Neural Relation Extraction

Swapnil Gupta, Karan Malhotra, Shubham Bansal
M.Tech, IISC Bangalore
Instructor: Dr. Partha Pratim Talukdar
{swapnilgupta.229,kmkaran212,shu09bei}@gmail.com

Abstract

In this report, we have analysed two of the most prominent neural models, PCNN and Bi-GRU with word attention, that have been shown to be very effective for the of Relation Extraction. In literature, both the models have been separately used for the task and combined as an ensemble. A key observation we found during the course of project is that PCNN being a window based method effectively captures representation of each word in a local context whereas Bi-GRU which captures the global representation of the word in the context of the entire sentence. Hence, both are having complementary properties. While PCNN captures syntax aware word representations from local windows, the global representation from Bi-GRU are more semantic in nature. In this project we have experimented with novel ways to combine these two models in an attempt to explore the effectiveness of the above interpretation. The key focus of the project is to work with the relation classification sub-task in relation extraction. For this project, SemEval-2010 Task 8 benchmark dataset (10) has been used for experimentation. On this dataset, (1) is the present state of the art with F-1 score of 85.9.

1 Introduction

Many tasks in natural language understanding require an understanding of the semantic relations between entities. Relation extraction (RE) is defined as the task of extracting semantic relations between given entity pairs from plain text.

For example, in the sentence "The **[fire]** inside WTC caused by exploding **[fuel]**", the entities **[fire]** and **[fuel]** are in a **Cause- Effect** relationship.

There is a considerable interest in automatic relation classification, both as an end in itself and as an intermediate step for many NLP applications such as Information extraction, Document summarization and automatic enrichment of existing knowledge bases.

Supervised methods based on neural networks have been found successful for relation extraction tasks (1),(2). But such methods require large-scale manually-constructed corpus, which is expensive to create. Recently, distant supervision has gained a lot of attention which automatically produce training corpus (11). In distant supervision, we simply label all the sentences containing entity e1 and e2 by their corresponding relation present in the knowledge base.

Given an entity pair $(e^{'}, e^{"})$ from a knowledge base such as Freebase, assuming that the predefined semantic relation on the KB is *r*, we simply label all sentences containing the two entities by label *r*. However, it has a major shortcoming that the distant supervision assumption is too strong and may cause the wrong label problem. It is possible that these two entities may simply share the same topic or maybe representing some other relation than that present in knowledge-base.

In order to address this problem, (12) proposed to model distant supervision as Multi-Instance Multi-Label (MIML) problem. It assumes that in a set of sentences corresponding to an entity pair, at least one sentence in that set should express the true relation assigned to the set. In a more recent work, (9) proposed sentence attention model to deal with the challenges associated with distant supervision.

Conventionally, the task of relation extraction is divided into two distinct parts. First, is to extract candidate sentences and entity pairs from a

plain text between whom we want to identify the relations and the second task is the relation classification task. In this work, our main focus is to perform relation extraction under supervised settings. In this project we have worked with two neural models, PCNN and Bi-GRU following implementation details given in (4). Taking a step further, we have proposed a novel interpretation to these two models and based on that interpretation proposed better hybrid models by combining these two models.

The report is organized as following. In section 2 we have briefly overviewed the relevant work. In the following section 3, we have described the baseline models used and details of our combination models. In section 4, experimentation and results are discussed and finally the report is concluded in section 5.

2 Related Work

As an important and fundamental task in NLP, relation extraction has been studied extensively. Initially in (5), handcrafted features were extracted using pre-existing tools but that lead to propagation of errors in the existing tools and hinders the performance of the system. Further, kernal-based methods (6) were also introduced which automatically generate features.

Recently, the neural network models have dominated the work of Relation Extraction because of higher performances. (2) Zeng et al., 2014 used a convolutional deep neural network (DNN) to extract lexical and sentence level features. This work also proposed the inclusion of position features for the task of relation extraction. These two levels of features are concatenated to form the final extracted feature vector. Softmax classier is used to predict the relationship between two marked entity. In an extension to this, (1) Yatian Shen, Xuanjing Huang, 2016 proposed a attention-based convolutional neural network architecture for this task that makes full use of word embedding, part-of-speech tag embedding and position embedding information. Further to incorporate distant supervision, (8) Zeng et al., 2015 combined the multi-instance learning with piecewise convolutional neural networks to learn more relevant features. (9) Lin et al., 2016 employed CNN with sentence-level attention over multiple instances to encode the semantics of sentences. (7) Miwa and Bansal, 2016 used a syntax-tree-based long short-term memory networks (LSTMs) on the sentence sequences. Further, (4) Sharmistha Jat et al., 2017 proposed a weighted ensemble model of a BiGRU-based word attention model and EA, an entity-centric attention model. The current state-of-art model for relation extraction is proposed in (3) Zhengqiu He et al., 2018 centered on the ideas of using tree-GRU based syntax aware embeddings.

In our work, we have adapted the implementation as described in (4) Sharmistha Jat et al., 2017 making changes wherever felt necessary. Further building upon these models we have built our hybrid model by combining the two models.

3 Proposed Methods

In this project, we have experimented with several models, starting with some which have been proposed in the literature and shown to be effective for the task of relation extraction and following them, we have implemented some novel combination models to boost the performance. As our component models, we have used a piece wise convolution model as proposed in (8) and a Bi-GRU word attention model as proposed in (4). This section contains the description of all these models.

All our models takes a sentence as input and learns useful representations of the sentence building on the initial representation that we feed into it. The initial representation is composed of two components: *Word vector representations, position features* as described below. This is inline with most recent research in the field.

3.1 Input Representations

The training data includes raw word tokens, but they can not be directly applied on neural nets. So, the words are transformed into a low dimension space using pre-trained word embedding lookup. Moreover, (2) have shown position features of each word in the sentence with respect to the entity locations to be effective for the task. Hence, position features are also added using position embedding and the final embedding for a word is concatenation of both the embeddings.

3.1.1 Word Embedding

The word embeddings helps to map words to a distributed k-dimension representation. These embeddings captures syntactic and semantic information of tokens and using these embeddings and

updating them while training has become common process for various nlp classification tasks. We have intitalized our word embeddings with pretrained word embedding and these are treated as model parameters and tuned during model training. The embeddings are pretrained using Word2Vec (13). The pretraining task aims at capturing the similarity between words based on the assumption that similar words occur in similar neighbourhood.

3.1.2 Postion Embedding

A Position feature is defined as the combination of the relative distances from the current word to entity1 and entity2. Two position embedding matrices similar to (2) are randomly initialized and the relative distances are transformed into vectors by looking in these two matrices. The aim of these features is to incorporate in our model the information that which word in the sentence are actually our target nouns/entities between which we want to find the relation using the context given by the sentence.

$$d_p(position\ embedding) = d_1 + d_2$$

$$d(word\ vector) = d_w + d_p$$

3.2 Piece-Wise CNN

A key challenge in the task of Relation Extraction is that of variable length of the sequences. Under such conditions, Convolution neural network along with pooling layers gives effective solutions to learn sentence representation. Conventionally, a max pool layer over all the tokens of the words is used for the task of sentence classification. But, this is not very effective in the context of relation extraction due to presence of entities have much more structure to it which needs to be effectively captured. Our framework needs to give special consideration to these words. Based on the above motivation (8) proposed a piece wise max pooling layer following the output of the convolution layer. The intuition behind using a piece wise CNN is to capture both internal and external context. The internal contexts consists the tokens between the two entities and the external contexts consists of the tokens around the entities. Assume a sentence consists of n words and each word is a d-dimensional vector i.e $x_i(R^{dx1})$. The representation of every word becomes w_i after convolution layer which is of dimension $N \times 1$.

N= number of filters
The sentence is segmented into three parts:
$s1 = [w_i.....w_{e1-1}]$, $s2 = [w_{e1}.....w_{e2}]$,
$s3 = [w_{e2+1}.....w_n]$

Now the piece wise max pooled sentence representation is :
$s_p = [max(s1), max(s2), max(s3)]$
s_p will be of dimension $3N \times 1$

Figure below shows the architecture of the PCNN model:

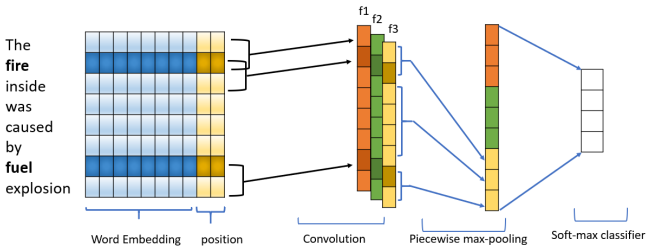

Figure 1: Model Architecture of PCNN

The hyper parameters for this model are listed below:
1.Number of filters
2.Filter Size

3.2.1 PCNN with Entity attention

(1) proposed incorporation of entity context vectors along with the CNN. The entity context vectors are formed using word level attention mechanism to determine which parts of the sentence are most influential with respect to the two entities of interest.

Let a sentence consist of k words $[w_1, w_2, ..., w_k]$ where each of w_i represents the corresponding embedding and $e1$ and $e2$ represents the embedding of the entities present in the sentence. Now, for entity attention, first, we find the concatenated embeddings given by c_{ij} where i represents the i^{th} word embedding and j represents j^{th} entity where $i \in [1, 2.....k]$ and $j \in [1, 2]$.
Now u_{ij} represents the entity-specific attention score for each word. u_{ij} is calculated as below:

$$c_{ij} = [w_i, e_j]$$

$$u_{ij} = c_{ij} \times r_j$$

where, r_j is the attention weights which are learned from the model. Further, we use softmax

to normalize u_{ij} for $j \in [1, 2]$ and finally we obtain attention weights a_{ij}. Now, we take the weighted average over all the words for each of the $j \in [1, 2]$ entity which gives us the entity context vector:

$$s_j = \sum_{i=0}^{k} a_{i,j} * w_i$$

Finally, we concatenate both the entity context vectors with PCNN sentence representation. Now, this is the representation which we use in this model to find the relation.

3.3 Bi-GRU based word attention

To highlight the motivation behind this model, consider the following example "The **fire** inside was *caused* by **fuel** explosion". In this example the word *cause* helps in identifying the relation *Cause-Effect* between the two entities in the above sentence.

The above example illustrates that in each sentence there are some key words which are more important in bringing out the relation between the entities. To capture this relative importance of the words, bidirectional Gated Recurrent Units (Bi-GRU) based word attention models have been effectively used in several applications. Moreover (Bi-GRU) cells captures long range structural dependencies, and thereby give better context embedding for a sentence. Figure 2 below shows the architecture corresponding to this model.

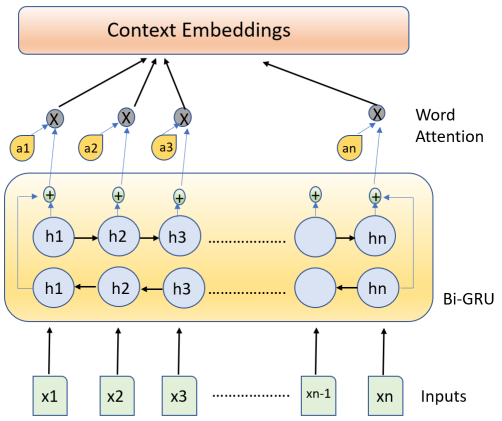

Figure 2: Model Architecture of BiGRU with word Attention

Assume a sentence consists of n words and each word is a d-dimensional vector i.e $x_i(R^{dx1})$. The representation output of the word is concatenation of both forward (h^f_i) and backward state (h^b_i) vector of the Bi-GRU each of length g/2 .The u_i is degree of relevance defined as:

$$w_i = [h_i^f h_i^b]; u_i = w_i \times A \times r$$
$$a_i = softmax(u_i); w_i' = a_i \times w_i$$

A is $g * g$ square matrix and $r(R^{gx1})$ is relation vector and both are learned. w'_i is the attention weight multiplied updated word representation of w_i.

Then we apply piece wise average pooling according to the entities position same as done in PCNN model. And the last stage is a single layer softmax classifier.

The model has a single hyper-parameter, the no. of hidden units in the state vector.

3.4 Combination Models

As discussed above, both PCNN and Bi-GRU models have complementary strengths. PCNN which learns word representations through a local window based context captures syntactic properties of the word whereas Bi-GRU captures long-term dependencies in the words capturing semantic representation of each word. The aim of this section is to experiment with different ways to combine these models.

3.4.1 Ensemble Model

Our first attempt is to make an ensemble model. Here, both the models independently compute the scores for each class which we get by passing the respective sentence representations through a dense layer, and we combine the two models by taking a weighted sum of the output scores. Where all the weight vectors for both the models and for all the classes are learnt in the training process. Now the softmax layer is applied to these ensembelled output scores to determine the relation in the sentence.

The weighted output can be found as follows:

$$o_e = o_p * \alpha 1 + o_g * \alpha 2$$

where o_e (ensembled output), o_p (PCNN output), o_g (GRU output), $\alpha 1$ and $\alpha 2$ are of dimension R^{nx1} and * represents element wise dot product between them. Then, we apply softmax to the ensembled output.

$$o_m = softmax(o_e)$$

o_m is considered as the final output of our model.

3.4.2 Local Global Word Representation Model

Next, we discuss a novel combination model which follows naturally from our above discussions in support of combination models. Since, the two models contrast the most in the way they compute the word representations. Hence, in this model we combined the two models at the word representation level. The idea is the combination gives us better word representations on which we process further. The figure 3 below gives an architecture of our local global word representation model:

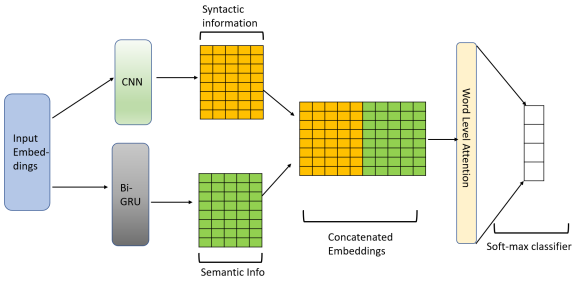

Figure 3: Model Architecture of local global word representation model

When an input sentence is passed to Bi-GRU and CNN both returns representation of sentence in a matrix in which i^{th} row is the representation of the i^{th} word in the sentence. The representations returned by both the models are concatenated at word level and then they are sent to a word attention layer for learning attention weights for the words. After the attention layer the sentence is segmented into three segments based on entities and the segments are pooled and then concatenated as being performed in PCNN. The last stage is Softmax Classifier.

3.4.3 Sequential Layer Combination

We tried two more combination models. Here we have sequentially combined the two model layers. In any multi-layer network, the aim of the first layer is to modify the input data in a way that the next layer gets a better representation of the input and next layer further modifies this representation before we apply the softmax layer. We have experimented with both the layer sequence. First one is a PCNN built on Bi-GRU and the second one is a Bi-GRU word attention model built on CNN layer.

A comparative analysis of all the models used is presented in section 4 below.

4 Experimental Results

4.1 Dataset

Our experiments in relation extraction are based on the SemEval-2010 Task 8 dataset (10). The dataset is freely available and contains 10,717 annotated examples, including 8,000 training instances and 2,717 test instances. There are 9 relationships (with two directions) and an undirected other class. The following are examples of the included relationships: Cause-Effect, Component-Whole and Entity-Origin.

4.2 Hyper-parameter settings

For all our models we have initialized word embeddings with pre-trained word vectors d_w of length 50. The dimension of position embeddings d_1 and d_2 corresponding to each entity is set at 5 and they are initialized from random normal distribution. This is a standard practice which is followed in all the relevant works and hence these parameters are directly taken without any validation. Now, final embedding d for each word in a sentence will be of length 60.

$$d_w = 50$$

$$d_p = d_1 + d_2 = 10$$

$$d = d_w + d_p = 60$$

The parameters used for various models are mentioned below:

- **Piece-Wise CNN:** Number of filters used are 230, context window size is 3 and is followed by piece wise max-pooling. To avoid overfitting dropout layer with probability of 0.5 is being used on max-pooling output. Final representation of sentence is of dimension 690.

- **Bi-GRU word attention model:** Number of hidden units in each GRU cell is 100 and GRU cell dropout probability is kept at 0.5. Output of BiGRU layer is of R^{nx200} , where n is the length of sentence. Attention layer is applied to this output and piecewise average pooling is done to obtain final representation of sentence of dimension 600.

Remaining combination models are built using the parameters of above mentioned individual models. Also, we have used L2-regualrization to update word embeddings, position embeddings, convolution layer filter weights, attention layer weights

and weights which connect sentence level representation to input of softmax. This regularization constant has been kept as 0.05.

4.3 Precision-Recall Curve

A standard practice followed in all the relevant literature is to use Precision-Recall curve as the performance metric. P-R curve is a useful measure of success of prediction when classes are imbalanced. Precision is a measure of result relevancy and recall is the measure of how many true relevant results are being returned. The P-R curve shows the trade-off between precision and recall for different value of threshold.Basically the P-R curve is plotted in case of 2 class classification but it can be easily generalized for multi-class by considering each element of the label indicator matrix as a binary prediction(micro averaging). The following figure shows the PR curve plotted for all the models built to solve the given task:

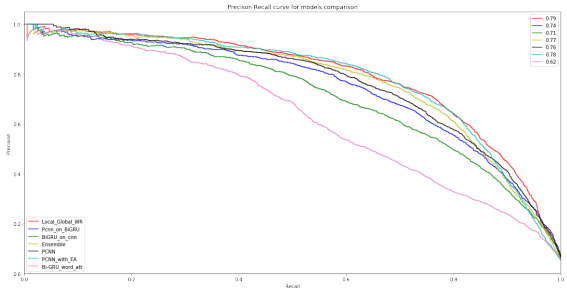

Figure 4: PR curves for model comparison

A high area under precision recall curve denotes both high recall and high precision which shows that the classifier is returning accurate results(high precision) as well as returning a majority of all positive results(high recall).
In the Figure 4 it is seen that the area under the curve for the Local_Global_WR which corresponds to our Local Global Word Representation Model is maximum which shows that this model is the best among all. In the Figure 4 top right box shows the average precision of all the models which is directly related to the accuracy of the models built. Though unexpectedly, the improvement over the baseline PCNN model is only marginal. This can be attributed to less effective training procedure since in combination models the no. of training parameters almost get doubled. But still we believe the idea of perceiving CNN as a syntactic feature extractor and Bi-GRU as a semantic feature extractor is significant and can be effectively used in several domains.

4.4 F1-Score

F1-score which is weighted harmonic mean of Precision and Recall is used as metric to evaluate the performance of models built.
F1-Score:

$$\frac{2 * Precision * Recall}{Precision + Recall}$$

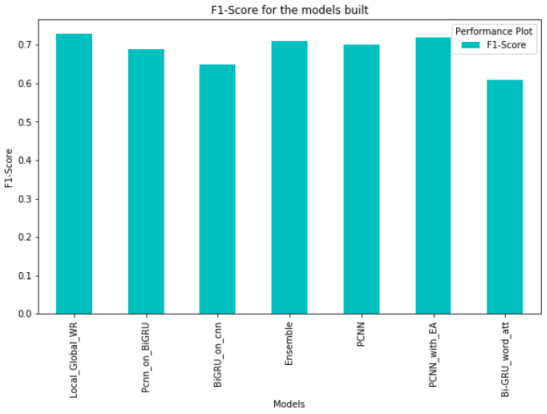

Figure 5: F1-Score for models

It can be inferred from the above figure that the Local_Global_WR model and PCNN with entity attention gives the best performance among all, if only F1-score measure is considered.

4.5 Result gradation table

Neural network models for relation extraction under supervised settings suffer from overfitting problem and may not generalize well due to lack of availability of large manually constructed corpus. Our models also faced similar issues. Hence, to avoid such problems, we used various techniques such as regularization, xavier initialization of weights and pretrained word embeddings. Also, using both word embeddings and position embeddings yielded better performance. Table 1 represents the F1-score performance for our Piecewise CNN model with such techniques being incorporated into our model one after other.

A key conclusion that can be drawn from the above experiment is that a model is only as effective as its tuning. Hyper-parameter tuning and proper regularization sometimes give better boost to the performance than going for more advanced models.

Gradation table	
gradation	F1-Scores
word embeddings	.46
word+pos embeddings	.54
word+pos embeddings + regularization	.63
pretrained word vectors + pos embeddings + regularization	.71

Table 1: F-1 scores of our PCNN model incorporating various techniques

4.6 Visualizing Word Attention

As discussed above the aim of word attention is to let the model identify which words in a sentence are more relevant to our classification and accordingly give more consideration to them.

The below figures demonstrates how effectively our model has been able to capture these salient features in a sentence through attention mechanism

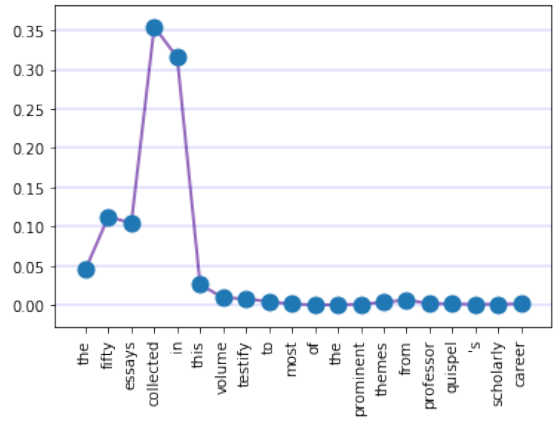

Figure 6

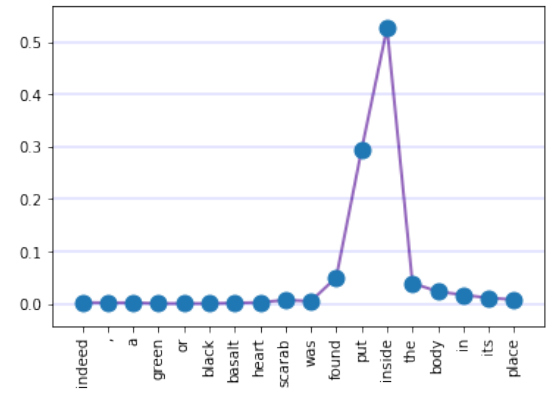

Figure 7

In Figure 6 the sentence contains a relation of *Member-Collection* between the entity pairs and *essays* and *volume* and our model is rightly giving the most attention to the word *collected*. While in Figure 7 the sentence contains a relation of *Content-Container* between the entity pairs *scarab* and *body* and our model is rightly giving the most attention to the phrase *put inside*.

5 Conclusion

In this project, we worked on the task of relation extraction. In the first glance, the task of relation extraction seems like any other sentence classification task. But as mentioned above, the presence of entities in each sentence adds much more salient features to the task. These salient features motivated the rise of ideas like position embeddings and piece-wise pooling in the research community. As our base models, we implemented two of the most successful models in the field namely PCNN and Bi-GRU with word attention. Observing the complementary strengths of both the models, we have experimented with some novel ideas in combining the two models. Our most successful model combines the syntactic and semantic word features we generate from CNN and Bi-GRU models respectively. Which we believe can be used in several other domains as well.

The project included several challenges. Key challenges include working with variable length input, avoiding over-fitting in a high dimensional neural model while training with a small dataset. Capturing the salient structural properties in a sentence which can aid our performance in the sentence specific manner.

This project is our first major exposure to apply deep learning techniques to an ongoing research problem and we had several key takeaways from this experience. First, we developed an understanding and comfort in using deep learning tools like tensorflow. Second, we gained practical incites on good practices to follow while applying deep learning models. And lastly, this experience of working on an ongoing research problem helped us to build a research acumen.

6 Acknowledgement

We would like to express gratitude to our course instructor, Dr. Partha Pratim Talukdar for giving us this opportunity to work on this project. The project gave us the opportunity to develop a bet-

ter understanding of all the theoretical concepts taught in class and a great research experience in a growing field of Natural Language. We strongly believe this exposure would be highly beneficial for our future research endeavours.

7 Source code

Source codes for our models implementation can be obtained from https://github.com/karanMal/Neural_Relation_Extraction

References

[1] Yatian Shen, Xuanjing Huang, "Attention-Based Convolutional Neural Network for Semantic Relation Extraction", in Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics.

[2] Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou and Jun Zhao, "Relation Classification via Convolutional Deep Neural Network", in Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics.

[3] Zhengqiu He, Wenliang Chen, Zhenghua Li, Meishan Zhang, Wei Zhang, Min Zhang, "SEE: Syntax-aware Entity Embedding for Neural Relation Extraction"

[4] Sharmistha Jat, Siddhesh Khandelwal, Partha Talukdar, "Improving Distantly Supervised Relation Extraction using Word and Entity Based Attention" in 6th Workshop on Automated Knowledge Base Construction (AKBC) at NIPS 2017, Long Beach, CA, USA.

[5] Nanda Kambhatla. 2004, "Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations" In Proceedings of the ACL on Interactive poster and demonstration sessions. Association for Computational Linguistics.

[6] Min Zhang, Jie Zhang, and Jian Su. 2006, "Exploring syntactic features for relation extraction using a convolution tree kernel" In Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics. Association for Computational Linguistics.

[7] Miwa, M., and Bansal, M. 2016, " End-to-end relation extraction using lstms on sequences and tree structures" In Proceedings of ACL, 11051116.

[8] Zeng, D.; Liu, K.; Chen, Y.; and Zhao, J. 2015, " Distant supervision for relation extraction via piecewise convolutional neural networks" In Proceedings of EMNLP, 17531762.

[9] Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016, " Neural relation extraction with selective attention over instances" In Proceedings of Association for Computational Linguistics, 2016.

.

[10] Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid O. S eaghdha, Sebastian Pad o, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010, " Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals" In Proceedings of the 5th International Workshop on Semantic Evaluation, SemEval 10, pages 3338.

[11] Mintz, M.; Bills, S.; Snow, R.; and Jurafsky, D. 2009, " Distant supervision for relation extraction without labeled data" In Proceedings of ACL, 10031011.

[12] Surdeanu, M.; Tibshirani, J.; Nallapati, R.; and Man-ning, C. D. 2012, " Multi-instance multi-label learning for relation extraction" In Proceedings of EMNLP, 455465.

[13] Tomas Mikolov et al. 2013, "Distributed Representations of Words and Phrases and their Compositionality"