Relation Extraction Using Sense Embeddings

Ahmed Elshinawi

October 15, 2018

1 Introduction

There exists a vast amount of unstructured data, including newswire, web, blogs, Email communications, governmental documents, logs, and so on. Hence the need to develop systems that help us to understand all of this data. One popular approach is to turn unstructured text into structured text via annotating semantic information. However, the huge volume and heterogeneity of data make manual annotation almost impossible. Instead, we would like to have a machine system proficient to annotate all data efficiently. Development of such systems requires it to understand relations between entities, such as persons, organizations, and locations. More formally Information extraction (IE) is the process of extracting information from text, which is often also referred to as text analytics. Such a process begins by extracting the entities followed by extracting their relationship such as located in, employed_by, part_of, married_to, ... The applications of IE are vast and is still considered an open research problem by the natural language processing community. Relation extraction plays a vital role in extracting structured information from unstructured sources such as raw text. For example, we may want to find interactions between drugs to build a medical database, shopping engines & product search, build scholarly literature databases: Google Scholar, CiteSeerX or extract relationships among people to build an easily searchable knowledge base.

As mentioned above the process starts with first finding names in the text and then classifying them by type. It is a tagging task very similar to part-of-speech (POS) tagging. The Standford (CRF-NER) [5] can automatically label data with high accuracy and is considered to be one of the best approaches for named entity recognizers (NER). The NER nowadays is commonly thought of as a solved problem. Existing relation extraction systems focus on extracting binary relations [6]. However, relation extraction methods developed over the years are subdivided into (a) Hand-built patterns, (b) Bootstrapping methods, (c) Supervised methods, (d) Distant supervision, (e) Unsupervised methods). This study project will focus on distant supervision methods for relation extraction using Multi-instance Multi-label learning framework as well as non-distant supervision approaches. Distant supervision for relation extraction (RE) is a very effective method in gathering training data by aligning a database of facts with text; it is an efficient approach to scale RE to thousands of different relations as compared to the traditional supervised approach where each example is explicitly mapped to a single label [7]. Multi-label Multi-instance training (MLMI) is a governing framework for labeled data such that the labels are not mutually exclusive.

The aim of this research project to develop a Neural Network architecture, in particular, a Convolutional Neural Network (CNN) and train it on RE datasets. Our proposed CNN model will be able to process dataset build from Wikipedia and Freebase using distant supervision methodology for relation extraction. The CNN model has to be able to incorporate any number of embedding layers such as Adaptive Skip-gram [2], Google Word2vec[4], sensegram [11] etc. The model also has to be able to count for MLMI framework. Our proposed model will also be able to process SemEval 2010 [9] dataset for RE.

2 Motivation & Contribution

"The coexistence of many possible meanings for a word or phrase" is called polysemy [1].

Vector-space models (VSM), despite their usefulness, share a common problem that each word is only represented with one vector, which clearly fails to capture homonymy and polysemy. Similar work by the authors in [1] touches upon the importance of Unsupervised word representations. In recent years, Neural Networks has gained increasing interest in the research community for performing various NLP tasks.

The novelty of our work is the implementation of a Neural Network architecture incorporated with an adaptive embedding layer that has the ability to represent polysemy. Our research is driven by the following main question:

"Could Improving Word Representations by learning different word representation contrary to current existing embedding models improve relation extraction task ?"

3 Related Work

This chapter introduces the background knowledge required for this scientific research work. It provides all the relevant information necessary to comprehend the approach proposed in this research study. This section explains the computational components forming the models like convolutional neural network (CNN) [13]. The current section aims to introduce the reader to CNN, relation extraction tasks in general and utilizing CNN for relation extraction as well.

3.1 Relation Extraction Approaches

The relation extraction task has applied several learning paradigms for extracting relational facts from texts, for example, learning that a person is employed by a particular company, or that a city is located in a particular country, etc. In the supervised methods, sentences in a corpus are labeled for the presence of entities and the relations between them. However, supervised relation extraction suffers from several problems. Acquiring labeled data for training is expensive to produce. An alternative approach is to use unsupervised paradigm, where the system extracts strings of words between entities in large amounts of text and performs clusters simplifying these word strings to produce relation-strings [14]. Although unsupervised approaches can use a very large volume of data and extract large numbers of relations, but the resulting relations may not be easy to map to relations needed for a particular knowledge base. A third approach is to use a very small number of seed instances to do bootstrap learning [15]. These seeds are utilized usually with a large corpus to extract a new set of patterns, which are used to extract even more instances, which are used to extract more patterns, all this process takes place in an iterative manner. The resulting patterns often suffer from very low precision and semantic drift [16]. A fourth approach is to use distant supervision, that combines some of the advantages of each of the previously discussed approaches. Distant supervision is often regarded as an extension of the paradigm used by Snow et al. [17] for exploring WordNet to extract hypernym (is-a) relations between entities. Distant supervision employees Freebase [18] a large semantic database, to provide distant supervision for relation extraction. Free-base contains 118 million instances of 7,500 relations between 12 million entities. The core idea of distant supervision is to extract any sentence that contains a pair of entities that are found in a known Freebase relation which is likely to express that relationship in some way. Since there might be many sentences containing a given entity pair, we can extract very large numbers of features that are combined in a logistic regression classifiers. Another interesting approach is the MLMI (Multi-Label learning Multi-Instance) framework. In the MLMI approach, an example is described by multiple instances and associated with multiple class labels. Compared to traditional learning frameworks, the MLMI framework is more convenient and natural for representing complicated objects which have multiple semantic meanings. The MLMI learning studies the problem where a real-world object described by one instance is associated with a number of class labels, which is different from multi-class learning because in multi-class learning each object is only associated with a single label. As explained in the previous section, in this work we use distance supervision as well as supervised learning approaches.

3.2 Convolutional Neural Network

Convolutional Neural Networks (CNNs / ConvNets) are feedforward Neural Networks that are made up of synaptic neurons that have learnable weights and biases. The convolutional neural networks are shift invariant due to its shared-weights architecture and translation invariance characteristics. In 1980, a researcher called Fukushima proposed the hierarchical neural network model which he named neocognitron [13]. The neocognitron was a hierarchical, multilayered artificial neural network which showed the ability of handwritten character recognition and other pattern recognition tasks, and served as the inspiration for convolutional neural networks. The convnet showed great success in classifying digits from hand-written numbers [13]. Many other variations of convnets have been introduced over the years, but they all do have the same set of layer structure with the only variation in layer sizes and the number of layers, for example, Inception-v3, Inception-v4, LeNet5, VGG-19, etc. The convolutional networks neurons connectivity pattern resembles the organization of the animal visual cortex, and thus are said to be biologically inspired [13].

3.2.1 ConvNets Architecture

Convolutional Neural Networks have a slightly different architecture than most feedforward neural networks. The Input layers of ConvNets are organized into width, height, and depth. Each input neuron receives the input usually an image or text embedding as in text analysis context and performs a dot product and optionally follows it with a non-linearity. Finally, the final output will be reduced to a single vector of probability scores. Similar to other neural networks models, convNets have a sequence of layers, and every layer of a ConvNet transforms activations to another through a differentiable function [13]. A convNet is usually composed of Convolutional Layer, Pooling Layer, and Fully-Connected Layer. Stacking these layers forms a full ConvNet architecture. In contexts like deep-reinforcement learning, max-pooling layers are not used in forming the convNet architecture so not to lose the features spatial position information as this information is vital in detecting states for the agent during the decision phase [13].

An Overview of ConvNets layers functionality:

- **Convolutional layers** This layer is responsible for applying convolution operation to the input, the layer parameters are made up of a set of learnable filters also called kernel, during the forward pass the kernel performs a convolutional operation on the input where it computes a dot product. As the kernel slides over the input, it will produce a 2-dimensional activation map also called feature map [13]. Basically, the convNet will activate when it detects low-level visual features such as contours, edges, etc.

- **Pooling Layer** These type of layers are usually inserted between successive Conv layers in a ConvNet. [13] Its primary function is to reduce the spatial size of the network representation to reduce the number of parameters and computation within the network, which in turns allow controlling overfitting. The pooling layers perform reduction by combining the outputs of the neurons into a single neuron in the next layer. For example, max pooling takes the maximum value from each cluster of neurons at the previous layer; other pooling units can perform other functions such as average where their operation takes the average value of the cluster of neurons [13].

- **Fully Connected** Neurons that exist in a fully connected layer have full connections to every neuron in another layer. This same principle is also applied in the multi-layer perceptron neural network (MLP) [13].

- **Weights** ConvNets share the same weights across convolutional layers.

ConvNets have a lot of variations that have been developed over the years as presented in [13].

3.3 Embeddings Layers

This section offers a very brief introduction to the embedding layers used during this research task.

Adagram The Adaptive Skip-gram model is a nonparametric Bayesian extension of Skip-gram with the capability to automatically learn the required number of representations for all words at desired semantic resolution. The adaptive skip gram model (Adagram) will serve as one of the embeddings layers for our proposed model architecture.

Sensgram Sensgram [10] is an effective approach that doesn't learn directly sense representations from corpora or from lexical resources but learns through clustering of ego-networks of the existing word embeddings.

JoBimText JoBimText [11] is an open source platform for large-scale distributional semantics based on graph representations, its aim is to address issues like lexical ambiguity and variability, word sense disambiguation.

Glove Glove (Global Vectors for Word Representation) [12] is an unsupervised learning algorithm that offers vectorized representations for words. Glove provides vector representation in two forms a 300d (dimension) and 100d.

Word2vec Word2vec [4] has initially been created by a group of researchers led by Tomas Mikolov at Google. Word2vec produces a vector space, usually of several hundred dimensions (300d), with each unique word in the corpus being assigned a corresponding vector in the newly created space. Word2vec has two model architectures to produce a vectorized representation of words: continuous bag-of-words (CBOW) or continuous skip-gram. In the continuous bag-of-words, the model predicts the current word from a window of surrounding context words. The order of context words does not influence prediction (bag-of-words assumption). In the continuous skip-gram architecture, the model uses the current word to predict the surrounding window of context words. The skip-gram architecture weighs nearby context words more heavily than more distant context words.

3.4 CNN for relation extraction

Deep learning has been applied to many NLP tasks. Rather than building hand-craft features, DNN (Deep Neural Network) builds features by automatic learning, fitting different domains well including relation extraction as well. Convolution network, for instance, has shown great performance by incorporating lexical features it learned from text [19]. The authors approach [19] built an end-to-end DNN network that first encodes the given input text sentence using word vectors and lexical features, which is then followed by a convolutional kernel layer and a softmax output layer to yield a probability distribution over all the related classes. The model used synonym vectors instead of word vectors, by assigning a single vector to each synonym class rather than giving individual word a vector. However, it fails to exploit the real representational power of word embeddings. The authors [20] utilized CNN to extract lexical and sentence level features by looking up transformed word tokens to vectors by looking up word embeddings. Then as a next step, the lexical level features are extracted according to the given nouns. In this approach, the sentence level features are learned using a convolutional approach. These extracted level features are concatenated to form the final extracted feature vector. Finally, the features are fed into a softmax classifier to predict the relationship between two marked nouns. In this research work our approach shows similarity to the previously described methods except instead of using CNN to perform convolution on the word embedding of Word2vec, we perform convolution on the sense embeddings.

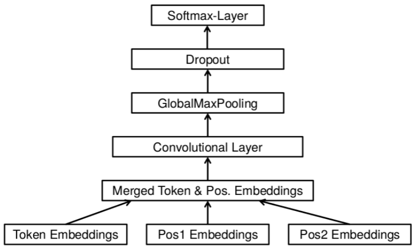
4 Methodology

This section illustrates the architectural setup of the models created during this research project.

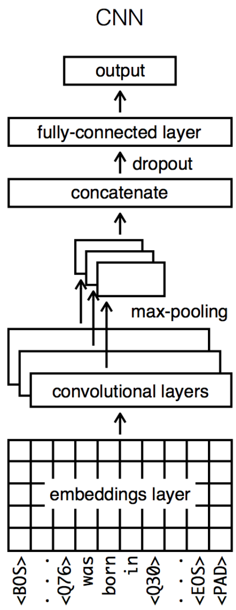
4.1 Proposed Model

We have developed two variations of our proposed model for the purpose of our research work, where both the model variations share the same architectural configuration except for the processing layers. Our first learning architecture consists of Convolutional neural network (CNN) [13] that allows for incorporating the embedding representations from several embeddings such as Adaptive Skip-gram model [2] and Word2vec [4]. etc. as well as it takes into consideration the MLMI representation that is produced to the model by the distant supervision approach. We have used this architectural representation when performing experiments on the Wikipedia dataset we built using distant supervision. Furthermore, details are explained in the next sections.

Our second learning architecture also consists of the same Convolutional neural network (CNN) [13] architecture that allows for incorporating the embedding representations from several embeddings such as Adaptive Skip-gram model [2] and Word2vec [4] etc. We have utilized the second architecture to perform experiments on the SemEval task-8 as detailed in the next section.



(a) Our proposed model with processing layer for Semeval



(b) Our proposed model with processing layer for MLMI

Figure 1: Our proposed models

4.2 mfs vs comb vs word embeddings

The section will introduce necessary notations that are important to look upon for the experiments section. (1) mfs: most frequent sense, (2) comb: combined senses. As explained above (see section 3.3) the sense embeddings have several representation for the same word for example the word apple in word2vec representation is in the form of a word and vector representation, however the word apple in sensegram has two different senses (while also being vectorized) as in the sense of apple the company and apple the fruit. As per our above-laid model architecture, we utilize the model ability to convolute over the word2vec representation and as well the various sense embeddings. In order to perform extensive experiments on the sense embeddings, we have utilized them in two forms. One form is that we have only picked up the most frequent sense (mfs) of a particular word and provided that to the model as a first layer. We have recorded our model performance during the experiments and provided its results in the experiments section (see section 6). In the second set of our conducted experiments we have utilized the second form of the different sense embeddings, we have used their combined senses (comb). Basically, we have used all the different senses for each and every word within our corpus and provided that as input to our model as explained earlier. The conducted experiments and their results are also presented in the experiments section.

5 Datasets and Evaluation

The section introduces the datasets used during this research project. First, the chosen datasets are presented in more detail for a well-suited understanding of the problem. Then, the different evaluation metrics utilized to evaluate the models created during the course of this work are discussed.

5.1 Datasets

We chose two datasets to evaluate our model. The Wikipedia dataset and SemEval 2010 task 8.

Wikipedia To test distant supervision Wikipedia dataset was our choice as in [3], a subset of English Wikipedia will be used for both training and evaluation. The process for building the dataset is performed in the following manner; we build the dataset by first downloading the raw text, then applying Standford NER [5] to find entities. After extracting all the possible combinations, we proceed to find the entity ID (using SPARQL on WikiData API) only at the end to find relation (using SPARQL on WikiData API) following the described process we ensure the dataset is build and ready for our model.

SemEval 2010 The second dataset chosen was the SemEval 2010 – Task 8 [9]. The dataset presents the text in the form of sentences and its target nominals for example (The [introduction] e1 in the [book] e2 is a summary of the text). Using SemEval dataset the task is then to classify the relationship between those target nominals, for example, consider the above example where the relation between the target nominals is (component-whole) [9].

5.2 Evaluation Metrics

The evaluation metrics that have been used to evaluate the proposed models are here discussed. To efficiently evaluate our proposed model a quantitative metric has been used. The quantitative evaluation metric was utilized to have an objective evaluation measure of our proposed architecture. The quantitative evaluation of our experimental setup is based on the classification accuracy information provided by utilizing the developed classifiers described in the previous sections (see section 4.1). In the quantitative evaluation, the classifiers are trained and tested on the described above datasets. In such classification tasks, a confusion matrix is widely used [8]. A confusion matrix is a matrix that is often used to report on the performance of a classification model, in our case, it will describe the performance of the developed classifiers. The confusion matrix is a performance measurement tool for machine learning models that depends on four main terminologies. The true positives TP_x (indicating are all samples that have been classified correctly as class x, while true negatives TN_x represent all samples that truly do not belong to class x and have been classified correctly as such. The false positives FP_x are the samples that have been wrongly classified as belonging to a different class also known as Type I error. The false negatives FN_x are the samples that do not belong to class x yet have been classified as members class x also known as TypeII error [8].

6 Experiments and Results

The aim of the experiments conducted is to identify the best performing embedding layer in the context of relation extraction. This section explains in detailed descriptions the conducted experiments and also presents the results of those experiments. All the experiments are explained in more details for reproducibility. Finally, visualizations are presented which allows for the analysis and interpretation of the proposed model's performance.

Several experiments were conducted to test our proposed models shown in figure 4.1, Experiment 1 aimed to evaluate the model performance in performing binary classification using the different embedding layers presented in the previous section (see section 3.3). While Experiment 2-MLMI was conducted to test the model performance of the model in predicting the relationship between two entities under the assumption of MLMI framework using the above-discussed embedding layers. Finally, Experiment 3 aimed to evaluate the model performance while performing a relation extraction task on the SemEval dataset task-8. To conduct experiment 1 and 2, the Wikipedia dataset has been used. The experiments description is discussed below:

6.1 Experiment 1

As seen in figure 4.1 our proposed CNN model allows to perform Binary classification of the existence of a relation between entities; namely, the positive label means that the entities have a specific relation, the negative label implies that the entities don't share that relation. Under this assumption, no MLMI framework is being used. This experiment was conducted on the Wikipedia dataset. The experiment aim was to evaluate the performance of CNN model in classifying two classes (Binary classification) using different embeddings layers.

6.1.1 Intrepation of Results of Experiment 1

Table 1: Experiment 1

	Precision	Recall	F1 score	Loss
RE-CNN G-word2vec	**0.9152**	**0.9174**	**0.9163**	0.3425
RE-CNN Glove 300d	0.9055	0.9080	0.8963	0.3212
RE-CNN Adagram_mfs	0.9126	0.9112	0.9119	0.3946
RE-CNN sensegram_mfs	0.8915	0.8979	0.8863	0.301
RE-CNN JoBimText_mfs	0.7508	0.7525	0.7205	0.380
RE-CNN Adagram_comb	0.5026	0.5202	0.5303	0.4146
RE-CNN sensegram_comb	0.5205	0.5189	0.5643	0.3519
RE-CNN JoBimText_comb	0.5305	0.5447	0.5549	0.4852

from the above comparison table 1, we can observe that several models show very close performance with slight deviation for the JoBimText.

6.2 Experiment 2-MLMI

The proposed model can also perform predictions on how likely a given relation is the correct relation for the given two entities. This experiment demonstrates the workings of the MLMI concept were the task is to predict the relationship between two entities. This experiment was conducted on the Wikipedia dataset. The experiment aim was to evaluate the performance of CNN model Figure 4.1 in predicting the relationship between two entities using different embeddings layers.

6.2.1 Intrepation of Results of Experiment 2

Table 2: Experiment 2

	Precision	Recall	F1 score	Loss
RE-CNN G-word2vec	**0.8491**	0.6391	0.7293	0.0819
RE-CNN Glove 300d	0.8355	0.6380	0.7163	0.0791
RE-CNN Adagram_mfs	0.8602	0.5870	0.6978	0.0873
RE-CNN sensegram_mfs	0.8055	**0.6780**	**0.7363**	0.1019
RE-CNN JoBimText_mfs	0.7912	0.6025	0.6963	0.1202
RE-CNN Adagram_comb	0.5342	0.5260	0.5568	0.1270
RE-CNN sensegram_comb	0.5435	0.5340	0.5343	0.1525
RE-CNN JoBimText_comb	0.5702	0.5585	0.5123	0.1402

from the above comparison table 2, we can observe that for this particular task the different embedding layers have near similar performance except for JoBimText.

6.3 Experiment 3

The proposed model performing relation extraction on the SemEval dataset task-8, during this experiment the embeddings described in section 3.3 are used as the model embedding layers.

6.3.1 Intrepation of results of Experiment 3

Table 3: Experiment 3

Emb. Layers	Accuracy	F1 score
RE-CNN G-word2vec	**0.7784**	**0.7469**
RE-CNN Glove 300d	0.7273	0.7106
RE-CNN Adagram_mfs	0.6261	0.5780
RE-CNN Sensgram_mfs	0.6699	0.6584
RE-CNN JoBimText_mfs	0.5742	0.5455
RE-CNN Adagram_comb	0.5541	0.5280
RE-CNN Sensgram_comb	0.5979	0.5879
RE-CNN JoBimText_comb	0.5432	0.5335

6.4 Error Analysis

This section is dedicated to present the error analysis for all the used embedding layers during the above-described experiments in terms of calculating known and unknown words. The numbers are rounded to the nearest percentage number.

Table 4: Error Analysis

Emb. Layers	UnKnown	Known
RE-CNN G-word2vec	10%	90%
RE-CNN Adagram	20%	80%
RE-CNN Sensgram	16%	84%
RE-CNN Glove 300d	13%	87%
RE-CNN JoBimText	31%	69%

7 Conclusion

This research project aimed to test the hypothesis that utilizing different embedding layers other than word2vec such sensegram, adagram, etc. might lead to better performance when performing relation extraction task. Thus to test this hypothesis quantitative evaluation has been conducted. The overall results show that word2vec is, in fact, outperforms all the other used embeddings. In conclusion, the focus of this study was on designing a learning architecture that is composed of a convolutional neural network and is able of processing several embeddings. Although we have been mainly focused on our work on relation extraction, we would like to try our proposed models in different domains in the future.

References

[1] Huang, E.H., Socher, R., Manning, C.D. and Ng, A.Y., 2012, July. Improving word representations via global context and multiple word prototypes. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1 (pp. 873-882). Association for Computational Linguistics, Stroudsburg, PA, USA.

[2] Bartunov, S., Kondrashkin, D., Osokin, A. and Vetrov, D., 2016, May. Breaking sticks and ambiguities with adaptive skip-gram. In Artificial Intelligence and Statistics (pp. 130-138), Cadiz, Spain.

[3] Surdeanu, M., Tibshirani, J., Nallapati, R. and Manning, C.D., 2012, July. Multi-instance multi-label learning for relation extraction. In Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning (pp. 455-465). Association for Computational Linguistics, Jeju Island, Korea.

[4] Mikolov, T., Chen, K., Corrado, G. and Dean, J., 2013. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

[5] Finkel, J.R., Grenager, T. and Manning, C., 2005, June. Incorporating non-local information into information extraction systems by gibbs sampling. In Proceedings of the 43rd annual meeting on association for computational linguistics (pp. 363-370). Association for Computational Linguistics, Stroudsburg, PA, USA.

[6] Bach, N. and Badaskar, S., 2007. A review of relation extraction. Literature review for Language and Statistics II, 2, Stroudsburg, PA, USA.

[7] Ji, G., Liu, K., He, S. and Zhao, J., 2017, February. Distant Supervision for Relation Extraction with Sentence-Level Attention and Entity Descriptions. In AAAI (pp. 3060-3066), San Francisco, California USA.

[8] Townsend, J.T., 1971. Theoretical analysis of an alphabetic confusion matrix. Perception & Psychophysics, 9(1), pp.40-50.

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

[10] Ustalov, D., Teslenko, D., Panchenko, A., Chernoskutov, M., Biemann, C. and Ponzetto, S.P., 2018. An Unsupervised Word Sense Disambiguation System for Under-Resourced Languages. arXiv preprint arXiv:1804.10686.

[11] Faralli, S., Panchenko, A., Biemann, C. and Ponzetto, S.P., 2016, October. Linked disambiguated distributional semantic networks. In International Semantic Web Conference (pp. 56-64). Springer, Cham, Kobe, Japan.

[12] Pennington, J., Socher, R. and Manning, C., 2014. Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543), Qatar.

[13] Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105), California, USA.

[14] Banko, M., Cafarella, M.J., Soderland, S., Broadhead, M. and Etzioni, O., 2007, January. Open information extraction from the web. In IJCAI (Vol. 7, pp. 2670-2676), USA.

[15] Jones, R., McCallum, A., Nigam, K. and Riloff, E., 1999, August. Bootstrapping for text learning tasks. In IJCAI-99 Workshop on Text Mining: Foundations, Techniques, and Applications (Vol. 1, No. 7), USA.

[16] Bunescu, R. and Mooney, R., 2007. Learning to extract relations from the web using minimal supervision. In Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (pp. 576-583), Prague, Czech Republic.

[17] Snow, R., Prakash, S., Jurafsky, D. and Ng, A.Y., 2007. Learning to merge word senses. In Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (emnlp-conll), Prague, Czech Republic.

[18] Bollacker, K., Evans, C., Paritosh, P., Sturge, T. and Taylor, J., 2008, June. Freebase: a collaboratively created graph database for structuring human knowledge. In Proceedings of the 2008 ACM SIGMOD international conference on Management of data (pp. 1247-1250). AcM, Vancouver, Canada.

[19] Zeng, D., Liu, K., Lai, S., Zhou, G. and Zhao, J., 2014. Relation classification via convolutional deep neural network. In Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers (pp. 2335-2344), Dublin, Ireland.

[20] Nam, Sangha, Kijong Han, Eun-kyung Kim, and Key-Sun Choi. "Distant supervision for relation extraction with multi-sense word embedding." In Global Wordnet Conference, Workshops on Wordnets and Word Embeddings, Singapore.