ISSN: 2229-6956 (ONLINE)
DOI: 10.21917/ijsc.2019.0263

RELATION EXTRACTION USING DEEP LEARNING METHODS - A SURVEY

C.A. Deepa[1], P.C. ReghuRaj[2] and Ajeesh Ramanujan[3]

[1,2]Department of Computer Science and Engineering, Government Engineering College Sreekrishnapuram, India
[3]Department of Computer Science and Engineering, College of Engineering Trivandrum, India

Abstract

Relation extraction has an important role in extracting structured information from unstructured raw text. This task is a crucial ingredient in numerous information extraction systems seeking to mine structured facts from text. Nowadays, neural networks play an important role in the task of relation extraction. The traditional non deep learning models require feature engineering. Deep Learning models such as Convolutional Neural Networks and Long Short Term Memory networks require less feature engineering than non-deep learning models. Relation Extraction has the potential of employing deep learning models with the creation of huge datasets using distant supervision. This paper surveys the current trend in Relation Extraction using Deep Learning models.

Keywords:

Relation Extraction, Deep Learning, LSTM, CNN, word Embeddings

1. INTRODUCTION

Relation Extraction (RE), a subtask of Information Extraction is an emerging area in the field of Natural Language Understanding. It is not a trivial task because it needs to identify the piece of text which contains the basic unit of information and requires to process them in order to obtain hidden information in the document. The basic units of information for RE are named entities, relations, and events. A Named Entity (NE) is often a word or phrase that represents a specific real-world object. For example, Sunder Pichai is an NE which has a specific mention in the sentence "Sunder Pichai is the Chief Executive Officer (CEO) of Google LLC". A relation occurs between two named entities or events. In the above sentence, CEO is the relation between two named entities, namely Sunder Pichai and Google LLC and represented as a binary relation CEO. Relation Extraction can be either at global level or at mention level [1].

A global level relation extraction task lists all pairs of entity mentions which hold a certain semantic relation while mention level relation extraction takes an entity pair and a sentence that contains the entity pair as input, and then predicts whether a certain relation exists between the specified entity pair. The task of relation extraction refers to predicting whether a relation occurs between a pair of entities in a document, modelled as a binary classification problem. If a pair of entities in a text is related, then the relation classifier will predict the relation from a predefined relation set. Supervised approaches for extracting relations build a multi-class relation classification model, with an extra class labelled "No Relation". Deep learning models are becoming important due to their demonstrated success at tackling complex learning problems [2].

Deep learning allows computational models that are composed of multiple processing layers, to learn representations of data with multiple levels of abstraction. This peculiarity of deep learning lets the model to extract relations with better accuracy. Deep learning discovers intricate structure in large data sets by using the back propagation algorithm [3]. The traditional non-deep learning methods depend upon existing natural language processing systems for feature extraction. Such a pipelined approach causes cascaded error and retards the performance of the system. Also, the manually constructed features may not capture all relevant information. These problems in relation extraction can be mitigated using deep learning techniques. This review focuses on the current trend in mention level relation extraction using deep learning models. The classification of deep learning models for relation extraction is shown in Fig.1.

The deep learning models for relation extraction are classified into end-to-end models, dependency models, and distantly supervised models.

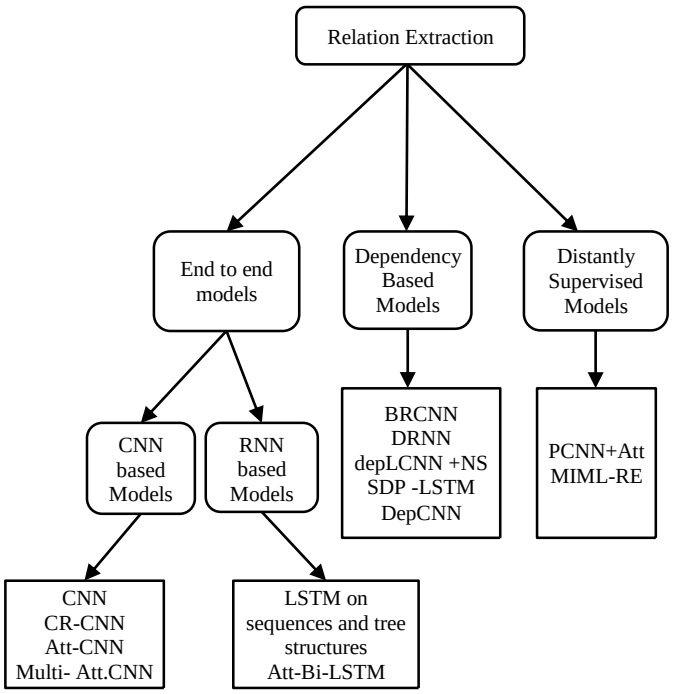

Fig.1. Classification of relation extraction models using deep learning

1.1 PROBLEM DEFINITION

The task of relation classification can be defined as follows. Given a sentence S with a pair of annotated entities e_1 and e_2, the task is to identify the semantic relation between e_1 and e_2 in accordance with a set of predefined relation classes (e.g., content-container, cause-effect).

The paper is organized as follows: Some basic concepts such as features and different models are discussed in section 2. Section 3 is about supervised methods and the relation extraction task in

detail is discussed in section 4. Section 5 analyses the results of various relation extraction models, which are reviewed in this paper followed by conclusion in section 6.

2. BASIC CONCEPTS

This section describes some common features used by most of the deep learning models for the task of relation extraction.

2.1 WORD EMBEDDINGS

Word embedding is the vector representation of a word in a *d* dimensional vector space, where *d* is a relatively a small number (typically between 50 and 1000). This distributed word representation allows words with similar meaning to have similar representations. The vectors of words with similar meaning are placed closer when projected onto a vector space [4]. These are low-dimensional vector representations from a corpus of text, which preserve the contextual similarity of words. Word embeddings are capable of capturing the context of a word in a document, semantic and syntactic similarities of words, relation with other words, etc. The basic idea of word embeddings is that any two words that have similar meaning will also have similar context words [4]. Two different approaches that leverage this principle are count based methods and prediction based methods [5]. Predictive methods predict a word from its neighbours in terms of learned small, dense embedding vectors. word2vec [6] and glove [7] are two methods for learning word embeddings from raw text.

2.2 WORD POSITION EMBEDDINGS

The information needed to determine the class of a relation between two target nouns normally comes from words which are close to the target nouns. Word Position Embedding (WPE) keeps track of the fact that how close the words are to the target nouns [8]. These are derived from the relative distances of current word to target nouns, say $Noun_1$; and $Noun_2$.

2.3 DEPENDENCY PARSE

Dependency parse is a common feature used in relation extraction tasks. The dependency parse trees reveal non-local dependencies within sentences, i.e. between words that are far apart in a sentence [9], [10]. So dependency parsers are used to capture long distance dependencies between two nominal. Dependency parse tree holds the grammatical structure of a sentence like subject, object etc. and thus it becomes an important feature in relation extraction.

Features derived from WordNet, named entity recognizers, and part of speech tags are also considered for Relation Extraction.

2.4 MODELS

Commonly used deep learning models for relation extraction are Convolutional Neural Networks and Long Shot Term Memory Networks. Variants of these deep learning models are used for relation extraction.

2.4.1 Convolutional Neural Network (CNN):

CNNs are inspired by the fact that the visual cortex of animals has a complex arrangement of cells, responsible for the detection of light in small local regions of the visual field [11]. Convolutional Neural Networks are very similar to ordinary Neural Networks, which are made up of neurons that have learnable weights and biases. A CNN, in particular, has one or more layers of convolution units. A convolution unit receives its input from multiple units of the previous layer which together create proximity. Therefore, the input units form a small neighbourhood to share their weights. The convolution units (as well as pooling units) are especially beneficial, because they reduce the number of units in the network and consider the context or shared information in the small neighbourhoods [12]. Nowadays, CNNs are commonly used for extracting semantic relationships [8], [13], [14], [15, [16].

2.4.2 Long Short Term Memory Networks (LSTM):

In conventional Back Propagation Networks or Real Time Recurrent Learning Networks, the error signals flowing backward in each time step tend to blow up or vanish. Learning to store information over extended time intervals via recurrent back propagation takes a very long time, mostly due to insufficient, decaying back propagation error. Long Short-Term Memory (LSTM) is a novel recurrent architecture designed to overcome these error back flow problems [17]. LSTM is a specific recurrent neural network (RNN) architecture that is designed to model temporal sequences and their long-range dependencies more accurately than conventional RNNs. LSTM does not use activation function within its recurrent components, the stored values are not modified, and the gradient does not tend to vanish during training. Usually, LSTM units are implemented in blocks with several units. These blocks have three or four gates: input gate, forget gate, output gate that control information flow drawing on the logistic function.

3. SUPERVISED LEARNING

The non-deep learning methods for relation extraction typically work in a supervised paradigm. Supervised approaches focus on mention level relation extraction [1]. It requires labelled data where each entity pair in the corpus is labelled with a predefined relation type. Supervised relation extraction can be divided into two classes as feature based methods and kernel based methods. Both of these methods depend on the existing NLP systems for tasks like named entity recognition. Such pipeline methods are prone to error propagation from the first step (i.e., extracting entity mentions) to the second step (i.e., extracting relations). Another problem with traditional supervised methods is that manually constructed features may not capture all the relevant information. In order to overcome these problems, a joint model for extraction of entities and relations is needed. This can be done by considering deep learning techniques [18].

4. RELATION EXTRACTION

Relation extraction using deep learning models can be classified into:

- End-to-End Models

- Dependency Models
- Distributed Supervised Models.

4.1 END-TO-END MODELS

Both CNN based models and RNN based models are used for relation extraction tasks.

4.1.1 CNN Based Models

Convolutional Neural Networks play an important role in relation extraction. Based on the kind of layers, there various kinds of CNNs are used for the task.

- *Convolutional Deep Neural Network (CDNN):* A convolutional deep neural network (CDNN) is used to extract lexical and sentence level features [13]. This method takes all of the word tokens as input without complicated syntactic or semantic pre-processing. These word tokens are then transformed into vectors by looking up word embeddings. Meanwhile, sentence level features are learned using a convolutional approach. A max-pooled convolutional neural network is used to offer sentence level representation. This CNN automatically extracts sentence level features. These lexical and sentence level features are concatenated to form the final extracted feature vector. Finally, these features are fed into a softmax classifier to predict the relationship between two marked nouns. A CNN for extracting sentence level features and a CNN for extracting relation between entities achieves state-of-the-art performance on the SemEval-2010 Task 8 dataset [19]. In the network, position features (PF) are used to specify the pairs of nominals. The system obtained a significant improvement when considering the position features. The automatically learned features yielded excellent results and replaced the elaborately designed features that are based on the outputs of existing NLP tools. The architecture of neural network for relation classification is shown in Fig.2.

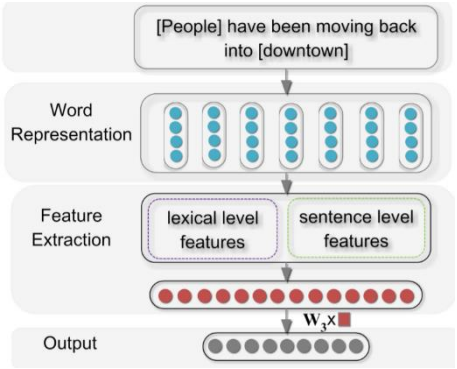

Fig.2. Architecture of CDNN used for relation classification [13]

- *Classification by Ranking using CNN (CR-CNN):* A method for relation extraction using CNN that performs classification by ranking is proposed in [8]. A new pairwise ranking loss function is proposed to reduce the impact of artificial classes (for example, the Other class in SemEval 2010 task 8 dataset). As in [13] input to the network is a tokenized sentence. The first layer of the model transforms words into real-valued feature vectors. The important features considered are word embeddings, word position embeddings and class embeddings. The convolutional layer constructs a distributed representation of sentence, r_x. In the final step, the CR-CNN computes a score for each class c, by performing a dot product between r_x and W^c, where W^c is an embedding matrix whose columns encode the distributed vector representations of the different class labels. The architecture of *CR-CNN* is shown in Fig.3. CR-CNN, we outperform the state-of-the-art for this dataset and achieve a F1 of 84.1 without using any costly handcrafted features. CR-CNN is more effective than CNN followed by a softmax classifier. Both precision and recall of the system was improved by omitting the representation of the artificial class *Other*. Using only the text between target nominals is almost as effective as using word position embeddings is demonstrated in [8].
- *Attention based CNN*: Attention based CNN model makes full use of word embedding, part-of-speech tag embedding and position embedding information for the task of relation extraction. A word-level attention mechanism to select relevant words with respect to the target entities is proposed in [14]. The attention model consists of heterogeneous models that are a sentence and two entities. Out of the four features, the learned position embedding features were effective for relation classification task. The proposed word level attention mechanism is able to better determine which parts of the sentence are most influential with respect to the two entities of interest. The architecture of attention based network is shown in Fig.4. The word attention mechanism to quantitatively model such contextual relevance of words with respect to the target entities. The weight of each word in the sentence is calculated by feeding each word and each entity in the sentence a to a multilayer perceptron (MLP).

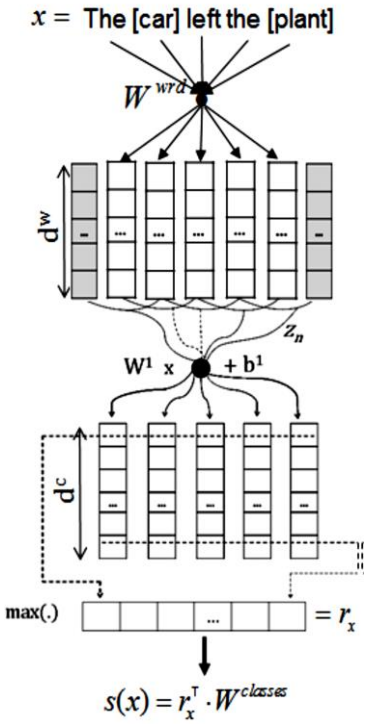

Fig.3. Architecture of CR-CNN for relation extraction [8]

- *Multi-Level Attention CNNs*: Multi-level attention mechanism is proposed in [15]. This multi-level attention

CNN enables end-to-end learning from task-specific labelled data, forgoing the need for external knowledge such as explicit dependency structures. Multi-level attention mechanism means applying attention in multiple layers. Multi-level attention is used to capture both entity-specific attention (primary attention at the input level, with respect to the target entities) and relation-specific pooling attention (secondary attention with respect to the target relations). This allows it to detect subtler cues despite the heterogeneous structure of the input sentences, enabling it to automatically learn which parts are relevant for a given classification. The results show that this simple but effective model is able to outperform previous work relying on substantially richer prior knowledge in the form of structured models and NLP resources [15].

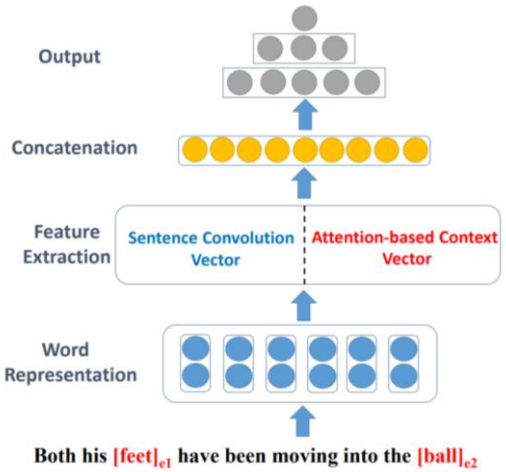

Fig.4. Architecture of attention based CNN [14]

4.1.2 RNN based Models (LSTM Models):

A particular shortcoming of CNN is that it is less powerful in modelling long-span relations [20]. Since the RNN model can deal with long-distance patterns, LSTMs will be suitable for learning long-span relations. Variants of LSTM network models are used for relation extraction task.

Step 1: *Bi-directional LSTMs (Bi-LSTM):* An inevitable challenge in Relation Extraction is that the important information can appear at any position in the sentence. Bi-directional LSTM networks are used to model the sentences with complete sequential information about all words before and after it. The overview of the methodology is as follows:

Step 2: *Initial Feature Extraction*: Extract feature such as word, POS tags, and hypernyms as features from input sentence.

Step 3: *Feature Embedding*: Represent initial extracted features as the real valued vectors.

Step 4: *Bi-LSTM based Sentence Level Representation*: Represent the feature representation from step 2, into a more abstract level.

Step 5: *Constructing Feature Vector*: Concatenate the feature vectors obtained from step 2 and step 3 to form a final feature vector.

Step 6: *Classification*: Classify relation from the pre-defined relation set. This final feature vector is fed into Multi-layer perceptron and softmax layer to get the probability distribution of relation types [21].

In addition to the features used in [21], word position feature is used by omitting the hypernym feature in [22].

- *LSTM on Sequences and Tree Structures*: An end-to-end neural model for relation extraction is pro-posed by Miwa et.al [23]. This method uses a recurrent neural network based model that captures both word sequence and dependency tree substructure information, by stacking bidirectional tree-structured LSTM-RNNs on bidirectional sequential LSTM-RNNs. The model mainly consists of three representation layers: a word embeddings layer (embedding layer), a word sequence based LSTM-RNN layer (sequence layer), and finally a dependency subtree based LSTM-RNN layer (dependency layer).

Step 1: *Embedding layer*: The embedding layer is responsible for embedding representations of words, part-of-speech (POS) tags, dependency types, and entity labels.

Step 2: *Sequence layer*: The word sequence in a sentence using representations obtained from embedding layer using bidirectional LSTM-RNNs.

Step 3: *Dependency layer*: The dependency layer represents a relation between a pair of entities in the dependency tree, and is in charge of relation specific representations.

This method uses a bidirectional tree structured LSTM-RNNs (i.e., top-down and bottom-up) to represent a relation candidate by capturing the dependency structure around the target word pair. This method employs three dependency models Shortest Path (SP) tree, SubTree and FullTree. SubTree is the part of dependency tree with common ancestor of the entity pair. Full tree is the full dependency tree. SubTree provides additional modifier information to the path and the word pair in SPTree where FullTree provides the entire context of the sentence. The important inference obtained is that selection of the appropriate tree structure representation of the input (i.e., the shortest path) is more important than the choice of the LSTM-RNN structure on that input (i.e., sequential versus tree-based) [23].

- *Attention based Bi-directional LSTM*: Bi-directional LSTM with attention mechanism is proposed in [24]. Attention based Bi-directional LSTM networks (Att-BLSTM) intended to capture the important semantic information at any position in the sentence. This model contains five components: Input layer, Embedding layer, LSTM layer, Attention layer, and Output layer. This paper describes BLSTM with attention mechanism, which can automatically focus on the words that have decisive effect on classification, to capture the most important semantic information in a sentence, without using extra knowledge and NLP systems [24]. Att-BLSTM model yields an F1-score of 84.0%. It outperforms most of the existing competing approaches, without using lexical resources such as WordNet or NLP systems like dependency parser and NER to get high-level features.

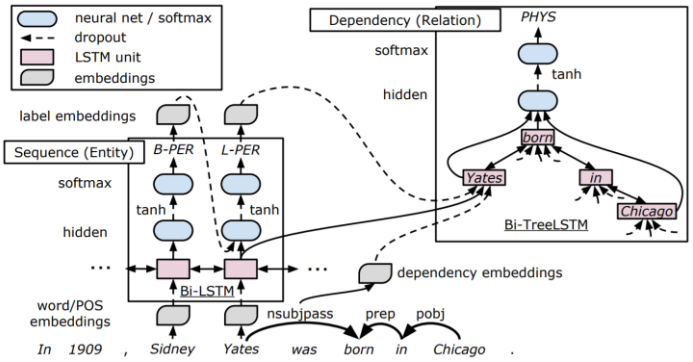

Fig.5. Architecture of end-to-end relation extraction model, with bidirectional sequential and bidirectional tree-structured LSTM-RNNs [23]

4.2 DEPENDENCY BASED RE MODELS

- *Dependency based Neural Network (DepNN)*: This paper proposes a new structure, termed augmented dependency path (ADP), which is composed of the shortest dependency path between two entities and the subtrees attached to the shortest path [20]. Architecture of DepNN is shown in Fig.6. The dependency-based framework where two neural networks are used to model shortest dependency paths and dependency subtrees separately. One convolutional neural network (CNN) is applied over the shortest dependency path, because CNN is suitable for capturing the most useful features in a flat structure. A recursive neural network (RNN) is used for extracting semantic representations from the dependency subtrees, since RNN is good at modelling hierarchical structures. To connect these two networks, each word on the shortest path is combined with a representation generated from its subtree, strengthening the semantic representation of the shortest path. In this way, the augmented dependency path is represented as a continuous semantic vector which can be further used for relation classification. The DepNN, is taking advantages of both convolutional neural network and recursive neural network.

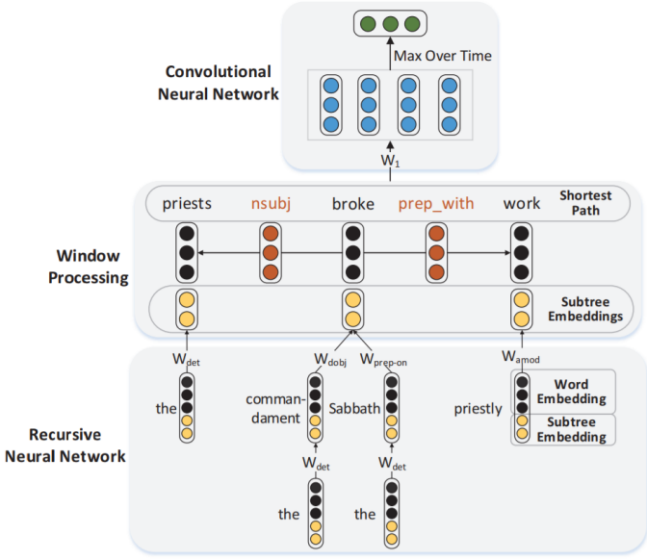

Fig.6. Illustration of dependency based neural network for relation extraction [20]

- *LSTM along with Shortest Dependency Paths (SDP-LSTM)*: A novel architecture of LSTM with shortest dependency paths to classify relationships between entities in a sentence is proposed in [9]. The proposed model has several distinct features: (1) The shortest dependency paths retain most relevant information (for relation classification), while eliminating irrelevant words in the sentence. (2) The multichannel LSTM networks allow effective information integration from heterogeneous sources over the dependency paths. (3) A customized dropout strategy regularizes the neural network to alleviate overfitting [9]. The overall architecture of the SDP-LSTM model is shown in Fig.7. The input sentence is parsed to a dependency tree, and then the shortest dependency path is extracted as the input. Four additional information such as word embeddings, POS tags, grammatical relations, and WorldNet hypernyms are considered in this work. Two Recurrent Neural Networks are used to capture the right and left sub tree information. LSTM units are used with Recurrent Neural Networks to propagate in-formation effectively. The max pooling layer gathers all the information from LSTM nodes. Finally, a softmax output layer is used for relation classification [9].

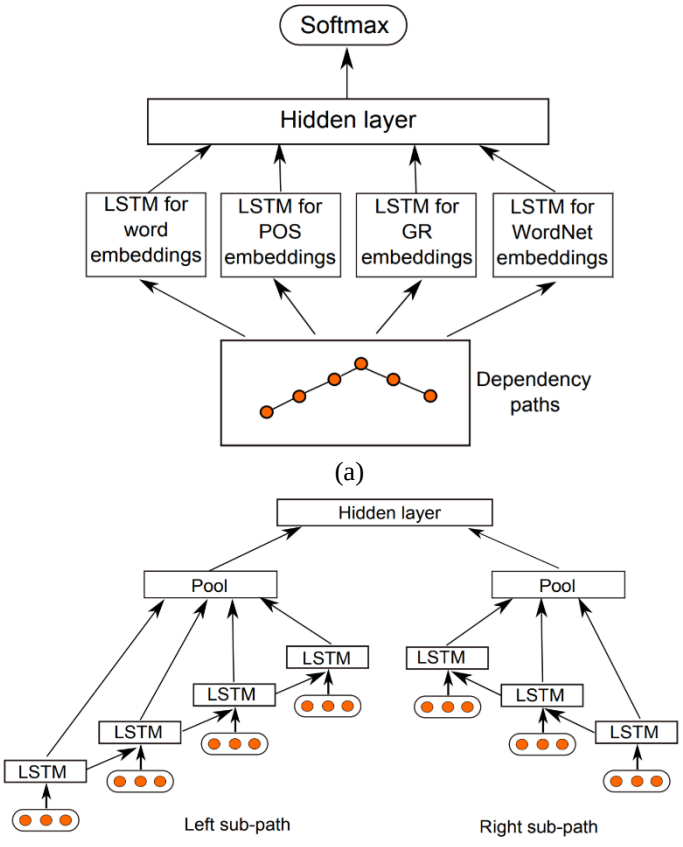

Fig.7. (a) Overall architecture of SDP-LSTM. (b) One channel of the recurrent neural networks built upon the shortest dependency path. The channels are words, part-of-speech (POS) tags, grammatical relations (abbreviated as GR), and WordNet hypernyms [9]

- *CNN with Simple Negative Sampling*: Unlike [13] and [15], this model [16] considers the shortest dependency path between two nominals e_1 and e_2 which describes the relationship. This is because if e_1 and e_2 are arguments of the same predicate, then their shortest path should pass through that predicate. Suppose if e_1 and e_2 belong to different predicate-argument structures, their shortest path will pass through a sequence of predicates, and any consecutive predicates will share a common argument [16]. The input to the model is the shortest dependency path from subject to object. This input passes through a look up table layer. Here each word and label in the dependency path is transformed into a vector by looking up in the embedding matrix $w_e \in \Re^{dx|V|}$ where d is the dimension of vector and V is the set of all nodes or words contain the relation mention. The feature vectors produced locally around each node are combined into a global feature vector. Finally, this feature vector is forwarded to a softmax layer for relation classification. The objective function for the training data is

$$J(\theta) = -\sum_{x}\sum_{k=1}^{K} t_k(x)\log d_k(x) + \lambda \| \theta \|^2 \qquad (1)$$

where $\theta = (W_e, W_1, W_2, W_3)$ is the set of model parameters to be learned, is a vector of regularization parameters, $T_k(x)$ is the probability based on dependency path of k^{th} relation, and $d_k(x)$ is the probability distribution of k^{th} class[16].

The dependency based representation is used to learn the assignments of subjects and objects. The opposite assignments of subjects and the objects are treated as negative samples. The dependency path of correct and incorrect assignments will be different and it can be crucial information to the model, in order to distinguish between the subject and object. This can be done by simply feeding the negative samples to the model and let the model to learn the correct assignments of subjects and objects. This method makes use of dependency path and captures the syntactic features for relation extraction.

- *Deep Recurrent Neural Networks (DRNNs)*: A deep recurrent neural networks (DRNNs) to classify relations is proposed in [25]. The deep RNNs can explore the representation space in different levels of abstraction and granularity. By visualizing how RNN units are related to the ultimate classification, we demonstrate that different layers indeed learn different representations: low-level layers enable sufficient information mix, while high- level layers are more capable of precisely locating the information relevant to the target relation between two entities. The overall architecture of DRNN is illustrated in Fig.8. Two recurrent neural networks pick up information along the shortest dependency path, separated by its common ancestor. Four information channels, namely words, part-of-speech tags, grammatical relations (GR), and WordNet hypernyms are used for capturing these features [25].

  In the relation classification task, words along SDPs provide information from different perspectives. On the one hand, the marked entities themselves are informative. On the other hand, the entities' common ancestor (typically verbs) tells how the two entities are related to each other. Such heterogeneous information might necessitate more complex machinery than a single RNN layer. When evaluated on the SemEval-2010 Task 8 dataset, the DRNNs model results in substantial performance boost. The performance generally improves when the depth increases; with a depth of 4, this model has the highest F1-measure.

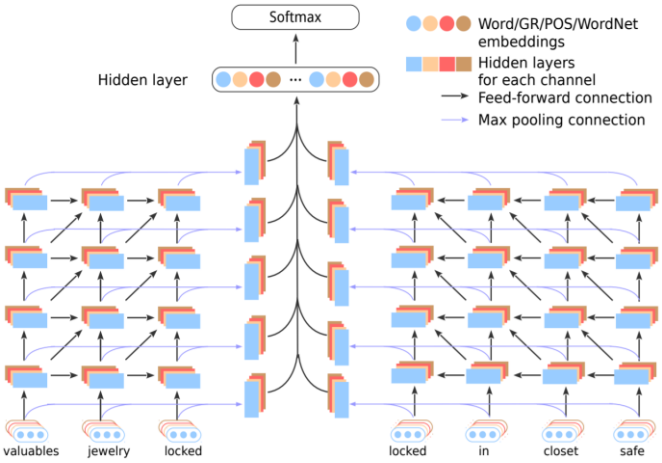

Fig.8. Overall architecture of DRNNs. Two recurrent neural networks pick up information along the shortest dependency path, separated by its common ancestor. This channel uses four information channels, namely words, part-of-speech tags, grammatical relations (GR), and WordNet hypernyms [25]

- *Bi-directional Recurrent CNN*: This paper explores how to make full use of the dependency relations information in the SDP, by combining convolutional neural networks and two-channel recurrent neural networks with long short term memory (LSTM) units [26]. A bidirectional architecture to learn relation representations with directional information along the SDP forwards and backwards at the same time, which benefits classifying the direction of relations. The first contribution is that recurrent convolutional neural network(RCNN) is proposed to encode the global pattern in SDP utilizing a two-channel LSTM based recurrent neural network and capture local features of every two neighbor words linked by a dependency relation utilizing a convolution layer. Given a sentence and its dependency tree, a neural network is built on its SDP extracted from the tree. Along the SDP, two recurrent neural networks with long short term memory units are applied to learn hidden representations of words and dependency relations respectively. A convolution layer is applied to capture local features from hidden representations of every two neighbour words and the dependency relations between them. A max pooling layer thereafter gathers information from local features of the SDP or the inverse SDP. A softmax output layer after pooling layer for classification is used in the unidirectional model RCNN [26]. Bi-directional RCNN (BRCNN) is built on the basis of RCNN with SDP and inverse SDP of a sentence as its input. The BRCNN model, consisting of two RCNNs, learns features along SDP and inversely at the same time. RCNN achieves a better performance at learning features along the shortest dependency path, compared with some common neural networks. A significant improvement is observed when BRCNN is used, outperforming state-of-the-art methods. The architecture of BRCNN is illustrated in Fig.9.

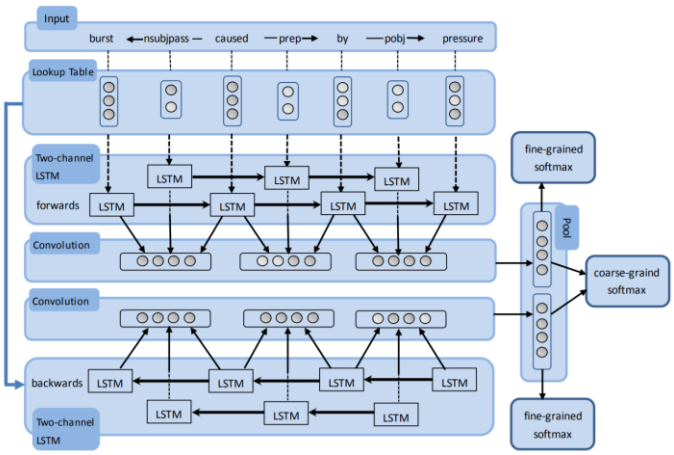

Fig.9. Overall architecture of BRCNN. Two-Channel recurrent neural networks with LSTM units pick up information along the shortest dependency path, and inversely at the same time. Convolution layers are applied to extract local features from the dependency units [26]

4.3 DISTANTLY SUPERVISED RE MODELS

The unavailability of huge dataset will lead to manual construction of datasets for training and testing a model. In order to avoid this problem, Mintz et al propose an alternative paradigm, known as Distant Supervision which does not need annotated data [27]. The distant supervision avoids the domain dependence of data. The approach of distant supervision takes advantages of both supervised and unsupervised paradigms. It combines many features using a probabilistic classifier as in the case of supervised learning. While, it extracts a large number of relations from large corpora of any domain as in the case of unsupervised learning. The idea is to use a large knowledge base to obtain relation labels automatically. Mintz et al. [27] used Freebase [28] as a knowledge base which keeps pairs of entities for various relations. Using knowledge base for labelling relations can be noisy due to the wrong labelling problem. Sometimes a sentence that mentions two entities does not necessarily express their relation as in a knowledge base. For example, consider the two sentences:

1) Steve Jobs was the co-founder and CEO of Apple and formerly Pixer.
2) Steve Jobs passed away the day before Apple unveiled iPhone 4S in late 2011.

Here, the first sentence expresses the relation "company/founder", while the second one does not hold the relation, though the same entity mentions appear in the sentence. The second sentence is wrongly labelled as "company/founder" and it will be considered as training data. Another problem with distant supervision is that it failed to model overlapping relations. Consider the same pair of entities, there can be multiple valid relations like founded by (Steve Jobs, Apple) and CEO (Steve Jobs, Apple). Multi-instance Multi-label (MIML) learning proposed by Hoffmann et al. [29] and Surdeanu et al. [18]. MIML for relation extraction uses a novel graphical model for representing multiple labels for multiple instances.

- *Distant Supervision via Piecewise Convolutional Neural Networks*: This paper [29] proposed a novel model that uses Piecewise Convolutional Neural Network (PCNN) with multi-instance learning to address the aforementioned problems. For wrong labeling problem, the distant supervised relation extraction is treated as a multi-instance problem similar to [18] and [17]. In multi-instance problem, the training set consists of many bags, and each contains many instances. The labels of the bags are known; however, the labels of the instances in the bags are unknown. During the training process, the uncertainty of instance labels is taken into account and thus alleviates the wrong labelling problem. The second problem is addressed by convolutional architecture to automatically learn relevant features without complicated NLP pre-processing as in [15]. To capture structural and other latent information, this method divides the convolution results into three segments based on the positions of the two given entities and devise a piecewise max pooling layer instead of the single max pooling layer [29]. PCNN with multi-instance learning outperforms the CNNs and PCNNs without multi-instance learning.
- *Distant Supervision with Sentence-Level Attention and Entity Descriptions*: A sentence-level attention model to select valid instances is proposed in [30], which makes full use of the supervision information from knowledge bases. The extraction of entity descriptions from Freebase and Wikipedia pages are used to provide background knowledge for the relation extraction task. The background knowledge provides more information for predicting relations and brings better entity representations for the attention module [30]. Sentence-level attention identifies and selects multiple valid instances for training, thus making full use of the supervision information.

5. DISCUSSIONS

All the approaches except distantly supervised approaches discussed in this paper use SemEval-2010 Task 8 dataset for evaluation. The evaluation metric used to compare reviewed papers is *F1-measure*. The aim of the reviewed papers is to predict the relation between a pair of entities in a text. Therefore, the performance analysis of the models is based on the fact that, from the predicted set of relations how many relations predicted by the system are correct. This will be measured by

$$Recall = \frac{correctly\_predicted\_relations}{predicted\_relations} \quad (2)$$

At the same time, the accurate prediction of existing relationship between a pair of characters is also important. Among the set of relationship between entities existing in the text, how much of the relations are correctly predicted by the system need to be considered. This will be evaluated by

$$Precision = \frac{correctly\_predicted\_relations}{actual\_relations} \quad (3)$$

Considering the importance of both precision and recall, the models took F1-measure as the evaluation metric.

$$F1 = 2 \times \frac{precision \times recall}{precision + recall} \quad (4)$$

F1-score is a better measure if the system needs a balance between precision and recall. Performance of the reviewed

systems on two datasets SemEval-2010 Task 8 and New York Times corpus are as follows.

5.1 SEMEVAL-2010 TASK 8 DATASET

SemEval-2010 Task 8 focuses on multi-way classification of semantic relations between pairs of nominals. The task was designed to compare different approaches to semantic relation classification and to provide a standard test bed for future research [19]. This dataset consists of 10,717 samples, in which 8000 for training and 2717 for testing. The nine relations of the SemEval-2010 Task 8 are cause-effect, instrument-agency, product-producer, content-container, entity-origin, entity-destination, component-whole, member-collection, and message-topic. Most of the methods discussed in this paper for relation extraction use the SemEval-2010 Task 8 dataset for training and testing. The Table.1 gives the performance (F1 Score) of different relation classification systems on SemEval-2010 Task 8 dataset. Multi-level attention mechanism applied to CNN shows better performance on SemEval-2010 Task 8 dataset. Most of the methods experimented with word embedding only as a feature for relation classification, and obtained state-of-the-art results. To obtain more accurate results, CNN models use other features such as word position embeddings, words around nominals, wordnet, hypernym-hyponym features, and word pairs. Using additional features improved the performance of relation extraction models, when comparing with their baseline model (Model which uses word embeddings only for training). LSTM networks are capable of learning dependencies in sequence prediction problem. The combination of convolutional neural networks with recurrent neural network have better performance in relation classification task. Models which consider dependency feature as a feature can have better results. Word position embeddings plays an important role in relation classification. When wpord position embeddings are considered along with word embeddings improved the performance of most of the models. Adding attention layers to the model also have an impact in the performance of the relation extraction systems.

Table.1. Performance of Relation Classification System on SemEval 2010 Task 8 Dataset

Method	**Feature Set**	**F1 Score**
SVM [13]	POS, WordNet, prefixes and other morphological features, dependency parse, Levin classes, PropBank, FrameNet, NomLex-Plus, Google n-grams, paraphrases, TextRunner	82.2
CNN + Softmax [13]	word embedding Word embedding, word pair, word around nominals, wordnet	78.9 82.7
CR-CNN [8]	word embedding word embeddings, word position embeddings	82.8 84.1
Att. CNN [14]	Word embedding word embedding, wordnet, words around nominals	84.3 85.9
Multi Att. CNN [15]	word embedding word embedding, word position embedding	82.8 87.5*
Bi-LSTM [21]	word embedding word embedding, position embedding, POS, NER, wordnet Synset, dependency parse	82.7 84.3
Bi-LSTM on Sequences and Tree structures [23]	Wordnet + SP tree	85.5
Att. Bi-LSTM [24]	Word embedding, word position embedding	84.0
DepNN[20]	Word embedding Word embedding, NER	83.0 83.6
SDP –LSTM [9]	Word embeddings Word embeddings, POS embeddings, WordNet embeddings, grammar relation embeddings	82.4 83.7
CNN-NS [16]	Word embedding Word embedding, WordNet,words around nominals	84.0 85.6
DRNN [25]	Word+POS+GR+WordNet embeddings w/o data augmentation + data augmentation	84.2 86.1
BRCNN [26]	Word embeddings + POS, NER, WordNet embeddings	85.4 86.3

(* Best performance among the models compared)

The convolutional neural network with multi-level attention mechanism has better performance when compared with the other models. The F1-measure of the system is 87.5. Bi directional recurrent convolutional neural network also has an F1-score of 86.1.

5.2 NEW YORK TIMES CORPUS

SemEval-2010 Task 8 is a small dataset for relation extraction. Deep learning models require huge datasets for training. These huge datasets can be built using distant supervision. The standard corpus for distantly supervised relationship extraction is the New York Times (NYT) corpus, published in [32]. This dataset contains text from the New York Times annotated corpus, in which named entities are extracted using Stanford NER system [33] and these entities are automatically linked to entities in the Freebase Knowledge base. Different papers have used different metrics for evaluation, making it difficult for direct comparison of systems. This review uses plots of precision and recall for evaluating the performance of distant supervision systems. The performance of various models using distant supervision is shown in Fig.10.

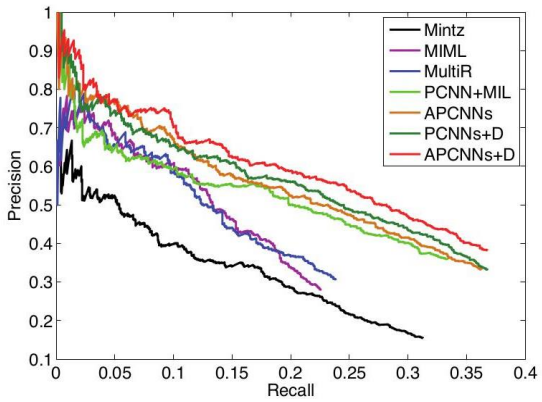

Fig.10. Performance analysis of distantly supervised models [30]

The Multi-Instance Multi-Label (MIML) mechanism with CNN improves over the performance when compared with MIML. Selective attention mechanism applied on PCNN model shows better performance while PCNN model with sentence attention mechanism and entity description has the best performance. Distant supervision is a good practice if there is no annotated corpus. The annotation is done with the help of an external knowledge base in the case of distant supervision. A large corpus can be built using distant supervision, which is needed for deep learning.

6. CONCLUSIONS

Convolutional Neural Networks and Long Short Term Memory Networks yield state-of-the-art results, by only considering word embeddings as a feature. LSTMs are better in handling long-span relations than CNNs. Multi-Instance Multi Label mechanism for relation extraction jointly models all the instances of a pair of entities in text and all their labels using a graphical model with latent variables. Combination of recurrent neural networks and convolution neural networks has good F1-Score, indicating that the model is good at relation classification. CNN with multi-level attention mechanism has the best performance in the task of relation extraction with an F1-score, 87.5. The surveyed relation extraction methods deal with small relation sets. But a large number of relations can exist between a pair of entities. The existing relation extraction methods focus on identifying and extracting relations between a pair of entities within in a sentence. Future work includes extending the scope of relation extraction between a pair of entities across sentences with a pre-defined relation set with large number of relations.

REFERENCES

[1] Sachin Pawar, Girish K. Palshikar and Pushpak Bhattacharyya, "Relation Extraction: A Survey", *Proceedings of International Conference on Computation and Language*, pp. 1-51, 2017.

[2] Marc Moreno Lopez and Jugal Kalita, "Deep Learning applied to NLP", *Proceedings of International Conference on Computation and Language*, pp. 1-15, 2017.

[3] Yan Lecun, Yoshua Bengio and Geoffrey Hinton, "Deep Learning", *Nature*, Vol. 521, pp. 436-444, 2015.

[4] Yoshua Bengio, Rejean Ducharme, Pascal Vincent and Christian Jauvin, "A Neural Probabilistic Language Model", *Journal of Machine Learning Research*, Vol. 3, pp. 1137-1155, 2003.

[5] Omer Levy, Yoav Goldberg and Ido Dagan, "Improving Distributional Similarity with Lessons Learned from Word Embeddings", *Transactions of the Association for Computational Linguistics*, Vol. 3, pp. 211-225, 2015.

[6] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado and Jeffrey Dean, "Distributed Representations of Words and Phrases and their Compositionality", *Proceedings of 26th International Conference on Neural Information Processing Systems*, Vol. 2, pp. 3111-3119, 2013.

[7] Jeffery Pennington, Richard Socher and Christopher D. Manning, "GloVe: Global Vectors for Word Representation", *Proceedings of International Conference on Empirical Methods in Natural Language Processing*, pp. 1532-1543, 2014.

[8] Cicero Dos Santos, Bing Xiang and Bowen Zhou, "Classifying Relations by Ranking with Convolutional Neural Networks", *Proceedings of 7th International Conference on Natural Language Processing*, pp. 626-634, 2015.

[9] Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng and Zhi Jin, "Classifying Relations via Long Short Term Memory Networks along Shortest Dependency Paths", *Proceedings of International Conference on Empirical Methods in Natural Language Processing*, pp. 1785-1794, 2015.

[10] Katrin Fundel, Robert Kuffner and Ralf Zimmer, "RelEx: Relation Extraction using Dependency Parse Trees", *Bioinformatics*, Vol. 23, No. 3, pp. 365-371, 2007.

[11] David H. Hubel and Torsten N. Wiesel, "Receptive Fields and Functional Architecture of Monkey Striate Cortex", *Journal of Physiology*, Vol. 195, No. 1, pp. 215-243, 1968.

[12] CS231n: Convolutional Neural Networks for Visual Recognition, Available at: http://cs231n.github.io/convolutional-networks/

[13] Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou and Jun Zhao, "Relation Classification via Convolutional Deep Neural Network", *Proceedings of 25th International Conference on Computational Linguistics: Technical Papers*, pp. 2335-2344, 2014.

[14] Yatian Shen and Xuanjing Huang, "Attention-Based Convolutional Neural Network for Semantic Relation Extraction", *Proceedings of 26th International Conference on Computational Linguistics: Technical Papers*, pp. 2526-2536, 2016.

[15] Linlin Wang, Zhu Cao, Gerardde Melo and Zhiyuan Liu, "Relation Classification via Multi-Level Attention CNNs", *Proceedings of 26th International Conference on Computational Linguistics: Technical Papers*, pp. 1298-1307, 2016.

[16] Kun Xu, Yansong Feng, Songfang Huang and Dongyan Zhao, "Semantic Relation Classification via Convolutional Neural Networks with Simple Negative Sampling", *Proceedings of International Conference on Empirical Methods in Natural Language Processing*, pp. 536-540, 2015.

[17] Sepp Hochreiter and Jrgen Schmidhuber, "Long Short-Term Memory", *Neural Computation*, Vol. 9, No. 8, pp. 1735-1780, 1997.

[18] Mihai Surdeanu, Julie Tibshirani, Ramesh Allapati and Christopher D Manning, "Multi-Instance Multi-Label Learning for Relation Extraction", *Proceedings of Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 455-465, 2012.

[19] Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva and Preslav Nakov, "SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations Between Pairs of Nominals", *Proceedings of 5th International Workshop on Semantic Evaluation*, pp. 333-338, 2010.

[20] Yang Liu, Furu Wei, Sujian Li, Heng Ji, Ming Zhou and Houfeng Wang, "A Dependency-Based Neural Network for Relation Classification", *Proceedings of 53rd Annual Meeting of the Association for Computational Linguistics and 7th International Joint Conference on Natural Language Processing*, pp. 285-290, 2015.

[21] Shu Zhang, Dequan Zheng, Xinchen Hu and Ming Yang, "Bi-Directional Long Short-Term Memory Networks for Relation Classification", *Proceedings of 29th Pacific Asia Conference on Language, Information and Computation*, pp. 73-78, 2015.

[22] Menglong Wu, Lin Liu, Wenxi Yao, Chunyong Yin and Jin Wang, "Semantic Relation Classification by Bi-Directional LSTM Architecture", *Advanced Science and Technology Letters*, Vol. 143, pp. 205-210, 2017.

[23] Makoto Miwa and Mohit Bansal, "End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures", *Proceedings of 54th Annual Meeting of the Association for Computational Linguistics*, Vol. 1, pp. 1105-1118, 2016.

[24] Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao and Bo Xu, "Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification", *Proceedings of 54th Annual Meeting of the Association for Computational Linguistics*, pp. 207-212, 2016.

[25] Nanyun Yan Xu, Ran Jia, Lili Mou, Ge Li, Yunchuan Chen, Yangyang Lu and Zhi Jin, "Improved Relation Classification by Deep Recurrent Neural Networks with Data Augmentation", *Proceedings of 26th International Conference on Computational Linguistics*, pp. 1-10, 2016.

[26] Rui Cai, Xiao dong Zhang and Houfeng Wang, "Bidirectional Recurrent Convolutional Neural Network for Relation Classification", *Proceedings of 54th Annual Meeting of the Association for Computational Linguistics*, pp. 756-765, 2016.

[27] Mike Mintz, Steven Bills, Rion Snow and Dan Jurafsky, "Distant Supervision for Relation Extraction without Labelled Data", *Proceedings of 4th International Conference on Natural Language Processing***,** pp. 1003-1011, 2009.

[28] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor, "Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge", *Proceedings of ACM International Conference on Management of Data*, pp. 1247-1250, 2008.

[29] Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer and Daniel S Weld, "Knowledge-based Weak Supervision for Information Extraction of Overlapping Relations", *Proceedings of 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 541-550, 2011.

[30] Daojian Zeng, Kang Liu, Yubo Chen and Jun Zhao, "Distant Supervision for Relation Extraction via Piecewise Convolutional Neural Networks", *Proceedings of International Conference on Empirical Methods in Natural Language Processing*, pp. 1753-1762, 2015.

[31] Guoliang Ji, Kang Liu, Shizhu He and Jun Zhao, "Distant Supervision for Relation Extraction with Sentence-Level Attention and Entity Descriptions", *Proceedings of 31st International Conference on Artificial Intelligence*, pp. 1132-1139, 2017.

[32] Sebastian Riedel, Limin Yao and Andrew McCallum, "Modeling Relations and their Mentions without Labeled Text", *Proceedings of European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 148-163, 2010.

[33] Jenny Rose Finkel, Trond Grenager and Christopher Manning, "Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling", *Proceedings of 43rd Annual Meeting of the Association for Computational Linguistics*, pp. 363-370, 2005.