ISSN: 2790-170X Vol 1, No. 4, 2023

The Influence of Word Embeddings on the Performance of Sentiment Classification

Rong Huang [1,*] , Qianyi Chen[2], Jun Tang[1], and Jianjie Song[1]

[1] School of Software, Hunan Vocational College of Science and Technology, Changsha, China;

[2] School of Computer and Communication Engineering, ChangSha University Of Science And Technology, Changsha, China.

[*]Corresponding Author Rong Huang ocean1205@163.com

Abstract

Word embeddings are widely used in natural language processing for mapping words into a numerical representation in vector space. Their quality can be influenced by a variety of factors such as training methods and corpus, which in turn impact machine learning performance. As a whole, larger corpora result in higher-quality word embeddings and improved classification accuracy when the training method is same and the corpus is different. However, the content of the corpus will also affect the classification performance. In this work, we study the relationship between several common word embeddings and sentiment classification models through a series of comparative experiments. Comparison results reveal that in addition to the training method and corpus size, the corpus content and dimensionality also play a significant role in determining the quality of word embeddings. Therefore, when dealing with specific tasks, it is necessary to comprehensively consider these factors, so as to obtain better results. This work provides an improved understanding of factors for consideration that may lead to more efficient sentiment classification.

Keywords

Word Embeddings; Sentiment Classification; Language processing; Data Training.

1. Introduction

In the era of Big Data, a vast amount of information is generated on the Internet every day. The analysis and utilization of massive amounts of data cannot be completed by humans alone. Computers have powerful computing capabilities, and various machine-learning algorithms can be used to solve people's various needs for data analysis. Simultaneously, massive amounts of data and increasing computing power have promoted the development of machine learning. In the past few years, numerous deep learning algorithms have proliferated, leading to increasingly powerful models that have demonstrated superior performance in a diverse range of tasks. In certain cases, these models surpass human abilities. For machine learning or deep learning, computers cannot directly process various types of images and text data. Particularly in the context of natural language, words must first be converted into data forms that are computationally understandable and calculable. To accomplish this goal, utilization of word embeddings or distributed representations of words is required[1]. Using word embeddings, words in natural language are mapped to a certain vector space. Each word has a unique vector corresponding to a specific vector space. This mapping is not random and must satisfy many conditions. For example, with regard to word embeddings, the distance between men and women should be shorter than the distance between men and the building.

Sentiment classification[2] is a common task in natural language processing (NLP) [3]. This is a type of text classification. The goal of this task was to divide each document into corresponding emotion categories for a series of emotional documents and several predefined emotion categories. The objective of this task was to divide the authors' tendencies, opinions, and attitudes. This method allows for efficient analysis of texts that include emotional qualities, providing users with quick access to relevant evaluation information that can be easily organized and analyzed. Sentiment classification is mainly used in the analysis and utilization of movie reviews, product evaluations, social opinions, etc., to assist decision-making. In addition to the classification model, the effectiveness of sentiment classification is related to the quality of text representation. In machine and deep learning, text is represented by word embeddings during model processing, such as word2vec[4], GloVe[5], and several typical classification models such as convolutional neural networks (CNN)[6]. ElMo[7] is a novel variant of word Embeddings in Sentiment classification , it represents each word with a Long Short-Term Memory (LSTM) [8] that is derived from the entire input sequence.

This study aims to address the issue of sentiment classification by utilizing widely used machine learning word embeddings. The impact of word embeddings on the efficiency of sentiment classification was systematically studied[9-13]. Through a series of comparative experiments, the following conclusions were drawn.

1. When the same corpus and classification model were used, the effect of GloVe was slightly better than that of word2vec.
2. In general, higher quality word embeddings can be obtained by utilizing larger corpora.
3. When the size of the corpus is sufficiently large, the similarity of the content of the corpus and the content of the task dataset will affect the representation performance of the word embeddings.
4. When training word embeddings, the dimensions of word embeddings need to be adjusted according to the size of the corpus.

2. Models and Methodology

This section introduces the word embeddings, sentiment classification models, and sentiment classification dataset used in the comparative experiment in this work.

First, the word embeddings used in this paper are introduced, which are influenced not only by the training method but also by the scale and corpus content. When the scale and content of the training corpus are different, the obtained word embeddings may be different for each word, even with the same method. In this study, three methods were used to obtain word embedding. First, we utilize word2vec; second, we use GloVe; Third, we utilize a pre-trained model. This method is currently popular for various NLP tasks. In this study, Bidirectional Encoder Representation from Transformers (BERT) [14] is used to encode words, and the encoding result is used as the word embedding.

For word2vec, word embeddings were trained based on the relationship between contexts. There are two training modes: skip gram [15] and continuous bag-of-words (CBOW). In the Skip gram model, the context is predicted based on the target words, whereas in the CBOW model, the target words are predicted based on the context. Finally, certain parameters of the model were used as word embeddings. The architecture of these two models is shown in Fig. 1 Two optimization methods were used during the training process. These two methods are the hierarchical softmax and negative sampling methods. For each word, the word and all other words must be calculated, and the hierarchical softmax can be used to significantly reduce the calculation time. Negative sampling is implemented to accelerate the training process and enhance the caliber of the produced word embeddings. Unlike the conventional approach, in which every training sample updates all weights, negative sampling updates only a limited set of weights at a time, thereby reducing the calculations involved in the gradient descent process. In this study, four corpora were used for training to obtain word embeddings. These word embeddings were used in subsequent classification models. These four types of corpora are Wikipedia Dependency, Wikipedia Gigaword, Twitter Tweets, and Google News. The Wikipedia Dependency corpus contains one billion tokens and 170,000 words sourced from Wikipedia. The Wikipedia Gigaword corpus combines the 2014 Wikipedia dump and Gigaword 5, resulting in a corpus containing approximately six billion tokens and 400,000 words. The Twitter Tweets corpus comprises two billion tweets, 27 billion tokens, and 1.2 million words. Google News is a vast text set with 100 billion tokens and vocabulary of three million words and phrases. For the word vector dimension, two scales were adopted: 200 and 300 dimensions. Using the above settings, eight different Word2vec word embeddings were obtained.

GloVe is a word representation tool that utilizes global word frequency statistics. It combines the advantages of the statistical information of global vocabulary co-occurrence and the local window context approach, resulting in a comprehensive synthesis.

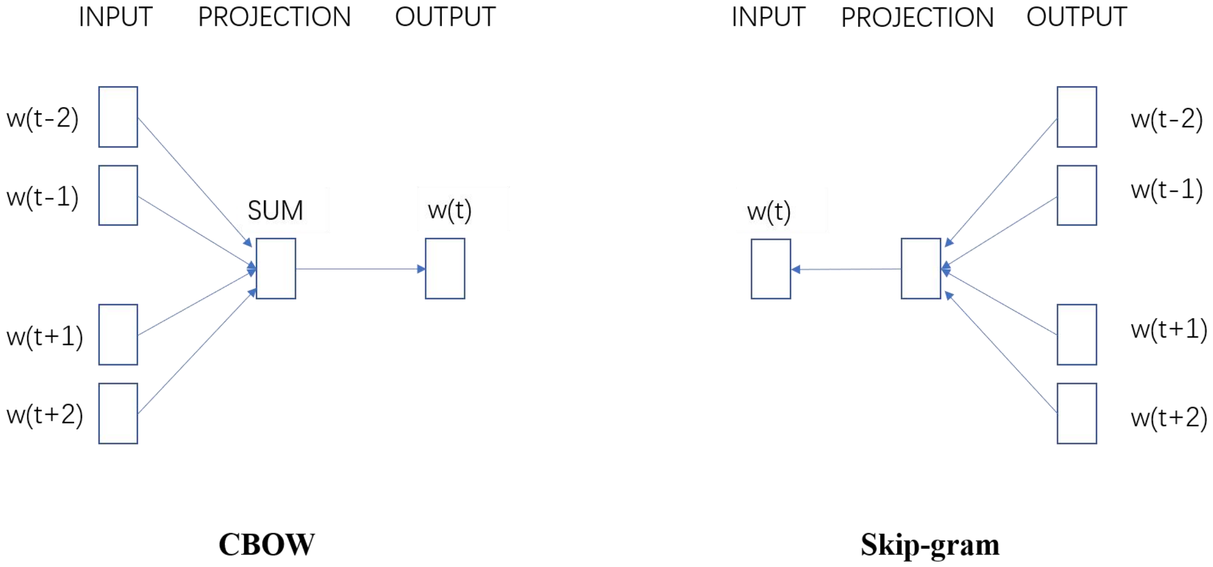

Figure 1: The architectures of two models in word2vec

However, as opposed to the global matrix factorization method, GloVe eliminates the need to calculate the co-occurrence frequency of words with zero count in the vocabulary. Therefore, this approach effectively decreased the computational and storage requirements for data analysis. By leveraging global prior statistical information, it accelerates model training and enables fine-tuned regulation of word importance. Its training process consists of three parts. First, a co-occurrence matrix was constructed based on the corpus, with each element representing the frequency of co-occurrence between the current word and its context words within a specified context window size. Typically, each co-occurrence is assigned a minimum value of1. However, GloVe employs a decay function to determine the weight based on the distance between the two words within the context window. The two words that were farther away had a lower weight in the overall count. The next step involves constructing an approximate relationship between the word vector and co-occurrence matrix. Finally, a loss function is constructed. To ensure the effect of the comparative experiment, the corpus in word2vec was also used for training, and the outcome of this process was the acquisition of four varieties of GloVe word embeddings with dimensions of both 200 and 300.

The BERT framework is shown in Fig. 2. BERT leverages a bidirectional transformer [16] encoder architecture and pre-trains the deep bidirectional representation by synchronously optimizing contextual information across all layers. The corpus is very large, including BooksCorpus 800 million words [17] and English Wikipedia 2.5 gillion words. Two tasks were used in the pretraining process. The first is the Masked Language Model [18] and the second is the next sentence prediction. For the first task, the method randomly masks some words (replaced with a unified mark [MASK]), and then predicts these masked words. The purpose of this task is to train a bidirectional language model and make the expression of each word refer to contextual information. For the second task, the method randomly replaced sentence B with 50\% of the sentence pairs in the input sequence, and then predicted whether B was the next sentence of A. The purpose of this task is to obtain information between sentences that is not directly captured by the language model. Owing to the difficulty of training and the high cost of this model, open-source trained models were directly used. For each word, the dimension of the mapped word vector was 1024. To ensure that the dimensions of the word embeddings trained by the three methods remained uniform, a matrix was added in front of the model to convert the dimensions. In addition, the word embeddings were trained using the classification model.

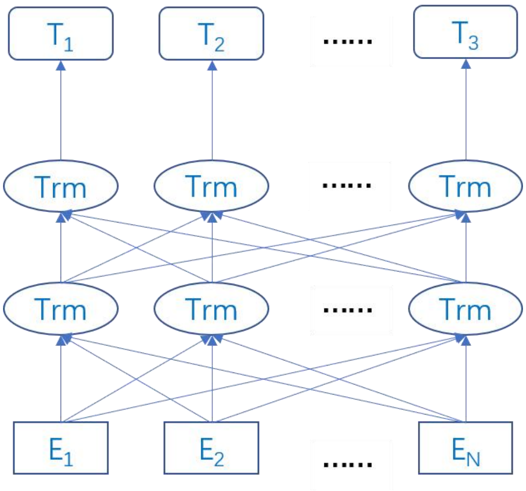

Figure 2: The framework of BERT

Next, the sentiment classification models used in this paper are introduced. A CNN is a neural network that uses multilayer supervised learning techniques. The crucial components of its feature extraction function are the convolutional and pooling layers. The CNN used for classification in this study contained four layers. The neural network model consists of four layers: the input layer, convolutional layer, pooling layer (utilizing the max-pooling technique), and fully connected layer. The softmax method was used for classification to obtain the category label. The advantage of a CNN is that it can obtain feature information that is important for sentence classification. However, its disadvantage is that it does not consider word-order information. LSTM is a unique type of Recurrent Neural Network (RNN) characterized by its focus on cell state. The cell state is updated using a gating mechanism that allows for the deletion or addition of information. The LSTM consists of three separate gates to control the state of the cells. The bidirectional LSTM classification model contained two layers of LSTM, one forward LSTM layer, and one backward LSTM layer. Finally, Softmax was used for classification.

In this study, the sentiment classification dataset consisted of three selected datasets. The three datasets were MR [19], SST-2 [20], and Subj [21]. The MR dataset consists of movie reviews containing a single sentence categorized as positive or negative. SST-2 is an extension of MR that includes separate sets for training, development, and testing as well as positive and negative categories. The Subj dataset aims to classify sentences as either subjective or objective.

3. Results and Discussion

This study uses the word embedding training methods mentioned in the previous section to train the corpus and obtain 16 types of word embeddings. In addition, BERT was used to obtain word embeddings, and a matrix was used to transform it into two dimensions: 200 and 300. Therefore, there were 18 types of word embedding. Eighteen types of word embeddings and the classification models and sentiment classification datasets introduced in the previous chapter were used for comparative experiments, and a series of experimental results were obtained with their comparative analysis outcomes presented in Fig. 3 and Fig. 4.

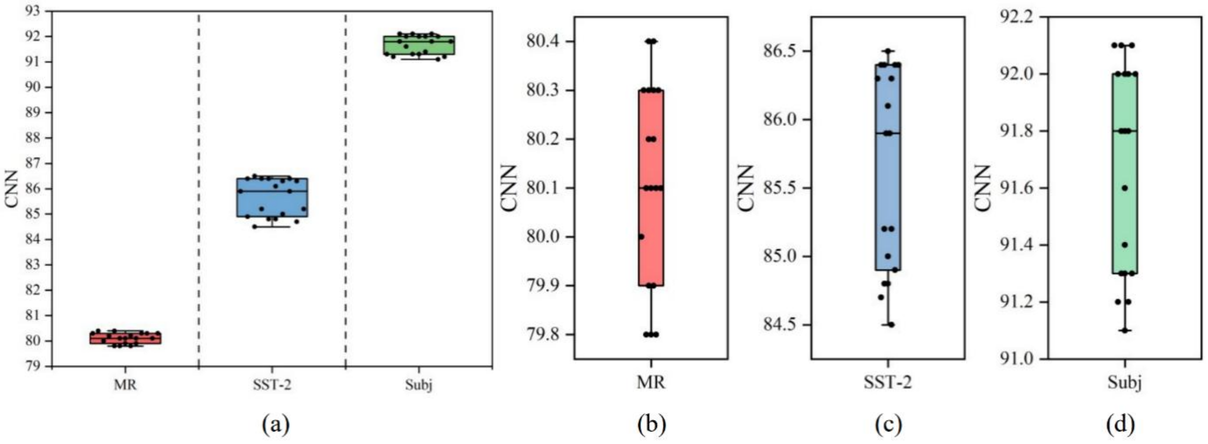

Figure 3: Analysis of CNN Classification Results

Fig. 3(a) shows a comparison of the prediction accuracy using CNN classification models on three different datasets: MR, SST-2, and Subj. The prediction accuracy of the different word-embedding algorithms on the MR dataset is shown in Fig. 3(b). Accuracy ranged from 79.80\% to 80.40\%, with a mean value of 80.11\% and a standard deviation of 0.21\%. The prediction accuracy of the different word-embedding algorithms in the SST-2 dataset is shown in Fig. 3(c). It has a higher mean value of 85.65\%, but also a larger standard deviation of 0.74\% (ranging from 84.50\% to 86.50\%). The prediction accuracy of different word embedding algorithms in the Subj dataset exhibited the highest accuracy value of 91.67\% within a narrower range from 91.10\% to 92.10\% (the standard deviation was 0.37\%). The results revealed that the Subj dataset outperformed both the MR and SST-2 datasets in terms of prediction accuracy. This can be attributed to the fact that larger corpora lead to higher quality word embeddings.

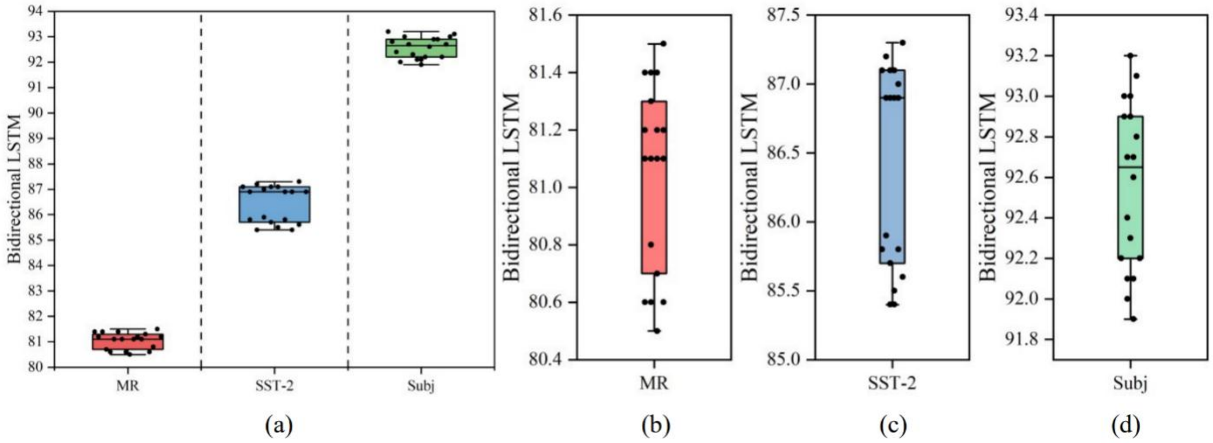

Figure 4: Analysis of BI-LSTM Classification Results

The prediction accuracy using the BI-LSTM classification model on three different datasets, MR, SST-2, and Subj, was further analyzed. The overall results are shown in Fig. 4(a), and the individual results are presented in Fig. 4(b) to (d), respectively. The three datasets give their mean accuracies at 81.04±0.33\%, 86.42±0.74\%, and 92.56±0.42\%, respectively. Compared to the results shown in Fig. 3, BI-LSTM classification model has a higher prediction accuracy for each dataset compared to the CNN model. This could be attributed to the BI-LSTM's unique composition of forward and backward LSTM, which has been found to capture information over longer distances. Its ability to recognize sequence annotation tasks with upper and lower relationships is a crucial aspect that makes it more suited for tasks such as CNN.

The detailed classification results obtained using the CNN are shown in Table 1, and the results obtained using the bidirectional LSTM are shown in Table 2. Observing the classification results, we can see that when using the same classification model, training corpus, and classification dataset, GloVe outperforms word2vec by a slight margin. The reason for this result may be that, compared with word2vec, GloVe adds global statistical information when training word embeddings, which makes the information contained in word embeddings more abundant.

Table 1: Classification results obtained by CNN

Classification model	Data Set	Word Embeddings	Accuracy (%)
CNN	MR	w2v wiki depend 200	79.9
		GloVe wiki depend 200	79.8

Classification model	Data Set	Word Embeddings	Accuracy (%)
		w2v wiki giga 200	80.1
		GloVe wiki giga 200	80.1
		w2v tweets 200	80.2
		GloVe tweets 200	80.4
		w2v news 200	80.1
		GloVe news 200	80.2
		BERT 200	80.3
		w2v wiki depend 300	79.8
		GloVe wiki depend 300	79.8
		w2v wiki giga 300	79.9
		GloVe wiki giga 300	80.0
		w2v tweets 300	80.3
		GloVe tweets 300	80.4
		w2v news 300	80.1
		GloVe news 300	80.3
		BERT 300	80.3
	SST-2	w2v wiki depend 200	84.8
		GloVe wiki depend 200	84.8
		w2v wiki giga 200	85.0
		GloVe wiki giga 200	85.2
		w2v tweets 200	86.1
		GloVe tweets 200	86.4
		w2v news 200	85.9
		GloVe news 200	86.3
		BERT 200	86.4
		w2v wiki depend 300	84.5
		GloVe wiki depend 300	84.7
		w2v wiki giga 300	84.9
		GloVe wiki giga 300	85.2
		w2v tweets 300	86.4
		GloVe tweets 300	86.5
		w2v news 300	85.9
		GloVe news 300	86.3
		BERT 300	86.4
	Subj	w2v wiki depend 200	91.3
		GloVe wiki depend 200	91.3
		w2v wiki giga 200	91.4
		GloVe wiki giga 200	91.6
		w2v tweets 200	91.8
		GloVe tweets 200	91.8
		w2v news 200	92.0
		GloVe news 200	92.1
		BERT 200	92.0

Classification model	Data Set	Word Embeddings	Accuracy (%)
		w2v wiki depend 300	91.1
		GloVe wiki depend 300	91.2
		w2v wiki giga 300	91.2
		GloVe wiki giga 300	91.3
		w2v tweets 300	92.0
		GloVe tweets 300	91.8
		w2v news 300	92.1
		GloVe news 300	92.1
		BERT 300	92.0

Table 2: Classification results obtained by BI-LSTM

Classification model	Data Set	Word Embeddings	Accuracy (%)
Bidirectional LSTM	MR	w2v wiki depend 200	80.5
		GloVe wiki depend 200	80.6
		w2v wiki giga 200	81.1
		GloVe wiki giga 200	81.1
		w2v tweets 200	81.2
		GloVe tweets 200	81.4
		w2v news 200	81.1
		GloVe news 200	81.1
		BERT 200	81.3
		w2v wiki depend 300	80.6
		GloVe wiki depend 300	80.6
		w2v wiki giga 300	80.7
		GloVe wiki giga 300	80.8
		w2v tweets 300	81.4
		GloVe tweets 300	81.5
		w2v news 300	81.2
		GloVe news 300	81.2
		BERT 300	81.4
	SST-2	w2v wiki depend 200	85.5
		GloVe wiki depend 200	85.7
		w2v wiki giga 200	85.8
		GloVe wiki giga 200	85.9
		w2v tweets 200	87.1
		GloVe tweets 200	87.1
		w2v news 200	86.9
		GloVe news 200	87.0
		BERT 200	86.9
		w2v wiki depend 300	85.4
		GloVe wiki depend 300	85.4
		w2v wiki giga 300	85.6
		GloVe wiki giga 300	85.8

Classification model	Data Set	Word Embeddings	Accuracy (%)
		w2v tweets 300	87.2
		GloVe tweets 300	87.3
		w2v news 300	86.9
		GloVe news 300	86.9
		BERT 300	87.1
	Subj	w2v wiki depend 200	92.1
		GloVe wiki depend 200	92.1
		w2v wiki giga 200	92.2
		GloVe wiki giga 200	92.3
		w2v tweets 200	92.6
		GloVe tweets 200	92.7
		w2v news 200	92.9
		GloVe news 200	93.0
		BERT 200	92.9
		w2v wiki depend 300	91.9
		GloVe wiki depend 300	92.0
		w2v wiki giga 300	92.2
		GloVe wiki giga 300	92.4
		w2v tweets 300	92.7
		GloVe tweets 300	92.8
		w2v news 300	93.0
		GloVe news 300	93.2
		BERT 300	93.1

Overall, larger corpora resulted in higher-quality word embeddings and improved classification accuracy when the training method was the same and the corpus was different. However, the content of the corpus also affects the classification performance. From the classification results of MR and SST-2, we find that although Twitter Tweets have a small corpus size, the classification performance is better. The main reason for this is that the content of Twitter Tweets is more similar to the content of movie reviews than the larger Google News corpus. Therefore, it can better express words in movie review classification datasets and achieve a better classification performance. For Subj, Twitter tweets do not have this advantage; therefore, the performance is not as good as the word embeddings trained by Google News.

Next, we analyzed the dimensionality of the word embeddings. According to the experimental results, when the corpus size is small, increasing the dimensionality of word embeddings can adversely impact the quality of results when working with smaller corpora. However, as the corpus size increases, expanding it further enhances the quality of the word embeddings. The reason for this result may be that, when the size of the corpus is small, increasing the dimension of the word embeddings increases redundant information and destroys the expression effect.

From the above conclusions, the results of the other classification models are consistent with the classification results of the CNN.

Finally, although the performance using word embeddings obtained by BERT is not the best, the gap is quite small compared with the best performance of other word embeddings. The size of the training corpus used by BERT was very large, and the training method was also very powerful. However, this effect is not optimal. This may be due to the added conversion dimension matrix. Although the matrix also participates in the training of classification models, it still causes damage to the representation of the word embeddings. However, compared with other word embeddings, the performance was still excellent.

4. Summary

Word embedding is a type of representation of words in NLP and is a distributed representation. Its quality affects the performance of the subsequent models and specific tasks. The quality of word embeddings can be influenced by numerous factors, such as the training methodology, scale of the training corpus, content of the training corpus, and dimensions. When dealing with specific tasks, it is necessary to consider these factors comprehensively to obtain better results.

Acknowledgements

This work was financially supported by the Natural Science Foundation of Hunan Province (2020JJ7042); the Hunan Provincial Education Science Planning Project (XJK21BZJ011).

References

[1] Bengio, Y.; Ducharme, R.; Vincent, P. A neural probabilistic language model. Advances in neural information processing systems 2000, 13.

[2] Dong, L.; Wei, F.; Liu, S.; Zhou, M.; Xu, K. A Statistical Parsing Framework for Sentiment Classification. Computational Linguistics 2014, 41, 293-336.

[3] Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; Kuksa, P. Natural language processing (almost) from scratch. Journal of machine learning research 2011, 12, 2493-2537.

[4] Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems 2013, 26.

[5] Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. In Proceedings of the Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532-1543.

[6] Kim, Y. Convolutional Neural Networks for Sentence Classification. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2014.

[7] Peters, M.E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep Contextualized Word Representations. In Proceedings of the North American Chapter of the Association for Computational Linguistics, 2018.

[8] Zhou, C.; Sun, C.; Liu, Z.; Lau, F. A C-LSTM neural network for text classification. arXiv preprint arXiv:1511.08630 2015.

[9] Çano, E.; Morisio, M. Quality of Word Embeddings on Sentiment Analysis Tasks. In Proceedings of the International Conference on Applications of Natural Language to Data Bases, 2017.

[10] Rudkowsky, E.; Haselmayer, M.; Wastian, M.; Jenny, M.; Emrich, S.; Sedlmair, M. More than Bags of Words: Sentiment Analysis with Word Embeddings. Communication Methods and Measures 2018, 12, 140-157.

[11] Rezaeinia, S.M.; Ghodsi, A.; Rahmani, R. Improving the Accuracy of Pre-trained Word Embed- dings for Sentiment Analysis. ArXiv 2017, abs/1711.08609.

[12] Çano, E.; Morisio, M. Word Embeddings for Sentiment Analysis: A Comprehensive Empirical Survey. ArXiv 2019, abs/1902.00753.

[13] Jiao, Q.; Zhang, S. A Brief Survey of Word Embedding and Its Recent Development. 2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC) 2021, 5, 1697-1701.

[14] Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. ArXiv 2019, abs/1810.04805.

[15] Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 2013.

[16] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. Advances in neural information processing systems 2017, 30.

[17] Zhu, Y.; Kiros, R.; Zemel, R.S.; Salakhutdinov, R.; Urtasun, R.; Torralba, A.; Fidler, S. Aligning Books and Movies: Towards Story-Like Visual Explanations by Watching Movies and Reading Books. 2015 IEEE International Conference on Computer Vision (ICCV) 2015, pp. 19-27.

[18] Taylor, W.L. “Cloze Procedure”: A New Tool for Measuring Readability. Journalism & Mass Communication Quarterly 1953, 30, 415-433.

[19] Pang, B.; Lee, L. Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales. In Proceedings of the Annual Meeting of the Association for Computational Linguistics, 2005.

[20] Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C.D.; Ng, A.; Potts, C. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2013.

[21] Pang, B.; Lee, L. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. arXiv preprint cs/0409058 2004.