Ferryman at SemEval-2020 Task 3: Bert with TFIDF-Weighting for Predicting the Effect of Context in Word Similarity

Weilong Chen, Xin Yuan, Sai Zhang, Jiehui Wu, Yanru Zhang, and Yan Wang*
University of Electronic Science and Technology of China
chenweilong1995,coma_uestc,dwight12,jiehuiwu@std.uestc.edu.cn
yanruzhang,yanbo1990@uestc.edu.cn

Abstract

Word similarity is widely used in machine learning applications like searching engine and recommendation. Measuring the changing meaning of the same word between two different sentences is not only a way to handle complex features in word usage (such as sentence syntax and semantics), but also an important method for different word polysemy modeling. In this paper, we present the methodology proposed by team Ferryman. Our system is based on the Bidirectional Encoder Representations from Transformers (BERT) model combined with term frequency-inverse document frequency (TF-IDF), applying the method on the provided datasets called CoSimLex, which covers four different languages including English, Croatian, Slovene, and Finnish. Our team Ferryman wins the the first position for English task and the second position for Finnish in the subtask 1.

1 Introduction

Word similarity quantitatively measures semantic similarity of two words by comparing word vectors or word embedding, multi-dimensional meaning representations of two words. Word embedding models can be trained and used to predict the value of word similarity, which are comparable to methods based on human judgments. However, most of these established methods are context-independent. Context plays a significant role for people to understand the meaning of words from different perspectives. The meaning of an ambiguous word can be inferred by a certain context, which is known as contextual selection of senses. In a single sense, context can modify the meaning of a word by highlighting certain semantic traits and backgrounding others, which is called the contextual modulation of meaning. To fully utilize context and avoid treating words in isolation, a model that can reflect similarity judgement of words in context is required, which takes word sense, discrete effects and other factors of context into consideration(Armendariz et al., 2020a).

In this paper, we primarily focus on quantifying the similarity of two words in different contexts. First, literature surveys have been conducted to obtain the similarity scores of two words in two different contexts. Then, the uncentered Pearson correlation coefficient between the scores output by our model and the scores output by the human annotators are calculated. According to the Pearson correlation coefficient, we can evaluate the predicted performance of our context-dependent embeddings model in human perception of similarity. We have pre-processed the data by replacing abbreviation and used TF-IDF(Das, 2007) to enrich the attributes of the pair of words to be evaluated in the mask layer of the bert. Then we fine-tune the BERT model(Devlin et al., 2018) with multi-dropout for predicting the absolute similarity rating for each pair of words in each context. Among all teams participating in CoSimLex, our models ranks the 1st place out of 14 on subtask 1.

2 Related Work

The effect of context in word similarity is a cutting-edge topic, which has drawn attention from both industry and academic realm. A number of works have made important contributions in the past several

*All the corresponding to Yan Wang.

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 281–285
Barcelona, Spain (Online), December 12, 2020.

years(Armendariz et al., 2020b). The dataset for this competition has been explained in CoSimLex(Carlos Santos Armendariz, 2019), which is suitable for intrinsic evaluation of state-of-the-art contextual word embedding models. Different embedding algorithms will drastically affect end results, so we need to choose one that will deliver the results we're hoping for. term frequency-inverse document frequency(TF-IDF)(Das, 2007) is a statistical measure that evaluates how relevant a word is to a document a collection of documents. The dimension reduction is also needed for the sparse and high-dimensional vectors, which are not conducive to calculation. Both Principal Component Analysis(PCA)(Wold et al., 1987) and Singular Value Decomposition(SVD)(Golub and Reinsch, 1971) methods can be adopted to get the corresponding k-dimensional vectors.

In order to catch the meaning of the same word in different sentences, one of the most effective strategies is to use computational methods to encoding sentence information containing specific word. Word2Vec(Goldberg and Levy, 2014) is a two-layer neural network that processes text by vectorizing words. The input of Word2Vec is a text corpus and its output is a set of feature vectors that present words in that corpus. Word2Vec has two training methods. The first one is continuous bag-of-word model (CBOW)(Wu et al., 2010) and the second one is skip-gram(Guthrie et al., 2006). There are more ways to train a word embedding. Global Vectors for Word Representation(GloVe)(Pennington J, 2014) is an unsupervised learning algorithm for obtaining vector representations for words. Training is performed on aggregated global word-word co-occurence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space. Both of Word2Vec and Glove provide the same core output, a vector per word. The difference between the two models is that Word2Vec does incremental, "sparse" training of a neural network, by repeatedly iterating over a training corpus, but GloVe works to fit vectors to model a gaint word co-occurence matrix built from corpus. Doc2vec(Q. Le, 2014) model is based on Word2vec, with only adding another vector(paragrah ID) to the input. More specifically, Doc2vec is based on the CBOW model, but instead of using just nearby words to predict the word, it also added another feature vector, which is document-unique. So when training the vectors W, the document vector D is trained as well, and in the end of training, it hold a numeric representation of the document.

Moreover, in order to deal with the situation that Word2vec cannot be used to efficiently measure the change in similarity socres between two contexts, a conrtuction method based on Word2vec and HowNet(Dong and Dong, 2003) has been proposed. This method uses the Word2vec training corpus to obtain the word vector and then obtains the 10 words closet to the candidate word. Then, the HowNet method is used to determine the polarity by calculating the semantic similarity between the candidate word and the seed word. The experimental results show that the construction method has high accuracy and availability.

Models such as Embeddings from Language Models(ELMo)(Peters and Zettlemoyer, 2018) and Bidirectional Encoder Representations from Transformers(BERT)(Devlin et al., 2018) are able to provide representations of words according to surrounding context, which considers not only of discrete differences in word sense but of the more graded effects of context. The Generative Pre-Training(GPT-2)(Radford et al., 2019) model uses conditional probability language modeling with a Transformer neural network architecture that relies on self-attention mechanisms in lieu of recurrence or convolution.

3 Methodology and data

3.1 Data Description

The dataset called CoSimLex(Carlos Santos Armendariz, 2019) is supported by the European Union Horizon 2020 research and innovation programme. CoSimLex is built on the familiar pairwise, graded similarity task of SimLex-999, and provides similarity measures dependent on context. It covers subtle and graded changes in word meaning even for words not necessarily considered polysemous. CoSimLex covers English, as well as less-resourced languages including Croatian, Estonian, Finnish and Slovene.

The goal of this competition is to build a system to predict the effect of context in human perception of similarity. The task has been divided into two different subtasks. Predicting changes is the topic of subtask 1 which our team participated in. The dataset has been released in two stages. As shown in Table 1, stage

1 released the trial data for practice that includes 10 English pairs, 5 Croatian pairs, and 5 Slovene pairs, which contains only a few sample of the actual dataset. Stage 2 released the actual dataset for evaluation consists of 340 English pairs, 112 Croatian pairs, 111 Slovene pairs, and 24 Finnish pairs. Each pair of words was rated within two different paragraphs, giving a total of 1554 scores of contextual similarity ratings by annotators. The subtask is a unsupervised task, so the gold standard human scores of the full dataset would not be opened before the finish of competition. At the end of the evaluation stage, the human annotator gold standard was released, which helps participants to evaluate models.

Language	**practice**	**evaluation**
English	10	340
Croatian	5	112
Finnish	0	24
Slovene	5	111

Table 1: Data Distribution for four language in the released dataset.

3.2 Data Preprocessing

We process the data by replacing abbreviation. Some abbreviations such as "What's" have the same meaning as "What is", but ignoring the differences will cause problems for the final results. First, we defined some replace words and got an abbreviation dictionary, where the form of the key-value data is like 'ain't:is not','I've:I have'.Then we replaced all the keys that appear in the sentence with corresponding values according to rule of the abbreviation dictionary so that we can easily unify the expressions of different words that means equally .

3.3 Methodology

3.3.1 TF-IDF

By using the TF-IDF score, we can calculate the relevance between a word and a particular document. This is done by multiplying two metrics: how many times a word appears in a document, and the inverse document frequency of the word across a set of documents. The score for a word *t* in the document *d* from the document *D* is calculated as follows:

$$tfidf\,(t,d,D) = tf\,(t,d) \cdot idf(t,D) \tag{1}$$

where:

$$tf\,(t,d) = \log\left(1 + freq\,(t,d)\right) \tag{2}$$

$$idf\,(t,D) = \log\left(\frac{N}{count\,(d \in D : t \in d)}\right) \tag{3}$$

We apply the TF-IDF score in the BERT mask layer, making the different attention score for the embedding crossing. It achieved a good performance in the English dataset.

3.3.2 BERT

Google search algorithm ingredient framework is widely known as Google BERT, which stands for Bidirectional Encoder Representations from Transformers. The model aims to help search engine better understand the nuance and context of words in Searches, and match those queries with helpful results. BERT it's pre-trained on huge corpus from different sources. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for this SemEval-2020 Task 3.

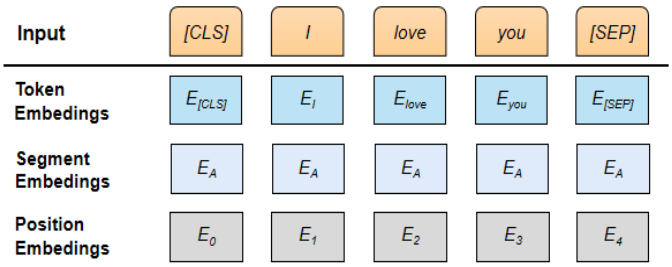

Figure 1: BERT input representation. The input embeddings are the sum of the token embeddings, the segmentation embeddings and the position embeddings.

In this paper, through the attention mechanism of BERT model, we converted the distance between two words at any position to 1, which effectively solves the difficult long-term dependence problem in NLP. So we can directly use the feature representation of BERT as the word embedding feature of the following task. At the same time, TF-IDF is used to evaluates how relevant a word is to a document in a collection of documents. The TF-IDF score can be fed to our Bert model , greatly improving the predicting performance.

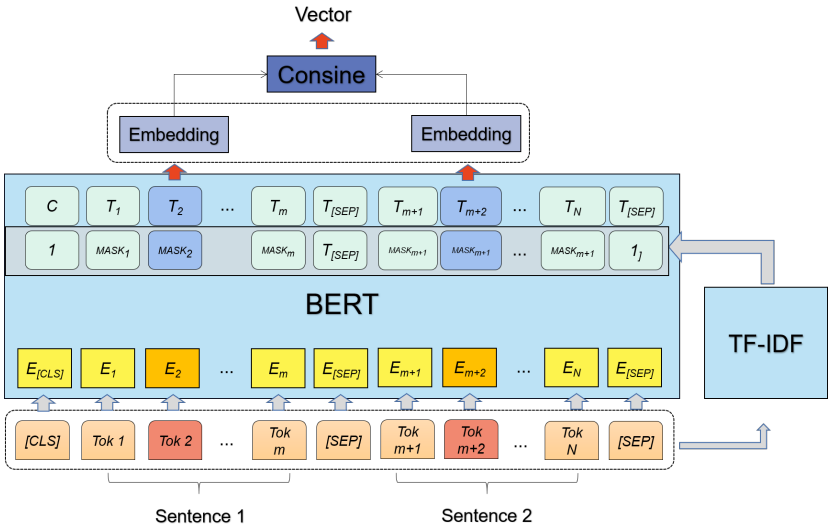

Figure 2: TF-IDF is used in Bert model training

4 Result

The evaluation metrics of this task is the uncentered variation of the Pearson correlation, calculated the standard deviation from zero rather than from the mean value. By using the metric, we can measure the change in similarity scores between two contexts, which is very important as it determines the direction of change prediceted.

The results on the official evaluation set for the subtask 1 is presented in Table 2. In the task, our BERT model, achieved a similarity score of 0.774 on English language, score of 0.634 on Croatian language, score of 0.681 on Finnish language and score of 0.606 on Slovene language.

5 Conclusion

In this study, we report our systems for CoSimLex. To solve the problem of context-dependent word embeddings, we preprocessed the data by replacing abbreviation and unfine-tuned a BERT-based model to predict the similarity of two words in two different contexts. In subtask 1, our model achieved excellent performence as the change of similarity produced by our model is very close to what produced by the

Language	bert_wiki_muti lingual_uncased	elmo char max	bert_cased	bert_uncased	elmo char mean	bert_uncas ed_TF-IDF
English	0.713	-0.121	0.686	0.664	0.529	**0.774**
Croatian	0.587	0.278	**0.634**	0.602	0.531	-0.032
Finnish	0.671	0.068	0.569	**0.681**	0.399	0.306
Slovene	0.603	0.345	**0.606**	0.579	0.51	0.128

Table 2: Scores of different models for each language.

survey annotators. The results indicate that our system is capable of quantifying the effect of different contexts on word similarity of the same set of words.

References

Carlos S. Armendariz, Matthew Purver, Senja Pollak, Nikola Ljubešić, Matej Ulčar, Marko Robnik-Šikonja, Ivan Vulić, and Mohammad Taher Pilehvar. 2020a. SemEval-2020 task 3: Graded word similarity in context (GWSC). In *Proceedings of the 14th International Workshop on Semantic Evaluation*.

Carlos S. Armendariz, Matthew Purver, Matej Ulčar, Senja Pollak, Nikola Ljubešić, and Mark Granroth-Wilding. 2020b. CoSimLex: A resource for evaluating graded word similarity in context. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 5878–5886, Marseille, France, May. European Language Resources Association.

Matej Ulčar Senja Pollak Nikola Ljubešić Marko Robnik-Šikonja Mark Granroth-Wilding Kristiina Vaik Carlos Santos Armendariz, Matthew Purver. 2019. Cosimlex: A resource for evaluating graded word similarity in context. *arXiv preprint arXiv:1912.05320*.

Martins A.F.T Das, D. 2007. A survey of automatic text summarization. *CMU technical report*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Zhendong Dong and Qiang Dong. 2003. Hownet - a hybrid language and knowledge resource. *International Conference on Natural Language Processing and Knowledge Engineering*, pages 820–824.

Yoav Goldberg and Omer Levy. 2014. word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*.

Gene H Golub and Christian Reinsch. 1971. Singular value decomposition and least squares solutions. In *Linear Algebra*, pages 134–151. Springer.

David Guthrie, Ben Allison, Wei Liu, Louise Guthrie, and Yorick Wilks. 2006. A closer look at skip-gram modelling. In *LREC*, pages 1222–1225.

Manning C Pennington J, Socher R. 2014. Glove: Global vectors for word representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Neumann M. Iyyer M. Gardner M. Clark C. Lee K. Peters, M. and L Zettlemoyer. 2018. Deep contextualized word representations. in proceedings of the 2018 conference of the north american chapter of the association for computational linguistics: Human language technologies. *Volume 1 (Long Papers), pages 2227– 2237. Association for Computational Linguistics*.

T. Mikolov Q. Le. 2014. Distributed represenations of sentences and documents. *In Proceedings of ICML 2014*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Svante Wold, Kim Esbensen, and Paul Geladi. 1987. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52.

Lei Wu, Steven CH Hoi, and Nenghai Yu. 2010. Semantics-preserving bag-of-words models and applications. *IEEE Transactions on Image Processing*, 19(7):1908–1920.