Leveraging Semantic and Sentiment Knowledge for User-Generated Text Sentiment Classification

Jawad Khan, Niaz Ahmad, Youngmoon Lee
Hanyang University
{jkhanbk1,niazahamd89,youngmoonlee}
@hanyang.ac.kr

Aftab Alam
Hamad Bin Khalifa University
afalam@hbku.edu.qa

Abstract

Sentiment analysis is essential to process and understand unstructured user-generated content for better data analytics and decision making. State-of-the-art techniques suffer from a high dimensional feature space because of noisy and irrelevant features from the noisy user-generated text. Our goal is to mitigate such problems using DNN-based text classification and popular word embeddings (Glove, fastText, and BERT) in conjunction with statistical filter feature selection (mRMR and PCA) to select relevant sentiment features and pick out unessential/irrelevant ones. We propose an effective way of integrating the traditional feature construction methods with the DNN-based methods to improve the performance of sentiment classification. We evaluate our model on three real-world benchmark datasets demonstrating that our proposed method improves the classification performance of several existing methods.

1 Introduction

Sentiment analysis is used to classify user-generated review/comments into positive and negative classes, and widely applied to various domains such as businesses and organizations, politics, health, education, etc. Existing proposals for text sentiment analysis can be mainly divided into lexicon-based and corpus-based approaches. Sentiment lexicons may ignore important domain-specific sentiment words incurring concerns with word coverage. Unlike lexicon-based approach, corpus-based approaches requires careful consideration of sentiment clues behind sentiment words, that is crucial for determining a text's sentiment orientation.

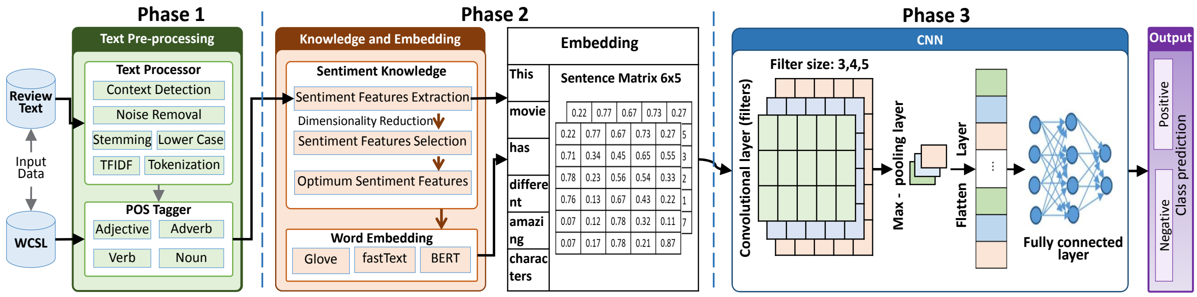
We propose an effective method for improving sentence-level classification performance by integrating the traditional feature construction method with the DNN-based method, while considering semantics, context and sentiment clue. First, we parse the review sentences and employ linguistic rules to identify mixed opinionated sentences. Then the POS tags are assigned to the sentiment bearing words: adjectives, adverbs, verbs, and nouns by Stanford POS tagger. Next we leverage the integrated wide coverage sentiment lexicon (WCSL) (Khan and Lee) as the semantic and sentiment information to identify and extract sentiment bearing words. After that, we employ statistical features reduction algorithms namely mRMR (Ding and Peng) and PCA (Wold et al.) for optimum features selection. Further we process the optimum sentiment features and convert them to word vector by employing word embedding methods (e.g., Glove, fastText, and BERT). Finally, we apply a CNN classifier to process the word vector/vector embedding and predict the sentiment class of each sentence.

Our main contribution is summarized as follow: (1) We use semantic and sentiment knowledge, linguistic rules, and integrated WCSL to identify and extract the sentiment features in the sentence. (2) We reduce the dimensionality of feature space by employing the mRMR and PCA statistical filter algorithms to filter out redundant features and select the optimum sentiment features. (3) The experimental results of our proposed method using three real word benchmark domain datasets show that the suggested sentiment analysis model improves the performance of several previous baseline methods significantly.

2 Related Work

Many traditional feature-based machine learning methods have been largely used for textual sentiment classification (Tripathy et al., 2016; Yousefpour et al., 2017; Chang et al., 2020). These approaches have employed Bag of Words, high order n-grams, Part of speech (POS) patterns and linguistic patterns for sentiment features representation and sentiment classification. While traditional feature-based selection approaches might lower the

Proceedings of the 2022 COLING Workshop: The 8th Workshop on Noisy User-generated Text (W-NUT 2022), pages 101–105



0.22	0.77	0.67	0.73	0.27
0.71	0.34	0.45	0.65	0.55
0.78	0.23	0.56	0.54	0.33
0.76	0.13	0.67	0.43	0.22
0.07	0.12	0.78	0.32	0.11
0.07	0.17	0.78	0.21	0.87

Figure 1: Proposed framework consists of three phases: text pre-processing, knowledge and embedding, and CNN.

dimensionality of textual data to improve classification performance, classifiers still face sparsity issues due to a lack of adequate data representation strategies. **Word embeddings**: Word2vec, Glove, fastText, and BERT (Zhang and Wallace, 2015; Mikolov et al., 2017; Kenton and Toutanova) are alternative approaches recently used for the dense representation of words of the text of analysis.

Deep neural network (DNN) models: CNN, BiLSTM, and BiGRU with word embeddings have achieved tremendous results in textual sentiment analysis (Kim; Rezaeinia et al., 2017; Lei et al.; Huang et al.; Khasanah, 2021). However, according to recent studies, DNN-based methods select some irrelevant and redundant features and also ignore the sentiment clue behind each sentiment word which affects its performance in terms of classification accuracy (Rezaeinia et al., 2017; Ayinde et al.; Denil et al., 2013). Although traditional feature-based methods have benefits in interpretability and time complexity, DNN-based methods outperform classic feature-based methods.

3 Methodology

Our proposed effective method for sentiment classification is composed of three main phases: (1) text pre-processing (2) knowledge and embedding (3) CNN architecture. The overall framework of our proposed method is shown in Figure 1.

3.1 Text Pre-processing

We employ the text pre-processing method to create the initial feature space. The review dataset is loaded first, followed by sentence parser and tokenizer. The noise removal and text transformer module is then used to remove noisy text (e.g., stop words, URLs, numeric symbols, etc.), and convert the text to lowercase respectively. Next the POS tagger is employed to assigns POS tags to the likely words such as adjectives, adverbs, verbs and nouns. Furthermore, these words are searched in the integrated WCSL to identify and extract the sentiment words. We also employ linguistic rules following the work of (Appel et al., 2016; Khan et al., 2021) to identify the context of text sentiment and discriminate synonyms from antonyms. Linguistic rules provide help to the context-based sentiment analysis that comprises differing viewpoints. For example, in the statement “the filmmaker is well-known but the film is dull" linguistic norms only consider the clause after “but" whereas the clause preceding “but" is omitted. It comprises certain words that can change the polarity of a statement, such as “but" “despite" “while" “unless" and so on.

3.2 Sentiment Knowledge and Embedding

We leverage semantic and sentiment knowledge using integrated wide coverage sentiment lexicons to identify, extract and select the relevant sentiment features for word embedding and sentiment classification (Khan and Lee).

Integrated Wide Coverage Sentiment Lexicons In literature different sentiment lexicons (Khan et al., 2021) such as AFFIN, OL, SO-CAL, WordNet-Affect, GI SentiSense, MPQA Subjectivity Lexicon, NRC Hashtag Sentiment Lexicon, SenticNet5, and SentiWordNet with different sizes have been built. There is no one-size-fits-all general sentiment lexicon that can be utilized for sentiment analysis. We standardize them by assigning scores, +1, -1, 0 to positive, negative, and neutral words respectively. Then for integration, we take the average of the sentiment score of the overlapping words, which produces a huge sentiment lexicon with more sentiment words that we called WCSL. In this study sentiment words in the review sentences are matched against integrated WCSL and then used for sentiment classification.

Sentiment Features Extraction For reliable model learning, it’s crucial to identify and extract the right features. Specifically, we employ Stanford POS tagger (Toutanvoa and Manning) to assign POS tags to the content words such as adjectives, adverbs, verbs, and nouns and then identify the sentiment orientation of these words/features in the integrated WCSL.

Sentiment Features Selection We utilize two statistical filter-based algorithms namely minimum redundancy-maximum relevance (mRMR) and Principal component analysis (PCA) for feature reduction and selection. We use the mRMR and PCA feature selection techniques to reduce the feature space and select the subset of most acceptable top k high ranked features.

3.3 Word Embedding

We employ popular word embedding methods (Glove, fastText and BERT) to convert words into real-valued, low-dimensional vectors and extract useful syntactic and semantic information from them. The BERT-generated word vector has better quality features. In this study, we utilize these embedding algorithms for vectorization and sentiment classification.

3.4 CNN Architecture

We train our proposed system employing the CNN model, which is made up of four layers.

Input layer In this layer the tokenized input sentence is represented in our model by the matrix $D \in R^{m \times d_i}$, where d_i is the word embedding vector dimension of each word and m is the number of words in the sentence. Each sentence is padded with a zero vector to ensure that all the review sentences are the same size. The embedding matrix for each word in the sentence D is expressed in the embedding layer as:

$$M_e = \{V_{t_1}, V_{t_2}, ..., V_{k_i}, ..., V_{k_m}\}, \qquad (1)$$

where V_{t_i} is the word vector and V_{k_i} is the placeholder for it in the embedding space.

Convolutional Layer The second layer is convolution layer and it is applied to the word embedding matrix M_e attained in the preceding layer. Assume that the convolution kernel $K^c \in R^{h \times l}$ has the following properties: c represents the number of convolution kernels, l indicates the length of the convolution kernel, and h represents the width of the convolution kernel. For the input matrix $D \in R^{m \times d_i}$, the feature map is created $P = \{p_1, p_2, ..., p_{n-h}\} \in R^{m-h+1}$ by repeatedly applying a convolution kernel R to perform convolution operation. Over the convolution output, the ReLU activation is applied.

Max-pooling Layer The max-pooling layer is the third layer, and it is applied to each feature map and takes the maximum value $\hat{c} = max\{c\}$(Collobert et al.). The max-pooling procedure is used in this study to save the most significant features (Kalchbrenner et al., 2014). These features are then concatenated and sent to the fully connected layer which is the final layer

Fully-connected Layer The main goal of a fully connected layer is to use the outputs of the convolution and pooling layer to processes and classify them into a label. A sigmoid function is utilized to get the final output. The probability distribution on the label is the output.

4 Experiments

Experimental Setup We tested our system using three real-world benchmark datasets: (1) Movie Reviews (MR) (Pang and Lee, 2005), (2) Stanford Sentiment Treebank (SST-2) datasets (Socher et al.), (3) Customer Review datasets (CR) (Hu and Liu). MR composed of 5331 positive and 5331 negative review samples. SST-2 contains positive and negative sentences, there are 9,613 single sentences in the dataset, which were obtained from movie reviews. CR consists of 14 products extracted from Amazon (Hu and Liu). SST-2 have standard training–test splits. MR and CR do not have such a standard split, we apply 10-fold cross validation, which is consistent with previous research (Huang et al.) on the dataset. We hold out 10 % of the training data for MR and CR for development purposes (e.g. for early stopping), we adopt classification accuracy as an evaluation measure. We generate 300-dimensional word vectors for GloVe and fastText embedding. The BERT-BASE model case version (network layers L = 12, hidden layer dimension H = 768, attention=12, total number of parameters surpass 110 M, Learning rate for Adam = 2e-5) was utilized as the pre-trained BERT model for word vectorization. We employed wide coverage sentiment lexicon (WCSL) for sentiment information extraction from review texts. We used mRMR and PCA filter-based feature selection algorithm for top k optimum feature selection. Top 2000 features of MR, 1500 features of SST-2, and 1000 features of CR dataset are feed to each channel in CNN. The dropout rate for each network's layer is 0.5, and

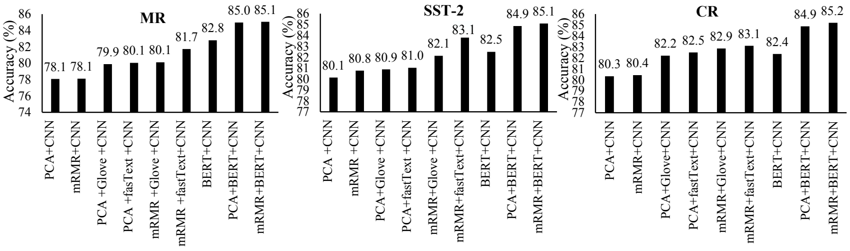


Figure 2: Ablation results of each component of (PCA, mRMR, Glove, fastText, BERT, and CNN) for different datasets (MR, SST-2, CR).

Table 1: Comparative Study

Model	Dataset	Accuracy
Rezaeinia *et al.* 2017	MR	79.80
Lei *et al.* 2018	MR	84.30
Huan *et al.* 2020	MR	79.45
Khasanah *et al.* 2021	MR	80.00
Our model	MR	85.12
Rezaeinia *et al.* 2017	SST-2	83.70
Huan *et al.* 2020	SST-2	84.34
Khasanah *et al.* 2021	SST-2	83.90
Our model	SST-2	85.10
Rezaeinia *et al.* 2017	CR	83.70
Our model	CR	85.20

the layers activation function is Rectified Linear Unit (ReLU). The sigmoid is used for the probability of class label in the fully connected layer. The proposed model and the other baseline models are implemented using the Rapidminer Studio (visual workflow designer) and tensorflow Keras library (High-level neural networks) in python. In our proposed model, the filter sizes of convolution1, convolution2, and convolution3 are 3,4, and 5, respectively, with 100 feature maps. The dropout rate is 0.5, l_2 constraint is (s) 3, mini-batch size is 5, and the layers activation function is Rectified Linear Unit (ReLU). The sigmoid is used for the probability of class label. We used the paired t-test (P<0.05) to calculate the evaluation measures of proposed model.

Experimental Results The ablation results of our proposed approach in terms of accuracy for each component with and without embeddings, with different feature selection methods is shown in Figure 2. From Figure 2, we can observe that selecting and representing relevant sentiment feature in a real valued vector/dense representation boost classification performance.We compare our approach with state-of-the-art DL approaches (Rezaeinia et al., 2017; Lei et al.; Huang et al.; Khasanah, 2021) that employed CNN-based model, multi-head attention convolutional network, and DNN models with fastText embedding respectively for sentence-level sentiment classification as shown in Table 1.

Model Analysis We explore the performance of our semantic and sentiment-aware CNN model. From Table 1, it is clear that our proposed model outperform baseline models on three benchmark datasets significantly. There are five reason why the proposed model achieves the best and comparable results. The first reason is that during text pre-processing, noisy and irrelevant features are removed from the text. The extraction and selection of relevant sentiment features is the second reason. The third reason is to classify mixed-opinionated texts using linguistic rules and semantic information. The integration of WCSL for sentiment features identification is the fourth reason. The fifth reason is the dense representation of sentiment features in a real valued vector, and fine tuning the proposed semantic and sentiment aware sentiment analysis model.

5 Conclusion

We propose an effective way of integrating the traditional feature construction method with the deep learning method to improve the overall performance of sentiment classification. To this end, we leverage semantic and sentiment knowledge using integrated WCSL to extract and select the relevant sentiment features for word embedding and sentiment classification. By employing mRMR and PCA filter-based algorithms and pre-trained embedding models (Glove, fastText, and BERT) to select optimum sentiment features and consider the semantics and context of words, we can filter out irrelevant and redundant features and reduce the dimensionality of feature space. In-depth experiments with three benchmark domain datasets demonstrate the effectiveness of the proposed model.

Acknowledgement

This work was supported by the research fund of Hanyang University (HY-2021-0372, HY-2022-0010, HY-2023)

References

Orestes Appel, Francisco Chiclana, Jenny Carter, and Hamido Fujita. 2016. A hybrid approach to the sentiment analysis problem at the sentence level. *Knowledge-Based Systems*, 108:110–124.

Babajide O Ayinde, Tamer Inanc, and Jacek M Zurada.

On correlation of features extracted by deep neural networks. In *IJCNN*.

Jing-Rong Chang, Hsin-Ying Liang, Long-Sheng Chen, and Chia-Wei Chang. 2020. Novel feature selection approaches for improving the performance of sentiment classification. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–14.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12.

Misha Denil, Babak Shakibi, Laurent Dinh, Marc'Aurelio Ranzato, and Nando De Freitas. 2013. Predicting parameters in deep learning. *arXiv preprint arXiv:1306.0543*.

Chris Ding and Hanchuan Peng. Minimum redundancy feature selection from microarray gene expression data. *Journal of bioinformatics and computational biology*, 3.

Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *SIGKDD*.

Hui Huang, Yueyuan Jin, and Ruonan Rao. Sentiment-aware transformer using joint training. In *ICTAI*.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *naacL*.

Jawad Khan, Aftab Alam, and Young-Koo Lee. 2021. Intelligent hybrid feature selection for textual sentiment classification. *IEEE Access*.

Jawad Khan and Young-Koo Lee. Lessa: A unified framework based on lexicons and semi-supervised learning approaches for textual sentiment classification. *Applied Sciences*, 9.

Isnaini Nurul Khasanah. 2021. Sentiment classification using fasttext embedding and deep learning model. *Procedia Computer Science*, 189:343–350.

Yoon Kim. Convolutional neural networks for sentence classification. In *EMNLP*.

Zeyang Lei, Yujiu Yang, and Min Yang. Saan: a sentiment-aware attention network for sentiment analysis. In *SIGIR*.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2017. Advances in pre-training distributed word representations. *arXiv preprint arXiv:1712.09405*.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. *arXiv preprint cs/0506075*.

Seyed Mahdi Rezaeinia, Ali Ghodsi, and Rouhollah Rahmani. 2017. Improving the accuracy of pre-trained word embeddings for sentiment analysis. *arXiv preprint arXiv:1711.08609*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*.

Kristina Toutanvoa and Christopher D Manning. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *SIGDAT*.

Abinash Tripathy, Ankit Agrawal, and Santanu Kumar Rath. 2016. Classification of sentiment reviews using n-gram machine learning approach. *Expert Systems with Applications*, 57:117–126.

Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2.

Alireza Yousefpour, Roliana Ibrahim, and Haza Nuzly Abdel Hamed. 2017. Ordinal-based and frequency-based integration of feature selection methods for sentiment analysis. *Expert Systems with Applications*, 75:80–93.

Ye Zhang and Byron Wallace. 2015. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*.