World Academy of Science, Engineering and Technology
International Journal of Computer and Systems Engineering
Vol:11, No:6, 2017

Causal Relation Identification Using Convolutional Neural Networks and Knowledge Based Features

Tharini N. de Silva, Xiao Zhibo, Zhao Rui, Mao Kezhi

***Abstract*—**Causal relation identification is a crucial task in information extraction and knowledge discovery. In this work, we present two approaches to causal relation identification. The first is a classification model trained on a set of knowledge-based features. The second is a deep learning based approach training a model using convolutional neural networks to classify causal relations. We experiment with several different convolutional neural networks (CNN) models based on previous work on relation extraction as well as our own research. Our models are able to identify both explicit and implicit causal relations as well as the direction of the causal relation. The results of our experiments show a higher accuracy than previously achieved for causal relation identification tasks.

***Keywords*—**Causal relation identification, convolutional neural networks, natural Language Processing, Machine Learning

I. Introduction

RELATION extraction from unstructured sources is an integral part of automated knowledge extraction. It remains an important and open challenge in Natural Language Processing (NLP). The complex syntax and semantics of natural language, its extensive and evolving vocabulary, as well as its ambiguous nature make relation extraction a daunting task. Despite its difficulties, relation extraction plays an important role in transforming unstructured text to machine readable data. Therefore, it has been garnering plenty of interests from research scientists in the past decade. With the popularity of machine learning and deep learning methods, researchers have managed to produce increasingly more accurate and efficient systems to perform relation extraction.

Causal-effect relations in particular are of great interest due to its application in question answering [1], decision making, and knowledge discovery. It introduces another facet to information extraction through its inherent ability to discover new knowledge. Causal relations extracted from a multitude of sources, for example the Word Wide Web or online journals, can be used to form causal chains which may lead to the discovery of previously unknown relations between entities. Causal chains are particularly useful in medicine and biology, where it can be used to find hitherto unknown connections between symptoms, diseases, and their drugs. Indeed, there have been several research studies done in the past which apply causal relation identification to medical text in order to extract knowledge [2].

The definition of causality itself remains somewhat of a controversy, so there can be disagreement even among experts about whether or not a certain sentence implies causality. In certain textual constructs, causality can be obvious; for example, in the sentence "*Bread <e1>shortages</e1> and mounting poverty cause <e2>riots</e2> in nearby Paris.*" it is clear that "bread shortages" and "poverty" have a causal relationship with "riots". In the other cases, however, causality can be much more subtle and ambiguous. Two people reading the same text may disagree on whether or not two events are causal. This is not surprising given the multitude of diverse methods in which causality can be expressed. This also makes the task of causal relation identification more difficult as the model must be capable of identifying a wide variety of syntax structures that imply causal relations.

Causal relations can be explicit or implicit. In explicit causal relations, both cause and effect are explicitly stated in the sentence. Furthermore, causal relations can be expressed using ambiguous or unambiguous keywords. Unambiguous keywords, such as 'because' and 'cause' always signifies causality, while ambiguous keywords like 'from' or 'after' sometimes expresses causality depending on its context. Implicit causal relations are instances where the cause or effect is not explicitly stated but is implied. For example, "The cat killed a bird" is implicit because the effect which is the cat's death is not explicitly stated. In our work, we seek to identify both implicit and explicit causal relations. Unambiguous causal words in particular pose a challenge because the words themselves are not sufficient to determine whether there's a causal relation.

Early work on causal relation identification was mainly based on hand-coded pattern matching to detect cause-effect relations. Studies by Garcia [3] and Khoo et al. used linguistic patterns to identify causal relations in text. Garcia attempted to discover causative verb patterns through linguistic indicators in French text. She reportedly discovered 25 causal relations achieving a precision of 85%. Khoo et al. use predefined linguistic patterns to extract causal relations from medical newspaper text, managing to achieve a precision of approximately 68%. Despite the seemingly good performance of the methods, hand-coded linguistic patterns require a significant amount of tedious manual work. It is also difficult to come up with a comprehensive manually-defined model based solely on syntactic patterns that would apply to all or most cases of causal relations.

In 2002, Girju and Moldovan [4] proposed a semi-supervised algorithm to automatically discover linguistic patterns that express causal relations. The method was evaluated on a diverse set of news articles and achieved an accuracy of around 65%. Several works published around this

Mao Kezhi is with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore (corresponding author, e-mail: EKZMao@ntu.edu.sg)

time used machine learning techniques to identify causal relations [5], [6], [1]. The methods were largely automated; however, they focused on extracting explicit causal relations, and do not address the more complex extraction of implicit causal relations. Moreover, most of the work focused on identifying whether a sentence or relation is causal, and little attention is given to determining the direction of causality, i.e. which entity is the cause and which is the effect.

In recent years, there has been a surge of interest in deep learning methods and its application to problems in NLP. Researches have had considerable success in applying deep neural networks to NLP tasks such as sentence classification, sentiment analysis, topic categorization [7] and relation classification. Although multi-way classification of semantic relations is a different problem than causal relation identification, they are closely related tasks, especially since cause-effect relations are often considered as one of the relations to be classified. Recent researches in relation classification using deep learning methods have managed achieve high accuracy, outperforming previous work in the field. Zeng et al. use convolutional deep neural networks (CNNs) to classify relations in the SemEval Task 8 dataset. Nguyen [9] further improves the performance of CNNs in relation extraction by introducing position embedding to the input sentence vector. Other works attempt to classify relations using deep recurrent neural networks [10] and LSTMs [11], and convolutional neural networks [12]. Nyugen and Grishman [13] use both CNNs and RNNs, combining their results by voting or log-linear modeling. The most recent work by Wang et al. applies a multi-level attention based CNN model to relation classification.

To the best of our knowledge, few works attempt to perform causal relation identification using deep learning methods. It should be emphasized that even though cause-effect relations may be a subset of the relations in the relation classification task, the features extracted for relation classification would be different from the features used for causal relation identifications. In later sections, we explore features that would be specifically useful in the task of identifying causal relations.

In this work, we introduce a feature based classification model to identify causality and causal direction of annotated entity pairs in sentences. Our model is able to identify both explicit and implicit causal relations. We also evaluate the application of CNNs to causal relation identification. We model the problem as a three-class classification of entity pairs in the context of a sentence. Class 1 indicates the annotated pair is causal with the causal direction entity1 -> entity2 (entity1 is the cause and entity2 is the effect). Class 2 entity pairs are causal with the causal direction entity2 -> entity1. Class 3 entity pairs are non-causal.

II. Knowledge Based Sentence Representation

The most important and challenging task of any text processing problem is to construct a comprehensive representation of the text using a feature vector. Based on experimentation with various potential features, we arrived at a feature vector that gives optimal performance for the problem. The features consist of four main components: causal words, prepositions and conjunctions, WordNet features, and TF-IDF.

A. Causal Words

Through our research into past work on causal relation identification/extraction, and our own study of public causal datasets, we compiled a list of words and phrases (ambiguous and unambiguous) that indicate causality (see Table I). While some words in the list like ‘because’, ‘cause’, and ‘since’ almost always indicate the existence of a causal entity pair in the sentence, other words such as ‘make’ are more ambiguous. Nevertheless, the existence of causal word(s) generally indicates a significant probability of there being a causal entity pair in a sentence. Therefore, the existence of causal words in the sentence is used as a feature for the classification model.

TABLE I
Examples of Causal Words

actuate	induce	pioneer
affect	influence	play
breed	initiate	promote
cause	inspire	prompt
compel	instigate	propel
complicate	kick	provide
decide	kill	provoke
decrease	lead	reduce
determine	make	regulate
effect	mold	result
effectuate	motivate	set
encourage	move	shape
facilitate	obligate	solicit
impel	oblige	spawn
impose	occasion	stimulate
increase	persuade	suborn
produce	generate	beget
create		

B. Prepositions and Conjunctions

In our classification problem, we have to not only determine whether an entity pair is causal, we must also determine the direction of causality. Through the study of causal sentences, it is evidential that prepositions and conjunctions between the two entities can give clues as to the causal direction. To illustrate how the existence of the preposition ‘by’ in conjunction with the causal word ‘caused’ can be used to determine direction of causality, consider the sentences:

- The <e1>closure</e1> caused a <e2>backup</e2> on the freeway for several miles northbound as cars were detoured to the Lost Hills Road exit and over to Agoura Road to head south on Las Virgenes
- The <e1>deficits</e1> are caused by <e2>people</e2> saving too much of their money
- Your <e1>doctor</e1> tells your blood pressure by using a <e2>sphygmomanometer</e2>, which is the instrument for measuring blood pressure.

The entity pair in the first sentence is clearly causal as indicated by the causal word ‘caused’. We can also infer that the direction of causality is from entity1 to entity2, ergo the entity pair belongs to class 1. In the second sentence, even though the same causal word is used to indicate causality, we

can see that the causal direction is reversed. The reversal of direction is mainly due to the conjunction of the word 'by' to the word 'caused'. The entity pair in the third sentence is not causal despite the appearance of 'by' between the two entities. The three sentences demonstrate how prepositions and conjunctions can give strong indications of the causal direction when used in conjunction with causal words. In addition to the word 'by', we identified several other prepositions such as "to", "from", "in", "as" and "of" that relate to the direction of causality.

The position of the preposition in the sentence is also important. In almost all cases in which it has a bearing on causal direction, the aforementioned prepositions appear in the word phrases between the two annotated entities. So, for each of the six prepositions, we add a feature to the feature vector to indicate the existence of the preposition between the two entities in a sentence.

C. *WordNet Features*

The features that we have mentioned so far use the word patterns and the syntactic structure of the sentence to determine the causality and direction of an entity pair. In some cases, however, word clues from the sentence are not sufficient to determine the causality of two entities; in these cases, we depend on our prior knowledge of the entities themselves to infer causality. For example, if we consider the sentence "<e1>Asteroid</e1> threatened Earth with <e2>disaster</e2>", the syntax of the sentence itself does not contain certain evidence of causality. Rather we use our inherent knowledge that asteroids may cause disaster to determine that this is a causal entity pair. In such cases, it is important to consider the semantics or the meaning of the entity pair. In order to encode semantic information about the entity pair, we employ WordNet hypernyms, which are parent nodes of a given node in the WordNet hierarchy. Hypernyms can be considered as progressively more generic categories that an entity belongs to. There are several ways in which hypernyms may assist in determining causality: if the entity pair has parent nodes, it indicates that they are related concepts that might potentially be causal. Furthermore, entities with certain hypernyms tend to have higher likelihood of being causal. For example, consider the sentence:

The river had now turned into full <e1>flood</e1> after the deluge of <e2>rain</e2> a few days ago.

The hypernyms for 'flood' are: 'geological phenomenon', 'natural phenomenon', 'phenomenon', 'process', 'physical entity' 'entity'.

The hypernyms for the entity 'rain' are: 'precipitation', 'weather', 'atmospheric phenomenon', 'physical phenomenon', 'natural phenomenon', 'phenomenon', 'process', 'physical entity', 'entity'.

As we can see, both entities are children of the parent node 'natural phenomenon', which indicate a close relation between the two entities. If the two entities or events are correlated, we may assume that there is a greater probability of causality.

For our model, we get a list of hypernyms in WordNet for each of the two annotated entities in the sentence. The vectorized hypernyms for entity1 and entity2 are added as separate features to the feature vector. We use Pywsd's [14] simple lesk algorithm [15] to perform word sense disambiguation. This enables us to detect the correct sense of the word used in the sentence for each entity before using it as input to WordNet.

D. *TF-IDF*

The final features used in our model are the TF-IDF of the entire sentence and the TF-IDF of only phrases between the entity pair. TF-IDF helps the classifier identify words that are not included in the aforementioned features but may potentially indicate causality and causal direction. Through our study of causal sentences, we realized that, with a few exceptions, most of the time causality is inferred by the words in between the entity pairs. Therefore, we append the TF-IDF of the phrase in between the entity pair to the feature list. This may appear to be redundant since we are already using the TF-IDF features of the entire sentence; however, the different corpus or 'documents' in each case result in a different TF-IDF value for each term, and hence a different set of features in each case.

The features causal words, prepositions, WordNet features and TF-IDF are combined to create a high dimensional feature vector for each sentence in the dataset. A linear SVM classifier is used to train the model for classification. Evaluation of the model through experiments is presented in section IV.

III. Convolutional Neural Networks

In the previous section, we use a knowledge based feature vector and a traditional classifier to build a model for causal relation identification. In this section, we explore the application of convolutional neural networks to identify cause relations

In recent years, we see a growing trend of applying deep learning methods to NLP tasks, relation extraction in particular. There has been a surge of research that uses CNNs for relation extraction. Here, we explore several models using CNN based on previous research as well as our own.

A. *The Basic Model*

The basic CNN model used in this work consists of four layers: input representation, the convolutional layer, pooling layer, and a fully connected layer (Fig. 1).

The function of the first input representation layer is to encode the input sentence using word vector representations. In our work, we use word embedding to represent each word in a sentence. Word embeddings are pre-trained vectors where each word from a vocabulary is mapped to a vector of real numbers in a low-dimensional space (relative to the vocabulary size). We use word embeddings that have been pre-trained on the Google News corpus of 3 billion words using Google's word2vec [16]. Each word is represented by a 300-dimension word vector. Therefore, for a 10-word

sentence, the output of this layer would be a 10x300 dimension matrix.

The convolutional layer applies a convolution operation over the input matrix to extract higher level features. The input to the layer is the word embedding matrix for the sentence, a window size w (ex: 3) and a filter. A filter is a randomly initialized weight matrix of size wx300. For a given window size w, an input matrix i and a filter f, we slide the window over the input matrix i and apply a convolution operation on the two matrices i and f to produce a score sequence. The sliding window over the word matrix emulates extracting n-grams from the sentence. The weights of the filter are trained by the model to function as feature detectors to recognize the hidden class of the n-grams.

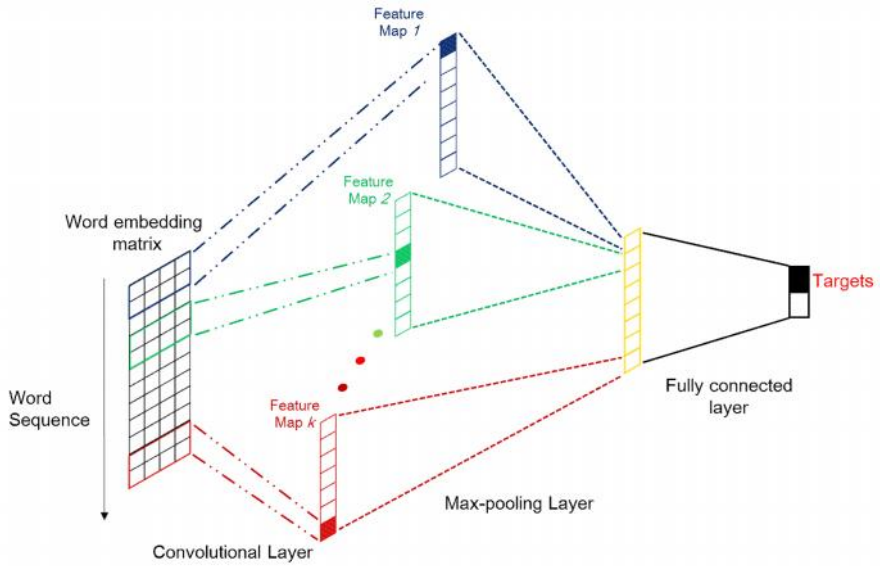

Fig. 1 A basic Convolutional Neural Network for causal relation identification

The pooling layer performs an aggregate function on the scores of each filter to produce a single number. The objective of this step is to avoid dependency on the absolute positioning of the n-grams in the sentence and provide abstraction to the features generated by the convolution layer. In our work, we use the max aggregation function as it has the effect of identifying the most important or relevant features from the score sequence. The pooling layer also reduces the variable sized output from the convolutional layer to a fixed sized output matrix. The pooling scores of each filter are concatenated to form a feature vector to represent the entity pair in the context of that particular sentence.

The final layer is a fully connected neural network with a Softmax classifier at the end. Before feeding the feature vector to the fully connected layer we execute a dropout for regularization. The dropout vector is then fed into the fully connected layer for classification.

B. Tri-Section

We experimented with several different methods using the basic CNN model. In our first approach, we divide the input sentences into three sections based on the position of the entity pair:

- Word phrases before and including entity1
- Word phrases between and including entity1 and entity2
- Word phrases after and including entity2.

For example, the sentence: "Dogs develop a <e1>fever</e1> from <e2>stress</e2> and/or pain such as in a severe flea infestation." would be divided into:

- "Dogs develop a <e1>fever</e1>"
- "<e1>fever</e1> from <e2>stress</e2>"
- "<e2>stress</e2> and/or pain such as in a severe flea infestation."

The objective of splitting up the sentence into three sections is to encode whether a word is before, after, or in-between the annotated entity pair.

The three phrases are input into three parallel branches of the CNN model (Fig. 2). Each branch consists of an embedding layer to encode each word with word embedding vectors from Word2Vec, a convolution layer and a max pooling layer to select the highest value feature vector. We use the window size of 3 in the convolution layer with 150 randomly initialized filters. The features extracted from each branch of the model are concatenated and fed into a fully connected layer with a Softmax classifier at the end. Back propagation is used to fine-tune the value of the filters.

C. Position Embedding

For the second method, we used position embedding to encode more specific relative positioning information for the CNN model [8]. For each word, we embed the position of that word in the sentence in relation to entity1 and entity2:

- For word xi, calculate relative position i-i1 and i-i2 where i1 and i2 is the position index of entity1 and entity2
- Map the resulting value to randomly initialized vectors in a position embedding table
- Each word in a sentence is now represented by:
- Word embedding

- Position embedding in relation to e1
- Position embedding in relation to e2

As in the previous model, CNN has four layers: embedding layer, convolution layer, max pooling layer, and a fully connected layer with a Softmax classifier.

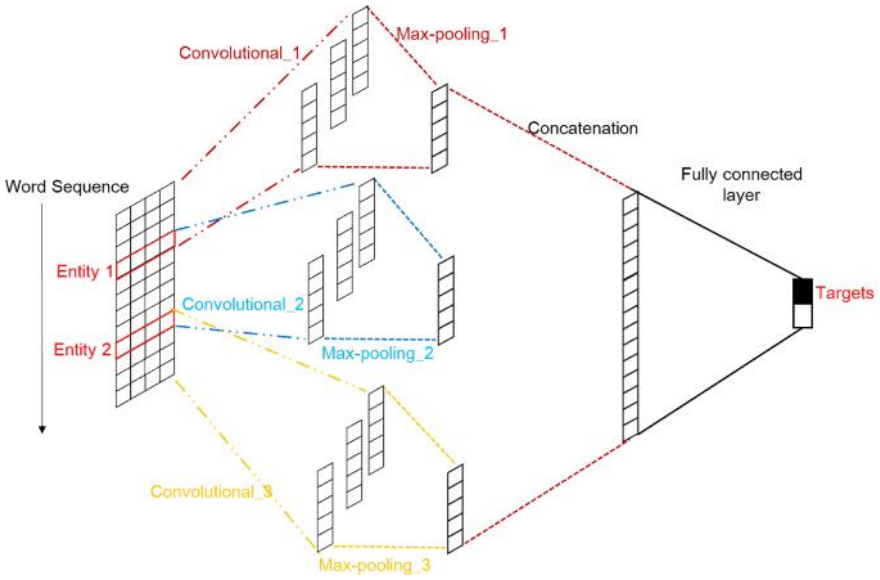

Fig. 2 The CNN tri-section model for causal relation identification. There are three parallel convolutions + pooling layers that process different parts of the sentence to extract feature maps which are later concatenated together

D. Ensemble Method

Finally, we integrate the knowledge based features defined in the section III with the CNN model to integrate causal words and semantic information. The feature matrix is merged with the features extracted by CNN before being passed to the final fully connected layer and Softmax classifier.

IV. Experiments

A. Dataset

We use labeled cause-effect data from two public datasets for evaluation: SemEval-2007 Task 4 dataset and SemEval-2010 Task 8 dataset. Both datasets are labeled for relation classification- the SemEval-2007 dataset has seven labeled relations including Cause-Effect and the SemEval-2010 dataset has nine. Since we are only interested in classifying causal relations, we extract cause-effect data from each dataset as positive labeled data and extract a random mix of other relations as negative data. Both datasets have separate samples for training and testing. The SemEval-2007 dataset has 140 training samples and 70 testing samples for cause-effect relation, while the SemEval-2010 dataset has 1003 training samples and 328 testing samples.

For the purpose of our work, we divide the cause-effect relations in the dataset into two classes based on direction of causality. Our final dataset has 479 total samples for class 1, 927 samples for class 2, and 982 samples for class 3 (class 3 samples are randomly selected from other relations). The imbalance between class1 and class2 is due to the fact that the causal direction entity2 -> entity1 is more common in text than the reverse.

B. Classifiers and Hyperparameters

For the knowledge-based features described in section III, we use a Linear SVM classifier to train the classification model.

For all our experiments using CNNs, we use RelU as the activation function. We use categorical cross entropy as the loss function and Rmsprop as the optimization function. The dimensionality of the word embedding vectors is 300, while the position embedding vectors have a dimension of 50. We use a fixed window size of 3 and 150 filters for the window size. We use a dropout rate of 0.5.

C. Evaluation

Table II shows the accuracy of each method:

TABLE II
Evaluation of Classification Models

Method	Accuracy
Knowledge based features	88.1%
CNN Tri-section	92.3%
CNN Position embedding	92.9%
CNN Position embedding + Knowledge based features	93%
CNN Tri-Section + Position embedding + Knowledge based features	93.2%

The results in Table II show that the CNN models have significantly higher accuracy than the SVM model using knowledge based features. The difference in accuracy between the CNN models appears to be marginal. Adding the knowledge based features to the CNN model does not appear to improve the accuracy significantly; this may be due to the fact that CNN has already extracted features similar to the knowledge based features.

Even though the CNN models appear to work best for this dataset, we discovered that they show more tendencies towards overfitting than the SVM model. We tested both the models with annotated sentences extracted from an earthquake corpus. The accuracy of the CNN models decreases with the new dataset. It can be argued that since the available training dataset is too small to accurately train a deep learning model without the risk of overfitting, typically deep learning models work best with large datasets.

V. Conclusion

In this paper, we have introduced two approaches to causal relation identification: a classification model based on knowledge based features and convolutional neural networks based method. For the latter method, we present several different models based on different methods of input encoding: the tri-section model, position embedding, and an ensemble method. Evaluation of the respective models on data from SemEval relation extraction tasks shows that all models achieve a considerably high accuracy with the CNN models giving the best performance. However, when testing the trained models on an external dataset, we discover that the CNN models are prone to overfitting and therefore achieve lower accuracy if the testing dataset is significantly different from the training data. The main reason for this is the limited amount of labeled training data available. In our future work, we hope to study how to overcome the short comings of the CNN model to achieve high accuracy in diverse datasets.

References

[1] R. Girju, "Automatic Detection of Causal Relations for Question Answering," in Proceedings of the ACL 2003 workshop on Multilingual summarization and question answering-Volume 12, 2003.
[2] C. S. Khoo, S. Chan and Y. Niu, "Extracting Causal Knowledge from a Medical Database Using Graphical Patterns," in Proceedings of the 38th Annual Meeting on Association for Computational Linguistics, 2000.
[3] D. Garcia, "COATIS, an NLP system to locate expressions of actions connected by causality links," in Knowledge Acquisition, Modeling and Mangement, 1997.
[4] R. Girju and D. Moldovan, "Text mining for causal relations," in FLAIRS Conference, 2002.
[5] E. Blanco, N. Castell and D. I. Moldovan, "Causal relation extraction," in LREC, 2008.
[6] D. Chang and K. Choi, "Causal relation extraction using cue phrase and lexical pair probabilities," in Natural Language Processing– IJCNLP 2004, 2004.
[7] Y. Kim, "Convolutional Neural Networks for Sentence Classification," in Proceedings of EMNLP, 2014.
[8] D. Zeng, K. Liu, S. Lai, G. Zhou and e. a. Jun Zhao, "Relation classification via convolutional deep neural network," in COLING, 2014.
[9] T. H. Nguyen, "Relation Extraction: Perspective from Convolutional Neural Networks".
[10] Y. Xu, R. Jia, L. Mou, G. Li, Y. Chen, Y. Lu and Z. Jin, "Improved relation classification by deep recurrent neural networks with data augmentation," 2016.
[11] S. Zhang, D. Zheng, X. Hu and a. M. Yang, "Bidirectional long short-term memory networks for relation classification," in Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation pages, 2015.
[12] Linlin Wang, Z. Cao, G. d. Melo and Z. Liu, "Relation Classification via Multi-Level Attention CNNs," 2016.
[13] T. H. Nguyen and R. Grishman, "Combining neural networks and log-linear models to improve relation extraction," 2015.
[14] L. Tan, "Pywsd: Python Implementations of Word Sense Disambiguation (WSD) Technologies," https://github.com/alvations/pywsd, 2014.
[15] S. Banerjee and T. Pedersen, "An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet," in In Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing, 2002.
[16] T. Mikolov, I. Sutskever, K. Chen, G. Corrado and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," in Proceedings of NIPS, 2013.