# Improving Simple-Word-Embedding-based Models on Sentiment Analysis

**Shihan Ran**
UC, San Diego
M.S. in Computer Science
Student ID: A53313589
sran@ucsd.edu

**Nan Wei**
UC, San Diego
M.S. in Computer Science
Student ID: A53317118
nwei@ucsd.edu

**Jikun Wang**
UC, San Diego
M.S. in Computer Science
Student ID: A53308689
wagjk@ucsd.edu

## 1 Problem Definition

For *Text Classification* and *Sentiment Analysis* tasks, modern models have achieved very impressive performance, e.g. RNNs or CNNs. However, they are usually quite computationally expensive to train due to the fact that the model needs to estimate an extensive number of parameters. On the contrary, Simple-Word-Embedding-based Models (SWEM) are relatively a lot faster and simpler to train by doing simple computations such as adding and averaging over the word embedding of word sequence elements, e.g. word2vec, Glove, etc.

In our project, we want to specifically take on SWEMs, and seek ways to improve its performance. Some studies have pushed its performance to a relatively high level using associate pooling mechanisms (Shen et al., 2018). Based on that, we want to apply some tricks such as feature engineering and regularization techniques to further improve its performance and stability.

### 1.1 Why is this NLP task important?

From the overall task point of view, *Text Classification* and *Sentiment Analysis* has always been a topic with high popularity and importance in *Natural Language Processing* (NLP). With the number of text files on the Internet increasing exponentially each day, the volume of information available online continues expanding. *Text Classification*, as the assignment of text files to one or more predefined categories based on information contained from text files, is an important component in information management tasks. *Sentiment Analysis*, as the process of determining the emotional tone behind a series of words, is extremely useful in social media monitoring since it allows us to gain an overview of the wider public opinion behind certain topics. The applications of sentiment analysis are broad and powerful. Businesses are trying to unlock the hidden value of text in order to understand their customer's opinions and needs, so as to make better decisions and improve customer services.

For the focus of our project, SWEMs are important for modern model developing. Although SWEMs are not performing as well as more complicated modern models like CNN and RNN, SWEMs have far fewer parameters and therefore faster to train. Therefore, While modern models like RNN and CNN are extensively used for applications, SWEMs are usually treated as the easy-to-get baseline for these more complicated models. As a result, they serve an important role as a baseline indicator, and their performance is highly important.

### 1.2 Why doesn't the problem have a trivial solution?

From the overall task point of view, *Text Classification* and *Sentiment Analysis* has always been a difficult task, mainly due to the fact that human languages are ambiguous. It not only depends on phrasing and sentence structure, but also depends on common sense and contextual knowledge. In addition, human language's wide diversity across genre, dialects and even personal styling contributes to the fact that the tasks are not trivial.

For the focus of our project, since we want to improve SWEMs' performance as a 'baseline' indicator without eliminating its traits, i.e. its simpleness and high speed to train, we need to put some constraints in our methodology, hence is where our challenges lie: we dont want to over-complicate the model. Therefore, based on the studies conducted on the topic (Shen et al., 2018), our current direction is to apply feature engineering techniques and regularization strategies to try to further improve the baseline performance without modifying the infrastructure too much.

## 2 Related Paper Summary

Related paper: *Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms* (Shen et al., 2018).

### 2.1 Contributions

The paper emphasized that SWEMs, i.e. word-embedding with pooling is still powerful in many NLP tasks, such as document categorization, text sequence matching, sentiment analysis, and so on. Compared to neural network models such as CNN and LSTM, SWEMs are easy and fast to train since it does not require extra parameters. The experiments on different tasks show that SWEMs exhibit comparable or even superior performance in the majority of cases considered.

Specifically, the paper proposed 3 pooling methods after word embedding. They are average-pooling, max-pooling, and hierarchical-pooling. The average pooling can be expressed as

$$z = \frac{1}{L}\sum_{i=1}^{L} v_i, \tag{1}$$

where $L$ is the sequence length and $v_i$ is the word vector for the i-th word.

And, max-pooling is like

$$z = \max(v_1, v_2, ..., v_L). \tag{2}$$

This method is motivated by the observation that, in general, only a small number of keywords contribute to final predictions. With this pooling operation, those words that are unimportant or unrelated to the corresponding tasks will be ignored in the encoding process.

Finally, the hierarchical-pooling combined the former two pooling methods. We omit the detail of this method here. The basic idea is to preserve the local spatial information while still focus on key features. This method has been proved to be very effective in sentiment analysis.

### 2.2 Critical Analysis

#### 2.2.1 Regularization strategies

We noticed that from the study mentioned above (Shen et al., 2018), the performance results on some relatively small datasets are sometimes sensitive to how we apply model regularization techniques because of some over-fitting issues. Therefore, one direction of improving the model is to experiment on some specific regularization strategies for the model on smaller datasets, e.g. dataset augmentation, bagging, dropout, to achieve better and more stable overall performance on small classification datasets.

#### 2.2.2 Feature Engineering

In this paper, the authors used GloVe (Pennington et al., 2014) word embedding as initialization for all models. Except for that, we could try to use some other word embedding methods (e.g. word2vec (Mikolov et al., 2013)) and other features that take word-order or spatial information into consideration as input.

#### 2.2.3 Classifier

The paper claimed to use a MLP as the classifier. However, there are still other choices such as SVM (Moraes et al., 2013), decision tree (Bilal et al., 2016), gradient tree boosting (Gupte et al., 2014) and so on. Some of these models are proved to be effective on sentiment analysis, trying different classifier may help to improve the models' performance.

## References


Muhammad Bilal, Huma Israr, Muhammad Shahid, and Amin Khan. 2016. Sentiment classification of roman-urdu opinions using naïve bayesian, decision tree and knn classification techniques. *Journal of King Saud University-Computer and Information Sciences*, 28(3):330–344.

Amit Gupte, Sourabh Joshi, Pratik Gadgul, Akshay Kadam, and A Gupte. 2014. Comparative study of classification algorithms used in sentiment analysis. *International Journal of Computer Science and Information Technologies*, 5(5):6261–6264.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Rodrigo Moraes, JoãO Francisco Valiati, and Wilson P GaviãO Neto. 2013. Document-level sentiment classification: An empirical comparison between svm and ann. *Expert Systems with Applications*, 40(2):621–633.

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

Dinghan Shen, Guoyin Wang, Wenlin Wang, Martin Renqiang Min, Qinliang Su, Yizhe Zhang, Chunyuan Li, Ricardo Henao, and Lawrence Carin. 2018. Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms. *arXiv preprint arXiv:1805.09843*.