Tech review of work embedding

Qiyang Chen qiyangc2@illinois.edu

1 Introduction

In this paper, the author aims to review the development of the work embedding technology in natural language processing (NLP) to improve the specific tasks and make a brief analysis on different work embedding algorithms.

2 Work embedding

Based on the content from CS447 Natural language processing, I know word embeddings are a type of word representation that allows words with similar meaning to have a similar representation. In addition, they are a distributed representation for text that is perhaps one of the key breakthroughs for the impressive performance of deep learning methods on challenging NLP problems. Bakarov [1] defined work embedding as real-valued word representations, which are able to capture lexical semantics and trained on natural language corpora.

In this book [4], work embeddings are defined as: "One of the benefits of using dense and low-dimensional vectors is computational: the majority of neural network toolkits do not play well with very high-dimensional, sparse vectors. ... The main benefit of the dense representations is generalization power: if we believe some features may provide similar clues, it is worthwhile to provide a representation that is able to capture these similarities."

Bengio [2] regarded word embeddings as each word mapped to one vector and the vector values are learned in a way that resembles a neural network, and hence the technique is often lumped into the field of deep learning.

Based on the different definition ways above, we could capture some similar concept of word embeddings: in word embeddings, each word is represented by a real-valued vector, often tens or hundreds of dimensions. In addition, Key to the approach is the idea of using a dense distributed representation for each word.

This is contrasted to the thousands or millions of dimensions required for sparse word representations, such as a one-hot encoding. The distributed representation is learned based on the usage of words. This method allows words that are used in similar ways to result in having similar representations, naturally capturing their meaning. This can be contrasted with the crisp but fragile representation in a bag of words model where, unless explicitly managed, different words have different representations, regardless of how they are used.

There is deeper linguistic theory behind the approach, namely the “distributional hypothesis” by Zellig Harris [5] that could be summarized as: words that have similar context will have similar meanings.

3 Word Embedding Algorithms

In this section, the author will introduce some word embedding algorithms in chronological order. For the previous, we already have a basic understanding about word embeddings, which are a kind of real-valued representations of words produced by distributional semantic models (DSMs), are one of the most popular tools in modern NLP. The term word embeddings first time was introduced by

Bengio [2] in 2003 who trained them in a neural language model together with the model's parameters. However, Collobert and Weston were arguably the first to demonstrate the power of pre-trained word embeddings in their work [3], in which they establish word embeddings as a highly effective tool when used in downstream tasks, while also announcing a neural network architecture that many of today's approaches were built upon.

It was Mikolov [6], however, who really brought word embedding to the fore through the creation of word2vec, a toolkit enabling the training and use of pre-trained embeddings in 2013. A year later, Pennington [7] introduced us to GloVe, a competitive set of pre-trained embeddings, suggesting that word embeddings was suddenly among the mainstream.

In current years, word embeddings are considered to be one of necessary part in NLP research. In one hand, they do not require pricey annotation is probably their main benefit, on the other hand, they can be derived from already available unannotated corpora. In addition, they could be modified by the researchers to meet their specific target, for example, researchers can add more semantic and syntactic information in word embeddings in some NLP tasks.

4 Conclusion and future work

Word embeddings have been found to be very useful for many NLP tasks, including but not limited to: (1) Chunking (Turian [10]); (2) Question Answering (Tellex [9]); and (3) Parsing and Sentiment Analysis (Socher [8]). Many of the mentioned advances seen in the literature have been incorporated in widely used toolkits, such as Word2Vec, gensim19, FastText, and GloVe, resulting in ever more accurate and faster word embeddings, ready to be used in NLP tasks.

In the future, the author will implement some of mentioned word embedding algorithms and compare them in the same NLP task. Then, the author will analyze the performance of each model to see the effect bring by the different word embedding algorithms.

References

[1] Felipe Almeida and Geraldo Xexéo. Word embeddings: A survey. *arXiv preprint arXiv:1901.09069*, 2019.

[2] Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. A neural probabilistic language model. *Journal of Machine Learning Research*, 3, 2003.

[3] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167, 2008.

[4] Yoav Goldberg. Neural network methods for natural language processing. *Synthesis lectures on human language technologies*, 10(1):1–309, 2017.

[5] Zellig S Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.

[6] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013.

[7] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[8] Richard Socher, Jeffrey Pennington, Eric H Huang, Andrew Y Ng, and Christopher D Manning. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the 2011 conference on empirical methods in natural language processing*, pages 151–161, 2011.

[9] Stefanie Tellex, Boris Katz, Jimmy Lin, Aaron Fernandes, and Gregory Marton. Quantitative evaluation of passage retrieval algorithms for question answering. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 41–47, 2003.

[10] Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394, 2010.