Deep Neural Approaches to Relation Triplets Extraction: A Comprehensive Survey

Tapas Nayak[†], Navonil Majumder[◇], Pawan Goyal[†], Soujanya Poria[◇]

[†] IIT Kharagpur, India
[◇] Singapore University of Technology and Design, Singapore
tnk02.05@gmail.com,
{navonil_majumder,sporia}@sutd.edu.sg,
pawang@cse.iitkgp.ac.in

Abstract

Recently, with the advances made in continuous representation of words (word embeddings) and deep neural architectures, many research works are published in the area of relation extraction and it is very difficult to keep track of so many papers. To help future research, we present a comprehensive review of the recently published research works in relation extraction. We mostly focus on relation extraction using deep neural networks which have achieved state-of-the-art performance on publicly available datasets. In this survey, we cover sentence-level relation extraction to document-level relation extraction, pipeline-based approaches to joint extraction approaches, annotated datasets to distantly supervised datasets along with few very recent research directions such as zero-shot or few-shot relation extraction, noise mitigation in distantly supervised datasets. Regarding neural architectures, we cover convolutional models, recurrent network models, attention network models, and graph convolutional models in this survey.

1 Introduction

A relation triplet consists of two entities and a relation between them. We can find such triplets in a structured format in several publicly available knowledge bases (KBs) such as, Freebase (Bollacker et al., 2008), DBpedia (Bizer et al., 2009), Wikidata (Vrandečić and Krötzsch, 2014), etc. These triplets are very useful for many natural language processing tasks such as machine reading comprehension (Qiu et al., 2019), machine translation (Zhao et al., 2020), abstractive summarization (huang et al., 2020), etc. However, building such knowledge bases is a daunting task. The aforementioned KBs are built by crowdsourcing, which may not be scalable. Although these KBs

contain a large number of triplets, they remain incomplete. On the other hand, relation triplets can be automatically distilled from the copious amount of free text on the Web. This can be leveraged for identifying missing links in the existing KBs or build a KB from scratch without human intervention.

There are two distinct research paradigms of relation extraction: open information extraction (Open IE) and supervised relation extraction. Banko et al. (2007), Christensen et al. (2011), Etzioni et al. (2011), and Mausam et al. (2012) use open information extraction (Open IE) to extract relation triplets from sentences where relations set is open. Open IE systems like KnowItAll (Etzioni et al., 2004), TEXTRUNNER (Yates et al., 2007), REVERB (Etzioni et al., 2011), SRLIE (Christensen et al., 2011), and OLLIE (Mausam et al., 2012) use rule-based methods to extract entities from the noun phrases and relations from the verb phrases present in sentences. These systems can extract a large number of triplets of diverse relations from text within a reasonable time frame. These models extract any verb phrase in the sentences as a relation thus yielding too many uninformative triplets. Also, a relation can be expressed in sentences with different surface forms (lives_in relation can be expressed with 'lives in', 'stays', 'settles', 'lodges', 'resident of', etc) and Open IE treats them as different relations which leads to duplication of triplets.

The problems of the Open IE can be addressed using supervised relation extraction. In supervised relation extraction, we consider a fixed set of relations, thus there is no need to do any normalization of the extracted relations. This approach requires a large parallel corpus of text and relation triplets for training. There are some annotated and some distantly supervised parallel cor-

Class	Sentence	Triplets
NEO	The original Joy of Cooking was published in 1931 by Irma Rombauer, a St. Louis housewife.	<Irma Rombauer, St. Louis, place_lived>
EPO	Berlin is the capital of Germany.	<Germany, Berlin, capital> <Germany, Berlin, contains> <Berlin, Germany, country>
SEO	Dr. C. V. Raman who was born in Chennai worked mostly in Kolkata.	<Dr. C. V. Raman, Chennai, birth_place> <Dr. C. V. Raman, Kolkata, place_lived>

Table 1: Examples of different classes of overlapping relation triplets. This table is taken from Nayak (2020).

pus of (text, triplets) available publicly that can be used for training the models. Creating annotated corpus is difficult and time-consuming, so datasets created in this way are relatively smaller in size. On the other hand, the distant supervision approach can be exploited to create a large training corpus automatically, but these datasets contain a significant amount of noisy labels. These noisy labels in the distantly supervised datasets can affect the performance of the models in a negative way. Several feature-based models and deep neural network-based are proposed in the last decade for relation extraction. In this survey, we discuss these datasets and models in detail in the remaining part of the paper.

Previously, Cui et al. (2017); Pawar et al. (2017); Kumar (2017); Shi et al. (2019); Han et al. (2020) presented survey of the research works in the relation extraction, but they mostly focused on pipeline-based relation extraction approaches at the sentence-level. Different from these survey papers, we extend the survey to document-level relation extraction and joint entity and relation extraction approaches. We also survey very recent research directions in this area such as zero-shot or few-shot relation extraction and noise mitigation in distantly supervised datasets. To the best of our knowledge, this is the first survey paper that covers so many different aspects of relation extraction in detail.

2 Task Description

Given a sentence and a set of relations R as input, the task is to extract a set of relation triplets, with relations from R, from the sentence. Pipeline-based relation extraction approaches divide the task into two sub-tasks: (i) entity recognition and (ii) relation classification. In the first sub-task, all the candidate entities are identified in a sentence. In the second sub-task, the relation between every possible ordered pair of candidate entities is determined — this relation may not exist (*None*).

Joint-extraction approaches, in contrast, jointly find the entities and relations. Joint models extract only the valid relational triplets and they do not need to extract the *None* triplets. Relation triplets may share one or both entities among them and this overlapping of entities makes this task challenging. Based on the overlap of entities, we divide the sentences into three classes: (i) *No Entity Overlap (NEO)*: A sentence in this class has one or more triplets, but they do not share any entities. (ii) *Entity Pair Overlap (EPO)*: A sentence in this class has more than one triplet, and at least two triplets share both the entities in the same or reverse order. (iii) *Single Entity Overlap (SEO)*: A sentence in this class has more than one triplet and at least two triplets share exactly one entity. It should be noted that a sentence can belong to both EPO and SEO classes. The goal is to extract all relation triplets present in a sentence.

3 Scope of this Survey

In this survey, we focus on the relation triplets concerning PERSON, ORGANIZATION, and LOCATION mainly. Many research works are published for domain-specific relation extraction such scientific articles (Luan et al., 2017; Jain et al., 2020), medical (Gu et al., 2016; Li et al., 2017; Choi, 2018; Thillaisundaram and Togia, 2019), legal (Andrew, 2018), finance (Vela and Declerck, 2009), etc. But in this survey, we do not include the research papers that only focus on a particular domain. Also, we only focus on relation extraction for the English language.

4 Challenges of Dataset Annotation

Existing KBs, such as Freebase, Wikidata, and DBpedia, are manually built which takes much effort and time. However, these KBs still have a large number of missing links. On the other hand, we can find evidence of a large number of relation triplets in free texts. We have included some examples of such triplets and texts in Table 2. If

Relation	Entity 1	Entity 2	Text
acted_in	Meera Jasmine	Sootradharan	Meera Jasmine made her debut in the Malayalam film "Soothradharan" .
located_in	Chakkarakadavu	Kerala	Chakkarakadavu is a small village to the east of the town of Cherai, on Vypin Island in Ernakulam district, Kerala, India .
birth_place	Barack Obama	Hawaii	Barack Obama was born in Hawaii .
plays_for	Moussa Sylla	Horoya AC	Fodé Moussa Sylla is a Guinean football player, who currently plays for Horoya AC .
owns	MTV Channel	Shakthi TV	MTV Channel (Pvt) Ltd is a Sri Lankan media company which owns three national television channels - Shakthi TV, Sirasa TV and TV 1 .

Table 2: Examples of relation triplets found in free texts. This table is taken from Nayak (2020).

we can extract relation triplets automatically from the text, we can build a KB from scratch or add new triplets to the existing KBs without any manual effort. But to achieve this goal, we need a large number of texts annotated with relation triplets, and creating such a corpus manually is a daunting task. One possible way to do the annotation is to identify the entities in the text and then for all possible pairs of entities, identify the relations from a pre-defined set of relations or *None* if none of the relations from this set holds in this text. The identification of the entities in a text is relatively easier, but the difficulty of identifying the relations from a set grows with the size of the relations set. For few relations such as 3/4/5, this may be easier, but when the number of relations grows to 20/30/40, it becomes very challenging. Marking the *None* relations in the case of large relations set is more difficult as the annotators have to make sure that none of the relations from the set holds between two entities in the text.

To overcome the dataset annotation problems, Mintz et al. (2009); Riedel et al. (2010); Hoffmann et al. (2011) proposed the idea of distant supervision to automatically obtain the text-triplet mapping without any human effort. In distant supervision, the triplets from an existing KB are mapped to a free text corpus such as Wikipedia articles or news articles (e.g., New York Times). The idea of distant supervision is that if a sentence contains two entities of a triplet from a KB, that sentence can be considered as the source of this KB triplet. On the other hand, if a sentence contains two entities from a KB and there is no relation between

these two entities in the KB, that sentence is considered as a source of *None* triplet between the two entities. These *None* samples are useful as distantly supervised models consider only a limited set of positive relations. Any relation outside this set is considered as *None* relation. This method can give us a large number of triplet-to-text mappings which can be used to build supervised models for this task. This idea of distant supervision can be extended easily to single-document or multi-document relation extraction.

But the distantly supervised data may contain many noisy samples. Sometimes, a sentence may contain the two entities of a positive triplet, but the sentence may not express any relation between them. These kinds of sentences and entity pairs are considered as noisy positive samples. Another set of noisy samples comes from the way samples for *None* relation are created. If a sentence contains two entities from the KB and there is no relation between these two entities in the KB, this sentence and entity pair is considered as a sample for *None* relation. But knowledge bases are often not complete and many valid relations between entities in a KB are missing. So it may be possible that the sentence contains information about some positive relation between the two entities, but since that relation is not present in the KB, this sentence and entity pair is incorrectly considered as a sample for *None* relation. These kinds of sentences and entity pairs are considered as noisy negative samples.

We include examples of clean and noisy samples generated using distant supervision in Table 3. The KB contains many entities out of which

Text	Entity 1	Entity 2	Distantly Supervised Relation	Actual Relation	Status
Barack Obama was born in Hawaii .	Barack Obama	Hawaii	birth_place	birth_place	Clean
Barack Obama visited Hawaii .	Barack Obama	Hawaii	birth_place	None	Noisy
Suvendu Adhikari was born at Karkuli in Purba Medinipur in West Bengal .	Karkuli	West Bengal	None	located_in	Noisy
Suvendu Adhikari, transport minister of West Bengal, visited Karkuli .	Karkuli	West Bengal	None	None	Clean

Table 3: Examples of distantly supervised clean and noisy samples. This table is taken from Nayak (2020).

four entities are *Barack Obama*, *Hawaii*, *Karkuli*, and *West Bengal*. *Barack Obama* and *Hawaii* have a *birth_place* relation between them. Karkuli and West Bengal are not connected by any relations in the KB. So we assume that there is no valid relation between these two entities. The sentence in the first sample contains the two entities *Barack Obama* and *Hawaii*, and it also contains information about *Obama* being born in *Hawaii*. So this sentence is a correct source for the triplet (*Barack Obama*, *Hawaii*, *birth_place*). So this is a clean positive sample. The sentence in the second sample contains the two entities, but it does not contain the information about *Barack Obama* being born in *Hawaii*. So it is a noisy positive sample. In the case of the third and fourth samples, according to distant supervision, they are considered as samples for *None* relation. But the sentence in the third sample contains the information for the actual relation *located_in* between *Karkuli* and *West Bengal*, even though the KB happens not to contain the *located_in* relation relating *Karkuli* and *West Bengal*. So the third sample is a noisy negative sample. The fourth sample is an example of a clean negative sample.

Despite the presence of noisy samples, relation extraction models trained on distantly supervised data have proven to be successful for relation extraction. These models can be used to fill the missing facts of a KB by automatically finding triplets from free texts. It can save much manual effort towards completing an existing KB.

5   Relation Extraction Datasets

Several datasets are available for the relation extraction task. Hendrickx et al. (2010) proposed a shared task on relation extraction in SemEval 2010 and released a dataset with 8,000 training sentences and 2,717 test instances across nine relations including *None*. The relations in this dataset are not taken from any knowledge base. They represent the relationship between two nominals in the sentences. Examples of such relations are *Cause-Effect, Component-Whole,* etc. Mintz et al. (2009) mapped Freebase (Bollacker et al., 2008) triplets to Wikipedia articles to obtain a dataset. Riedel et al. (2010) (NYT10) and Hoffmann et al. (2011) (NYT11) mapped Freebase triplets to the New York Times (NYT) articles to obtain a similar dataset. These two datasets are used extensively by researchers for their experiments. They have 52 and 24 valid relations respectively. The training and test data in NYT10 are distantly supervised, whereas in NYT11, the test data is annotated and training data is distantly supervised. Recently, Zhu et al. (2020) created an annotated test dataset for the NYT10 dataset with a subset of its relations set. This annotated test set contains 22 relations. They used a binary strategy to annotate each instance either the distantly supervised relation is present or not in the sentences. But this test dataset does not include any *None* samples which makes it unsuitable for the relation extraction task. ACE04 (Doddington et al., 2004) and ACE05 (Walker et al., 2006) are two datasets containing 7 relations. These two datasets focus on both named entity recognition and relation extraction tasks. CoNLL04 (Roth and Yih, 2004) and GDS (Jat et al., 2017) are two other datasets with 5 and 4 valid relations respectively. ACE04, ACE05, CoNLL04, and GDS datasets are manually annotated but they contain few relations in comparison to distantly supervised datasets. TACRED (Zhang et al., 2017) is another dataset for relation extraction that has manually annotated training and test data. TACRED contains 41 relations similar

Dataset Name	Level	# Valid Relations	# Train	# Test	Manual Annotation
SemEval 2010 Task 8	sentence	18	8,000	2,717	Yes
NYT10	sentence	52	455,412	172,415	No
NYT11	sentence	24	335,843	1,450	Test
NYT29	sentence	29	63,306	4,006	No
NYT24	sentence	24	56,196	5,000	No
WebNLG	sentence	216	5,519	703	Yes
ACE05	sentence	7	9,038	1,535	Yes
CoNLL04	sentence	5	1,153	288	Yes
GDS	sentence	4	13,161	5,663	Yes
TACRED	sentence	41	90,755	15,509	Yes
FewRel 2.0	sentence	100	56,000	14,000	Yes
WikiReading	document	884	14.85M	3.73M	No
DocRED	document	96	4,053	1,000	Yes

Table 4: The details of relation extraction datasets.

to that of the distantly supervised datasets. So that makes this dataset very suitable for comparing models in this task. Automatic evaluation of the models can be carried out on this dataset easily. FewRel 2.0 (Gao et al., 2019) is a few-shot relation extraction dataset. WebNLG (Zeng et al., 2018) is another dataset that contains 216 relations. Recently, this dataset has been used for joint entity and relation extraction. It is curated from the original WebNLG dataset of Gardent et al. (2017). NYT24 (Zeng et al., 2018; Nayak and Ng, 2020) and NYT29 (Takanobu et al., 2019; Nayak and Ng, 2020) are two other popular datasets for joint extraction task. These two datasets are curated from the NYT11 and NYT10 datasets respectively after removing the sentences that do not contain any valid relation triplets. These datasets are created at the sentence level.

WikiReading (Hewlett et al., 2016) and Do-cRED (Yao et al., 2019) are two document-level relation extraction datasets created using Wikipedia articles and Wikidata items. WikiReading is a slot-filling dataset where a document of an entity and the name of a property (same as the relation) is given to the models as input to predict the second entity. This dataset does not have any *None* instances. Each document in the dataset corresponds to one instance of training or testing. DocRED, on the other hand, is a relation extraction dataset. Training data contains 4,053 documents and test data contains 1,000 documents. Each document contains multiple instances and test data is blind. Nayak (2020) proposed an idea of extending the relation extraction task to multi-documents. They created a 2-hop relation extraction dataset from a multi-hop question answering dataset WikiHop (Welbl et al., 2018) that contains more relations than the previous sentence-level or document-level datasets. Their idea can be extended to create an N-hop dataset to cover more relations. The details of these datasets are included in Table 4.

6 Evaluation Metrics

In the pipeline approach, the assumption is that entities are already identified and models need to classify the relation or no relation (*None*) between the pairs of entities. There are two ways in which the performance of models can be measured: (i) At sentence-level (ii) At bag-level. In the case of the sentence-level, each sentence with an entity pair is considered as a test instance. At the bag-level, a bag of sentences where each sentence must contain the same entity pair is considered as a test instance. In both ways, models are evaluated using precision, recall, and F1 scores after removing the *None* labels. A confidence threshold is used to decide if the relation of a test instance belongs to the set of relations R or *None*. If the model predicts *None* for a test instance, then it is considered as *None* only. But if the network predicts a relation from the set R and the corresponding softmax score is below the confidence threshold, then the final prediction label is changed to *None*. This confidence threshold is the one that achieves the highest F1 score on the validation dataset. Since most of the test datasets in this task are distantly supervised and they contain noisy samples, automatic evaluation metric such as the F1 score may not be suitable. The precision-recall curve is a popular automatic metric for the evaluation of distantly supervised test datasets. The area under the precision-recall curve (AUC) indicates the performance measure of the models. Precision@K is another metric used for evaluation on such test datasets, but it requires manual effort.

For the joint extraction approaches, models are evaluated based on the number of the correct triplets extracted from the sentences. The extracted triplets are considered as a set and duplicate triplets are removed. An extracted triplet is considered correct if the corresponding entity names are correct and the relation is also correct. Precision, recall, and F1 scores are measured based on that. There are two variants of matching the entity names. The first one is partial matching (P) where only the last token of the entity names is matched. The second one is exact matching (E) where the full entity names are matched.

7 Relation Extraction Models

Relation extraction models can be categorized into two sets: (i) pipeline extraction approaches (ii) joint extraction approaches. We have included several state-of-the-art models of both the categories below.

7.1 Pipeline Extraction Approaches

At the beginning of relation extraction research, pipeline approaches were quite popular. A pipeline approach has two steps: (i) First, a named entity recognizer is used to identify the named entities in a text. (ii) Next, a classification model is used to find the relation between a pair of entities. The named entities identified in the first step are mapped to the KB entities. There are several state-of-the-art NER models available as proposed by Huang et al. (2015); Ma and Hovy (2016); Lample et al. (2016); Chiu and Nichols (2016) can be used for this purpose. Contextualized word embeddings based model such as ELMo (Peters et al., 2018), BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and SpanBERT (Joshi et al., 2019) can also be used for named entity recognition. In the next step, different classification models are proposed to find the relations between entity pairs and we describe them in detail in the following subsections.

7.1.1 Feature-Based Models

Mintz et al. (2009) proposed a feature-based relation classification model for this task. They used lexical features such as the sequence of words between two entities and their part-of-speech (POS) tags, a flag indicating which entity appears first, k tokens to the left of entity 1 and k tokens to the right of entity 2, syntactic features such as dependency path between two entities, and named

entity types of the two entities in their model. Riedel et al. (2010) proposed multi-instance learning for this task to mitigate the problem of noisy sentences obtained using the distant supervision method. They used a factor graph to explicitly model the decision of whether two entities are related and whether this relation is mentioned in a given sentence. Also, they applied constraint-driven semi-supervision to train their model without any knowledge about which sentences express the relations. Their multi-instance learning model significantly improves the performance over the model proposed by Mintz et al. (2009).

Hoffmann et al. (2011) and Surdeanu et al. (2012) proposed the idea of multi-instance multi-labels (MIML) to solve the problem of overlapping relations. They used probabilistic graphical models that take a bag of sentences containing two entities as input and find all possible relations between them. Similarly, Ren et al. (2017) used a feature-based model to jointly predict the relation between two entities and their fine-grained types. They used features like the head tokens of two entities, tokens of two entities, tokens between the two entities, their POS tags, ordering of the two entities, the distance between them, and the Brown cluster[1] of each token in their model. They proposed a joint optimization framework to learn the entity embeddings, relation embeddings, and fine-grained type embeddings of the entities together.

7.1.2 CNN-Based Neural Models

Distributed representations of words as word embeddings have transformed the way that natural language processing tasks like IE can be tackled. Word2Vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014) are two sets of large and publicly available word embeddings that are used for many NLP tasks. Most neural network-based models for information extraction have used the distributed representation of words as their core component. The high dimensional distributed representation of words can encode important semantic information about words, which is very helpful for identifying the relations among the entities present in a sentence. Initially, neural models also follow the pipeline approach to solve this task.

Zeng et al. (2014) used a convolutional neural network for relation extraction. They used the pre-trained word embeddings of Turian et al.

---

[1] https://github.com/percyliang/brown-cluster

(2010) to represent the tokens in a sentence and used two distance embedding vectors to represent the distance of each word from the two entities. They used a convolutional neural network (CNN) and max-pooling operation to extract a sentence-level feature vector. This sentence representation is passed to a feed-forward neural network with a softmax activation layer to classify the relation.
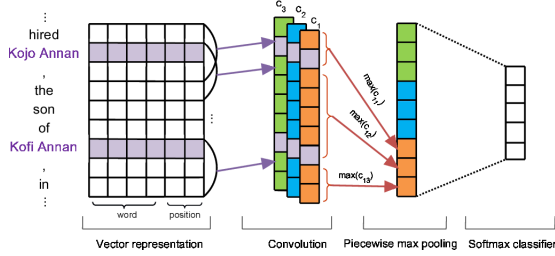

Figure 1: The architecture of the PCNN model (Zeng et al., 2015).

Zeng et al. (2015) introduced a piecewise convolutional neural network (PCNN) to improve relation extraction. Zeng et al. (2014) applied the max-pooling operation across the entire sentence to get the single important feature from the entire sentence for a particular convolutional filter. In PCNN, the max-pooling operation is not performed for the entire sentence. Instead, the sentence is divided into three segments: from the beginning to the argument appearing first in the sentence, from the argument appearing first in the sentence to the argument appearing second in the sentence, and from the argument appearing second in the sentence to the end of the sentence. Max-pooling is performed in each of these three segments and for each convolutional filter to obtain three feature values. A sentence-level feature vector is obtained by concatenating all such feature values and is given to a feed-forward neural network with a softmax activation layer to classify the relation.

7.1.3 Attention-Based Neural Models

Recently, attention networks have proven very useful for different NLP tasks. Shen and Huang (2016), Wang et al. (2016), Zhang et al. (2017), and Jat et al. (2017) used word-level attention model for single-instance sentence-level relation extraction. Shen and Huang (2016) proposed a combination of a convolutional neural network model and an attention network. First, a convolution operation with max-pooling is used to extract the global features of the sentence. Next, attention is applied to the words of the sentence based on the two entities separately. The word embedding of the last token of an entity is concatenated with the embedding of every word. This concatenated representation is passed to a feed-forward layer with tanh activation and then another feed-forward layer with softmax to get a scalar attention score for every word of that entity. The word embeddings are averaged based on the attention scores to get the attentive feature vectors. The global feature vector and two attentive feature vectors for the two entities are concatenated and passed to a feed-forward layer with softmax to determine the relation.

Wang et al. (2016) used multi-level attention CNNs for this task. Their model achived very high F1 score on the SemEval 2010 Task 8 dataset. Zhang et al. (2017) proposed a position-aware attention mechanism over the LSTM sequences for this task. Earlier Zeng et al. (2014) and Zeng et al. (2015) use the position information as dense embedding in the network for feature extraction, whereas Zhang et al. (2017) used it in attention modeling for the same task.
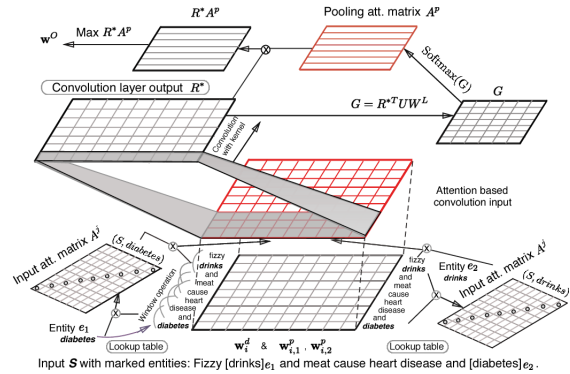

Figure 2: The architecture of the multi-level attention CNN model (Wang et al., 2016).

Jat et al. (2017) used a bidirectional gated recurrent unit (Bi-GRU) (Cho et al., 2014) to capture the long-term dependency among the words in the sentence. The tokens vectors $\mathbf{x}_t$ are passed to a Bi-GRU layer. The hidden vectors of the Bi-GRU layer are passed to a bi-linear operator which is a combination of two feed-forward layers with softmax to compute a scalar attention score for each word. The hidden vectors of the Bi-GRU layer are multiplied by their corresponding attention scores for scaling up the hidden vectors. A piecewise convolution neural network (Zeng et al., 2015) is applied to the scaled hidden vectors to obtain the

feature vector. This feature vector is passed to a feed-forward layer with softmax to determine the relation. Nayak and Ng (2019) used dependency distance based multi-focused attention model for this task. Dependency distance helps to identify the important words in the sentences and multi-factor attention helps to focus on multiple pieces of evidence for a relation. Bowen et al. (2019) used segment-level attention in their model rather than using traditional token-level attention for this task. Zhang et al. (2019) proposed an attention-based capsule network for relation extraction.

Lin et al. (2016) have used attention model for multi-instance relation extraction. They applied attention over a bag of independent sentences containing two entities to extract the relation between them. First, CNN-based models are used to encode the sentences in a bag. Then a bi-linear attention layer is used to determine the importance of each sentence in the bag. This attention helps to mitigate the problem of noisy samples obtained by distant supervision to some extent. The idea is that clean sentences get higher attention scores over the noisy ones. The sentence vectors in the bag are merged in a weighted average fashion based on their attention scores. The weighted average vector of the sentences is passed to a feed-forward neural network with softmax to determine the relation. This bag-level attention is used only for positive relations and not used for *None* relation. The reason is that the representations of the bags that express no relations are always diverse and it is difficult to calculate suitable weights for them.

Ye and Ling (2019) used intra-bag and inter-bag attention networks in a multi-instance setting for relation extraction. Their intra-bag attention is similar to the attention used by Lin et al. (2016). Additionally, they used inter-bag attention to address the noisy bag problem. They divide the bags belonging to a relation into multiple groups. The attention score for each bag in a group is obtained based on the similarity of the bags to each other within the group. This inter-bag attention is used only during training as we do not know the relations during testing. Similarly Yuan et al. (2019) proposed a cross-relation and cross-bag attention for multi-instance relation extraction. Li et al. (2020b) proposed an entity-aware embeddings and self-attention (Vaswani et al., 2017) enhanced PCNN model for relation extraction.

7.1.4 Dependency-Based Neural Models

Some previous works have incorporated the dependency structure information of sentences in their neural models for relation extraction. Xu et al. (2015) used a long short-term memory network (LSTM) (Hochreiter and Schmidhuber, 1997) along the shortest dependency path (SDP) between two entities to find the relation between them. Each token along the SDP is represented using four embeddings – pre-trained word vector, POS tag embedding, embedding for the dependency relation between the token and its child in the SDP, and embedding for its WordNet (Fellbaum, 2000) hypernym. They divide the SDP into two sub-paths: (i) The left SDP which goes from entity 1 to the common ancestor node (ii) The right SDP which goes from entity 2 to the common ancestor node. This common ancestor node is the lowest common ancestor between the two entities in the dependency tree. The token vectors along the left SDP and right SDP are passed to an LSTM layer separately. A pooling layer is applied to the hidden vectors to extract the feature vector from the left SDP and right SDP. These two vectors are concatenated and passed to a classifier to find the relation.

Liu et al. (2015) exploited the shortest dependency path (SDP) between two entities and the sub-trees attached to that path (augmented dependency path) for relation extraction. Each token in the SDP is represented using its pre-trained embedding and its sub-tree representation. The sub-tree representation of a token is obtained from the sub-tree of the dependency tree where the token is the root node. The dependency relations are represented using trainable embeddings. Each node in the sub-tree of a token receives information from its children including the dependency relations. The sub-tree representation of the token is obtained by following the sub-tree rooted at the token from its leaf nodes to the root in a bottom-up fashion. Next, they use CNN with max-pooling on the vectors of the sequence of the tokens and dependency relations across the SDP. The output of the max-pooling operation is passed to a classifier to find the relation.

Miwa and Bansal (2016) used a tree LSTM network along the shortest dependency path (SDP) between two entities to find the relation between them. They used a bottom-up tree LSTM and top-down tree LSTM in their model. In the bottom-up
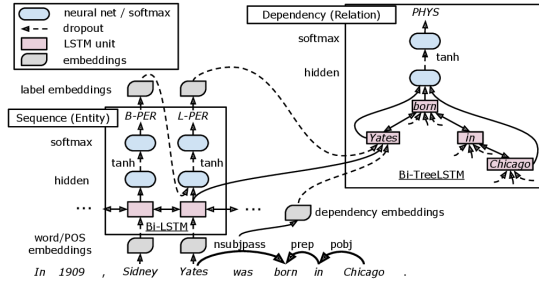
Figure 3: The architecture of the relation extraction model using LSTMs on sequences and tree structures (Miwa and Bansal, 2016).

tree LSTM, each node receives information from all of its children. The hidden representation of the root node of this bottom-up tree LSTM is used as the final output. In the top-down tree LSTM, each node receives the information from its parent node. The hidden representations of the head token of two entities are the final output of this tree LSTM. The representations of the bottom-up tree LSTM and top-down tree LSTM are concatenated and passed to a classifier to find the relation. They showed that using the SDP tree over the full dependency tree is helpful as unimportant tokens for the relation are ignored in the process. Veyseh et al. (2020) proposed a ON-LSTM (Shen et al., 2019) based relation extraction model to preserve the syntax consistency in the model.

7.1.5 Graph-Based Neural Models

Graph-based models are popular for many NLP tasks as they work on non-linear structures. Quirk and Poon (2017) proposed a graph-based model for cross-sentence relation extraction. They built a graph from the sentences where every word is considered as a node in the graph. Edges are created based on the adjacency of the words, dependency tree relations, and discourse relations. They extract all the paths from the graph starting from entity 1 to entity 2. Each path is represented by features such as lexical tokens, the lemma of the tokens, POS tags, etc. They use all the path features to find the relation between the two entities. Peng et al. (2017) and Song et al. (2018) used a similar graph for N-ary cross-sentence relation extraction. Rather than using explicit paths, they used an LSTM on a graph. A graph LSTM is a general structure for a linear LSTM or tree LSTM. If the graph contains only the word adjacency edges, then the graph LSTM becomes a linear LSTM. If the graph contains the edges from the

dependency tree, it becomes a tree LSTM. A general graph structure may contain cycles. So Peng et al. (2017) divides this graph into two directed acyclic graphs (DAG), where the forward DAG contains only the forward edges among the tokens and the backward DAG contains only the backward edges among the tokens. Each node has a separate forget gate for each of its neighbors. It receives information from the neighbors and updates its hidden states using LSTM equations (Hochreiter and Schmidhuber, 1997). If we only consider the word adjacency edges, this graph LSTM becomes a bi-directional linear LSTM. Song et al. (2018) did not divide the graph into two DAGs, but directly used the graph structure to update the states of the nodes. At time step t, each node receives information from its neighbor from the previous time step and update its hidden states using LSTM equations. This process is repeated k number of times where k is a hyper-parameter.

Kipf and Welling (2017) and Veličković et al. (2018) proposed a graph convolutional network (GCN) model which used simple linear transformations to update the node states, unlike the graph LSTMs used by Peng et al. (2017) and Song et al. (2018). Kipf and Welling (2017) gave equal weights to the edges, whereas Veličković et al. (2018) used an attention mechanism to assign different weights to the edges. Vashishth et al. (2018), Zhang et al. (2018), and Guo et al. (2019) used graph convolutional networks for sentence-level relation extraction. They considered each token in a sentence as a node in the graph and used the syntactic dependency tree to create a graph structure among the nodes. Vashishth et al. (2018) used the GCN in a multi-instance setting. They used a Bi-GRU layer and a GCN layer over the full dependency tree of the sentences to encode them. The sentence representations in a bag were aggregated and passed to a classifier to find the relation. Following Miwa and Bansal (2016), Zhang et al. (2018) used only the shortest dependency path (SDP) tree to build the adjacency matrix for the graph. Along with the SDP tree, they included the edges that are distance K away from the SDP where K is a hyper-parameter. Guo et al. (2019) proposed a soft pruning strategy over the hard pruning strategy of Zhang et al. (2018) in their GCN model. They considered the full dependency tree to build the adjacency matrix but using a multi-head self attention-based soft pruning

strategy, they can identify the important and unimportant edges in the graph. Mandya et al. (2020) proposed GCN over multiple sub-graphs for this task. They created such sub-graphs based on the shortest dependency path between two entities and the tokens associated with the two entities.
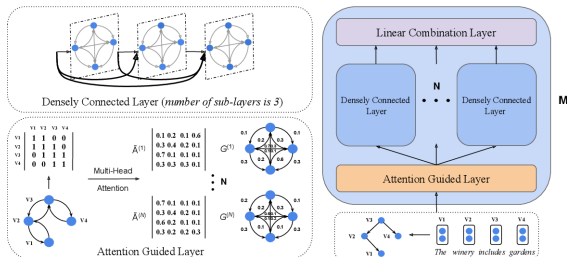

Figure 4: The architecture of the attention guided graph convolutional network for relation extraction (Guo et al., 2019).

Sahu et al. (2019); Christopoulou et al. (2019); Nan et al. (2020) used GCN for document-level relation extraction. Sahu et al. (2019) considered each token in a document as a node in a graph. They used syntactic dependency tree edges, word adjacency edges, and coreference edges to create the connections among the nodes. Christopoulou et al. (2019) considered the entity mentions, entities, and sentences in a document as nodes of a graph. They used rule-based heuristics to create the edges among these nodes. In their graph, each node and each edge were represented by vectors. GCN was used to update the vectors of nodes and edges. Finally, the edge vector between the two concerned entities was passed to a classifier to find the relation. Nan et al. (2020) considered the entity mentions, entities, and tokens on the shortest dependency path between entity mentions as nodes in a graph. They used a structure induction module to learn the latent structure of the document-level graph. A multi-hop reasoning module was used to perform inference on the induced latent structure, where representations of the nodes were updated based on an information aggregation scheme. Zeng et al. (2020b) proposed a graph aggregation and inference network for document-level relation extraction. They construct an entity-mention level graph to capture their interaction in the document and an entity-level graph to aggregate the mention-level information. Wang et al. (2020a) used a global graph similar to Christopoulou et al. (2019) to model the entities in a document and then used a multi-head

attention network (Vaswani et al., 2017) to aggregate the information in the global graph. Zhou et al. (2020) proposed multi-head attention guided graph convolution network and Li et al. (2020a) proposed GCN-based dual attention network for document level relation extraction.

7.1.6 Contextualized Embedding-Based Neural Models

Contextualized word embeddings such as ELMo (Peters et al., 2018), BERT (Devlin et al., 2019), and SpanBERT (Joshi et al., 2019) can be useful for relation extraction. These language models are trained on large corpora and can capture the contextual meaning of words in their vector representations. All neural models that are proposed for relation extraction use word representations such as Word2Vec (Mikolov et al., 2013) or GloVe (Pennington et al., 2014) in their word embedding layer. Contextualized embeddings can be added in the embedding layer of the relation extraction models to improve their performance further. The SpanBERT model shows significant improvement in performance on the TACRED dataset. Joshi et al. (2019) replaced the entity 1 token with its type and SUBJ such as PER-SUBJ and entity 2 token with its type and OBJ such as LOC-OBJ in the sentences to train the model. Finally, they used a linear classifier on top of the CLS token vector to find the relation. Baldini Soares et al. (2019) also proposed a BERT based model where they used special marker for entity 1 and entity 2 in the sentences. Then they used the vector of the start token of the entity 1 and entity 2 for relation classification.

Wang et al. (2019) proposed two-step fine-tuning of BERT for document-level relation extraction on the DocRED dataset. In the first step, they used BERT to identify whether or not there is a relation between two entities. In the second step, they used BERT to classify the relation. Nan et al. (2020) also used BERT in their model to show that it significantly improved the performance on the DocRED dataset compared to GloVe vectors. Han and Wang (2020) used BERT to identify all possible relations among the entity pairs in documents in a single pass. They used entity types and special tokens to mark all the entity mentions in documents. All entity mentions of an entity received the same special token. Documents were passed to a pre-trained BERT model. An entity mention vector was obtained by averaging the BERT out-

puts of the entity mention tokens. An entity vector was obtained by averaging all the entity mention vectors of that entity. A bilinear classifier was used to classify the relation between two entities. Tang et al. (2020) proposed a hierarchical inference network for document-level relation extraction. They also showed that using BERT in their model improved performance significantly.

7.2 Noise Mitigation for Distantly Supervised Data

The presence of noisy samples in distantly supervised data adversely affects the performance of models. Researchers have used different techniques in their models to mitigate the effects of noisy samples to make them more robust. Multi-instance relation extraction is one of the popular methods for noise mitigation. Riedel et al. (2010), Hoffmann et al. (2011), Surdeanu et al. (2012), Lin et al. (2016), Yaghoobzadeh et al. (2017), Vashishth et al. (2018), Wu et al. (2019), and Ye and Ling (2019) used this multi-instance learning concept in their proposed relation extraction models. For each entity pair, they used all the sentences that contained these two entities to find the relation between them. Their goal was to reduce the effect of noisy samples using this multi-instance setting. They used different types of sentence selection mechanisms to give importance to the sentences that contained relation-specific keywords and ignored the noisy sentences. Ren et al. (2017) and Yaghoobzadeh et al. (2017) used the multi-task learning approach for mitigating the influence of noisy samples. They used fine-grained entity typing as an extra task in their model.

Wu et al. (2017) used an adversarial training approach for the same purpose. They added noise to the word embeddings to make the model more robust for distantly supervised training. Qin et al. (2018a) used a generative adversarial network (GAN) to address the issue of noisy samples in relation extraction. They used a separate binary classifier as a generator in their model for each positive relation class to identify the true positives for that relation and filter out the noisy ones. Qin et al. (2018b) used reinforcement learning to identify the noisy samples for the positive relation classes. Jia et al. (2019) proposed an attention-based regularization mechanism to address the noisy samples issue in distantly supervised relation extraction. They used the attention to identify the relation pat-
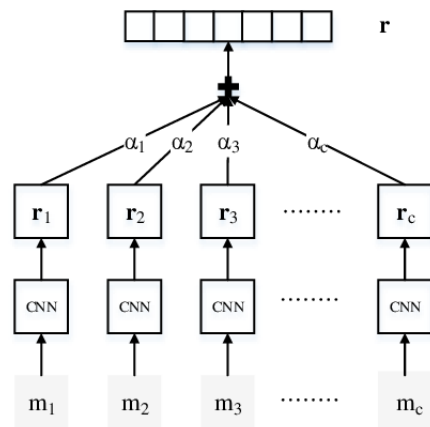

Figure 5: The architecture of the attention over sentences model for bag-level relation extraction (Lin et al., 2016).

terns in the sentences and sentences which do not contain such patterns are considered as noisy samples. He et al. (2020) used reinforcement learning to identify the noisy samples for the positive relations and then used the identified noisy samples as unlabeled data in their model. Shang et al. (2020) used a clustering approach to identify the noisy samples. They assigned the correct relation label to these noisy samples and used them as additional training data in their model.

7.3 Zero-Shot and Few-Shot Relation Extraction

Distantly supervised datasets cover a small subset of relations from the KBs. Existing KBs such as Freebase, Wikidata, and DBpedia contain thousands of relations. Due to the mismatch of the surface form of entities in KBs and texts, distant supervision cannot find adequate training samples for most relations in KBs. It means that distantly supervised models cannot fill the missing links belonging to these uncovered relations. Zero-shot or few-shot relation extraction can address this problem. These models can be trained on a set of relations and can be used for inferring another set of relations.

Levy et al. (2017) and Li et al. (2019) converted the relation extraction task to a question-answering task and used the reading comprehension approach for zero-shot relation extraction. In this approach, entity 1 and the relation are used as questions, and entity 2 is the answer to the question. If entity 2 does not exist, the answer is *NIL*. Levy et al. (2017) used the BiDAF model

Model	Prec.	Rec.	F1
SDP-LSTM (Xu et al., 2015)	66.3	52.7	58.7
Tree-LSTM (Tai et al., 2015)	66.0	59.2	62.4
GCN (Zhang et al., 2018)	69.8	59.0	64.0
PA-LSTM (Zhang et al., 2017)	65.7	64.5	65.1
AGGCN (Guo et al., 2019)	69.9	60.9	65.1
C-GCN (Zhang et al., 2018)	69.9	63.3	66.4
GCN + PA-LSTM (Zhang et al., 2018)	71.7	63.0	67.1
C-GCN + PA-LSTM (Zhang et al., 2018)	71.3	65.4	68.2
C-AGGCN (Guo et al., 2019)	73.1	64.2	69.0
BERT (Devlin et al., 2019)	69.1	63.9	66.4
$BERT_{EM}$ (Baldini Soares et al., 2019)			70.1
SpanBERT (Joshi et al., 2019)	70.8	70.9	70.8
$BERT_{EM}$ + MTB (Baldini Soares et al., 2019)			71.5

Table 5: Current State-of-the-art on TACRED dataset.

Model	F1
SVM (Rink and Harabagiu, 2010)	82.2
CNN (Zeng et al., 2014)	82.7
PA-LSTM (Zhang et al., 2017)	82.7
SDP-LSTM (Xu et al., 2015)	83.7
SPTree (Miwa and Bansal, 2016)	84.4
C-GCN (Zhang et al., 2018)	84.8
C-AGGCN (Guo et al., 2019)	85.7
Att-Input-CNN (Wang et al., 2016)	87.5
Att-Pooling-CNN (Wang et al., 2016)	88.0
$BERT_{EM}$ (Baldini Soares et al., 2019)	89.2
$BERT_{EM}$ + MTB (Baldini Soares et al., 2019)	89.5

Table 6: Current State-of-the-art on SemEval 2010 Task 8 dataset.

(Seo et al., 2017) with an additional *NIL* node in the output layer for this task on the WikiReading (Hewlett et al., 2016) dataset with additional negative samples. They used a set of relations during training and another set of relations during testing. Li et al. (2019) used templates to create the question using entity 1 and the relation. They modified the machine-reading comprehension models to a sequence tagging model so that they can find multiple answers to a question. Although they did not experiment with the zero-shot scenario, this approach can be used for zero-shot relation extraction too. FewRel 2.0 (Gao et al., 2019) is a dataset for few-shot relation extraction. In few-shot relation extraction, training and test relations are different just like zero-shot extraction. But during testing, a few examples of the test relations are provided to the model for better prediction.

7.4 Joint Extraction Approaches

All the previously mentioned works on relation extraction assume that entities are already identified by a named entity recognition system. They classify the relation between two given entities at the sentence level or the bag-of-sentences level.

These models depend on an external named entity recognition system to identify the entities in a text. Recently, some researchers (Katiyar and Cardie, 2016; Miwa and Bansal, 2016; Bekoulis et al., 2018; Nguyen and Verspoor, 2019) tried to remove this dependency. They tried to bring the entity recognition and relation identification tasks closer by sharing their parameters and optimizing them together. They first identify all the entities in a sentence and then find the relation among all the pairs of identified entities. Although they identify the entities and relations in the same network, they still identify the entities first and then determine the relation among all possible pairs in the same network. So these models miss the interaction among the relation triplets present in a sentence. These approaches resemble the pipeline approach to some extent.

Zheng et al. (2017) first proposed a truly joint extraction model for this task. They used a sequence tagging scheme to jointly extract the entities and relations. They created a set of tags derived from the Cartesian product of entity tags and relation tags. These new tags can encode the entity information and relation information together. But this strategy does not work when entities are shared among multiple triplets, as only one tag can be assigned to a token. Zeng et al. (2018) proposed an encoder-decoder model with a copy mechanism to extract relation triplets with overlapping entities. Their model has a copy network to copy the last token of two entities from the source sentence and a classification network to classify the relation between copied tokens. Their model cannot extract the full entity names of the triplets. Their best performing model uses a separate decoder to extract each triplet. During training, they need to fix the maximum number of decoders and dur-

Model	Prec.	Rec.	F1	Entity Matching Type
Tagging (Zheng et al., 2017)	0.624	0.317	0.420	P
CopyR (Zeng et al., 2018)	0.610	0.566	0.587	P
GraphR (Fu et al., 2019)	0.639	0.600	0.619	P
CopyMTL$_{Mul}$ (Zeng et al., 2020a)	0.757	0.687	0.720	E
MrMep (Chen et al., 2019)	0.779	0.766	0.771	E
HRL (Takanobu et al., 2019)	0.781	0.771	0.776	E
ETLSpan (Bowen et al., 2020)	0.855	0.717	0.780	E
PNDec (Nayak and Ng, 2020)	0.806	0.773	0.789	E
WDec (Nayak and Ng, 2020)	0.881	0.761	0.817	E
CasRel$_{LSTM}$ (Wei et al., 2020)	0.842	0.830	0.836	P
TPLinker$_{LSTM}$(Wang et al., 2020b)	0.860	0.820	0.840	E
RSAN (Yuan et al., 2020)	0.857	0.836	0.846	E
RIN (Sun et al., 2020)	0.839	0.855	0.847	E
CGT$_{BERT}$ (Ye et al., 2021)	0.947	0.842	0.891	E
CasRel$_{BERT}$ (Wei et al., 2020)	0.897	0.895	0.896	P
TPLinker$_{BERT}$(Wang et al., 2020b)	0.914	0.926	0.920	E
SPN$_{BERT}$ (Sui et al., 2021)	0.925	0.922	0.923	E

Table 7: Current state-of-the-art performance on NYT24 datasets for the joint extraction task. P=Partial entity matching, E=Exact entity matching.

Model	Prec.	Rec.	F1	Entity Matching Type
Tagging (Zheng et al., 2017)	0.593	0.381	0.464	E
CopyR (Zeng et al., 2018)	0.569	0.452	0.504	P
SPTree (Miwa and Bansal, 2016)	0.492	0.557	0.522	E
HRL (Takanobu et al., 2019)	0.692	0.601	0.643	E
MrMep (Chen et al., 2019)	0.717	0.635	0.673	E
PNDec (Nayak and Ng, 2020)	0.732	0.624	0.673	E
WDec (Nayak and Ng, 2020)	0.777	0.608	0.682	E

Table 8: Current state-of-the-art performance on NYT29 datasets for the joint extraction task.

ing inference, their model can only extract up to that fixed number of triplets. Also, due to the use of separate decoders for each triplet, their model misses the interaction among the triplets.
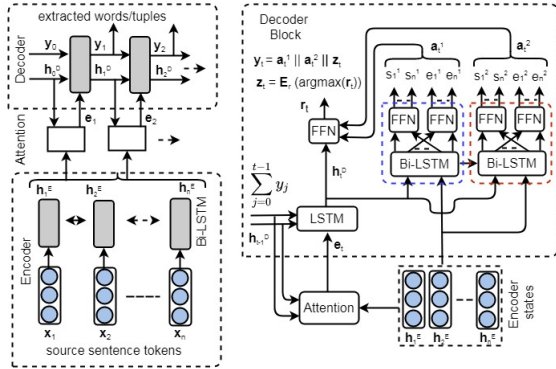

Figure 6: The architecture of the joint entity and relation extraction model as proposed in Nayak and Ng (2020).

Takanobu et al. (2019) proposed a hierarchical reinforcement learning-based (RL) deep neural model for joint entity and relation extraction. A high-level RL is used to identify the relation based on the relation-specific tokens in the sentences. After a relation is identified, a low-level

RL is used to extract the two entities associated with the relation using a sequence labeling approach. This process is repeated multiple times to extract all the relation triplets present in the sentences. A special *None* relation is used to identify no relation situation in the sentences. Entities extracted associated with the *None* relations are ignored. Fu et al. (2019) used a graph convolutional network (GCN) where they treated each token in a sentence as a node in a graph and edges were considered as relations. Trisedya et al. (2019) used an N-gram attention mechanism with an encoder-decoder model for the completion of knowledge bases using distantly supervised data. Chen et al. (2019) used the encoder-decoder framework for this task where they used a CNN-based multi-label classifier to find all the relations first, then used multi-head attention (Vaswani et al., 2017) to extract the entities corresponding to each relation. Nayak and Ng (2020) used encoder-decoder network for this joint extraction task. They proposed a word-level decoding framework and a pointer network-based decoding framework for the same.

CopyMTL model (Zeng et al., 2020a) was proposed to address the issues of CopyR (Zeng et al., 2018) model. CopyR model can only extract

Model	Prec.	Rec.	F1	Entity Matching Type
Tagging (Zheng et al., 2017)	0.525	0.193	0.283	P
CopyR (Zeng et al., 2018)	0.377	0.364	0.371	P
GraphR (Fu et al., 2019)	0.447	0.411	0.429	P
CopyMTL$_{One}$ (Zeng et al., 2020a)	0.578	0.601	0.589	E
HRL (Takanobu et al., 2019)	0.695	0.629	0.660	E
MrMep (Chen et al., 2019)	0.694	0.770	0.730	E
RIN (Sun et al., 2020)	0.773	0.768	0.770	E
RSAN (Yuan et al., 2020)	0.805	0.838	0.821	E
ETLSpan (Bowen et al., 2020)	0.843	0.820	0.831	E
CasRel$_{LSTM}$ (Wei et al., 2020)	0.869	0.806	0.837	P
TPLinker$_{LSTM}$(Wang et al., 2020b)	0.919	0.816	0.864	E
CGT$_{BERT}$ (Ye et al., 2021)	0.929	0.756	0.834	E
TPLinker$_{BERT}$(Wang et al., 2020b)	0.889	0.845	0.867	E
CasRel$_{BERT}$ (Wei et al., 2020)	0.934	0.901	0.918	P
SPN$_{BERT}$ (Sui et al., 2021)	0.931	0.936	0.934	P

Table 9: Current state-of-the-art performance on WebNLG datasets for the joint extraction task.
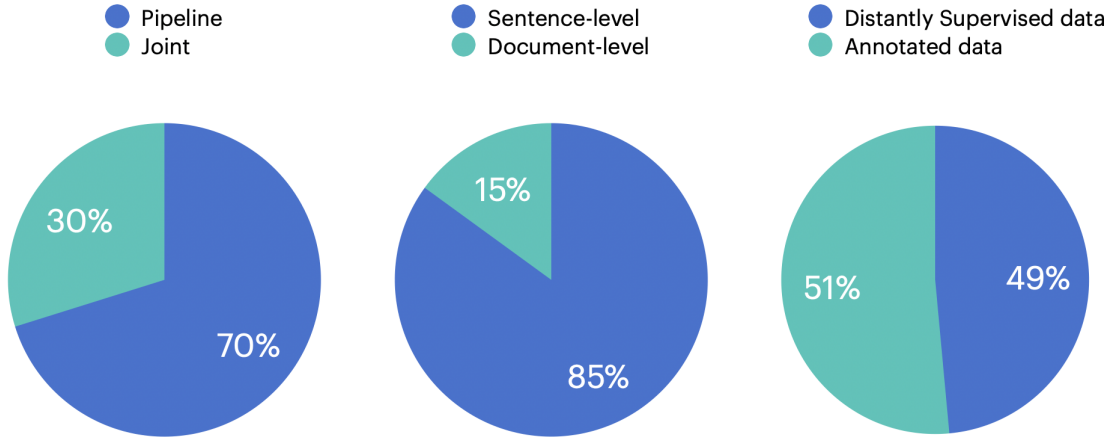

Figure 7: The statistics of the research articles published in year 2019 (CoNLL, ACL, EMNLP, AAAI, IJCAI) and 2020 (COLING, ACL, EMNLP, AAAI, IJCAI). The left one shows the pipeline vs joint extraction models, the middle one shows the sentence-level vs document-level extraction models, and the right one shows the use of distantly supervised datasets vs annotated datasets.

the last token of the entities, whereas CopyMTL model used a sequence tagging approach to extract the full entity names. Bowen et al. (2020) decomposed the joint extraction task into two subtasks: (i) head entity extraction (ii) tail entity and relation extraction. They used a sequence tagging approach to solve these two sub-tasks. Similarly, Wei et al. (2020) proposed a sequence tagging approach for this task. They first identified the head entities and then for each head entity and each relation, they identified the tail entities using a sequence tagging approach. They used pre-trained BERT (Devlin et al., 2019) in their model to improve the performance. Yuan et al. (2020) used a relation-specific attention mechanism with sequence labeling to jointly extract the entities and relations. Wang et al. (2020b) proposed a single-stage joint extraction model using

entity-pair linking. They aligned the sentence tokens using the Cartesian product so that the boundary tokens of the subject and object entities are aligned. Then they used a classifier to tag each token-pair as entity head, entity tail, subject head, subject tail, object head, and object tail for each relation separately. This scheme can identity the multiple triplets with overlapping entities easily. Sui et al. (2021) proposed a bipartite matching loss in the encoder-decoder network which considers the group of relation triplets as a set, not as a sequence. Ye et al. (2021) transformer-based generative model for this task. They used negative triplets to train the transformer model in contrastive settings. Wang and Lu (2020) proposed a table-sequence encoder model where the sequence encoder captures the entity-related information and the table encoder captures the relation-specific

information. Sun et al. (2020) proposed a recurrent multi-task learning architecture to explicitly capture the interaction between entity recognition task and relation classification task. Ji et al. (2020) proposed a span-based multi-head attention network for joint extraction task. Each text span is a candidate entity and each text span pairs is a candidate for relation triplets.

8 Current State-of-the-art & Trends

NYT10 is the most popular dataset for experiments in pipeline-based relation extraction. Since the test dataset of NYT10 is not manually annotated, researchers mostly report a precision-recall curve to compare the models (Vashishth et al., 2018; Ye and Ling, 2019; Li et al., 2020b). TACRED and SemEval 2010 Task 8 datasets are manually annotated and can be used for automatic evaluation. We have included the current state-of-the-art on these two dataset in Table 5 and Table 6. DocRED[2] and FewRel[3] datasets have manually annotated testset and their they have a leaderboard where current state-of-the-art can be found. For the joint extraction task researchers used NYT24, NYT29, and WebNLG datasets which have a considerably large number of relations. We have included the current state-of-the-art performance of the models on NYT24, NYT29, and WebNLG datasets in Table 7, Table 8, and Table 9 respectively.
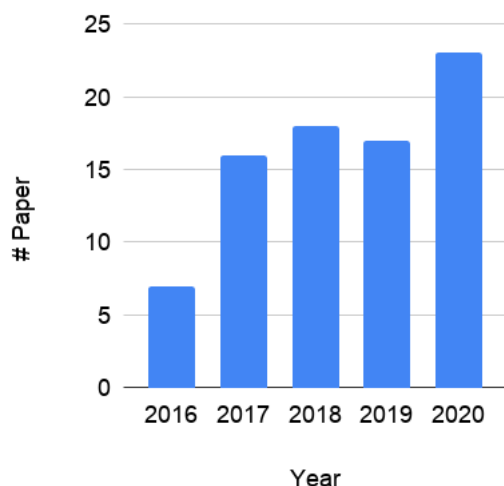

Figure 8: Publication trend of relation extraction research at ACL, EMNLP, AAAI, and IJCAI in 2016–2020.

We analyze the research articles published in 2019 (CoNLL, ACL, EMNLP, AAAI, IJCAI) and 2020 (COLING, ACL, EMNLP, AAAI, IJCAI) and include statistics in Figure 7. We see that majority of the research focuses on pipeline-based approaches on sentence-level relation extraction. We also see that the use of distantly supervised datasets and annotated datasets for experiments is evenly distributed among the published articles. We also show the increasing trends of yearly publications in relation extraction in Figure 8 over the last 5 years period (2016-2020).

9 Future Research Directions

With the progress of deep learning algorithms, significant advances have been made in the relation extraction task. However, many challenges remain in this area. In the pipeline approaches, since we need to find relations among all pairs of entities, there can be a very large number of *None* instances. This *None* class is challenging to identify as it is not a single relation but any relation outside the set of positive relations. Erroneous detection of *None* relation reduces the precision of the model and can add many wrong triplets to the KB. To build a cleaner KB, models have to perform very well to detect the *None* relation along with classifying the positive relations correctly.

Regarding the joint extraction approach, researchers do not include sentences with zero triplets in training or testing. NYT24 and NYT29 datasets are created after removing the sentences with zero triplets from the original NYT11 and NYT10 datasets. NYT11 and NYT10 datasets contain many sentences that do not have any relation triplets. So in the future, detecting sentences with no relation triplets must be handled in the joint extraction approaches.

Current relation extraction models deal with very few relations whereas existing knowledge bases have thousands of relations. In the future, we should focus more on document-level relation extraction or possibly relation extraction across documents. Following the idea proposed in Nayak (2020), we should extend the task of relation extraction to N-hop to cover more relations from the KB. However, it may not be easy to extend the task as the inclusion of more documents in the chain may make the data noisier. It will be challenging to create a clean dataset for N-hop relation extraction. Also, we need to explore zero-shot or few-

shot relation extraction to cover the relations for which we cannot obtain enough training data using distant supervision.

10   Conclusion

In this survey paper, we detail the recent progress in neural network-based relation extraction research that includes both pipeline-based and joint extraction-based relation extraction approaches. Furthermore, we describe different relation-extraction datasets and setup baselines to facilitate future research. Key issues with the current distantly-supervised datasets are also pointed out. We finally conclude with the possible future research directions to advance this field.

References

Judith Jeyafreeda Andrew. 2018. Automatic extraction of entities and relation from legal documents. In *Proceedings of the Seventh Named Entities Workshop*.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *ACL*.

Michele Banko, Michael J Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *IJCAI*.

Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. 2018. Joint entity recognition and relation extraction as a multi-head selection problem. *Expert Systems with Applications*.

Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. 2009. DBpedia-A crystallization point for the web of data. *Web Semantics: Science, Services and Agents on the World Wide Web*.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *SIGMOD*.

Yu Bowen, Zhenyu Zhang, Tingwen Liu, Bin Wang, Sujian Li, and Q. Li. 2019. Beyond word attention: Using segment attention in neural relation extraction. In *IJCAI*.

Yu Bowen, Zhenyu Zhang, Jianlin Su, Yubin Wang, Tingwen Liu, Bin Wang, and Sujian Li. 2020. Joint extraction of entities and relations based on a novel decomposition strategy. In *ECAI*.

Jiayu Chen, Caixia Yuan, Xiao-Jie Wang, and Ziwei Bai. 2019. MrMep: Joint extraction of multiple relations and multiple entity pairs based on triplet attention. In *CoNLL*.

Jason Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional LSTM-CNNs. In *TACL*.

Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. In *Workshop on Syntax, Semantics and Structure in Statistical Translation*.

Sung-Pil Choi. 2018. Extraction of protein–protein interactions (ppis) from the literature by deep convolutional neural networks with various feature embeddings. *Journal of Information Science*.

Janara Christensen, Mausam, Stephen Soderland, and Oren Etzioni. 2011. An analysis of open information extraction based on semantic role labeling. In *K-CAP*.

Fenia Christopoulou, Makoto Miwa, and Sophia Ananiadou. 2019. Connecting the dots: Document-level neural relation extraction with edge-oriented graphs. In *EMNLP and IJCNLP*.

Meiji Cui, L. Li, Zhihong Wang, and Mingyu You. 2017. A survey on relation extraction. In *CCKS*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

George R Doddington, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie M Strassel, and Ralph M Weischedel. 2004. The automatic content extraction (ACE) program-tasks, data, and evaluation. In *LREC*.

Oren Etzioni, Michael Cafarella, Doug Downey, Stanley Kok, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S Weld, and Alexander Yates. 2004. Web-scale information extraction in KnowItAll:(preliminary results). In *WWW*.

Oren Etzioni, Anthony Fader, Janara Christensen, Stephen Soderland, and Mausam. 2011. Open information extraction: The second generation. In *IJCAI*.

Christiane Fellbaum. 2000. WordNet: An electronic lexical database. *Language*.

Tsu-Jui Fu, Peng-Hsuan Li, and Wei-Yun Ma. 2019. GraphRel: Modeling text as relational graphs for joint entity and relation extraction. In *ACL*.

Tianyu Gao, Xu Han, Hao Zhu, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2019. FewRel 2.0: Towards more challenging few-shot relation classification. In *EMNLP and IJCNLP*.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. Creating training corpora for nlg micro-planners. In *ACL*.

Jinghang Gu, Longhua Qian, and Guodong Zhou. 2016. Chemical-induced disease relation extraction with various linguistic features. *Database: The Journal of Biological Databases and Curation*.

Zhijiang Guo, Yan Zhang, and Wei Lu. 2019. Attention guided graph convolutional networks for relation extraction. In *ACL*.

Xiaoyu Han and Lei Wang. 2020. A novel document-level relation extraction method based on BERT and entity information. *IEEE Access*.

Xu Han, Tianyu Gao, Yankai Lin, H. Peng, Y. Yang, Chaojun Xiao, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2020. More data, more relations, more context and more openness: A review and outlook for relation extraction. In *AACL and IJCNLP*.

Zhengqiu He, Wenliang Chen, Yuyi Wang, Wei Zhang, Guanchun Wang, and Min Zhang. 2020. Improving neural relation extraction with positive and unlabeled learning. In *AAAI*.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *SemEval*.

Daniel Hewlett, Alexandre Lacoste, Llion Jones, Illia Polosukhin, Andrew Fandrianto, Jay Han, Matthew Kelcey, and David Berthelot. 2016. WikiReading: A novel large-scale language understanding task over Wikipedia. In *ACL*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *ACL*.

Luyang huang, L. Wu, and L. Wang. 2020. Knowledge graph-augmented abstractive summarization with semantic-driven cloze reward. In *ACL*.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *ArXiv*.

Sarthak Jain, Madeleine van Zuylen, Hannaneh Hajishirzi, and Iz Beltagy. 2020. SciREX: A challenge dataset for document-level information extraction. In *ACL*.

Sharmistha Jat, Siddhesh Khandelwal, and Partha Talukdar. 2017. Improving distantly supervised relation extraction using word and entity based attention. In *AKBC*.

Bin Ji, Jie Yu, Shasha Li, Jun Ma, Q. Wu, Yusong Tan, and Huijun Liu. 2020. Span-based joint entity and relation extraction with attention-based span-specific and contextual semantic representations. In *COLING*.

Wei Jia, Dai Dai, Xinyan Xiao, and Hua Wu. 2019. ARNOR: Attention regularization based noise reduction for distant supervision relation classification. In *ACL*.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2019. SpanBERT: Improving pre-training by representing and predicting spans. *TACL*.

Arzoo Katiyar and Claire Cardie. 2016. Investigating LSTMs for joint extraction of opinion entities and relations. In *ACL*.

Thomas Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.

Shantanu Kumar. 2017. A survey of deep learning methods for relation extraction. *ArXiv*.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *NAACL-HLT*.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke S. Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *CoNLL*.

Bo Li, Wei Ye, Zhonghao Sheng, Rui Xie, Xiangyu Xi, and Shikun Zhang. 2020a. Graph enhanced dual attention network for document-level relation extraction. In *COLING*.

Fei Li, Meishan Zhang, G. Fu, and D. Ji. 2017. A neural joint model for entity and relation extraction from biomedical text. *BMC Bioinformatics*.

Xiaoya Li, Fan Yin, Zijun Sun, Xiayu Li, Arianna Yuan, Duo Chai, Mingxin Zhou, and Jiwei Li. 2019. Entity-relation extraction as multi-turn question answering. In *ACL*.

Yang Li, Guodong Long, Tao Shen, Tianyi Zhou, L. Yao, Huan Huo, and Jing Jiang. 2020b. Self-attention enhanced selective gate with entity-aware embedding for distantly supervised relation extraction. In *AAAI*.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *ACL*.

Y. Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, M. Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*.

Yang Liu, Furu Wei, Sujian Li, Heng Ji, Ming Zhou, and Houfeng Wang. 2015. A dependency-based neural network for relation classification. In *ACL and IJCNLP*.

Yi Luan, Mari Ostendorf, and Hannaneh Hajishirzi. 2017. Scientific information extraction with semi-supervised neural tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*.

Xuezhe Ma and Eduard H. Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *ACL*.

Angrosh Mandya, Danushka Bollegala, and F. Coenen. 2020. Graph convolution over multiple dependency sub-graphs for relation extraction. In *COLING*.

Mausam, Michael Schmitz, Stephen Soderland, Robert Bart, and Oren Etzioni. 2012. Open language learning for information extraction. In *EMNLP-CoNLL*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *ACL and IJCNLP*.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using LSTMs on sequences and tree structures. In *ACL*.

Guoshun Nan, Zhijiang Guo, Ivan Sekulic, and Wei Lu. 2020. Reasoning with latent structure refinement for document-level relation extraction. In *ACL*.

Tapas Nayak. 2020. Deep neural networks for relation extraction. *NUS Scholar Bank*.

Tapas Nayak and Hwee Tou Ng. 2019. Effective attention modeling for neural relation extraction. In *CoNLL*.

Tapas Nayak and Hwee Tou Ng. 2020. Effective modeling of encoder-decoder architecture for joint entity and relation extraction. In *AAAI*.

Dat Quoc Nguyen and Karin Verspoor. 2019. End-to-end neural relation extraction using deep biaffine attention. In *ECIR*.

S. Pawar, Girish Keshav Palshikar, and P. Bhattacharyya. 2017. Relation extraction : A survey. *ArXiv*.

Nanyun Peng, Hoifung Poon, Chris Quirk, Kristina Toutanova, and Wen-tau Yih. 2017. Cross-sentence n-ary relation extraction with graph LSTMs. *TACL*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *EMNLP*.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *NAACL-HLT*.

Pengda Qin, Weiran Xu, and William Yang Wang. 2018a. DSGAN: Generative adversarial training for distant supervision relation extraction. In *ACL*.

Pengda Qin, Weiran Xu, and William Yang Wang. 2018b. Robust distant supervision relation extraction via deep reinforcement learning. In *ACL*.

Delai Qiu, Yuanzhe Zhang, Xinwei Feng, Xiangwen Liao, Wenbin Jiang, Yajuan Lyu, Kang Liu, and Jun Zhao. 2019. Machine reading comprehension using structural knowledge graph-aware network. In *EMNLP and IJCNLP*.

Chris Quirk and Hoifung Poon. 2017. Distant supervision for relation extraction beyond the sentence boundary. In *EACL*.

Xiang Ren, Zeqiu Wu, Wenqi He, Meng Qu, Clare R Voss, Heng Ji, Tarek F Abdelzaher, and Jiawei Han. 2017. CoType: Joint extraction of typed entities and relations with knowledge bases. In *WWW*.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *ECML and KDD*.

Bryan Rink and Sanda Harabagiu. 2010. UTD: Classifying semantic relations by combining lexical and semantic resources. In *Proceedings of the 5th International Workshop on Semantic Evaluation*.

Dan Roth and Wen-tau Yih. 2004. A linear programming formulation for global inference in natural language tasks. In *CoNLL*.

Sunil Kumar Sahu, Fenia Christopoulou, Makoto Miwa, and Sophia Ananiadou. 2019. Inter-sentence relation extraction with document-level graph convolutional neural network. In *ACL*.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In *ICLR*.

Yuming Shang, He-Yan Huang, Xian-Ling Mao, Xin Sun, and Wei Wei. 2020. Are noisy sentences useless for distant supervised relation extraction? In *AAAI*.

Yatian Shen and Xuanjing Huang. 2016. Attention-based convolutional neural network for semantic relation extraction. In *COLING*.

Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron C. Courville. 2019. Ordered neurons: Integrating tree structures into recurrent neural networks. In *ICLR*.

Y. Shi, Y. Xiao, and L. Niu. 2019. A brief survey of relation extraction based on distant supervision. In *ICCS*.

Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. 2018. N-ary relation extraction using graph state LSTM. In *EMNLP*.

Dianbo Sui, Yubo Chen, Kang Liu, Jun Zhao, Xiangrong Zeng, and Shengping Liu. 2021. Joint entity and relation extraction with set prediction networks. In *AAAI*.

Kai Sun, Richong Zhang, Samuel Mensah, Yong yi Mao, and Xudong Liu. 2020. Recurrent interaction network for jointly extracting entities and classifying relations. In *EMNLP*.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D. Manning. 2012. Multi-instance multi-label learning for relation extraction. In *EMNLP and CoNLL*.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *ACL and IJCNLP*.

Ryuichi Takanobu, Tianyang Zhang, Jiexi Liu, and Minlie Huang. 2019. A hierarchical framework for relation extraction with reinforcement learning. In *AAAI*.

Hengzhu Tang, Yanan Cao, Zhenyu Zhang, Jiangxia Cao, Fang Fang, Shigang Wang, and Pengfei Yin. 2020. HIN: Hierarchical inference network for document-level relation extraction. *Advances in Knowledge Discovery and Data Mining*.

Ashok Thillaisundaram and Theodosia Togia. 2019. Biomedical relation extraction with pre-trained language representations and minimal task-specific architecture. *ArXiv*.

Bayu Distiawan Trisedya, Gerhard Weikum, Jianzhong Qi, and Rui Zhang. 2019. Neural relation extraction for knowledge base enrichment. In *ACL*.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *ACL*.

Shikhar Vashishth, Rishabh Joshi, Sai Suman Prayaga, Chiranjib Bhattacharyya, and Partha Talukdar. 2018. RESIDE: Improving distantly-supervised neural relation extraction using side information. In *EMNLP*.

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

M. Vela and Thierry Declerck. 2009. Concept and relation extraction in the finance domain. In *IWCS*.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *ICLR*.

Amir Pouran Ben Veyseh, Franck Dernoncourt, D. Dou, and T. Nguyen. 2020. Exploiting the syntax-model consistency for neural relation extraction. In *ACL*.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A free collaborative knowledge base. *Communications of the ACM*.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. Ace 2005 multilingual training corpus. In *Linguistic Data Consortium*.

D. Wang, Wei Hu, E. Cao, and Weijian Sun. 2020a. Global-to-local neural networks for document-level relation extraction. In *EMNLP*.

Hong Wang, Christfried Focke, Rob Sylvester, Nilesh Mishra, and William W. J. Wang. 2019. Fine-tune BERT for DocRED with two-step process. *ArXiv*.

Jue Wang and Wei Lu. 2020. Two are better than one: Joint entity and relation extraction with table-sequence encoders. In *EMNLP*.

Linlin Wang, Zhu Cao, Gerard de Melo, and Zhiyuan Liu. 2016. Relation classification via multi-level attention CNNs. In *ACL*.

Yucheng Wang, Bowen Yu, Y. Zhang, Tingwen Liu, Hongsong Zhu, and L. Sun. 2020b. Tplinker: Single-stage joint extraction of entities and relations through token pair linking. In *COLING*.

Zhepei Wei, Jianlin Su, Yue Wang, Yuan Tian, and Yi Chang. 2020. A novel cascade binary tagging framework for relational triple extraction. In *ACL*.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. In *TACL*.

Shanchan Wu, Kai Fan, and Qiong Zhang. 2019. Improving distantly supervised relation extraction with neural noise converter and conditional optimal selector. In *AAAI*.

Yi Wu, David Bamman, and Stuart Russell. 2017. Adversarial training for relation extraction. In *EMNLP*.

Yuning Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. 2015. Classifying relations via long short term memory networks along shortest dependency paths. In *EMNLP*.

Yadollah Yaghoobzadeh, Heike Adel, and Hinrich Schütze. 2017. Noise mitigation for neural entity typing and relation extraction. In *EACL*.

Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. 2019. DocRED: A large-scale document-level relation extraction dataset. In *ACL*.

Alexander Yates, Michael Cafarella, Michele Banko, Oren Etzioni, Matthew Broadhead, and Stephen Soderland. 2007. TEXTRUNNER: Open information extraction on the web. In *NAACL-HLT*.

Hongbin Ye, Ningyu Zhang, Shumin Deng, M. Chen, Chuanqi Tan, Fei Huang, and Huajun Chen. 2021. Contrastive triple extraction with generative transformer. *AAAI*.

Zhi-Xiu Ye and Zhen-Hua Ling. 2019. Distant supervision relation extraction with intra-bag and inter-bag attentions. In *NAACL-HLT*.

Y. Yuan, Liyuan Liu, Siliang Tang, Zhongfei Zhang, Y. Zhuang, S. Pu, Fei Wu, and Xiang Ren. 2019. Cross-relation cross-bag attention for distantly-supervised relation extraction. In *AAAI*.

Yue Yuan, Xiaofei Zhou, Shirui Pan, Qiannan Zhu, Zeliang Song, and Li Guo. 2020. A relation-specific attention network for joint entity and relation extraction. In *IJCAI*.

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *EMNLP*.

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

Daojian Zeng, Haoran Zhang, and Qianying Liu. 2020a. CopyMTL: Copy mechanism for joint extraction of entities and relations with multi-task learning. In *AAAI*.

Shuang Zeng, Runxin Xu, Baobao Chang, and Lei Li. 2020b. Double graph based reasoning for document-level relation extraction. In *ACL*.

Xiangrong Zeng, Daojian Zeng, Shizhu He, Kang Liu, and Jun Zhao. 2018. Extracting relational facts by an end-to-end neural model with copy mechanism. In *ACL*.

Xinsong Zhang, P. Li, W. Jia, and Zhao Hai. 2019. Multi-labeled relation extraction with attentive capsule network. In *AAAI*.

Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. In *EMNLP*.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware attention and supervised data improve slot filling. In *EMNLP*.

Yang Zhao, Jiajun Zhang, Yin qing Zhou, and Chengqing Zong. 2020. Knowledge graphs enhanced neural machine translation. In *IJCAI*.

Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu. 2017. Joint extraction of entities and relations based on a novel tagging scheme. In *ACL*.

Huiwei Zhou, Yibin Xu, W. Yao, Zhe Liu, Chengkun Lang, and H. Jiang. 2020. Global context-enhanced graph convolutional networks for document-level relation extraction. In *COLING*.

Tong Zhu, Haitao Wang, Junjie Yu, Xiabing Zhou, Wenliang Chen, Wei Zhang, and Min Zhang. 2020. Towards accurate and consistent evaluation: A dataset for distantly-supervised relation extraction. In *COLING*.