Using Convolution Neural Network and Piecewise Convolution Neural Network for Relation Extraction.

Melwin Poovakottu[1]

[1] School of Information, University of Berkeley, Berkeley
E-mail: melwinj7@berkeley.edu

Abstract

Since the advent of Internet, there has been an explosion in the amount of digital text generated in the form of news articles, research publications, blogs, question answering forums and social media. Many techniques have been developed which aim at extracting information automatically from these documents, since lot of important information is hidden within the text. Natural Language contains "Entities" (for example: persons, organizations, places, etc.) which form one of the basic units of the information. In addition to identifying the entities present in text, one of the most important task in language understanding is to understand how entities relate to each other. For example: Consider the sentence: "Barack Obama is married to Michelle Obama." Here the relation extraction aims at predicting the relation of "spouse". Relation extraction is the key module in constructing knowledge graphs, and it is a vital component of many natural language processing applications such as structured search, sentiment analysis, question answering, and summarization. Until recently, relation extraction systems have made extensive use of hand-crafted features generated by linguistic analysis modules. Errors inherent in the modules and hand-crafted features lead to errors in relation detection and classification. In this paper we try to use Neural Network architectures to improve the accuracy of the relation extraction task.

Keywords: Keywords: Relation Extraction, Supervised Learning, Semi-supervised Learning, Convolution Neural Networks

1. Introduction

Extracting relations between entity pairs from text plays an important role in information extraction, knowledge base creation, question answering etc., to name a few. The relation extraction (RE) task can be divided into two steps: First, detecting if the entities of interest mentioned in the sentences are linked by some relation. Second, classifying the detected relation mentions into some predefined classes representing each relation. In the last decade, the supervised relation extraction literature has been dominated by two methods, distinguished by the nature of relation extraction: the feature-based method (Kambhatla, 2004; Boschee et al., 2005; Zhou et al., 2005; Grishman et al., 2005; Jiang and Zhai, 2007; Chan and Roth, 2010; Sun et al., 2011; Nguyen and Grishman, 2014) and the kernel-based method (Zelenko et

al., 2003; Culotta and Sorensen, 2004; Bunescu and Mooney, 2005a; Bunescu and Mooney, 2005b; Zhang et al., 2006; Zhou et al., 2007; Qian et al., 2008; Nguyen et al., 2009; Sun and Han, 2014). There have been Semi Supervised methods used such as bootstrapping algorithm named DIPRE i.e. Dual Iterative Pattern Relation Expansion(Brin 1999) and Unsupervised Relation Extraction methods like Clustering based approaches (Hasegawa et al. 2004). The common characteristic of these methods is the leverage of a large body of linguistic analysis and knowledge resources to transform relation mentions into some rich representation to be used by some statistical classifier such as Support Vector Machines (SVM) or Maximum Entropy (MaxEnt). The NLP analysis pipeline which is hand-designed itself includes tokenization, part of speech tagging, chunking, name tagging and parsing. These are often performed by existing natural language processing (NLP) modules or toolkits. Since the NLP toolkits used to create the hand designed features are themselves imperfect, the downstream process used for relation extraction is subjected to error propagation introduced by these NLP toolkits.

In this paper, I target a supervised RE system that adopts a Convolutional Neural Net architecture to automatically learn relevant features without complicated NLP pre-processing. This paper is inspired by Zeng et al. (2014) and Zeng et al. (2015) and in the paper I try to reproduce the proposed architecture with CNN and Piecewise CNN but discard the multi-instance approach present in the original paper.

CNN avoids complicated feature engineering as well as minimizes the reliance on the NLP modules for feature extraction, alleviating the error propagation and advancing performance. Our relation extraction system is provided only with raw sentences marked with the positions of the two entities of interest (in CNN). Another novel approach that we are employing in the paper is Piecewise CNN. In a typical CNN a single max pooling operation is utilized to determine the most significant features. Although this operation has been shown to be effective, it reduces the size of the hidden layers too rapidly and cannot capture the structural information between two entities.

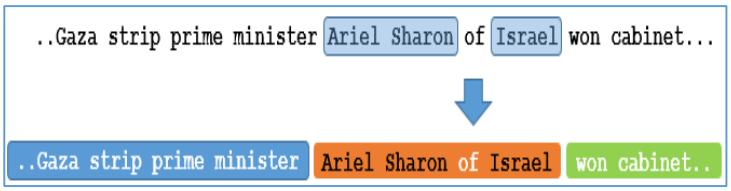

Figure 1: The structure of a sentence can be divided into three parts depending on location of entities. Relation: /people/person/nationality

For example, to identify the relation between Ariel Sharon and Israel in Figure 1, we need to specify the entities and extract the structural features between them. Traditional approaches have employed hand crafted features to model such structural information. These approaches usually consider both internal and external contexts.

A sentence is inherently divided into three segments according to the position of entity pairs in the sentence. The internal context includes the words/phrases between the two entities, and the external context involves the words/phrases around the two entities, as shown in Figure 1. Using single max pooling for each sentence is not enough to capture such structural information. To capture structural and other latent information, we divide the convolution results into three segments based on the positions of the entities and devise three piecewise max pooling node for each segment instead of the single max pooling node. Hence the piecewise max pooling procedure returns the maximum value in each segment instead of a single maximum value over the entire sentence. The expectation is using Piecewise pooling will be able to capture the structural information, and thus it is expected to exhibit superior performance compared to traditional methods.

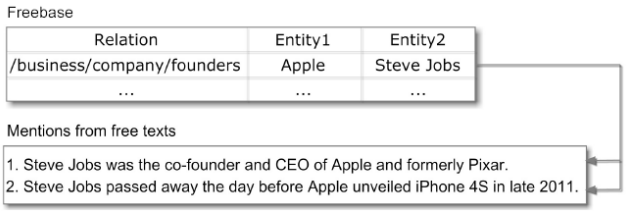

Figure 2: As shown in the image the dataset contains the entities and relationships based on Freebase

This paper evaluates the relation extraction task using the New York Times(NYT10) corpus. NYT10 is used as a distantly supervised dataset which was originally released with the paper Riedel S (2010). Note: The dataset is created using a distantly supervised manner and it provides the entity pairs and the relationships (which were created using freebase) See example in Figure 2.

2. Related Work

Relation extraction is one of the most important tasks in NLP, and has been applied in many practical scenarios (Kordjamshidi et al., 2011; Madaan et al., 2016). In the introduction we already spoke about some of the methodology which has been used for Relation Extraction. Supervised methods have relatively higher performance, but require massive human annotation, which is both expensive and time consuming. Distant supervision is an approach essentially similar to the traditional systems in representing relation mentions but attempts to generate training data automatically by leveraging large knowledge bases of facts and corpus. As mentioned before the NYT10 Dataset used in the paper was created using distant supervision solves the above mentioned problem by using heuristic assumptions to align triples in a knowledge base with sentences in real-world text corpus. A well-known approach in distant supervision is Mintz et al. (2009), which aligns Freebase with Wikipedia articles and extracts relations with logistic regression. Follow-ups studies use the feature set developed in this approach, but with deeper understanding on the nature of distant supervision. For example, Riedel et al. (2010) relaxes the assumption used in Mintz et al. (2009) and formulates distant supervision as a multi-instance learning issue; Hoffmann et al. (2011) and Surdeanu et al. (2012) consider overlapping relations between an entity pair. Further effects are also made to model missing data (Ritter et al., 2013), reduce noise (Roth et al., 2013), inject logical background knowledge (Rocktaschel et al., 2015), etc. In recent years, deep neural network has proven its ability to learn task-specific representation automatically, so that avoiding error propagation suffered by traditional feature-based models. In particular, many neural network approaches have been proposed and shown better performance in relation classification (Zeng et al., 2014; Liu et al., 2015; Xu et al., 2015) and relation extraction (Nguyen and Grishman, 2015).

3. Methodology

Our convolutional neural network for relation extraction consists of four main layers:

1. The embedding layer which consists of a look-up tables to encode words in sentences using pre-trained word2vec word embedding.
2. The word embedding are concatenated with positional embedding which are essentially vectors representing the relative positions of the entity pairs
3. The input vectors are feed into a convolutional layer to extract the important latent features hidden within the sentences
4a. A max pooling layer to determine the most relevant features
4b. In piecewise CNN divide the sentence into three parts based on the structure of the sentence and max pooling on each of the parts.
5. A fully connected neural network with a softmax at the end to perform classification

Figure 3 gives an overview of the CNN network and Figure 4 gives overview of Piecewise CNN

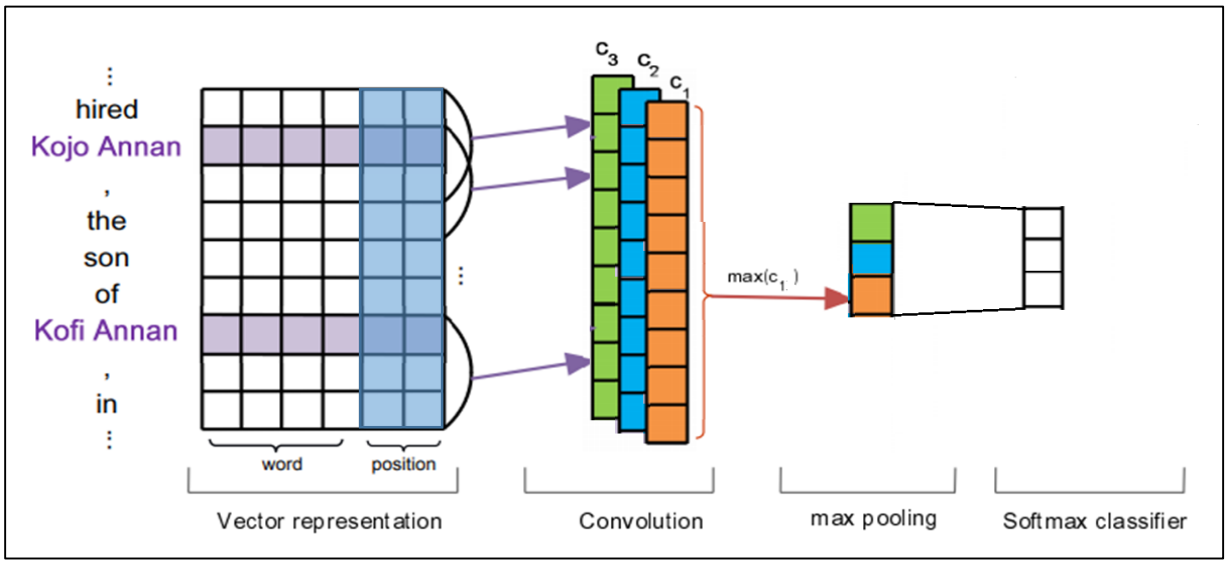

Figure 3: Convolution Neural Networks for Relation Extraction

3.1 Vector Representation

The inputs data is the the form or raw text tokens as mentioned in the introduction .When using neural networks, we transform word tokens into high-dimensional vectors called word embeddings. Each input word token is transformed into a vector by looking up pre-trained word embeddings for the corresponding word token. Additionally, we use position embeddings (PFs) to specify location of the entity pairs, which are also transformed into vectors by looking up position embeddings.

3.2 Word Embeddings

Word embeddings are distributed representations of words that map each word in a text to a 'k'dimensional real-valued vector. They capture both semantic and syntactic information about words very well, and is being used in almost every setting in several NLP tasks (Mikolov et al., 2013; Pennington et al., 2014). In this paper, we use the pre-trained word2vec word vector representations which is created using the Skip-gram model.

3.3 Position Embeddings

In relation extraction, we focus on assigning labels to entity pairs. We use Positional Embedding (PFs) to specify the location of the entity pairs. A PF is defined as the combination of the relative distances from the current word to the two entities in consideration within the sentence i.e. e1 and e2. For instance, in the sentence:

"hired[4] Kojo[3] Annan[2] , the[1] son[0] of[-1] Kofi[-2] Annan[-3]"

The relative distances (in square brackets) from the word "son" to entity 1 (Kojo Annan) and entity 2 (Kofi Annan) are 3 and -2, respectively. The position embedding matrixes is a randomly initialized matrixed. The relative distances are converted into real valued vectors by looking up the position embedding matrixes. We set the size of the word embedding as dw= 50 and that the size of the position embedding is dp = 5. We concatenate the word embedding and position embedding with respective to both the entities pairs to create the vector representation for a word. The total vector representation vector has

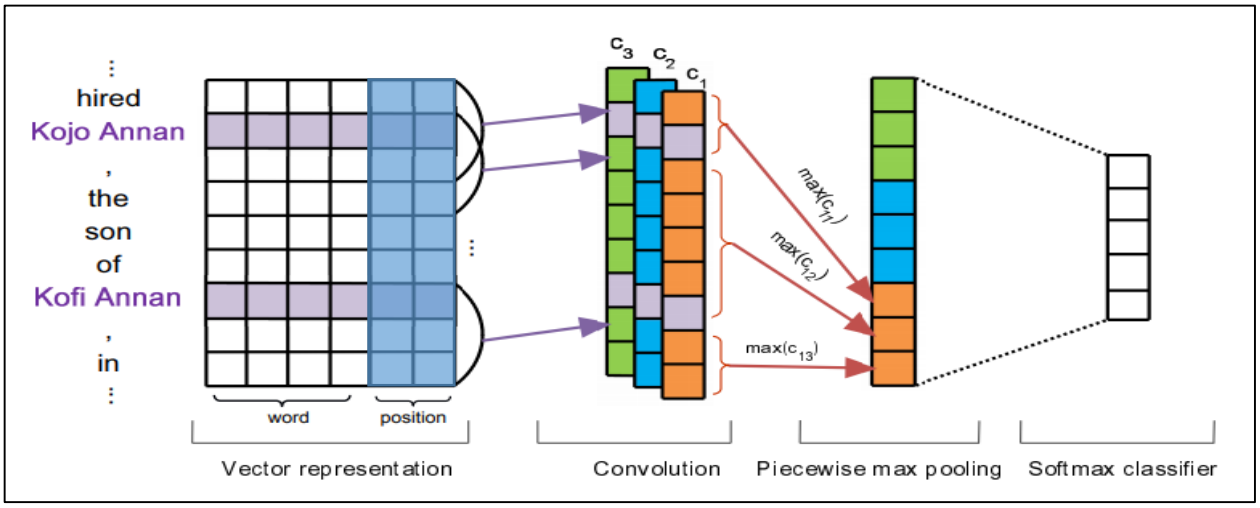

Figure 2: Piecewise Convolution Neural Networks for Relation Extraction. PCNN divides the sentences in three segments and max pools each segment

D = dw + dp ∗ 2 for each word in the sentence. If the sentence length is L then the input vector is a matrix with dimension L x D. This matrix is subsequently fed into the convolution network.

3.3 Convolution

The word representation can only capture contextual information for the word. In relation classification, along the with entity pairs, an input sentence contains contextual features and patterns spread throughout the sentence which help identify the relation. Thus, it might be necessary not only utilize context of the individual words as well as context around phrases and sentence structure and then predict a relation globally. When using a neural network, the convolution approach is a natural means of merging all these local features to create a global feature.

The core of the convolution layer is obtained from the application of the convolutional operator on the two matrices which is the input vector representation matrix and a filter matrix to produce a score sequence s = [s1, s2, s3 . . . , sn]

$$S_i = g(F . X + b)$$

where b is a bias term and g is some non-linear function.

F is the filter and *X* is the input vector representation. Since convolution layer has a fixed length the sentences shall have a maximum length which will be feed into the convolutional layer. Multiple filters (or feature maps) in the convolution layers are applied to increase the feature extracting power. It is instructive to think about the filter *F* as a weight matrix which scores the words/phrases depending upon the possibility of the sentence belonging to the corresponding hidden class (although these scores are not probabilities at all).The trained weights of the filter *F* would then amount to a feature detector that learns to recognize the hidden class of the words/phrases (Kalchbrenner et al., 2014).

3.4 Max Pooling and Piecewise Max Pooling

The size of the convolution output depends on the number of tokens in the sentence that is fed into the network and number of filters applied. To apply subsequent layers, the features that are extracted by the convolution layer must be combined to extract the most important features. In traditional Convolution Neural Networks (CNNs), max pooling operations are often applied for this purpose. The idea is to capture the most

significant features (with the highest values) in each feature map. This approach is insufficient for relation extraction. As described in the introduction section, single max pooling reduces the size of the hidden layers too rapidly and is too coarse to capture fine-grained features for relation extraction. In addition, single max pooling is not sufficient to capture the structural information between two entities. In relation extraction, an input sentence can be divided into three segments based on the sentence structure around the two entities in consideration. In piecewise convolutional neural networks the output of each convolutional filter is divided into three segments. First segment is before the first entity of the concerned entity pairs, the second segment is in between the first entity and the second entity and the third segment is all the words after the second entity. We apply max pooling to each of these segments. The output of each filter after max pooling gives three max pooled values which are concatenated with max pooled values of other filters to for the output of the layer. This is fed to a Softmax layer.

3.4 Softmax Output

To compute the confidence of each relation, the feature vector g is fed into a softmax classifier.

$$O = \sigma (W. X + b)$$

W is the transformation matrix or the weight matrix, O is the final output of the network and b is the bias term. The output O has the dimensionality of Nx1, where N is the number of classes. We use dropout to prevent overfitting by randomly dropping the weights to 0. The output score can be interpreted as a conditional probability by applying a softmax operation i.e. σ

4. Results and Comparision

Both CNN and PCNN were created using the following parameters for the Convoulutional layer.

Parameter Name	Value
Word embedding dimension	50
Positional embedding dimension	5
No of Conv Filters	230
Kernel Size	3
Filter stride	1

On comparing CNN with Piecewise CNN we can see that Piecewise CNN performed better than simple CNN. The AUC for both CNN and PCNN are shown below

Architecture	AUC
Convolutional Neural Nets	0.45
Piecewise CNN	0.71

Also in the Figure 4 the precision vs recall curve. More the area under the curve better the model performs. We can see that the piecewise CNN clearly out performs CNN.

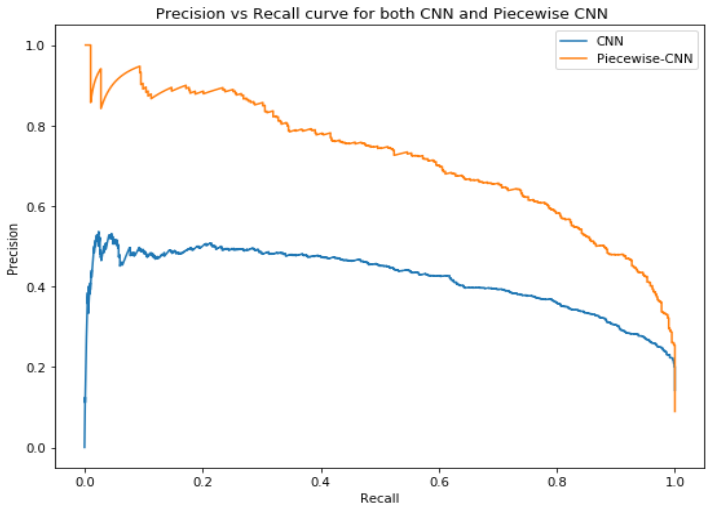

Figure 4: Precision vs Recall curve for CNN vs Piecewise CNN

This means that we cannot capture all the useful information by simply max pooling the filters.
The results demonstrate that the proposed piecewise max pooling technique is beneficial and can effectively capture structural information for relation extraction. Please note that the dataset used in the paper PCNN was initially proposed (Zeng 2014) uses a different paid dataset and hence we cannot use the compare the results from the original paper.

References

[1]Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In Proceedings of COLING, pages 2335–2344.

[2]Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In Proceedings of EMNLP.

[3]Riedel S., Yao L., McCallum A. (2010) Modeling Relations and Their Mentions without Labeled Text. In: Balcázar J.L., Bonchi F., Gionis A., Sebag M. (eds) Machine Learning and Knowledge Discovery in Databases. ECML PKDD 2010. Lecture Notes in Computer Science, vol 6323. Springer, Berlin, Heidelberg

[4]Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In Proceedings of NIPS, 2013.

[5]Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant Supervision for Relation Extraction without Labeled Data. In Proceedings of ACL, 2009.

[6] Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel Weld. 2011. Knowledgebased Weak Supervision for Information Extraction of Overlapping Relations. In Proceedings of ACL 2011.

[7]Chunyang Liu, Wenbo Sun, Wenhan Chao, Wanxiang Che. ADMA 2013. Convolution Neural Network for Relation Extraction

[8] Thien Huu Nguyen, Ralph Grishman. NAACL-HLT 2015 Relation Extraction: Perspective from Convolutional Neural Networks