Improving the Relation Classification Using Convolutional Neural Network

S Kamath, K G Karibasappa, Anvitha Reddy, Arati M Kallur, B B Priyanka, Bhagya B P
KLE Technological University, Hubballi, Karnataka, India

Abstract - Relation extraction has been the emerging research topic in the field of Natural Language Processing. The proposed work classifies the relations among the data considering the semantic relevance of words using word2vec embeddings towards training the convolutional neural network. We intended to use the semantic relevance of the words in the document to enrich the learning of the embeddings for improved classification. We designed a framework to automatically extract the relations between the entities using deep learning techniques. The framework includes pre-processing, extracting the feature vectors using word2vec embedding, and classification using convolutional neural networks. We perform extensive experimentation using benchmark datasets and show improved classification accuracy in comparison with the state-of-the-art methodologies using appropriate methods and also including the additional relations.

Keywords: Relation extraction, NLP, CNN, Word2vec embedding, Deep Learning.

1. Introduction

The task of determining semantic relations from a text is known as relation extraction. Extracted relations are those that exist between two or more entities of the same type and fit into one of the several semantic groups. It is an important part of natural language processing systems that needs to extract explicit facts from text, to ease the task of question answering systems[12] and knowledge base completion. Information extraction, Automatic content extraction [1] document summarization, machine translation, and the creation of thesauri and semantic networks are some of the uses of automatic semantic relations recognition.

The task of assigning specified relation labels to entity pairings that appear in texts is referred to as relationship classification. The following is an analysis of the Relation Extraction (RE) task: We want to figure out the link between e1 and e2 given a sentence S containing two nominals e1 and e2.

For example: "The<e1>burst</e1>has been caused by water hammer<e2>pressure</e2>."

e1 = burst

e2 = pressure

Cause-Effect (e2, e1).
The goal would be to automatically recognize that this sentence expresses Entity-Destination relationship between e1 and e2, for which we use the notation Entity-Destination (e1, e2). Accurate relation classification facilitates precise sentence interpretations, discourse processing, and higher level NLP tasks.

2. Related Work

One of the important factors seen in the Relation extraction is the availability of annotated datasets. Previously distinct supervision paradigm [3] were used instead of depending on the annotated datasets for training also several multilingual datasets [11] were developed for such tasks. With the advancement in the neural network and Deep learning techniques [13] several relation extraction classification models have been proposed various features. while syntactic features in the sentences [4] where the distance between the entities play a vital role, several other features include dependency parse features, incorporating knowledge through WordNet, Frame Net and POS [9] are also considered to bring significant improvements in accuracy of extraction. Yet another work considering lexical and sentence level features was proposed in [14] where these approaches depend on additional features.

Relation Extraction has not only seen its advancement in the task of relation classification but also has been used as a pipeline model or pre-processing model for many applications. Extraction task can be used to extract triplets to produce knowledge graphs [5] of relational facts between person location and organization in the text. However, in the wild it is not just the relation between these Named entities but different ways of expressing the common class relations between the entities are used [8]. Another visible change in the progress of Relation Extraction has been in considering the external knowledge [7] through knowledge base to boost the performance of the model. But in real world scenario the prior knowledge resources may not be consistent and may lead to ambiguity between the existing relations. Thus, with the combined features of the syntactic and semantic along with the external knowledge sources several relation extraction models have gained added accuracy.

Until recently the work of Relation Extraction has seen tremendous growth in binary classification where relation between the entities is learnt within the sentences but however when we consider the document as a discourse then the fragments of text [15] are classified as potential relations limiting the total classifications to be very minimal however for the relations across the sentences [6] there are plentiful of possible relations across and within the documents where utmost care must be given to avoid false positives in the relationship across sentences.

3. System model

The block diagram of the proposed system is shown in figure 1 below. The selected dataset from the user is pre-processed using the toolkit followed by the annotation of the dataset, which includes identification of the entities e1, e2, and labels. Further, these annotated sentences are given to the CNN model which uses pre-trained word2vec embeddings, leading to relation extraction between entities in the sentences.
Figure 1. System Design for Relation Extraction

4. Architecture

Many Natural Language Processing tasks have been solved using Deep Neural Networks. Instead of extracting features manually, Deep Neural Network is used to perform optimized feature extraction for effective model construction, and hence it is a well-suited approach used in various domains. In this paper, a convolutional neural network is used to extract relationships using linguistic features and word embedding using a word table to generate feature vectors of the entities in the sentence. Pre-processing is performed to extract the labels from the text and generate tokens for the sentence. As mentioned in figure 2 below it can be observed that the model takes the complete sentence with mentioned entities as an input and output a probability vector corresponding to all possible relation types. Each feature is represented by a vector that is randomly initialized. A pertained word vector is employed for word embedding. The embedding layer maps every feature value with its corresponding feature vector and concatenates them, to get local features from each part of the sentence.

Figure 2. Convolution Neural Network for relation extraction
To get global features through all filters, we apply maximum pooling over time. The idea of max-pooling is to take into account only the most useful features from the full sentence. To make inferences, the pooled features are input into a fully linked feed-forward neural network. In the output layer, we use a softmax classifier with the number of outputs equal to the number of possible relations between entities.

5. Implementation

DATASET: SEMEVAL 2010 task 8 dataset was used for the proposed model with nine common relations mentioned in the state-of-art work as well as many other works. The state-of-the-art was further improvised by introducing 3 new relations into the dataset namely Phase-Of, Participates-In, Located-In. This introduction of new relations was done to increase the efficiency of the model as the same as mentioned in other similar works. Manual annotation was carried out for the new datasets introduced. Datasets with different types of English preposition relations were not considered as they require human annotation including annotating the entities as e1 and e2 and labelling which is a tedious task. The distribution of the dataset among training and testing, for the relations, is mentioned in table 1.

| Relation          | Train Data | Test Data | Total Data |
|-------------------|------------|-----------|------------|
| Cause-Effect      | 1003       | 328       | 1331       |
| Component-Whole   | 941        | 312       | 1253       |
| Content-Container | 540        | 192       | 732        |
| Entity-Destination| 845        | 292       | 1137       |
| Entity-Origin     | 716        | 258       | 974        |
| Instrument-Agency | 504        | 156       | 660        |
| Member-Collection | 690        | 233       | 923        |
| Message-Topic     | 634        | 261       | 895        |
| Product-Producer  | 717        | 231       | 948        |
| Phase-Of          | 430        | 70        | 500        |
| Participates-In   | 430        | 70        | 500        |
| Located-In        | 430        | 70        | 500        |
| Others            | 1410       | 454       | 1864       |
| Total             | 9290       | 2927      | 12217      |

The train and test files are loaded to the model to generate labels and sentences text files. Vocabulary files words.txt and labels.txt containing the words and labels in the dataset are generated and stored. The model also saves a dataset_params.json with some extra information. A base model directory is created under the experiments directory which contains a file params.json which sets the hyper-parameters for the experiment. During training, a model is instantiated to train on the training dataset following the hyper-parameters specified in params.json.

Further evaluation is carried out using the given metrics on the development set.

Evaluation on the test set is carried after experimenting and selecting the best model and hyper-parameters based on the performance on the development set, the further performance of the model is evaluated on the test set and overall model performance is analysed after completing many experiments based on the training set's performance. F1-score combines precision and recall, and
works also for cases where the datasets are imbalanced as it requires both precision and recall to have a reasonable value. Accuracy is always good to check as one option. F1-score is another. The model achieved Precision: 76.95, Recall:84.30, F1 score: 88.66, and Accuracy: 75.22, after being trained on the training dataset. Summarizing the performance of the model on the test dataset, the model achieved Precision: 78.08, Recall:83.64, F1 score:80.76, and Accuracy:75.44. The performance of the model on the revised train dataset is that the model achieved Precision:78.96, Recall:84.57, F1 score:81.66, and Accuracy:76.69. The performance of the model on the revised test dataset as compared with other datasets is that the model achieved Precision:80.05, Recall:84.73, F1 score:82.32, and Accuracy:77.14.

6. Results and Discussion

A brief description of the comparison of the proposed model with other models and the output of the system is summarized in this section. Table 2 shows the variation of Precision, Recall, F1 score, and Accuracy with different datasets experimented. As stated in [2] increase in the size of data would lead to better results, the same could be observed in the proposed work. The loss is calculated on training and validation and its interpretation is based on how well the model is performing. Loss value implies how well a model behaves after each iteration of optimization. The loss graph was plotted for understanding the behaviour of the model. The graph depicted that for each epoch during training, the loss was decreasing which is ideal for the model to perform in a better way.

| Dataset                  | Precision | Recall | F1 Score | Accuracy |
|--------------------------|-----------|--------|----------|----------|
| SEM-EVAL 2007 task 4     | 79.00     | 79.16  | 79.08    | 79.00    |
| SEM-EVAL 2010 task 7     | 78.08     | 83.61  | 80.76    | 75.44    |
| SEM-EVAL 2010 task 8 with 3 additional relations | 80.06 | 84.73 | 82.32 | 77.14 |

7. Conclusion

In this paper, we have used SEM-EVAL 2010 TASK 8 dataset. The model was trained to learn the relationship among the entities in a sentence using a convolution neural network. To further improve the performance of extracting relations, a pre-trained Word2Vec embedding approach was integrated into the system which improved the f1 score to 80.76. To explore more relations among entities an additional 500 sentences each of 3 new relations was introduced into our dataset. Our results illustrated that the proposed model can keep high-precision over iterations, and adding more data and relations gives improved results. Hence we achieved the improved f1 score i.e, 82.32. Although there are other approaches, which seem to be much more complicated when compared to the proposed model which is simple and significantly gives a better F1 score.

8. References

[1] Stephanie Strassel, Mark Przybocki, Kay Peterson, Zhiyi Song, and Kazuaki Maeda. Linguistic resources and evaluation techniques for evaluation of cross-document automatic content extraction. In Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC’08), Marrakech, Morocco, May 2008. European Language Resources Association (ELRA).

[2] Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenzo Romano, and Stan Szpakowicz. SemEval- 2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In Proceedings of the 5th International Workshop on Semantic Evaluation, pages 33–38, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

[3] Sebastian Riedel, Limin Yao, and Andrew McCallum. Modeling relations and their mentions without labeled text. In José Luis Balcázar, Francesco Bonchi, Aristides Gionis, and Michèle.
Sebag, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 148–163, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[4] Kun Xu, Yansong Feng, Songfang Huang, and Dongyan Zhao. Semantic relation classification via convolutional neural networks with simple negative sampling. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 536–540, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[5] Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[6] Linlin Wang, Zhu Cao, Gerard de Melo, and Zhiyuan Liu. Relation classification via multi-level attention CNNs. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1298–1307, Berlin, Germany, August 2016. Association for Computational Linguistics.

[7] Abhyuday Bhartiya, Kartikeya Badola, Mausam. DiS-ReX: A Multilingual Dataset for Distantly Supervised Relation Extraction, 17th April 2021.

[8] Jiayang Cheng, Haiyun Jiang, Deqing Yang, and Yanghua Xiao. A Question-answering Based Framework for Relation Extraction Validation, 7th April 2021.

[9] Hailin Wang, Ke Qin, Rufai Yusuf Zakari, Guisong Liu, Guoming Lu. Deep Neural Network Based Relation Extraction: An Overview, 6th Jan 2021.