University of Sussex

My Theory of Everything

How it all works

Joe Bloggs

Submitted for the degree of Doctor of Philosophy

University of Sussex

September 2008

Declaration

I hereby declare that this thesis has not been and will not be submitted in whole or in part to another University for the award of any other degree.

Signature:

Joe Bloggs

UNIVERSITY OF SUSSEX

Joe Bloggs, Doctor of Philosophy

My Theory of Everything
How it all works

Summary

Acknowledgements

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Word embeddings is a method of word representation mainly used for natural language processing. Text data appears differently in computer from images and sounds. Both images and sounds can be easily represented as mathematic models either using analog or digital signals but not with text data (Li and Yang, 2017). Text data consists of strings that can only be modeled using one-hot vector. This one-hot vector does not have any information that infers connection between one to the other. Hence that, vector representations that maps semantic and syntactic information given one-hot vectors in a euclidean space is introduced (Li and Yang, 2017). This positional information then can be used to infer interconnection between words, whether its similarities or usage of the word in a sentence (S. Harris, 1954). To obtain word embeddings, a model is created to extract features of a word from a corpus and map its location in euclidean space based on the features found. These word embeddings then can be used to do many downstream tasks, such as postagging, named entity recognition (NER), and sentiment analysis (Ling et al., 2015; Lample et al., 2016). In general, large corpus with many words and examples of word usage is prefered because the size of the vocabularies will increases and more words connection can be inferred (Kutuzov and Kunilovskaya, 2018). However, it is not posible to have such corpus since the language itself is evolving and there are many cases of typography in a documents and these words maybe not present in the corpus hence not included in the vocabulary. This words are called as *out-of-vocabulary* (OOV) words. Due to this problems, simple approach by asigning unique random embedding or by replacing OOV with an unknown $<UNK>$ token. While in some cases using these simple approaches can produce acceptable results {CITE}, further improvement on downstream tasks can be achieved by using machine learning method to infer OOV embeddings.

To infer OOV embeddings, the model is built over quasi-generative perspective. Only

knowing the vocabularies and its embedding, the model tried to generate embedding for OOV words, the results are tested for downstream task namely postagging and compared to previous state of the art to search for improvement over previous models.

Chapter 2

Literature Review

Word2vec is one of word embedding that is trained using skip-gram model (Mikolov et al., 2013). A word $w(t)$ used as an input and its context word, for example context word with windows of 4 are $w(t-2), w(t-1), w(t+1)$, and $w(t+2)$, used as the target and the projection from input to the output is used as the representation of the input $w(t)$ that is usefull to predict the context words. This model is highly dependant on the corpus completeness. More examples and vocabulary a corpus has the better the representation of the embeddings since more information will be able to be learned. Word2vec model has no oov handling, meaning either random vector or unknown $<UNK>$ embedding will be used.

Part-of-Speech-Tagging (postagging) is a process of determining grammatical category (tag) of given word in a certain sentence. In English, exist words that has ambiguous grammatical category, such as word "tag" can be either noun or verb depends on the usage of it (Cutting et al., 1992). To tackle this problems, many researchers proposed to use mathematical models or statistical models namely hidden markov model (Cutting et al., 1992), n-grams (Brants, 2000), and neural network model (Ling et al., 2015). In this research, the neural network model will be implemented to serve as the downstream task. This model is a recurrent neural network (RNN) that took sequence of word embeddings representing a sentence or parts of sentence then categorize each word embedding for its tag.

As aforementioned above, some word embedding model such as Word2vec has no oov handling, thus creating a model to predict such word becomes research interest. One of the model that tries to tackle this problem successfully used bi-LSTM to predict embedding given sequence of characters from a word from a pretrained embedding (Pinter et al., 2017). As a result, oov embeddings are able to be predicted without the needs of knowing

lexicon or model used for creating the word embedding. The results then tested to do downstream task namely postagging.

N-grams often used to capture word features. N-grams relies on characters that make up a word, later on a sentence. With character embedding and convolution neural network (CNN), n-grams can be calculated by convoluting sets of characters embedding with the kernel size of $n \times d$ for n is the number of grams and d is the dimension of the embeddings. This CNN n-grams then can be used to create a neural language model (Kim et al., 2015).

Chapter 3

Method

OOV problem is handled from quasi-generative perspective as aforementioned in 1 by using neural language model under assumption that there is a form that could generate embedding for the original embedding. Hence that, the original embedding is used for training the model to generate the embedding. This model then will be used to estimate OOV embeddings. In other words, given sets of vocabulary $\mathcal{V}$ with size V and pretrained embeddings $\mathcal{W}^{V \times d}$ for each word $w_i \in \mathcal{V}$ that is represented as a vector e_i with d dimension, the model is trained to map function $f : \mathcal{V} \rightarrow \mathbb{R}^d$ that minimizes $|f(w_i) - e_i|^2$.

N-grams CNN is used to capture features of a word by using kernel with size of $n \times d$.

Chapter 4

Conclusion

I was right all along.

4.1 What was I right about?

I was right about the following things.

4.1.1 Previous theories were wrong

People thought they understood, but they didn't.

4.1.2 My new idea is right

Of course.

Bibliography

Brants, T. (2000). Tnt: A statistical part-of-speech tagger. In *Proceedings of the Sixth Conference on Applied Natural Language Processing*, ANLC '00, pages 224–231, Stroudsburg, PA, USA. Association for Computational Linguistics. 3

Cutting, D., Kupiec, J., Pedersen, J., and Sibun, P. (1992). A practical part-of-speech tagger. In *Proceedings of the Third Conference on Applied Natural Language Processing*, ANLC '92, pages 133–140, Stroudsburg, PA, USA. Association for Computational Linguistics. 3

Kim, Y., Jernite, Y., Sontag, D., and Rush, A. M. (2015). Character-aware neural language models. *CoRR*, abs/1508.06615. 4

Kutuzov, A. and Kunilovskaya, M. (2018). *Size vs. Structure in Training Corpora for Word Embedding Models: Araneum Russicum Maximum and Russian National Corpus*, pages 47–58. 1

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., and Dyer, C. (2016). Neural architectures for named entity recognition. *CoRR*, abs/1603.01360. 1

Li, Y. and Yang, T. (2017). *Word Embedding for Understanding Natural Language: A Survey*, volume 26. 1

Ling, W., Dyer, C., Black, A. W., Trancoso, I., Fermandez, R., Amir, S., Marujo, L., and Luis, T. (2015). Finding function in form: Compositional character models for open vocabulary word representation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1520–1530, Lisbon, Portugal. Association for Computational Linguistics. 1, 3

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546. 3

Pinter, Y., Guthrie, R., and Eisenstein, J. (2017). Mimicking word embeddings using subword rnns. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 3

S. Harris, Z. (1954). Distributional structure. *Word*, 10:146–162. 1

Appendix A

Code

```
10 PRINT "HELLO WORLD"
```