ASTES

Advances in Science, Technology and Engineering Systems Journal
Vol. 4, No. 2, 65-69 (2019)
www.astesj.com
Special Issue on Recent Advances in Engineering Systems

ASTES Journal
ISSN: 2415-6698

A Relation Extraction System for Indian Languages

Ajees Arimbassery Pareed*, Sumam Mary Idicula

Department of Computer Science, CUSAT, 682022, India

ARTICLEINFO

Article history:
Received: 04 January, 2019
Accepted: 28 February, 2019
Online: 15 March, 2019

Keywords:
Convolutional Neural Networks
Natural Language Processing
Word embedding
Relation extraction.

ABSTRACT

Relation Extraction is an important subtask of Information Extraction that involves extracting significant facts from natural language text. Extracting structured information from the plaintext is the ultimate goal of IE systems. The Indian language content on the internet is increasing day to day. Extracting relevant information from this huge unstructured data is a challenging task especially when the business firms are interested in ascertaining public view on their products and processes. The primary objective of relation extraction systems is to find those entities which can be targeted through social networking and digital marketing. Cannibalisation of the product is nowadays done using these Social Networks. Different methods are proposed and experimented for Relation extraction problems. In this paper, we propose a Relation Extraction system using Convolutional Neural Networks. Deep learning based methods have produced state of the art results in many domains. Training and testing are conducted using the shared corpus provided by 'ARNEKT-IECSIL 2018' competition organisers. The evaluation results show that the proposed system could outperform most of the reported methods in the competition.

1 Introduction

Internet is the fastest growing resource on the planet. Lots of information are added to the web every second. However, this information is stored in an unstructured manner. Retrieving the relevant information from this unstructured text is a challenging task that invites the focus of Language researchers. Information extraction, a branch of Artificial Intelligence deals with this challenge [1]. IE transforms the unstructured text into a structured form that can be easily handled by machines. Relation Extraction is one of the subdomains of IE. It is the process of identifying the relation between two entities in a document. There are two major types of RE namely-closed domain and open domain RE. Closed domain RE considers only a closed set of relationships between two arguments while the open domain RE systems use an arbitrary phrase to specify a relationship [2].

Relation Extraction is one of the subtasks of Information Extraction. The occurrence of entities in a sentence is always through well-defined relationships. Automatic identification of such relationships is what we call as the task of relation extraction. Relation Extraction also helps in getting the structured information from the unstructured text. It is very similar to Information Extraction with the exception that IE additionally requires the removal of repeated relations. Applications of Relation Extraction systems include construction of knowledge bases, Question answering systems, text summarisation, etc. Construction of knowledge bases is a laborious, time-consuming project that demands domain expertise. Automatic extraction of relationships and concepts from text documents helps to reduce the time and domain expertise needed for the task. Question answering systems also make use of relation extraction systems since relations can provide a clue about the answers to most of the questions. Similarly, relation extraction systems can be employed in areas like textual entailment, gene-disease prediction, protein-protein interaction, etc.

Text classification is the primary area of research where the supervised machine learning algorithms are explored in Natural Language Processing. It is a very active research area both in industry and academia. Examples of such classification tasks include sentiment analysis from social media text, detection of spam emails, categorisation of customer queries, auto tag-

*Corresponding Author: Department of Computer Science, CUSAT, 9061859697 & ajeesap87@gmail.com

https://dx.doi.org/10.25046/aj040208

ging of news articles, etc. In this work, we have tried to move towards a direction which is not much explored in the case of Indian languages.

The structure of this article is as follows. Section 2 briefly reviews the related works and details about the dataset. Section 3 explains the proposed method and section 4 illustrates the experiments and results. Finally, section 5 concludes the article along with some routes for future works.

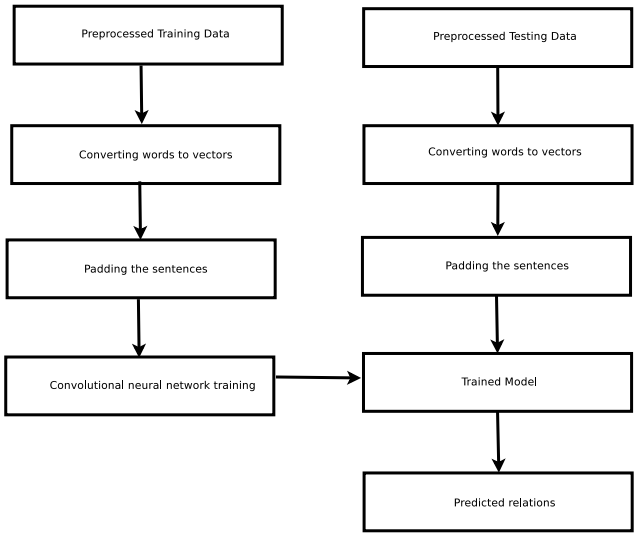

Figure 1: Architecture of the proposed system.

2 Related Works

Various studies are conducted to extract the semantic relations between entities in a text document. Starting from the straightforward rule-based approach to more complex supervised approaches and also semi-supervised approaches are explored in the literature [3]. Rule-based systems employ a naive approach towards the relation extraction task. They construct a set of possible rules for relation extraction by looking at some examples. Hearst, who applied this approach to find the hyponyms of words could achieve an accuracy of 66% on his task [4]. The major issue associated with this technique is that it needs a lot of rules to accomplish the task. We have to continuously update the rule set whenever we find an exception to the existing rules. And more notably, we have to redo this entire work for the other kinds of relations.

On the other hand, supervised relation extraction systems build the model with the help of already tagged corpus. Set of features are designed for each sample in the training data with the help of domain experts. This feature set act as input to the learning algorithms. The key idea behind the supervised learning algorithms is to model the relation extraction task as a classification problem and train the classifier with different algorithms (SVM, Naive Bayes, etc.) available for supervised learning. These classifiers can be trained using a different set of features selected after performing a textual analysis of the labeled data. Different features considered for the study include lexical features, syntactic features, dependency features, entity features, etc [3].

Supervised methods suffer from the problem of availability of enough labelled data. If we do not have enough labelled data to train our classifier, results will be poor. The solution to this issue is the bootstrapping technique. In this technique, we will start with some seed instances of training data, which is manually tagged data used for the first phase of training. We train our classifier with seed instances and learn the classifier. This classifier is used to test more unlabelled data, and get more train examples by adding the test results to the training set. Thus, the training set will expand up to a sufficient amount. This approach is called a semi-supervised learning. 'DIPRE' is an example of the semi-supervised system employed for the relation extraction task [5]. It tries to extract author-book relationship from web text. However, the current trend in relation extraction is based on reinforcement learning. Xiangrong Zeng et al.[6] reports an improvement of 13.36% over the baseline models with the help of reinforcement learning.

The shared task is divided into two subparts say task-A and task-B [7]. Task-A deals with the identification of named entities from the raw text and task-B deals with extracting relation amongst the entities in a sentence. Both these tasks come under the domain of Information Extraction (IE), which is an area under constant research. The growth of research in this area leads to the advancement of applications like information search, question answering, document summarization, etc. Five Indian languages are considered for this shared task. They are Tamil, Hindi, Kannada, Telugu, and Malayalam. It is well known that IE works significantly well with languages like English from applications like Google search, frameworks like Stanford CoreNLP, OpenNLP and many more. The same does not hold good for Indian Languages due to its morphologically rich nature and agglutinative structure. Hence, the objective of this shared task is to improve the Information Extraction systems for Indian languages.

The shared dataset contains data from five different Indian languages [8]. The training data for task-B is a set of files in plain text format. Each file consists of sentences and their corresponding labels. Each language has more than 25000 samples of training data. Statistics of the training data for the task-B is shown in figure 2. The testing data contains two files say test1 and test2. Test1 is for pre-evaluation, and test2 is for final evaluation. The statistics of the test data is given in table 1.

3 Proposed Method

The problem is framed as a sentence classification problem with classes as relations. The number of classes is equal to the number of relations in the tagged data. Convolutional neural networks are used to build the model. They are a class of neural networks that have

Table 1: Data statistics

Language	Train	Pre-eval	Final-eval
Hindi	56775	18925	18926
Kannada	6637	2213	2213
Malayalam	28287	9429	9429
Telugu	37039	12347	12347
Tamil	64833	21611	21612

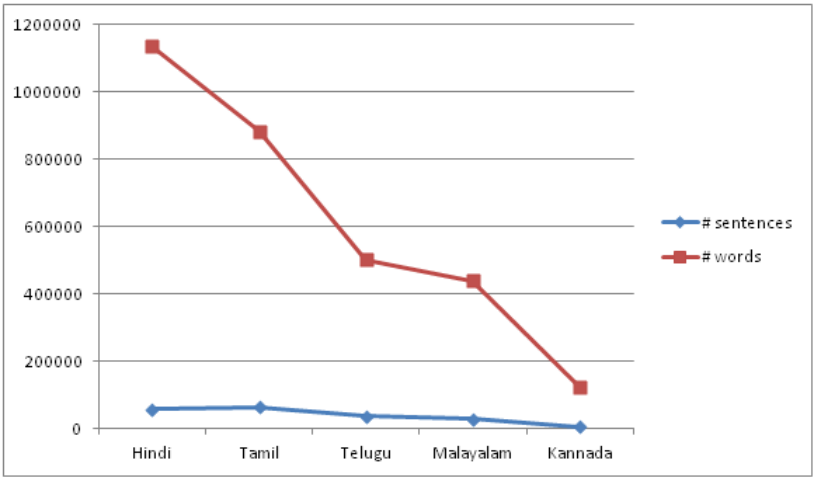

Figure 2: Statistics of the training data

proven very effective in areas such as image classification and pattern recognition. Figure 3 depicts the easiness of deep learning based text processing systems over the traditional machine learning approaches. They can capture the local texture within the text and can be used to find the representative patterns in a text document [9]. The most important property of CNN is preserving the 2D spatial orientation in computer vision problems. But these orientations have a one-dimensional structure in the case of texts.

The architecture of the proposed method is shown in figure 1. The first phase is the preprocessing phase, where words in each sentence are separated using NLTK word splitter. Since words are symbolic units, it can't be directly fed to neural networks. Hence words are converted into numeric values(vectors) using word2vec [10]. The sequences of vectors are then padded with zeros to make it of uniform length. Finally, the padded sentences and their corresponding labels are provided to CNN for training. After training, the model file is saved for testing. In the testing phase, the test data is passed through operations similar to that of the training data and are provided to the saved model for prediction. The model then predicts the label for each sentence indicating the relation within the sentences.

4 Experiments and Results

The preprocessing stage contains operation like trimming, stopword removal, etc. Trimming is the process of removing unwanted symbols from the text. And the most common words are removed in the stop word removal stage. Inputting sequences of raw human alike words will make no sense to computers. For that reason, the raw words are converted into vectors of numeric value using word2vec [11]. Word2vec model is built using a manually created corpus of 27 lakhs words. Skip-gram configuration of Word2vec is used to build the model. The model is constructed with a context window size of 10. The dimensionality of the word embedding is fixed as 100 to cope up with the processing power of the machine.

Keras sequential model is used to construct the classifier [12]. The configuration of the constructed convolutional neural network is shown in table 2. The network is designed with four convolutional layers, two max-pooling layers, and two dense layers. The first layer is a convolutional layer for its ability to capture the local context. The following layers are alternate max-pooling and convolutional layers for acquiring the hidden patterns within the sentence. We have used 'Relu' as the activation function to bring nonlinearity. The number of filters used in the first two convolutional layers is 756. And the kernel size is fixed at 7 for the first two convolutional layers and 3 for the remaining layers. The final dense layer is associated with softmax activation units. During the training phase, filters slide over full rows of the word embeddings. CNN automatically learns the values of its filters based on the labels on the training samples.

In our experiments, we have selected the first 90% of the sentences as training data, and the remaining is selected as the testing data. The batch size is fixed at 100. Categorical cross entropy is used as the loss function. We used Adam, the efficient gradient descent algorithm as the optimizer. Dropout is used to

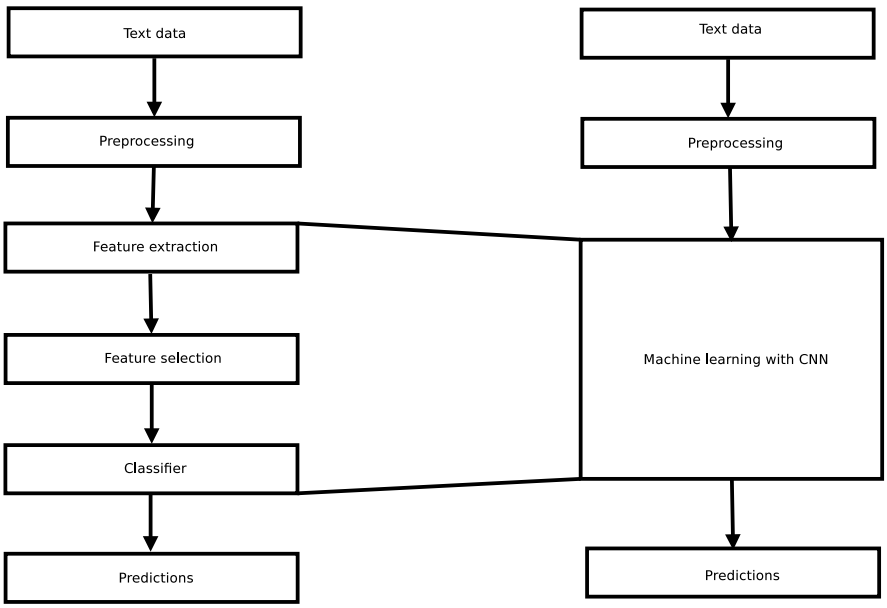

Figure 3: Comparative architecture of deep learning based and non-deep learning based systems

Table 2: Configuration of the CNN architecture

Layers	Output shape	Configuration
conv1d_1	30×756	$7 \times 1, strides1$
max_pooling1d_1	15×756	$3 \times 1, strides1$
conv1d_2	15×756	$7 \times 1, strides1$
max_pooling1d_2	7×756	$3 \times 1, strides1$
conv1d_3	7×256	$3 \times 1, strides1$
conv1d_4	7×256	$3 \times 1, strides1$
flatten_1	1792	–
dense_1	128	Relu activation
dropout_2	128	0.5
dense_2	14	Softmax activation

prevent overfitting [13]. Model is compiled using Tensorflow in the backend. The network is trained for five epochs, and the model file is saved for the testing. Due to the lack of pre-trained word embeddings, we could not complete our work on languages other than Malayalam using pre-trained word embeddings. For Malayalam, we were able to simulate word vectors using our corpus. Moreover, the publicly available pre-trained word embeddings were out of the scope of our machine memory. For languages other than Malayalam, we used an additional embedding layer in front of the convolutional layers.

The proposed system is tested with two test datasets(pre-evaluation and final evaluation). Our system predicts the relation in each input sentence. Table 3 demonstrates the results of our system on both the datasets. Since the final evaluation phase is not available for real-time evaluation, we were not able to get the results for test 2 dataset for languages other than Malayalam. It is clear from the results that our system performance is promising as compared with the performance of other methods reported in the competition. The performance of different relation extraction systems reported in the competition (on test 1) is shown in figure 4. Our system records an average accuracy of 85.62% on test-1 data. From the experimental results, it is obvious that the performance of the system increases with the increase in the training data size. This point is evident from the results of Kannada and other languages. Kannada, which contains the least number of training samples, records the least performance as compared with other languages in the dataset.

5 Conclusion

In this paper, we have discussed a CNN based Relation Extraction system for Indian languages. The exclusive feature of our technique is the use of CNN for relation extraction in Indian languages. The main reason we preferred CNN rather than other traditional feature-based methods is their ability to capture the relations within the sentences. Since deep learning methods require a sufficient amount of training data, the performance of the system can still be improved by increasing the training data size. The performance of the system can also be improved by incorporating word embedding based cluster features into the word vectors. Due to the lack of enough computational power, we could not accomplish that task. Apart from Rela-

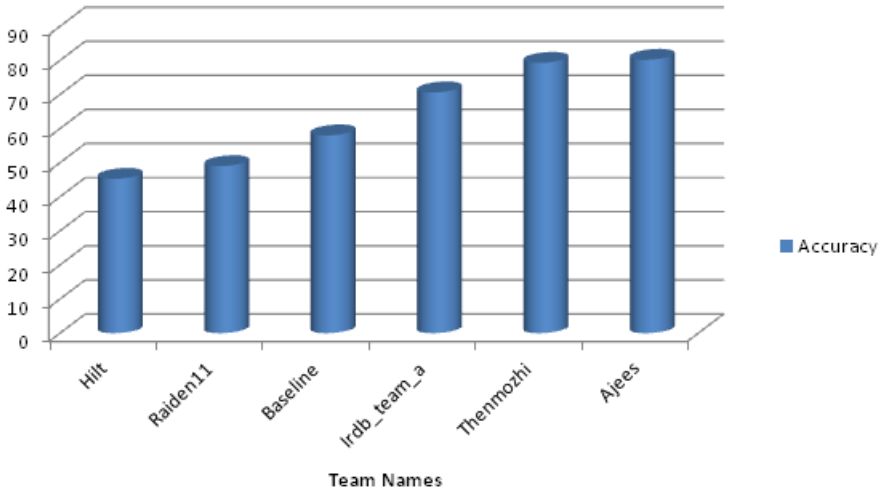

Figure 4: Performance of different relation extraction systems on the shared corpus

Table 3: Results

Test data	Hindi	Kannada	Malayalam	Tamil	Telugu	Average
Test 1 (Accuracy %)	94.72	50.64	80.46	85.76	84.76	85.62
Test 2 (Accuracy %)	NA	NA	77.77	NA	NA	15.55

tion Extraction, Convolutional Neural Networks based deep learning methods can also be applied to various NLP applications like Text classification, sentiment analysis, document labelling, etc.

References

[1] Nita Patil, Ajay S Patil, and BV Pawar. Survey of named entity recognition systems with respect to indian and foreign languages. *International Journal of Computer Applications*, 134(16), 2016.

[2] Manaal Faruqui and Shankar Kumar. Multilingual open relation extraction using cross-lingual projection. *arXiv preprint arXiv:1503.06450*, 2015.

[3] Kush Goyal and Pushpak Bhattacharyya. Literature survey on relation extraction and relational learning.

[4] Barbara Rosario and Marti Hearst. Classifying the semantic relations in noun compounds via a domain-specific lexical hierarchy. In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, 2001.

[5] Jinxiu Chen, Donghong Ji, Chew Lim Tan, and Zhengyu Niu. Relation extraction using label propagation based semi-supervised learning. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 129–136. Association for Computational Linguistics, 2006.

[6] Zeng Xiangrong, Liu Kang, He Shizhu, Zhao Jun, et al. Large scaled relation extraction with reinforcement learning. 2018.

[7] HB Barathi Ganesh, KP Soman, U Reshma, Kale Mandar, Mankame Prachi, Kulkarni Gouri, and Kale Anitha. Overview of arnekt iecsil at fire-2018 track on information extraction for conversational systems in indian languages. In *FIRE (Working Notes)*, 2018.

[8] HB Barathi Ganesh, KP Soman, U Reshma, Kale Mandar, Mankame Prachi, Kulkarni Gouri, and Kale Anitha. Information extraction for conversational systems in indian languages - arnekt iecsil. In *Forum for Information Retrieval Evaluation*, 2018.

[9] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[10] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[11] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. `http://is.muni.cz/publication/884893/en`.

[12] François Chollet et al. Keras. 2015.

[13] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.