can lead to ambiguities. For example, in the CTD knowledge base (Chapter 6), the data encodes the fact that the chemical Zonisamide both treats *and* causes the disease symptom Tremors. By tracing these facts back to their provenance, its clear that what is missing is the context. In experiments on rats, they found Zonisamide to be a treatment, while in experiments on humans, they found the opposite.

This example helps to illustrate the fact that KGs would benefit from more explicitly capturing context and modeling not just abstract general concepts, but also the specific concrete instances they occur in. Our work in Chapter 4 took steps in this direction by modeling general entity representations as a function of their concrete instances. Chapters 6 and 7 also considered this question by modeling larger textual contexts. Future work can leverage these ideas further to create contextual knowledge graphs that capture concepts and relationships and multiple levels of abstraction (See Figure 8.1).

8.1.1.1 Temporal Knowledge Graphs

Another often neglected concept in knowledge representation research is a notion of knowledge over time. In particular, research nearly always focuses on a a static world at a given single time slice. However, this neglects a lot of valuable information. One component of this is how facts evolve over time. In the simplest case this could be facts that are only true at a particular time step, such as a person's current job or location. But this can also be extended to the evolution of relationships between entities or groups of entities as well as events and their temporal ordering and

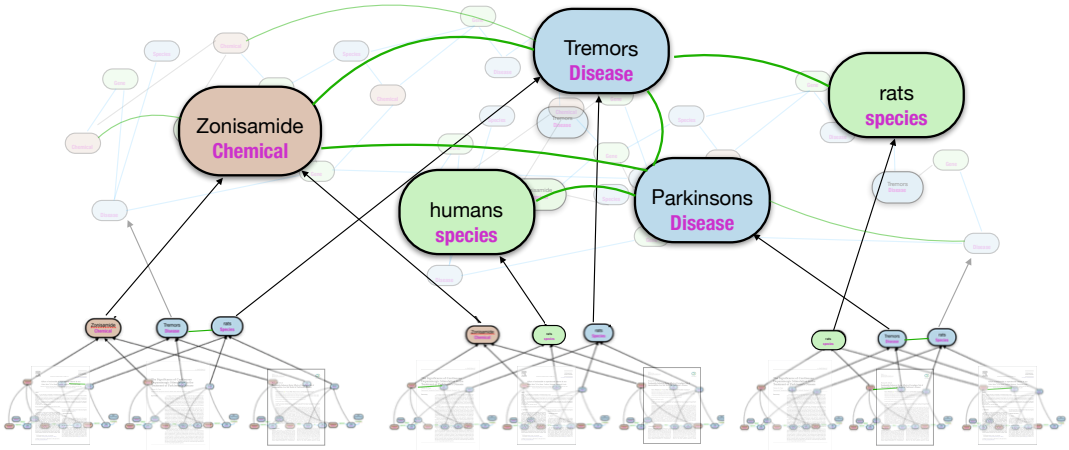


Figure 8.1: A schematic of a contextual knowledge graph incorporating the various ideas presented in this thesis. The highest level corresponds to the current instantiation of a knowledge graph consisting of generalized representations of concepts and entities abstracted away from specific concrete instances. At the bottom level are individual mentions contained within specific documents. Sentence and document level graphs are built up at the lower levels building upon the ideas in Chapters 6 and 7. The lower more concrete levels connect to higher more abstract levels with hypernym edges, as instances of concepts can be thought of as satisfying this hypernym property (as in Chapter 5). These higher level concept representations could potentially be only implicitly instantiated and actually represented as a function over their concrete instances (similar to the aggregation functions in Chapter 4). Finally, all of these representations can exist within a shared embedding space of structure and text encoded with neural architectures (Chapter 3), enabling interpretability, injecting useful biases, and human intervention.

dependencies. Moving forward, part of representing more contextualized knowledge could include representations of temporal properties.

8.1.1.2 Cognitively Inspired Models

Another fruitful future direction could be to have a greater focus on cognitively inspired models of knowledge representation. There has been substantial research over the years in developing cognitive architectures for more biologically plausible representations of knowledge (Anderson, 1983; Laird et al., 1987; Anderson, 1996; Laird et al., 2017). While building purely biologically plausible systems can enhance our understanding of human memory systems, there can also be benefit from a softer fo-

cus on modeling knowledge representation systems with inspiration from our existing understanding human cognition. One of these for example is related to the contextual knowledge discussed in Section 8.1.1. Rather than having a purely instance based or abstract based memory systems, humans retain both episodic and semantic memories capable of capturing different aspects of concepts and events (Tulving et al., 1972). Notions of episodic memory have begun to be used in reinforcement learning agents (Hassabis et al., 2017) and could potentially be leveraged for the type of general knowledge representation systems that we have been discussing.

8.1.2 Explicit vs Implicit Structure and Representations

Recent advances in neural modeling have called into question whether or not we need to be explicitly modeling structure at all. For example, open domain question answering systems have coupled an information retrieval system with neural reader module (Chen et al., 2017; Clark and Gardner, 2018). In this paradigm, their is no explicit structure or representation of knowledge. Instead, the knowledge is the text itself along with the parameters of the model. A related strand of research has shown that extremely large language models (Peters et al., 2018; Devlin et al., 2018; Radford et al., 2019) can capture a surprising amount of information and can be leveraged by machine reading models. These two developments along with their natural combination has led many researchers to question whether this constitutes the correct path forward for knowledge representation and reasoning research.

However, these models still have many shortcomings as they are less interpretable, difficult to augment with human knowledge, and have not yet been shown to be as capable at reasoning over many sources of information at once (though this is another area of active research (Welbl et al., 2018; Yang et al., 2018)). It will be interesting to see how knowledge representation research evolves in the future and to what extent explicit structured representations will play a part in the long term. There is reason

to believe that structure will continue to play a part going forward, particularly in the near term. Explicit structure will remain extremely useful in low resource domains that can benefit from priors and human knowledge, as well as in high value areas like medical treatments where interpretable predictions and provenance are both crucial and necessary for enhancing human decision making.

APPENDIX A

COMPOSITIONAL UNIVERSAL SCHEMA SUPPLEMENTARY

A.1 Additional Qualitative Results

Our model jointly embeds KB relations together with English and Spanish text. We demonstrate that plausible textual patterns are embedded close to the KB relations they express. Table A.1 shows top scoring English and Spanish patterns given sample relations from our TAC KB.

A.2 Implementation and Hyperparameters

We performed a small grid search over learning rate 0.0001, 0.005, 0.001, dropout 0.0, 0.1, 0.25, 0.5, dimension 50, 100, ℓ_2 gradient clipping 1, 10, 50, and epsilon 1e-8, 1e-6, 1e-4. All models are trained for a maximum of 15 epochs. The CNN and LSTM both use 100d embeddings while USchema uses 50d. The CNN and LSTM both learned 100-dimensional word embeddings which were randomly initialized. Using pre-trained embeddings did not substantially affect the results. Entity pair embeddings for the baseline USchema model are randomly initialized. For the models with LSTM and CNN text encoders, entity pair embeddings are initialized using vectors from the baseline USchema model. This performs better than random initialization. We perform ℓ_2 gradient clipping to 1 on all models. Universal Schema uses a batch size of 1024 while the CNN and LSTM use 128. All models are optimized using ADAM (Kingma and Ba, 2014a) with $\epsilon = 1e - 8$, $\beta_1 = 0.9$, and $\beta_2 = 0.999$ with

a learning rate of .001 for USchema and .0001 for CNN and LSTM. The CNN and LSTM also use dropout of 0.1 after the embedding layer.

A.3 Details Concerning Cosine Similarity Computation

We measure the similarity between r_{text} and r_{schema} by computing the vectors' cosine similarity. However, such a distance is not well-defined, since the model was trained using inner products between entity vectors and relation vectors, not between two relation vectors. The US likelihood is invariant to invertible transformations of the latent coordinate system, since $\sigma\left(u_{s,o}^{\top} v_r\right) = \sigma\left((A^{\top} u_{s,o})^{\top} A^{-1} v_r\right)$ for any invertible A. When taking inner products between two v terms, however, the implicit A^{-1} terms do not cancel out. We found that this issue can be minimized, and high quality predictive accuracy can be achieved, simply by using sufficient ℓ_2 regularization to avoid implicitly learning an A that substantially stretches the space.

A.4 Data Pre-processing, Distant Supervision and Extraction Pipeline

We replace tokens occurring less than 5 times in the corpus with UNK and normalize all digits to # (e.g. Oct-11-1988 becomes Oct-##-####). For each sentence, we then extract all entity pairs and the text between them as surface patterns, ignoring patterns longer than 20 tokens. This results in 48 million English 'relations'. In Section A.6, we describe a technique for normalizing the surface patterns. We filter out entity pairs that occurred less than 10 times in the data and extract the largest connected component in this entity co-occurrence graph. This is necessary for the baseline US model, as otherwise learning decouples into independent problems per connected component. Though the components are connected when using sentence encoders, we use only a single component to facilitate a fair comparison between modeling approaches. We add the distant supervision training facts from the Rela-

tionFactory system, i.e. 352,236 entity-pair-relation tuples obtained from Freebase and high precision seed patterns. The final training data contains a set of 3,980,164 (KB and openIE) facts made up of 549,760 unique entity pairs, 1,285,258 unique relations and 62,841 unique tokens.

We perform the same preprocessing on the Spanish data, resulting in 34 million raw surface patterns between entities. We then filter patterns that never occur with an entity pair found in the English data. This yields 860,502 Spanish patterns. Our multilingual model is trained on a combination of these Spanish patterns, the English surface patterns, and the distant supervision data described above. We learn word embeddings for 39,912 unique Spanish word types. After parameter tying for translation pairs (Section 3.2.5), there are 33,711 additional Spanish words not tied to English.

A.5 Generation of Cross-Lingual Tied Word Types

We follow the same procedure for generating translation pairs as Mikolov et al. (2013b). First, we select the top 6000 words occurring in the lowercased Europarl dataset for each language and obtain a Google translation. We then filter duplicates and translations resulting in multi-word phrases. We also remove English past participles (ending in -ed) as we found the Google translation interprets these as adjectives (e.g., 'she read the borrowed book' rather than 'she borrowed the book') and much of the relational structure in language we seek to model is captured by verbs. This resulted in 6201 translation pairs that occurred in our text corpus. Though higher quality translation dictionaries would likely improve this technique, our experimental results show that such automatically generated dictionaries perform well.

A.6 Open IE Pattern Normalization

To improve US generalization, our US relations use log-shortened patterns where the middle tokens in patterns longer than five tokens are simplified. For each long pattern we take the first two tokens and last two tokens, and replace all k remaining tokens with the number $\log k$. For example, the pattern **Barack Obama** *is married to a person named* **Michelle Obama** would be converted to: **Barack Obama** *is married [1] person named* **Michell Obama**. This shortening performs slightly better than whole patterns. LSTM and CNN variants use the entire sequence of tokens.

per:sibling
arg1, según petición the primeros ministro, su hermano gemelo *arg2* *arg1*, sea the principal favorito para esto oficina que también ambiciona su hermano *arg2* *arg1*, y su hermano gemelo, the primeros ministro *arg2*
arg1, for whose brother *arg2* *arg1* inherited his brother *arg2* *arg1* on saxophone and brother *arg2*
org:top_members_employees
arg2, presidente y director generales the *arg1* *arg2*, presidente of the negocios especializada *arg1* *arg2* (CIA), the director of the entidad, *arg1*
arg2, vice president and policy director of the *arg1* *arg2*, president of the German Soccer *arg1* *arg2*, president of the quasi-official *arg1*
per:alternate_names
arg1 (como también son sabido para *arg2* *arg2*-cuyos verdaderos nombre sea *arg1* *arg1* también sabido como *arg2*
arg1 aka *arg2* *arg1*, who also creates music under the pseudonym *arg2* *arg1* (of Modern Talking fame) aka *arg2*
per:cities_of_residence
arg1, poblado dónde vive *arg2* *arg1*, una ciudadano naturalizado american y nacido in *arg2* *arg1*, que vive in *arg2*
arg1 was born Jan. # , #### in *arg2* *arg1* was born on Monday in *arg2* *arg1* was born at Keighley in *arg2*

Table A.1: Top scoring patterns for both Spanish (top) and English (bottom) given query TAC relations.

APPENDIX B

HIERARCHICAL MODELING SUPPLEMENTARY

B.1 MedMentions Additional Details

Statistic	Train	Dev	Test
#Abstracts	2,964	370	370
#Sentences	28,457	3,497	3,268
#Mentions	199,977	24,026	22,141
#Entities	22,416	5,934	5,521

Table B.1: MedMentions statistics.

B.2 TypeNet Construction

Freebase type: musical_chord
Example entities: psalms_chord, power_chord harmonic_seventh_chord
chord.n.01: a straight line connecting two points on a curve
chord.n.02: a combination of three or more notes that blend harmoniously when sounded together
musical.n.01: a play or film whose action and dialogue is interspersed with singing and dancing

Table B.2: Example given to TypeNet annotators. Here, the Freebase type to be linked is musical_chord. This type is annotated in Freebase belonging to the entities psalms_chord, harmonic_seventh_chord, and power_chord. Below the list of example entities are candidate WordNet synsets obtained by substring matching between the Freebase type and all WordNet synsets. The correctly aligned synset is chord.n.02 shown in bold.

B.3 Model Implementation Details

For all of our experiments, we use pretrained 300 dimensional word vectors from Pennington et al. (2014). These embeddings are fixed during training. The type

Typeset	Count	Depth	Gold KB links
CoNLL-YAGO	4	1	Yes
OntoNotes 5.0	19	1	No
Gillick et al. (2014)	88	3	Yes
Figer	112	2	Yes
Hyena	505	9	No
Freebase	2k	2	Yes
WordNet	16k	14	No
TypeNet*	1,941	14	Yes

Table B.3: Statistics from various type sets. TypeNet is the largest type hierarchy with a gold mapping to KB entities. *The entire WordNet could be added to TypeNet increasing the total size to 17k types.

Freebase Types	1081
WordNet Synsets	860
child-of links	727
equivalence links	380
parent-of links	13
Freebase-Freebase links	614

Table B.4: Stats for the final TypeNet dataset. child-of, parent-of, and equivalence links are from Freebase types → WordNet synsets.

vectors and entity vectors are all 300 dimensional vectors initialized using Glorot initialization Glorot and Bengio (2010). The number of negative links for hierarchical training $n \in \{16, 32, 64, 128, 256\}$.

For regularization, we use dropout Srivastava et al. (2014b) with p $\in \{0.5, 0.75, 0.8\}$ on the sentence encoder output and L2 regularize all learned parameters with $\lambda \in \{1e\text{-}5, 5e\text{-}5, 1e\text{-}4\}$. All our parameters are optimized using Adam (Kingma and Ba, 2014b) with a learning rate of 0.001. We tune our hyper-parameters via grid search and early stopping on the development set.

B.4 FIGER Implementation Details

To train our models, we use the mention typing loss function defined in Section-5.2. For models with structure training, we additionally add in the hierarchical loss, along with a weight that is obtained by tuning on the dev set. We follow the same inference

time procedure as Shimaoka et al. (2017) For each mention, we first assign the type with the largest probability according to the logits, and then assign additional types based on the condition that their corresponding probability be greater than 0.5.

B.5 Wikipedia Data and Implementation Details

At train time, each training example randomly samples an entity bag of 10 mentions. At test time we classify bags of 20 mentions of an entity. The dataset contains a total of 344,246 entities mapped to the 1081 Freebase types from TypeNet. We consider all sentences in Wikipedia between 10 and 50 tokens long. Tokenization and sentence splitting was performed using NLTK Loper and Bird (2002). From these sentences, we considered all entities annotated with a cross-link in Wikipedia that we could link to Freebase and assign types in TypeNet. We then split the data by entities into a 90-5-5 train, dev, test split.

B.6 UMLS Implementation details

We pre-process each string by lowercasing and removing stop words. We consider ngrams from size 1 to 5 and keep the top 100,000 features and the final vectors are L2 normalized. For each mention, In our experiments we consider the top 100 most similar entities as the candidate set.

B.6.1 Candidate Generation Details

Each mention and each canonical entity string in UMLS are mapped to TFIDF character ngram vectors. We pre-process each string by lowercasing and removing stop words. We consider ngrams from size 1 to 5 and keep the top 100,000 features and the final vectors are L2 normalized. For each mention, we calculate the cosine similarity, csim, between the mention string and each canonical entity string. In our experiments we consider the top 100 most similar entities as the candidate set.

APPENDIX C

BRAN SUPPLEMENTARY

C.1 BRAN Implementation Details

The model is implemented in Tensorflow (Abadi et al., 2015) and trained on a single TitanX gpu. The number of transformer block repeats is $B = 2$. We optimize the model using Adam Kingma and Ba (2014a) with best parameters chosen for ϵ, β_1, β_2 chosen from the development set. The learning rate is set to 0.0005 and batch size 32. In all of our experiments we set the number of attention heads to $h = 4$.

We clip the gradients to norm 10 and apply noise to the gradients Neelakantan et al. (2015b). We tune the decision threshold for each relation type separately and perform early stopping on the development set. We apply dropout Srivastava et al. (2014a) to the input layer randomly replacing words with a special UNK token with keep probability .85. We additionally apply dropout to the input T (word embedding + position embedding), interior layers, and final state. At each step, we randomly sample a positive or negative (NULL class) minibatch with probability 0.5.

C.2 Chemical Disease Relations Dataset

Token embeddings are pre-trained using skipgram Mikolov et al. (2013a) over a random subset of 10% of all PubMed abstracts with window size 10 and 20 negative samples. We merge the train and development sets and randomly take 850 abstracts for training and 150 for early stopping. Our reported results are averaged over 10 runs and using different splits. All baselines train on both the train and development set. Models took between 4 and 8 hours to train.

ϵ was set to 1e-4, β_1 to .1, and β_2 to 0.9. Gradient noise $\eta = .1$. Dropout was applied to the word embeddings with keep probability 0.85, internal layers with 0.95 and final bilinear projection with 0.35 for the standard CRD dataset experiments. When adding the additional weakly labeled data: word embeddings with keep probability 0.95, internal layers with 0.95 and final bilinear projection with 0.5.

C.3 Chemical Protein Relations Dataset

We construct our byte-pair encoding vocabulary using a budget of 7500. The dataset contains annotations for a larger set of relation types than are used in evaluation. We train on only the relation types in the evaluation set and set the remaining types to the Null relation. The embedding dimension is set to 200 and all embeddings are randomly initialized. ϵ was set to 1e-8, β_1 to .1, and β_2 to 0.9. Gradient noise $\eta = 1.0$. Dropout was applied to the word embeddings with keep probability 0.5, internal layers with 1.0 and final bilinear projection with 0.85 for the standard CRD dataset experiments.

C.4 Full CTD Dataset

We tune separate decision boundaries for each relation type on the development set. For each prediction, the relation type with the maximum probability is assigned. If the probability is below the relation specific threshold, the prediction is set to NULL. We use embedding dimension 128 with all embeddings randomly initialized. Our byte pair encoding vocabulary is constructed with a budget of 50,000. Models took 1 to 2 days to train.

ϵ was set to 1e-4, β_1 to .1, and β_2 to 0.9. Gradient noise $\eta = .1$.Dropout was applied to the word embeddings with keep probability 0.95, internal layers with 0.95 and final bilinear projection with 0.5

BIBLIOGRAPHY

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Ammar, W., Peters, M., Bhagavatula, C., and Power, R. (2017a). The ai2 system at semeval-2017 task 10 (scienceie): semi-supervised end-to-end entity and relation extraction. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 592–596.

Ammar, W., Peters, M. E., Bhagavatula, C., and Power, R. (2017b). The ai2 system at semeval-2017 task 10 (scienceie): semi-supervised end-to-end entity and relation extraction. *nucleus*, 2(e2):e2.

Anderson, J. R. (1983). The architecture of cognition.

Anderson, J. R. (1996). Act: A simple theory of complex cognition. *American Psychologist*, 51(4):355.

Angeli, G., Gupta, S., Jose, M., Manning, C. D., Ré, C., Tibshirani, J., Wu, J. Y., Wu, S., and Zhang, C. (2014). Stanford's 2014 slot filling systems. *TAC KBP*.

Apweiler, R., Bairoch, A., Wu, C. H., Barker, W. C., Boeckmann, B., Ferro, S., Gasteiger, E., Huang, H., Lopez, R., Magrane, M., et al. (2004). Uniprot: the universal protein knowledgebase. *Nucleic acids research*, 32(suppl_1):D115–D119.

Ba, J. L., Kiros, J. R., and Hinton, G. E. (2016). Layer normalization. *arXiv preprint arXiv:1607.06450*.

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural Machine Translation by Jointly Learning to Align and Translate. In *3rd International Conference for Learning Representations (ICLR)*, San Diego, California, USA.

Banko, M., Cafarella, M. J., Soderland, S., Broadhead, M., and Etzioni, O. (2007). Open information extraction from the web. In *International Joint Conference on Artificial Intelligence*.

Bansal, T., Das, M., and Bhattacharyya, C. (2015). Content driven user profiling for comment-worthy recommendations of news and blog articles. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pages 195–202. ACM.

Bentor, Y., Viswanathan, V., and Mooney, R. (2014). University of texas at austin kbp 2014 slot filling system: Bayesian logic programs for textual inference. In *Proceedings of the Seventh Text Analysis Conference: Knowledge Base Population (TAC 2014)*.

Bodenreider, O. (2004). The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research*, 32(suppl_1):D267–D270.

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. (2008a). Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. ACM.

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. (2008b). Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. AcM.

Bordes, A., Chopra, S., and Weston, J. (2014). Question answering with subgraph embeddings. *arXiv preprint arXiv:1406.3676*.

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., and Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, pages 2787–2795.

Bragin, E., Chatzimichali, E. A., Wright, C. F., Hurles, M. E., Firth, H. V., Bevan, A. P., and Swaminathan, G. J. (2013). Decipher: database for the interpretation of phenotype-linked plausibly pathogenic sequence and copy-number variation. *Nucleic acids research*, 42(D1):D993–D1000.

Buchanan, B. (1984). Rule based expert systems. *The MYCIN Experiments of the Stanford Heuristic Programming Project*.

Bunescu, R. and Mooney, R. (2007). Learning to extract relations from the web using minimal supervision. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 576–583.

Bunescu, R. C. and Pasca, M. (2006). Using encyclopedic knowledge for named entity disambiguation. In *Eacl*, volume 6, pages 9–16.

Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka, E. R., and A (2010). Toward an architecture for never-ending language learning. In *AAAI*, Atlanta, Georgia, USA.

Caruana, R. (1993). Multitask learning: A knowledge-based source of inductive bias. *Proceedings of the Tenth International Conference on International Conference on Machine Learning (ICML)*, pages 41–48.

Chatr-aryamontri, A., Oughtred, R., Boucher, L., Rust, J., Chang, C., Kolas, N. K., O'Donnell, L., Oster, S., Theesfeld, C., Sellam, A., et al. (2017). The biogrid interaction database: 2017 update. *Nucleic acids research*, 45(D1):D369–D379.

Chen, D., Fisch, A., Weston, J., and Bordes, A. (2017). Reading wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879.

Chen, Q., Peng, Y., and Lu, Z. (2018). Biosentvec: creating sentence embeddings for biomedical texts. *arXiv preprint arXiv:1810.09302*.

Clark, C. and Gardner, M. (2018). Simple and effective multi-paragraph reading comprehension. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 845–855.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537.

Craven, M. and Kumlien, J. (1999). Constructing biological knowledge bases by extracting information from text sources. In *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, pages 77–86. AAAI Press.

Cucerzan, S. (2007). Large-scale named entity disambiguation based on wikipedia data. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*.

Dalton, J., Dietz, L., and Allan, J. (2014). Entity query feature expansion using knowledge base links. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 365–374. ACM.

Dalvi, B., Minkov, E., Talukdar, P. P., and Cohen, W. W. (2015). Automatic gloss finding for a knowledge base using ontological constraints. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, pages 369–378. ACM.

Das, R., Neelakantan, A., Belanger, D., and McCallum, A. (2017a). Chains of reasoning over entities, relations, and text using recurrent neural networks. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 132–141, Valencia, Spain. Association for Computational Linguistics.

Das, R., Neelakantan, A., Belanger, D., and McCallum, A. (2017b). Chains of reasoning over entities, relations, and text using recurrent neural networks. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, volume 1, pages 132–141.

Das, R., Zaheer, M., Reddy, S., and McCallum, A. (2017c). Question answering on knowledge bases and text using universal schema and memory networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 358–365, Vancouver, Canada. Association for Computational Linguistics.

Davis, A. P., Grondin, C. J., Johnson, R. J., Sciaky, D., McMorran, R., Wiegers, J., Wiegers, T. C., and Mattingly, C. J. (2018). The comparative toxicogenomics database: update 2019. *Nucleic acids research*, 47(D1):D948–D954.

Davis, A. P., Murphy, C. G., Saraceni-Richards, C. A., Rosenstein, M. C., Wiegers, T. C., and Mattingly, C. J. (2008). Comparative toxicogenomics database: a knowledgebase and discovery tool for chemical–gene–disease networks. *Nucleic acids research*, 37(suppl_1):D786–D792.

Davis, A. P., Wiegers, T. C., Rosenstein, M. C., and Mattingly, C. J. (2012). Medic: a practical disease vocabulary used at the comparative toxicogenomics database. *Database*, 2012.

Del Corro, L., Abujabal, A., Gemulla, R., and Weikum, G. (2015). Finet: Context-aware fine-grained named entity typing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Dettmers, T., Minervini, P., Stenetorp, P., and Riedel, S. (2018). Convolutional 2d knowledge graph embeddings. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Doddington, G., Mitchell, A., Przybocki, M., Ramshaw, L., Strassel, S., and Weischedel, R. (2004). The automatic content extraction (ace) program tasks, data, and evaluation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation*.

Doğan, R. I., Leaman, R., and Lu, Z. (2014). Ncbi disease corpus: a resource for disease name recognition and concept normalization. *Journal of biomedical informatics*, 47:1–10.

Dong, L. (2017). Amazon product graph.

dos Santos, C., Xiang, B., and Zhou, B. (2015a). Classifying relations by ranking with convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 626–634, Beijing, China. Association for Computational Linguistics.

dos Santos, C. N., Xiang, B., and Zhou, B. (2015b). Classifying relations by ranking with convolutional neural networks. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing ACL*.

Dozat, T. and Manning, C. D. (2017). Deep biaffine attention for neural dependency parsing. *5th International Conference on Learning Representations*.

Durrett, G. and Klein, D. (2014). A joint model for entity analysis: Coreference, typing, and linking. *Transactions of the Association for Computational Linguistics*, 2:477–490.

Etzioni, O., Banko, M., Soderland, S., and Weld, D. S. (2008). Open information extraction from the web. *Communications of the ACM*, 51(12):68–74.

Faruqui, M., Dodge, J., Jauhar, S. K., Dyer, C., Hovy, E., and Smith, N. A. (2014). Retrofitting word vectors to semantic lexicons. *arXiv preprint arXiv:1411.4166*.

Faruqui, M. and Kumar, S. (2015). Multilingual open relation extraction using cross-lingual projection. *arXiv preprint arXiv:1503.06450*.

Feigenbaum, E. A. (1980). Expert systems: looking back and looking ahead. In *GI-10. Jahrestagung*, pages 1–14. Springer.

Fellbaum, C. (1998). *WordNet*. Wiley Online Library.

Francis-Landau, M., Durrett, G., and Klein, D. (2016). Capturing semantic similarity for entity linking with convolutional neural networks. In *Proceedings of NAACL-HLT*, pages 1256–1261.

Gabrilovich, E., Ringgaard, M., and Subramanya, A. (2013). Facc1: Freebase annotation of clueweb corpora, version 1 (release date 2013-06-26, format version 1, correction level 0). http://lemurproject.org/clueweb09/FACC1/.

Gage, P. (1994). A new algorithm for data compression. *The C Users Journal*, 12(2):23–38.

Ganea, O.-E. and Hofmann, T. (2017). Deep joint entity disambiguation with local neural attention. *arXiv preprint arXiv:1704.04920*.

García-Durán, A., Bordes, A., Usunier, N., and Grandvalet, Y. (2016). Combining two and three-way embedding models for link prediction in knowledge bases. *Journal of Artificial Intelligence Research*, 55(1):715–742.

Gardner, M., Talukdar, P., Krishnamurthy, J., and Mitchell, T. (2014). Incorporating vector space similarity in random walk inference over knowledge bases. In *Empirical Methods in Natural Language Processing*.

Gillick, D., Lazic, N., Ganchev, K., Kirchner, J., and Huynh, D. (2014). Context-dependent fine-grained entity type tagging. *CoRR*, abs/1412.1820.

Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Glorot, X., Bordes, A., and Bengio, Y. (2011). Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 315–323.

Google (2012). Introducing the knowledge graph: things, not strings.

Gouws, S., Bengio, Y., and Corrado, G. (2015). B IL BOWA : Fast Bilingual Distributed Representations without Word Alignments. *Icml*, pages 1–10.

Graves, A., Wayne, G., and Danihelka, I. (2014). Neural Turing Machines. *arXiv preprint arxiv:1410.5401*.

Greenberg, N., Bansal, T., Verga, P., and McCallum, A. (2018). Marginal likelihood training of bilstm-crf for biomedical named entity recognition from disjoint label sets. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2824–2829.

Grishman, R. and Sundheim, B. (1996). Message understanding conference-6: A brief history. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*, volume 1.

Groza, T., Köhler, S., Doelken, S., Collier, N., Oellrich, A., Smedley, D., Couto, F. M., Baynam, G., Zankl, A., and Robinson, P. N. (2015). Automatic concept recognition using the human phenotype ontology reference and test suite corpora. *Database*, 2015.

Gu, J., Qian, L., and Zhou, G. (2016). Chemical-induced disease relation extraction with various linguistic features. *Database*, 2016.

Gu, J., Sun, F., Qian, L., and Zhou, G. (2017). Chemical-induced disease relation extraction via convolutional neural network. *Database*, 2017.

Gu, K., Miller, J., and Liang, P. (2015). Traversing knowledge graphs in vector space. *arXiv preprint arXiv:1506.01094*.

Gupta, N., Singh, S., and Roth, D. (2017a). Entity linking via joint encoding of types, descriptions, and context. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2671–2680, Copenhagen, Denmark. Association for Computational Linguistics.

Gupta, N., Singh, S., and Roth, D. (2017b). Entity linking via joint encoding of types, descriptions, and context. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2681–2690.

Gurulingappa, H., Rajput, A. M., Roberts, A., Fluck, J., Hofmann-Apitius, M., and Toldo, L. (2012). Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports. *Journal of biomedical informatics*, 45(5):885–892.

Hamosh, A., Scott, A. F., Amberger, J. S., Bocchini, C. A., and McKusick, V. A. (2005). Online mendelian inheritance in man (omim), a knowledgebase of human genes and genetic disorders. *Nucleic acids research*, 33(suppl_1):D514–D517.

Hassabis, D., Kumaran, D., Summerfield, C., and Botvinick, M. (2017). Neuroscience-inspired artificial intelligence. *Neuron*, 95(2):245–258.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Hearst, M. A. (1992). Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the International Conference on Computational Linguistics (COLING)*.

Hendrickx, I., Kim, S. N., Kozareva, Z., Nakov, P., Ó Séaghdha, D., Padó, S., Pennacchiotti, M., Romano, L., and Szpakowicz, S. (2009). Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*, pages 94–99. Association for Computational Linguistics.

Hermann, K. M. and Blunsom, P. (2014). Multilingual models for compositional distributed semantics. *arXiv preprint arXiv:1404.4641*.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., et al. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6):82–97.

Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(2):107–116.

Hochreiter, S. and Schmidhuber, J. (1997a). Long short-term memory. In *Neural Computation*.

Hochreiter, S. and Schmidhuber, J. (1997b). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Hoffmann, R., Zhang, C., Ling, X., Zettlemoyer, L., and Weld, D. S. (2011). Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 541–550. Association for Computational Linguistics.

Ji, H., Grishman, R., Dang, H. T., Griffitt, K., and Ellis, J. (2010). Overview of the tac 2010 knowledge base population track. In *Third Text Analysis Conference (TAC 2010)*, volume 3, pages 3–3.

Kalchbrenner, N., Grefenstette, E., and Blunsom, P. (2014). A convolutional neural network for modelling sentences. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*.

Kim, J.-D., Ohta, T., and Tsujii, J. (2008). Corpus annotation for mining biomedical events from literature. *BMC bioinformatics*, 9(1):10.

Kim, Y. (2014). Convolutional neural networks for sentence classification. *EMNLP*.

Kingma, D. P. and Ba, J. (2014a). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kingma, D. P. and Ba, J. (2014b). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Kiperwasser, E. and Goldberg, Y. (2016). Simple and accurate dependency parsing using bidirectional lstm feature representations. *Transactions of the Association for Computational Linguistics*, 4:313–327.

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86. Citeseer.

Köhler, S., Carmody, L., Vasilevsky, N., Jacobsen, J. O. B., Danis, D., Gourdine, J.-P., Gargano, M., Harris, N. L., Matentzoglu, N., McMurry, J. A., et al. (2018). Expansion of the human phenotype ontology (hpo) knowledge base and resources. *Nucleic acids research*, 47(D1):D1018–D1027.

Krallinger, M., Rabal, O., Akhondi, S. A., et al. (2017a). Overview of the biocreative vi chemical-protein interaction track. In *Proceedings of the sixth BioCreative challenge evaluation workshop*, volume 1, pages 141–146.

Krallinger, M., Rabal, O., Akhondi, S. A., Pérez, M. P., Santamaría, J., Rodríguez, P. G., Tsatsaronis, G., Intxaurrondo, A., López, J. A., Nandal, U., Buel, E. V., Chandrasekhar, A., Rodenburg, M., Laegreid, A., Doornenbal, M., Oyarzabal, J., Lourenço, A., and Valencia, A. (2017b). Overview of the biocreative vi chemical-protein interaction track. *Proceedings of the BioCreative VI Workshop*, page 140.

Krallinger, M., Rabal, O., Leitner, F., Vazquez, M., Salgado, D., Lu, Z., Leaman, R., Lu, Y., Ji, D., Lowe, D. M., et al. (2015). The chemdner corpus of chemicals and drugs and its annotation principles. *Journal of cheminformatics*, 7(S1):S2.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Kulick, S., Bies, A., Liberman, M., Mandel, M., Winters, S., and White, P. (2012). Integrated annotation for biomedical information extraction. *HLT/NAACL Workshop: Biolink*.

Lafferty, J., McCallum, A., and Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Laird, J. E., Lebiere, C., and Rosenbloom, P. S. (2017). A standard model of the mind: Toward a common computational framework across artificial intelligence, cognitive science, neuroscience, and robotics. *AAAI AI Magazine*, 38:13–26.

Laird, J. E., Newell, A., and Rosenbloom, P. S. (1987). Soar: An architecture for general intelligence. *Artificial intelligence*, 33(1):1–64.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., and Dyer, C. (2016). Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Larochelle, H., Erhan, D., and Bengio, Y. (2008). Zero-data learning of new tasks. In *National Conference on Artificial Intelligence.*

Le, P. and Titov, I. (2018). Improving entity linking by modeling latent relations between mentions. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1595–1604.

Leaman, R. and Gonzalez, G. (2008). Banner: an executable survey of advances in biomedical named entity recognition. In *Biocomputing 2008*, pages 652–663. World Scientific.

Leaman, R., Islamaj Doğan, R., and Lu, Z. (2013). Dnorm: disease name normalization with pairwise learning to rank. *Bioinformatics*, 29(22):2909–2917.

Leaman, R. and Lu, Z. (2016). Taggerone: joint named entity recognition and normalization with semi-markov models. *Bioinformatics*, 32(18):2839–2846.

Leaman, R., Wei, C.-H., and Lu, Z. (2015). tmchem: a high performance approach for chemical named entity recognition and normalization. *Journal of cheminformatics*, 7(1):S3.

Lenat, D. B., Guha, R. V., Pittman, K., Pratt, D., and Shepherd, M. (1990). Cyc: toward programs with common sense. *Communications of the ACM*, 33(8):30–49.

Li, F., Zhang, M., Fu, G., and Ji, D. (2017). A neural joint model for entity and relation extraction from biomedical text. *BMC bioinformatics*, 18(1):198.

Li, J., Sun, Y., Johnson, R. J., Sciaky, D., Wei, C.-H., Leaman, R., Davis, A. P., Mattingly, C. J., Wiegers, T. C., and Lu, Z. (2016a). Biocreative v cdr task corpus: a resource for chemical disease relation extraction. *Database*, 2016.

Li, X., Taheri, A., Tu, L., and Gimpel, K. (2016b). Commonsense knowledge base completion. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1445–1455, Berlin, Germany. Association for Computational Linguistics.

Lin, H., Zhao, Z., Jia, Y., Wang, Y., Xiong, J., and Li, X. (2014). OpenKN at TAC KBP 2014.

Lin, Y., Liu, Z., Sun, M., Liu, Y., and Zhu, X. (2015). Learning entity and relation embeddings for knowledge graph completion. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 2181–2187, Austin, Texas, US.

Lin, Y., Shen, S., Liu, Z., Luan, H., and Sun, M. (2016). Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2124–2133, Berlin, Germany. Association for Computational Linguistics.

Ling, X. and Weld, D. S. (2012a). Fine-grained entity recognition. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*.

Ling, X. and Weld, D. S. (2012b). Fine-grained entity recognition. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*, Toronto, Ontario, CA.

Liu, S., Shen, F., Wang, Y., Rastegar-Mojarad, M., Elayavilli, R. K., Chaundary, V., and Liu, H. (2017). Attention-based neural networks for chemical protein relation extraction. *Proceedings of the BioCreative VI Workshop*.

Loper, E. and Bird, S. (2002). Nltk: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics-Volume 1*, pages 63–70. Association for Computational Linguistics.

Luan, Y., He, L., Ostendorf, M., and Hajishirzi, H. (2018). Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3219–3232.

Luong, T., Pham, H., and Manning, C. D. (2015). Bilingual word representations with monolingual quality in mind. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 151–159.

Mallory, E. K., Zhang, C., Ré, C., and Altman, R. B. (2015). Large-scale extraction of gene interactions from full-text literature using deepdive. *Bioinformatics*, 32(1):106–113.

McCallum, A., Schultz, K., and Singh, S. (2009). FACTORIE: Probabilistic programming via imperatively defined factor graphs. In *Neural Information Processing Systems (NIPS)*.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Le, Q. V., and Sutskever, I. (2013b). Exploiting Similarities among Languages for Machine Translation. In *arXiv preprint arXiv:1309.4168v1*, pages 1–10.

Min, B., Grishman, R., Wan, L., Wang, C., and Gondek, D. (2013). Distant supervision for relation extraction with an incomplete knowledge base. In *HLT-NAACL*, pages 777–782.

Mintz, M., Bills, S., Snow, R., and Jurafsky, D. (2009a). Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, Suntec, Singapore. Association for Computational Linguistics.

Mintz, M., Bills, S., Snow, R., and Jurafsky, D. (2009b). Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics.

Miwa, M. and Bansal, M. (2016a). End-to-end relation extraction using lstms on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1116, Berlin, Germany. Association for Computational Linguistics.

Miwa, M. and Bansal, M. (2016b). End-to-end relation extraction using lstms on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1105–1116.

Murty, S., Verga, P., Vilnis, L., Radovanovic, I., and McCallum, A. (2018). Hierarchical losses and new resources for fine-grained entity typing and linking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 97–109.

Neelakantan, A. and Chang, M.-W. (2015a). Inferring missing entity type instances for knowledge base completion: New dataset and methods. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 515–525, Denver, Colorado. Association for Computational Linguistics.

Neelakantan, A. and Chang, M.-W. (2015b). Inferring missing entity type instances for knowledge base completion: New dataset and methods. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 515–525, Denver, Colorado. Association for Computational Linguistics.

Neelakantan, A., Le, Q. V., and Sutskever, I. (2016). Neural Programmer: Inducing latent programs with gradient descent. In *4th International Conference for Learning Representations (ICLR)*, San Juan, Puerto Rico.

Neelakantan, A., Roth, B., and McCallum, A. (2015a). Compositional vector space models for knowledge base completion. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 156–166.

Neelakantan, A., Shankar, J., Passos, A., and McCallum, A. (2014). Efficient non-parametric estimation of multiple embeddings per word in vector space. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1059–1069, Doha, Qatar. Association for Computational Linguistics.

Neelakantan, A., Vilnis, L., Le, Q. V., Sutskever, I., Kaiser, L., Kurach, K., and Martens, J. (2015b). Adding gradient noise improves learning for very deep networks. *arXiv preprint arXiv:1511.06807*.

Newell, A., Shaw, J. C., and Simon, H. A. (1959). Report on a general problem-solving program. In *Proceedings of the International Conference on Information Processing*.

Newell, A. and Simon, H. (1956). The logic theory machine–a complex information processing system. *IRE Transactions on information theory*, 2(3):61–79.

Nickel, M. and Kiela, D. (2017). Poincar\'e embeddings for learning hierarchical representations. *arXiv preprint arXiv:1705.08039*.

Nickel, M., Murphy, K., Tresp, V., and Gabrilovich, E. (2015). A review of relational machine learning for knowledge graphs: From multi-relational link prediction to automated knowledge graph construction. *arXiv preprint arXiv:1503.00759*.

Nickel, M., Tresp, V., and Kriegel, H. (2011a). A three-way model for collective learning on multi-relational data. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Nickel, M., Tresp, V., and Kriegel, H.-P. (2011b). A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 809–816, Bellevue, Washington, USA.

Orr, D., Subramanya, A., Gabrilovich, E., and Ringgaard, M. (2013). 11 billion clues in 800 million documents: A web research corpus annotated with freebase concepts. http://googleresearch.blogspot.com/2013/07/11-billion-clues-in-800-million.html.

Pantel, P., Lin, T., and Gamon, M. (2012). Mining entity types from query logs via user intent modeling. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 563–571, Jeju Island, Korea. Association for Computational Linguistics.

Park, S.-T. and Chu, W. (2009). Pairwise preference regression for cold-start recommendation. In *Proceedings of the third ACM conference on Recommender systems*, pages 21–28, New York, NY, USA. ACM.

Pavan, S., Rommel, K., Marquina, M. E. M., Höhn, S., Lanneau, V., and Rath, A. (2017). Clinical practice guidelines for rare diseases: the orphanet database. *PloS one*, 12(1):e0170365.

Peng, N., Poon, H., Quirk, C., Toutanova, K., and Yih, W.-t. (2017). Cross-sentence n-ary relation extraction with graph lstms. *Transactions of the Association for Computational Linguistics*, 5:101–115.

Peng, Y., Wei, C.-H., and Lu, Z. (2016). Improving chemical disease relation extraction with rich features and weakly labeled data. *Journal of cheminformatics*, 8(1):53.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proc. of NAACL*.

Piñero, J., Bravo, À., Queralt-Rosinach, N., Gutiérrez-Sacristán, A., Deu-Pons, J., Centeno, E., García-García, J., Sanz, F., and Furlong, L. I. (2017). Disgenet: a comprehensive platform integrating information on human disease-associated genes and variants. *Nucleic acids research*, 45(D1):D833–D839.

Poon, H., Toutanova, K., and Quirk, C. (2014). Distant supervision for cancer pathway extraction from text. In *Pacific Symposium on Biocomputing Co-Chairs*, pages 120–131.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 14(3):130–137.

Pyysalo, S., Ginter, F., Heimonen, J., Björne, J., Boberg, J., Järvinen, J., and Salakoski, T. (2007). Bioinfer: a corpus for information extraction in the biomedical domain. *BMC bioinformatics*, 8(1):50.

Quirk, C. and Poon, H. (2017). Distant supervision for relation extraction beyond the sentence boundary. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1171–1182, Valencia, Spain. Association for Computational Linguistics.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners.

Raiman, J. R. and Raiman, O. M. (2018). Deeptype: multilingual entity linking by neural type system evolution. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Ratinov, L. and Roth, D. (2009). Design challenges and misconceptions in named entity recognition. In *Proceedings of the thirteenth conference on computational natural language learning*, pages 147–155. Association for Computational Linguistics.

Ren, X., He, W., Qu, M., Voss, C. R., Ji, H., and Han, J. (2016). Label noise reduction in entity typing by heterogeneous partial-label embedding. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1825–1834.

Rendle, S., Freudenthaler, C., Gantner, Z., and Schmidt-Thieme, L. (2009). Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 452–461, Montreal, QC, Canada.

Riedel, S., McClosky, D., Surdeanu, M., McCallum, A., and D. Manning, C. (2011). Model combination for event extraction in bionlp 2011. In *Proceedings of BioNLP Shared Task 2011 Workshop*, pages 51–55, Portland, Oregon, USA. Association for Computational Linguistics.

Riedel, S., Yao, L., and McCallum, A. (2010). Modeling relations and their mentions without labeled text. *Machine learning and knowledge discovery in databases*, pages 148–163.

Riedel, S., Yao, L., McCallum, A., and Marlin, B. M. (2013a). Relation extraction with matrix factorization and universal schemas. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 74–84.

Riedel, S., Yao, L., McCallum, A., and Marlin, B. M. (2013b). Relation extraction with matrix factorization and universal schemas. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 74–84, Atlanta, Georgia. Association for Computational Linguistics.

Rocktaschel, T., Singh, S., and Riedel, S. (2015). Injecting logical background knowledge into embeddings for relation extraction. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386.

Roth, B., Barth, T., Chrupała, G., Gropp, M., and Klakow, D. (2014a). Relationfactory: A fast, modular and effective system for knowledge base population. *EACL 2014*, page 89.

Roth, B., Barth, T., Wiegand, M., Singh, M., and Klakow, D. (2014b). Effective slot filling based on shallow distant supervision methods. *arXiv preprint arXiv:1401.1158*.

Roth, B., Monath, N., Belanger, D., Strubell, E., Verga, P., and McCallum, A. (2015). Building knowledge bases with universal schema: Cold start and slot-filling approaches.

Rttmelhart, D., McClelland, J., Group, P. R., et al. (1986). Parallel distributed processing–explorations in the microstructure of cognition, vol. 1-2.

Schein, A. I., Popescul, A., Ungar, L. H., and Pennock, D. M. (2002). Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 253–260. ACM.

Segura-Bedmar, I., Martínez, P., and Herrero Zazo, M. (2013). Semeval-2013 task 9 : Extraction of drug-drug interactions from biomedical texts (ddiextraction 2013). In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 341–350, Atlanta, Georgia, USA. Association for Computational Linguistics.

Sennrich, R., Haddow, B., and Birch, A. (2015). Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Shen, W., Wang, J., and Han, J. (2015). Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering*, 27(2):443–460.

Shimaoka, S., Stenetorp, P., Inui, K., and Riedel, S. (2016). An attentive neural architecture for fine-grained entity type classification. In *Proceedings of the 5th Workshop on Automated Knowledge Base Construction*, pages 69–74, San Diego, CA. Association for Computational Linguistics.

Shimaoka, S., Stenetorp, P., Inui, K., and Riedel, S. (2017). Neural architectures for fine-grained entity type classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1271–1280, Valencia, Spain. Association for Computational Linguistics.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. (2017). Mastering the game of go without human knowledge. *Nature*, 550(7676):354.

Socher, R., Chen, D., Manning, C. D., and Ng, A. (2013). Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems*, pages 926–934, Lake Tahoe, Nevada, USA.

Spitkovsky, V. I. and Chang, A. X. (2012). A cross-lingual dictionary for english wikipedia concepts.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014a). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014b). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*.

Strubell, E., Verga, P., Belanger, D., and McCallum, A. (2017). Fast and accurate entity recognition with iterated dilated convolutions. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2670–2680.

Suchanek, F. M., Kasneci, G., and Weikum, G. (2007a). Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706, Banff, Alberta, Canada. ACM.

Suchanek, F. M., Kasneci, G., and Weikum, G. (2007b). Yago: a core of semantic knowledge. In *Proceedings of the International Conference on World Wide Web (WWW)*.

Suchanek, F. M., Kasneci, G., and Weikum, G. (2007c). Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706. ACM.

Sukhbaatar, S., Weston, J., Fergus, R., et al. (2015). End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448, Montreal, QC, Canada.

Surdeanu, M. and Ji., H. (2014). Overview of the english slot filling track at the tac2014 knowledge base population evaluation. *Proc. Text Analysis Conference (TAC2014)*.

Surdeanu, M., Tibshirani, J., Nallapati, R., and Manning, C. D. (2012). Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 455–465. Association for Computational Linguistics.

Sutskever, I., Vinyals, O., and Le, Q. V. V. (2014). Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*.

Swampillai, K. and Stevenson, M. (2011). Extracting relations within and across sentences. In *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*, pages 25–32, Hissar, Bulgaria. RANLP 2011 Organising Committee.

Tjong Kim Sang, E. F. and De Meulder, F. (2003). Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 142–147. Association for Computational Linguistics.

Toutanova, K., Chen, D., Pantel, P., Poon, H., Choudhury, P., and Gamon, M. (2015). Representing text for joint embedding of text and knowledge bases. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1499–1509, Lisbon, Portugal. Association for Computational Linguistics.

Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., and Bouchard, G. (2016). Complex embeddings for simple link prediction. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Tulving, E. et al. (1972). Episodic and semantic memory. *Organization of memory*, 1:381–403.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017a). Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017b). Attention is all you need. In *Conference on Advances in Neural Information Processing (NIPS)*.

Vendrov, I., Kiros, R., Fidler, S., and Urtasun, R. (2016). Order-embeddings of images and language. *ICLR*.

Verga, P., Belanger, D., Strubell, E., Roth, B., and McCallum, A. (2016a). Multilingual relation extraction using compositional universal schema. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 886–896, San Diego, California. Association for Computational Linguistics.

Verga, P., Belanger, D., Strubell, E., Roth, B., and McCallum, A. (2016b). Multilingual relation extraction using compositional universal schema. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 886–896, San Diego, California. Association for Computational Linguistics.

Verga, P., Belanger, D., Strubell, E., Roth, B., and McCallum, A. (2016c). Multilingual relation extraction using compositional universal schema. In *Proceedings of NAACL-HLT*, pages 886–896.

Verga, P. and McCallum, A. (2016). Row-less universal schema. In *Proceedings of the 5th Workshop on Automated Knowledge Base Construction*, pages 63–68, San Diego, CA. Association for Computational Linguistics.

Verga, P., Neelakantan, A., and McCallum, A. (2017a). Generalizing to unseen entities and entity pairs with row-less universal schema. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 613–622, Valencia, Spain. Association for Computational Linguistics.

Verga, P., Neelakantan, A., and McCallum, A. (2017b). Generalizing to unseen entities and entity pairs with row-less universal schema. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, volume 1, pages 613–622.

Verga, P., Strubell, E., and McCallum, A. (2018). Simultaneously Self-attending to All Mentions for Full-Abstract Biological Relation Extraction. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT)*, New Orleans, Louisiana.

Vilnis, L. and McCallum, A. (2015). Word representations via gaussian embedding. *ICLR*.

Vinyals, O., Kaiser, L., Koo, T., Petrov, S., Sutskever, I., and Hinton, G. (2014). Grammar as a foreign language. In *CoRR*.

Wang, L., Cao, Z., de Melo, G., and Liu, Z. (2016). Relation classification via multi-level attention cnns. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1298–1307.

Wang, Z., Zhang, J., Feng, J., and Chen, Z. (2014). Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 1112–1119, Quebec City, QC, Canada.

Wei, C.-H., Kao, H.-Y., and Lu, Z. (2013a). Pubtator: a web-based text mining tool for assisting biocuration. *Nucleic Acids Research*, 41.

Wei, C.-H., Kao, H.-Y., and Lu, Z. (2013b). Pubtator: a web-based text mining tool for assisting biocuration. *Nucleic acids research*, 41(W1):W518–W522.

Wei, C.-H., Kao, H.-Y., and Lu, Z. (2015a). Gnormplus: an integrative approach for tagging genes, gene families, and protein domains. *BioMed research international*, 2015.

Wei, C.-H., Peng, Y., Leaman, R., Davis, A. P., Mattingly, C. J., Li, J., Wiegers, T. C., and Lu, Z. (2015b). Overview of the biocreative v chemical disease relation (cdr) task. In *Proceedings of the fifth BioCreative challenge evaluation workshop*, pages 154–166.

Wei, C.-H., Peng, Y., Leaman, R., Davis, A. P., Mattingly, C. J., Li, J., Wiegers, T. C., and Lu, Z. (2016). Assessing the state of the art in biomedical relation extraction: overview of the biocreative v chemical-disease relation (cdr) task. *Database*, 2016.

Weissenborn, D. (2016). Embedding entity pairs through observed relations for knowledge base completion. Unpublished manuscript, OpenReview.

Welbl, J., Stenetorp, P., and Riedel, S. (2017). Constructing datasets for multi-hop reading comprehension across documents. *arXiv preprint arXiv:1710.06481*.

Welbl, J., Stenetorp, P., and Riedel, S. (2018). Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association of Computational Linguistics*, 6:287–302.

Weston, J., Weiss, R. J., and Yee, H. (2013). Nonlinear latent factorization by embedding multiple user interests. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 65–68, Hong Kong, China. ACM.

Wu, C., Tygert, M., and LeCun, Y. (2017). Hierarchical loss for classification. *CoRR*, abs/1709.01062.

Xie, R., Liu, Z., Jia, J., Luan, H., and Sun, M. (2016). Representation learning of knowledge graphs with entity descriptions. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Yaghoobzadeh, Y., Adel, H., and Schütze, H. (2017a). Corpus-level fine-grained entity typing. *arXiv preprint arXiv:1708.02275*.

Yaghoobzadeh, Y., Adel, H., and Schütze, H. (2017b). Noise mitigation for neural entity typing and relation extraction. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1183–1194, Valencia, Spain. Association for Computational Linguistics.

Yang, B., Yih, W., He, X., Gao, J., and Deng, L. (2015a). Embedding entities and relations for learning and inference in knowledge bases. In *3rd International Conference for Learning Representations (ICLR)*, San Diego, California, USA.

Yang, B., Yih, W., He, X., Gao, J., and Deng, L. (2015b). Embedding entities and relations for learning and inference in knowledge bases. In *3rd International Conference for Learning Representations (ICLR)*, San Diego, California, USA.

Yang, B., Yih, W.-t., He, X., Gao, J., and Deng, L. (2014). Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*.

Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W. W., Salakhutdinov, R., and Manning, C. D. (2018). Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.

Yao, L., Riedel, S., and McCallum, A. (2010). Collective cross-document relation extraction without labelled data. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1013–1023. Association for Computational Linguistics.

Yao, L., Riedel, S., and McCallum, A. (2013). Universal schema for entity type prediction. In *Proceedings of the 2013 workshop on Automated knowledge base construction*, pages 79–84. ACM.

Yates, A. and Etzioni, O. (2007). Unsupervised resolution of objects and relations on the web. In *North American Chapter of the Association for Computational Linguistics*.

Yosef, M. A., Bauer, S., Hoffart, J., Spaniol, M., and Weikum, G. (2012). Hyena: Hierarchical type classification for entity names. In *Proceedings of the International Conference on Computational Linguistics (COLING)*.

Zeng, D., Liu, K., Chen, Y., and Zhao, J. (2015a). Distant supervision for relation extraction via piecewise convolutional neural networks. *EMNLP*.

Zeng, D., Liu, K., Chen, Y., and Zhao, J. (2015b). Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1753–1762, Lisbon, Portugal. Association for Computational Linguistics.

Zhou, H., Deng, H., Chen, L., Yang, Y., Jia, C., and Huang, D. (2016a). Exploiting syntactic and semantics information for chemical–disease relation extraction. *Database*, 2016.

Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., and Xu, B. (2016b). Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 207–212, Berlin, Germany. Association for Computational Linguistics.