Literally the Littest Lit Review: Learning lexical language relations from laughably large lists of sentences

Cole Winstanley
colew@stanford.edu

Ethan Shen
ezshen@stanford.edu

Eli Pugh
epugh@stanford.edu

1 Motivation and Task Definition

As information services became more and more integral to the online ecosystem in the early 2000s, it became apparent that representing facts and information at scale was a pressing problem for a variety of applications. Much of the online information was encoded as natural language, in everything from Wikipedia pages to newspaper articles. Extracting this information in a systematic way was critically important to services like Google, who needed a way to identify and organize facts and information for quick retrieval, and also for others such as firms engaged in electronic securities trading based on real-time information found on the internet. Despite massive amounts of unstructured text data on the internet today, accessible structured information is limited in most domains. Even state-of-the-art knowledge bases are far from comprehensive and lack detailed domain-specific knowledge.

In this paper we review the current literature surrounding relation extraction, a form of extracting structured information from text where we are looking for relation triples. A relation triple consists of two entities and a directional relation between them. For example, given the sentence "The event was held Indiana's capital, Indianapolis." we can extract a triple (capitol_of, Indianapolis, Indiana).

Relation extraction can be performed in unsupervised, supervised, or distant-supervised settings, depending on the available data and desired use of the resulting knowledge base.

2 Supervision methods

2.1 Automatically modeling relation patterns

One important step towards creating a systematic way to store such information was the creation of a semantic taxonomy, for example WordNet [6], which stores a vast trove of data on the definitions of and relationships between words and other lexical entities. The database could be used as a tool to decode meaning of natural language. However, as a database WordNet is expensive and time-consuming to build and maintain. It would be more useful if there was an analogue to WordNet that could be built automatically, by learning directly from natural language. Even though some early work had used small sets of regular expressions for recognizing language that could indicate semantic relationships, this approach could not hope to recognize even a nonnegligible portion of the ways that natural language could express semantic relationships.

Recognizing the challenge of quickly building vast information repositories from natural language, Snow et al. in 2005 presented an article entitled, "Learning syntactic patterns for automatic hypernym discovery" [24] which tackled the problem of automatically extending semantic taxonomies from a core set of pre-made or hand-labeled data. Snow et al. take the approach of extracting "dependency paths" from sentence parse trees to automatically learn patterns that signify hypernym pairs ("is-a" relationships). Their algorithm uses statistical methods to automatically identify syntactic forms that indicate hypernymy relations using a set of pre-defined relations in WordNet. Then,

they use syntactic information from a corpus with these terms to build more generalized models of syntax that indicate hypernymy. This becomes a "hypernym classifier" that can take any parse tree of a sentence with novel terms and structure and decide whether a pair of given terms are in the hypernym relation. This allows them to turn to a new text, and potentially one in a different domain from the original training data, and search for entity pairs that are likely to be hypernyms.

Snow et al. evaluate their approach using a hand-labeled dataset of hypernym relations between terms in news articles. They train their model on a corpus of six million news articles, using known hypernym and non-hypernym pairs from WordNet as the pre-defined relations. They evaluate their model both on its ability to reproduce the hypernym relations defined in WordNet in novel sentences, and also its ability to classify randomly-selected noun pairs from the corpus as hypernyms or non-hypernyms.

The Snow et al. model significantly outperformed WordNet in terms of both precision and recall. In particular, their regression model based on learning hypernym patterns from WordNet achieved a 16% improvement in F-score over models based on WordNet alone. A model based on learning hypernyms from the Wikipedia corpus as well achieved a 54% improvement in F-score. This demonstrated, for the first time, that automatic methods of relation extraction were a viable alternative to manually-generated databases such as WordNet, suggesting that performing natural language relation extraction at scale, even in specialized domains, was possible.

2.2 Discovering relation patterns without direct supervision

The next milestone in automatic relation extraction from natural language text was demonstrated by Mintz et al. in a 2009 article entitled "Distant supervision for relation extraction without labeled data" [14]. This work gives a new algorithm that does not require a the hand-labeled corpus of relations required by Snow et al. Instead, they use the vast trove of relations provided by Freebase [4], an openly-available knowledge graph containing millions of word relation pairs. Additionally, they take the assumption that when two words have relation X and appear together in a sentence, that sentence is likely to expression relation X. They admit that this results in noisy data for each relation, but this is compensated for by the vast number of "examples" they can form, and for the fact that they do not need a new labeled corpus for each relation they want to apply the algorithm to.

After creating the examples for a given relation, Mintz et al. use a similar algorithm to Snow et al. to generate pattern features from the examples and ultimately a classifier for each of the relations. The authors term this pipeline "distant supervision" because it "combines the advantages of supervised IE (combining 400,000 noisy pattern features in a probabilistic classifier) and unsupervised IE (extracting large numbers of relations from large corpora of any domain)."

Mintz et al. were able to achieve high precision for a number of the relations in the Freebase database. They present a number of conclusions, as well as analysis of relations that their algorithm does not perform well on. In particular, they highlight some benefits and limitations of their syntactic pattern paradigm, where they use rigid syntactic patterns for recognizing relations in sentences. This foreshadows the success of more flexible parsing and embedding strategies, as introduced by RNNs and transformer networks discussed below.

The most significant achievement of Mintz et al. is that they were able to use a much larger core corpus to start the training process than previous approaches, due to their assumption that noisy pattern examples would work just as well hand-picked ones. The result is that their work can be extended quickly and easily to domains where there are not huge datasets of labeled relations, such as the biology dataset we aim to tackle. We discuss further below how these results are relevant, but we plan to use some of the same principles that Mintz et al. lean on in their work to perform ours.

2.3 Reducing noise in the Mintz et al. model

After the Mintz et al. article, there were a few methods for relation extraction that built off of their approach, mostly introducing incremental improvements to accuracy while preserving the flexibility of the original approach. One such improvement was introduced by Riedel et al. in their article "Modeling Relations and Their Mentions without Labeled Text" [19]. They specifically tackle

problems that occur when the knowledgebase (the role of Freebase in the Mintz et al. article) is not directly related to the domain of the corpus being trained on.

To attempt to handle situations where relations between entities in the knowledgebase are not expressed in sentences involving both of the entities, Riedel et al. introduce a factor graph approach that, instead of assuming that every sentence with the pair expresses the relation, that at least one sentence with the pair expresses the relation. The authors train this graphical model using typical constraint satisfaction techniques (in particular SampleRank [27]), predicting for each sentence in the corpus whether a given relation is expressed.

Intuitively, the assumption that at least one (or a fraction of, a constraint the authors also experimented with) sentence with a given pair expresses the relation is a more plausible assumption than assuming that all the given sentences are examples of the relation. This also works empirically: the authors achieve a 31% reduction in error through their improvement. This corresponds to impressive 91% precision for the New York Times dataset [21].

The Riedel et al. approach also has drawbacks, in particular with efficiency. Factor graphs are combinatorially difficult to solve, and even with a framework like SampleRank they present difficulty with scale. It would be a logical improvement to, instead of using a constraint satisfaction model like a factor graph, to use a statistical model, making the assumption that about a certain fraction of the sentences with a given pair express a known relation. This fraction could be learned along with the patterns for the relations, yielding an even more flexible algorithm for performing distant supervision for relation extraction.

3 Training Methods

3.1 Word Embeddings

Word embeddings are an important part of almost all NLP models, and there are many different successful embedding techniques. Word embeddings represent words as distributed vectors in order to capture semantic information about the word. Popular embedding methods in relation extraction include GloVe [16], Word2Vec [13], and fastText [3]. While these embeddings each provide advantages in different situations or contexts, their use in relation extraction doesn't differ much from their use in other NLP tasks. Because of this, we won't spend much time discussing their relative advantages.

In Section 4 we talk about modern state-of-the-art embedding methods including BERT contextual embeddings and featurization using OpenAI GPT. The implications of contextual embeddings in a relation extraction task is more interesting, and is covered in more detail in section 4.

3.2 Further Featurization

In many relations, the position of the two entities is relatively important and can help indicate whether words are related. This makes position a very useful tool in relation extraction, and most successful models use positional embeddings. First proposed by Zeng et al. in 2014, positional embeddings help a model keep track of how close words are to each other. [28] Other powerful featurization techniques include convolution over words and compositional vector representations created using parse trees. [25]

Zeng et al. discuss feature extraction for the relation extraction task. Using novel feature extraction techniques, they were able to significantly outperform state-of-the-art results with a relatively simple neural architecture. They begin by looking up pre-trained word embeddings for each word without much preprocessing. Lexical features are then computed using a few different techniques, including compositional vector representations proposed by Socherer et al. [25] Along with word and lexical features, sentence features are also utilized. Sentence features include part-of-speech labelling to disambiguate words, positional features, output of a convolutional layer, and a nonlinear transform of the sentence embeddings. All of the word, lexical, and sentence representations are then concatenated and passed through a softmax classifier. [28] The most interesting and impactful

featurization techniques Zeng et al. uses are the compositional vector representations by Socherer et al., positional embedding, and convolutional sentence structure.

Compositional vector representations are trained using an RNN, and produce vector and matrix representations for each node (word) in a parse tree. The vector is a simple embedding of the word, whereas the matrix is a linear transformation showing how the word modifies its neighbors. This is a very powerful representation for relation extraction because the matrix representations help show each word's relation to words nearby.

Positional embeddings encode the relation of each word to key entities in a sentence. The way Zeng et al. performs this is by predicting key entities and composing a vector of offsets from each important entity. [28] CNNs benefit from embedded position since they aren't an autoregressive architecture and don't have inherent ordering structure. Many modern methods in NLP such as transformers also take advantage of positional embeddings, which has sparked research efforts into the most effective positional encoding techniques. One popularly used method is adding sinusoidal curves with different periods to each embedding. For example, for the i-th word of an example we might append $\sin(.25i)$, $\sin(0.5i)$, and $\sin(i)$ to its embedding. [26]

After encoding position into their embeddings, Zheng et al. creates sentence-level embeddings using a convolution over words. These convolutions have a similar effect to taking n-grams of words, and then taking a weighted sum of those words' embeddings. This captures some structure in the sentence past individual word relations.

It's worth noting that these featurization techniques are not at odds with one another, but each provide valuable semantic or interrelation information that empirically improves most models. Not only are these techniques all useful and popular, but there are several other useful and popular methods that we don't have space to mention. The most successful approaches use many of these featurization techniques in concert, and choosing features could be a paper by itself.

3.3 Model Architectures

There are many common variations of model architectures used for relation extraction. In Convolution Neural Network for Relation Extraction, Liu et al. (2013) [12] discuss previous popular architectures as well as propose a novel use of a deeper CNN to perform relation extraction. Before discussing their approaches, we can use their discussion of related work to summarize the evolution of popular model architectures in relation extraction.

Early relation extraction methods heavily relied on regular expressions. Because of the high variability of sentence structures, these models had very low recall. Regular expressions were replaced for deep learning approaches on very simple featurization structures, such as simple word embedding lookups. These methods were soon replaced by similar classifiers that also used features such as part-of-speech, relation, and parse trees as discussed in section 3.2. These are all very important methods, but creating parse trees and using more complex structures like compositional vectors takes time and complexity. [12]

In order to speed up and improve upon current methods, Liu et al. proposes a deep learning approach to automatically learn features. They first reduce the sentence to a maximum length and pad, since CNNs don't handle variable-length inputs when feeding to a fixed-size classifier. After fixing the length of each sentence, they feed it into an embedding lookup table and encode a few other very simple lexical features. They then feed this to a convolutional layer, which simulates again the weighted combinations of words within a sliding window. This allows the model to gain information about how words are related. They then max pool over the convolution output and feed this into a fully-connected network structure that outputs to a softmax function. This gives probabilities of each relation as desired. [12]

This new architecture proposed by Liu et al. was utilized and improved upon in many later papers, such as Zeng et al. (2014) [28]. In addition, Nguyen and Grishman (2015) [15] used this structure with many different convolutional kernel sizes to gain interrelation between words at different distances.

More recently, attention mechanisms have yielded great improvements when using CNNs, especially with distant supervision as discussed in section 2. Since distant supervision relation extraction introduces noise in the form of false positive labels, sentence-level attention can shift focus away from these examples. Lin et al. (2016) first showed this to be effective in "Neural Relation Extraction with Selective Attention over Instances". [11] They first use a CNN sentence encoder with max-pooling and a non-linearity, as is commonly done in previous work. This creates a distributed representation of the sentence that is then passed to an attention mechanism that helps select the sentences which express a fixed relation. This then allows attention to recognize where distant supervision has wrongly labelled a sentence as one that expresses a relation by placing a much lower priority there. This is more effective than the traditional method of counting each sentence example equally since it often overlooks false positives. [11]

While there are many different shifts in methodology over time, the relation extraction task seems to be fairly dominated by CNN methods and composition of many different clever features. The more recent introduction of attention mechanisms has shown success as well, and reduces noise introduced from distant supervision techniques, allowing for the utilization of more data. While these techniques are different, they are again not at odds with one another, but instead improve on one another and show progress in understanding across the field.

4 Utilizing Modern NLP Tools

4.1 Unsupervised Pre-Training: BERT and GPT

The introduction of new language representation models has led to a paradigm shift in NLP. Here, we provide a brief overview of these models and their potential application to our projects.

In the Generative Pre-Training (GPT) model [18], Radford et. al. introduce language model pre-training, which uses minimal task-specific parameters in an unsupervised deep language modeling task to generate learned embeddings of context.

Bidirectional Encoder Representations from Transformers (BERT) [5] furthers this work by training a bidirectional transformer network. The key insight lies in its training procedure, which can be outlined in the following.

They first pre-training by random selection of 15% of the training corpus. They then replace 80% of these selected words with a masked token, 10% with a random word, and leave the other 10% as is. They then train the model to correct these, therefore learning about words from their surrounding context on each side. The second pre-training step is next sentence prediction. Given input sentences, BERT is trained to predict whether input 2 immediately follows input 1 in context. This allows BERT to learn about sentence relations, which is useful for extending to other NLP tasks.

Both GPT and BERT are task-agnostic models that can be fine-tuned to surpass a wide range of benchmarks for NLU while requiring minimal changes to the model architecture.

4.2 Improving Relation Extraction by Pre-trained Language Representations

In recent years, state-of-the-art performance has been achieved by careful tuning of neural models which incorporate lexical and syntactic features, such as part-of-speech tags and dependency trees. Given the effectiveness of pre-training BERT encodings for various other NLP tasks, Alt et. al. proposes a natural question: can we leverage these pretrained models to further push the state of the art in relation extraction and semantic role labeling, without relying on lexical or syntactic features? Using this strategy, they introduce TRE, a Transformer for Relation Extraction, and achieve state-of-the-art results using a simple neural network layer on top of OpenAI GPT encodings.

In this paper, the authors do the following: (1) describe TRE, a Transformer based relation extraction model that, unlike previous methods, relies on deep language representations instead of explicit

linguistic features, (2) show experiments outperforming state-of-the-art methods on the TACRED dataset, (3) report ablation studies showing that pre-trained language representations prevent overfitting and achieve better generalization in the presence of complex entity mentions. They also make the code open-source, which we will adapt for our project.

The TRE architecture is split into two modules – unsupervised pre-training of language representations and supervised fine-tuning on relation extraction. The pretraining step is described in [18], and reviewed above. Importantly, given a corpus $C = \{c_1, ..., c_n\}$ of tokens c_i, the language modeling objective maximizes the likelihood

$$L_1(C) = \sum_i \log P(c_i | c_{i-1}, ..., c_{i-k}; \theta)$$

and uses a weighted softmax function to predict the next word given a sequence of words. In the supervised module, the pre-trained embeddings from GPT is fed through another softmax weighted linear model to get the relation prediction to optimize the following objective:

$$L_2(D) = \sum_i^{|D|} \log P(r_i | x_i^1, ..., x_i^m)$$

where $D = ([x_i^1, ...x_i^m], a_i^1, a_i^2, r_i)$ is the labeled relation extraction dataset. The paper mentions that introducing language modeling as an auxiliary objective during fine-tuning improves generalization and leads to faster convergence, resulting in the following overall objective:

$$L(D) = \lambda * L_1(D) + L_2(D)$$

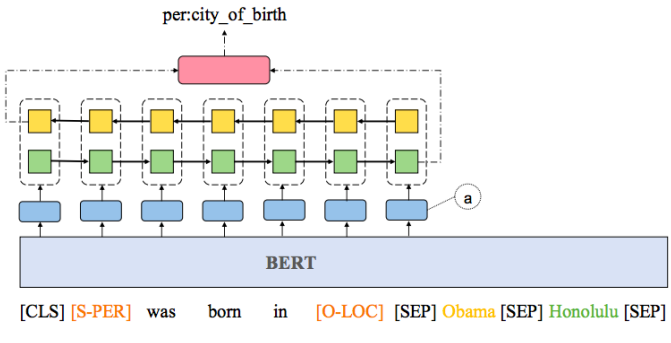
where λ is a scalar language model weight.

Beyond showing SOTA performance on TACRED and other datasets, the authors also show analysis and ablation experiments to demonstrate the contributions of different parts of their models to success. Comparing a model with pre-trained representations to a model with randomly initialized representations, they show that the TACRED validation F1 score decreases from 63.3 to 20. Furthermore, they show that entity masking has an important regularizing affect on model generalization. Briefly, entity masking limits information about entities in order to prevent overfitting. These strategies include replacing entity mentions with UNK tokens, named entity types (NE strategy), and/or grammatical roles (GR). Experimentally, they see that no entity masking gives the highest precision, though combining NE + GR masking strategies gives the highest Recall and F1 scores.

Overall, Alt. et. al. was the first work to apply language model pre-training to relation extraction, and demonstrate the effectiveness of this strategy.

4.3 Simple BERT Models for Relation Extraction

Given the results above, it is a natural extension to explore the results of better context-dependent embeddings with BERT. Here, Shi et. al. apply the trained BERT encodings to relation extraction and Semantic Role Labeling experiments.
Here, we focus on their relationship extraction experiments and describe their model. They first constructed the input sequence [[CLS] sentence [SEP] subject [SEP] object [SEP]]. After tokenization,



4.3 Simple BERT relation extraction model [23]

they fed this into the BERT encoder to get contextual representations, and created positional representations by encoding the distance of each word to subjects and objects. They then fed the

concatenated contextual and positional representations into a single-layer BiLSTM, and output predictions using a single-layer MLP (Figure 4.3).

On the TAC Relation Extraction Dataset (TACRED) [29], the Shi model achieves single-model SOTA with an F1 score of 67.8, beating Alt. et al's [1] F1 of 67.4. Overall, the simplicity of the BERT model provides a strong baseline for future studies in relation extraction, though this paper is still a work in progress.

5 Conclusion

In this review we have covered three major aspects of the relation extraction task, namely training paradigms, models, and modern NLP tools. Within each section we compared and contrasted methods from multiple papers and how they built upon each other. Here we will observe the general trend of relation extraction approaches over the last decade.

Since 2009, distant supervision has began to dominate in a field with insufficient labelled data, and many methods have been developed to reduce noise this introduces. Since 2011, many systems have focused on developing lexical features and representing sentences with a convolution over the words. With the arrival of pre-trained contextual embeddings in 2017, relation extraction has began to shift from more complex lexical features to deep unsupervised ones like BERT or GPT.

6 Future Work

While relation extraction has evolved very rapidly over the past decade, there is still significant room for improvement in domain-specific text. Several challenges arise, such as embedding lookup misses from domain-specific terms and relations that are more nuanced than in general text.

A specific direction of research that hasn't been thoroughly explored is the use of contextual embeddings that are trained on large corpora of domain-specific knowledge. In addition, domain-specific text is less likely to have relation labelling specific to this task, and thus distant supervision is certainly necessary. This challenge means that new advances in attention mechanisms are likely to yield great results on domain-specific corpora.

One example of such a task is an ongoing effort to build a knowledge base for the Intelligent LIFE biology textbook. [20] This research, led by Vinay Chaudhri, is working to overcome limited labelled relations and a base of entities that aren't found in most embedding lookup schemes. One possible approach with this would be to combine SCIBERT embeddings as well as more traditional relation extraction features in order to improve on this task. Since distant supervision seems to be a sensible approach to this task, a selective attention layer over distributed sentence representations should show significant improvements. Though this is one specific task, if successful, these methods can be applied to many types of situations where domain-specific text is used to build a knowledge base.

References

Note: references analyzed in-depth are marked with a *

[1] * Christoph Alt, Marc Hübner, and Leonhard Hennig. Improving relation extraction by pre-trained language representations. 2018.

[2] * Iz Beltagy, Arman Cohan, and Kyle Lo. Scibert: Pretrained contextualized embeddings for scientific text. *CoRR*, abs/1903.10676, 2019.

[3] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *CoRR*, abs/1607.04606, 2016.

[4] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, pages 1247–1250, New York, NY, USA, 2008. ACM.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[6] Christiane Fellbaum. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.

[7] Daniel Jurafsky and James Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, volume 2. 02 2008.

[8] Taku Kudo. Subword regularization: Improving neural network translation models with multiple subword candidates. *CoRR*, abs/1804.10959, 2018.

[9] Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *CoRR*, abs/1808.06226, 2018.

[10] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *CoRR*, abs/1901.08746, 2019.

[11] * Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2124–2133, Berlin, Germany, August 2016. Association for Computational Linguistics.

[12] * ChunYang Liu, WenBo Sun, WenHan Chao, and Wanxiang Che. Convolution neral network for relation extraction. *International Conference on Advanced Data Mining and Applications*, Springer, pages 231-242., 2013.

[13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013.

[14] * Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 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, August 2009. Association for Computational Linguistics.

[15] Thien Huu Nguyen and Ralph Grishman. Relation extraction: Perspective from convolutional neural networks. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 39–48, Denver, Colorado, June 2015. Association for Computational Linguistics.

[16] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[17] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *CoRR*, abs/1802.05365, 2018.

[18] * Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding with unsupervised learning. *Technical report, OpenAI.*, 2018.

[19] * Sebastian Riedel, Limin Yao, and Andrew McCallum. Modeling relations and their mentions without labeled text. In *ECML/PKDD*, 2010.

[20] David E. Sadava, David M. Hillis, H Craig Heller, and Sally D. Hacker. *Life: The Science of Biology*, volume 11. 2017.

[21] E. Sandhaus. The new york times annotated corpus. *Linguistic Data Consortium*, 2008.

[22] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *CoRR*, abs/1508.07909, 2015.

[23] * Peng Shi and Jimmy Lin. Simple BERT models for relation extraction and semantic role labeling. *CoRR*, abs/1904.05255, 2019.

[24] Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. Learning syntactic patterns for automatic hypernym discovery. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 1297–1304. MIT Press, 2005.

[25] Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1201–1211, Jeju Island, Korea, July 2012. Association for Computational Linguistics.

[26] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[27] Michael Wick, Khashayar Rohanimanesh, Kedar Bellare, Aron Culotta, and Andrew McCallum. Samplerank: Training factor graphs with atomic gradients. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pages 777–784, USA, 2011. Omnipress.

[28] * Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jian Zhao. Relation classification via convolutional deep neural network. In *COLING*, 2014.

[29] Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, pages 35–45, 2017.