Convolutional Neural Network Based Sentiment Analysis using Adaboost Combination

Yazhi Gao[1], Wenge Rong[1], Yikang Shen[2], Zhang Xiong[1]
[1]School of Computer Science and Engineering, Beihang University, Beijing 100191, China
[2]Sino-French Engineer School, Beihang University, Beijing 100191, China
{yazhi.gao, w.rong, yikang.shen, xiongz}@buaa.edu.cn

***Abstract*—Sentimental polarity detection has long been a hot task in natural language processing since its applications range from product feedback analysis to user statement understanding. Recently a lot of machine learning approaches have been proposed in the literature, e.g., SVM, Naive Bayes, recursive neural network, auto-encoders and etc. Among these different models, Convolutional Neural Network (CNN) architecture have also demonstrated profound efficiency in NLP tasks including sentiment classification. In CNN, the width of convolutional filter functions alike number N in N-grams model. Thus, different filter lengths may influence the performance of CNN classifier. In this paper, we want to study the possibility of leveraging the contribution of different filter lengths and grasp their potential in the final polarity of the sentence. We then use Adaboost to combine different classifiers with respective filter sizes. The experimental study on commonly used datasets has shown its potential in identifying the different roles of specific N-grams in a sentence respectively and merging their contribution in a weighted classifier.**

***Keywords*-Sentiment Analysis, Convolutional Neural Networks, Adaboost**

I. Introduction

With the proliferation of social network based services, a considerable quantity of information has been accumulated on blogs, tweets or other social media expressing people's opinions and emotions. How to efficiently extract polarity orientations from these statements, paragraphs, and sentences has become an essential challenge [1]. As an important mechanism to study documents' polarities, sentiment analysis has achieved increasing popularity among scientific and industrial research and has been widely used to predict people's attitudes in different domains, e.g., product recommendation [2], social event identification [3], political election prediction [4], and etc.

To better analyze sentimental polarity of online information, a lot of approaches have been proposed in the literature and can be roughly divided into two categories, i.e., linguistics oriented methods and machine learning oriented methods [5]. Compared with linguistics methods which mainly employ individual words' pre-defined polarity scores to predict the document's overall polarity, machine learning based approaches mainly employ learning models and consider sentimental analysis as a binary classification task [6].

In the machine learning oriented approaches, there are normally two main important tasks, i.e., how to efficiently represent the documents, and how to design an effective learning model. As to the document representation, traditional Bag-of-Words (BoW) has been widely employed due to its simplicity and robustness. However, BoW based approaches suffer from several problems such as difficulty in obtaining syntactic and semantic information of the words [7]. To overcome the syntactic deficiency, several approaches have been proposed to employ parse trees or dependency trees for the syntactic structure analysis [8]. Similarly, a lot of methods have also proposed to add semantic information e.g., LDA [9], word embeddings [10], and etc.

In the process of developing sophisticated models, a lot of efforts have been devoted e.g., Maximum Entropy, Naive Bayes, SVM [11], [12], and have been proven successful in sentimental analysis. Recently, with the popularity of deep learning concept, several deep learning architectures have also been proposed in the literature, e.g., auto-encoder, recursive neural network and long short term memory network [13], [14], [15]. These kinds of methods try to integrate the features extractor inherently and to learn representations of profound complexity to undertake the complicated natural language processing tasks.

As an important deep learning technique, the convolutional neural network (CNN) has recently been well studied and widely employed for search query retrieval [16], sentence modelling [17], sentence matching [18], question/answering matching [19] and document modelling [20]. Due to its capability of deep representation learning and sentence-level information awareness, it has also been adopted for sentimental analysis [21], [22].

The CNN model can use filters of different length to generate different feature maps. With regard to language processing tasks, the filters of various lengths can be considered as different kinds of N-gram scanners which differently interpret and parse sentence phrases. Features from N-grams of varying lengths might play different roles in the ultimate decision of sentiment classification. Considering a sentence "this loose collection of largely improvised numbers would probably have worked better as a one-hour TV documentary", while 3-grams would probably detect "have worked better" as a major contributor to positive feelings, 5-grams would span their vision and see "would probably have worked better" as a negative indicator. As such it is argued that when words are scanned in a proper combination, they might engender better features and thereby improving the overall polarity detection

978-1-5090-0620-5/16/$31.00 ©2016 IEEE

[23].

In this paper, following this inspiration, we tackle the sentiment analysis problem by proposing a boosted CNN architecture consisting of different filters. Unlike the aforementioned work [23], which concatenates features of different kinds together and inputs them into a single classifier, our proposed model injects different features into different classifiers with different convolutional layers followed respectively by their own pooling and softmax layers. After each classifier obtains the preliminary result, the Adaboost [24] mechanism will be employed to produce the overall classification result since Adaboost is able to combine weak classifiers into a strong classifier because it learns the classification error of each weak classifier and adjusts their contribution weights to the final classification [25]. Experimental studies on commonly used Cornell Movie Review and IMDB have shown its promising potential.

The remainder of this paper is organized as follows. In Section 2, we put attention to the recent advances of sentiment analysis techniques. Section 3 will focus on illustrating the proposed model in details. Section 4 will elaborate the experimental study. Section 5 concludes this paper and points out some potential future work.

II. Related Work

Due to its importance in gathering people's opinions and emotions, a lot of research has been proposed for the task of sentiment analysis, including computational linguistics methods and machine learning based approaches. Since machine learning based methods do not need hand-crafted features, it has shown promising potential in dealing with polarity detection task. Recently with the popularity of deep learning technology, a lot of deep architectures have been proposed, e.g., auto-encoder, recursive neural network and etc [13], [14], [15], among which the convolutional neural network (CNN) has been attached much attention to.

CNN based approaches have proven their capability in analyzing not only short texts like twitter corpus [26], [22], but also ordinary sentences like movie reviews [27]. Specialized for extracting the limited information from tweets shorter than 140 characters, [26] presented a model in which they incorporate two fully connected layers with non-linear functions. Through careful pre-training of the parameters, the model achieved promising results in SemEval Task 10. They also conducted experiments comparing the classification efficiencies by using different pre-trained word embeddings.

A model [22] was presented for its simple structure of single convolutional layer, fine-tuning of the embeddings and the careful initialization of the parameters. Convolutional neural networks which are susceptible to local parameters optima, call for special attention when initializing their parameters. To train a model of good generalization, they aimed to get high quality word embeddings. Other than only using **word2vec** neural language model [10] to train the word embeddings, they also use the distant supervision approach to further tune the embedddings to teem with sentiment information. The model nailed high ranks in **Semeval-2015**[1] task on twitter dataset.

Also, to deal with the challenge in analyzing short texts like tweets, a deep CNN model [21] was advanced which exploits character-to-sentence-level information to aid the sentiment analysis. This model maps the words in a sentence onto a concatenation of both word-level embeddings and character-level embeddings to fully utilize the scanty contextual information tweets have to offer. The character-level embeddings are reported capable of capturing the morphological and shape information in the sentence. The word embeddings and character embeddings are passed to the convolutional layers to generate the final features - the sentence embedding vectors. This kind of features demonstrated strong generalization of the sentiment information in the sentence and improved the classifiers' accuracy on several corpora.

Recently a multi-channel based CNN [23] has shown great promise in sentiment classification. By setting up a single convolutional layer structure, it adds the word embeddings to the model parameters and tunes them along training to obtain task-specific vectors thus boosting the classification. The model also adopts the convention of using filters of varying length to extract the sentiment features. It generates three kinds of feature and places them together to form an input for the classifier.

III. Model Architecture

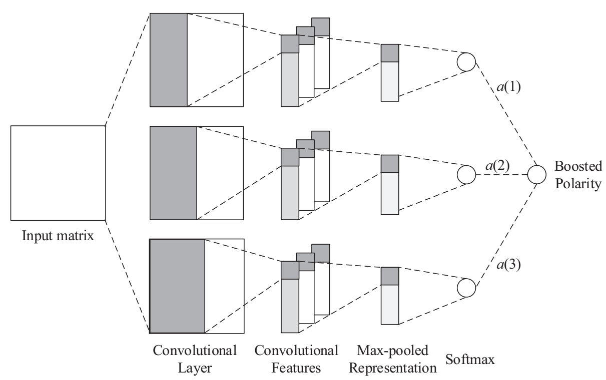

Fig. 1. Boosted Convolution Neural Network for Sentimental Analysis

In this paper, the proposed model with CNN and Adaboost is as shown in Fig.1, where different filters with varying windows sizes to scan the input sentences were employed. Each CNN has its own convolutional layer, one pooling layer and one softmax layer for classifying sentences into positive and negative classes. Afterwards, Adaboost [24] aggregating method will be used to construct the element-wise product of the classification results with different classifiers weights, thereby achieving the final polarity decision.

[1] http://alt.qcri.org/semeval2015/task10/

A. Sentence Matrix Input

To conduct sentimental analysis, the first step is to construct a proper representation of the target documents. In this research, the input to the proposed model is sentences in terms of sequences of words, and presented as their word embeddings [10] to form the input matrix.

Assuming that we have a sentence of n token words $\mathcal{S} = \{w_1, \cdots, w_n\}$, words are corresponded with their pre-trained vectors $w \in \mathbb{R}^{1 \times d}$ looked up in a vocabulary of the embeddings matrix $W \in \mathbb{R}^{d \times V}$, where V is the total count of the words vocabulary. This matrix is a stack of all the embeddings in the vocabulary. In this research, we map words onto the indices $1, \cdots, V$ in the vocabulary. After the words processing, a sentence input is treated as a concatenation of word embeddings $\mathcal{S} \in \mathbb{R}^{d \times n}$, where each column i vector stands for a word's embedding $w_i \in \mathbb{R}^{1 \times d}$. Afterwards we can now input this matrix into the proposed model for further processing.

B. Convolution Operation

The convolution operation is implemented to go across the whole input matrix and to extract sentiment features from the N-grams which the filters convolved themselves with. A filter $F \in \mathbb{R}^{d \times h}$ of size h convolves the weights with the window of word embeddings $w_{i:i+h-1}$ to generate the features. Learning the weights F and biases b is the main process of convolutional layers training. Here g is a non-linear function such as hyperbolic tangent in our experiment. A feature c_i is generated as follows:

$$c_i = g(F * w_{i:i+h-1} + b) \tag{1}$$

The filter scans across all the N-grams of size h in the sentence, the segmented sequence of the sentence of length n $\{w_{1:h}, w_{2:h+1}, \cdots, w_{n-h+1:n}\}$. Thus the filtering procedure produces a feature of size n-h+1 over one sentence input. A set of filters generates feature maps that need pooling operation to extract the most sensitive responses over the features. In this paper, we use filters of varying sizes to extract features from different views of the sentences.

C. Pooling

The feature maps are then passed onto the pooling operation layer to aggregate the best responses while keeping the sequential information of the features. We apply the max-over-time pooling method [28] and take the maximum feature value c_{max} among one map C :

$$c_{max} = \max\{C\} = \max\{c_1, \cdots, c_{n-h+1}\} \tag{2}$$

This method renders the architecture aware of both the spacial order and distribution of the sentiment information of the whole sentence and this scheme can enable us to deal with variable sentence lengths since the number of features is now aligned with the number the filters. The pooled features are then processed by a non-linear before injection into the classifier.

D. Regularization and Softmax

To deal with the problem of parameter overfitting which is always the weakness of neural models, dropout regularization [29] is utilized in this research. Dropout can randomly drop out some parameters of the hidden units in the classifier. That is, for the features pooled from the previous layer $C_{max} = [c^1_{max}, c^2_{max}, \cdots, c^m_{max}]$ (suppose that we have m filters in the convolutional layer), we set some portion of them to be zero when we pass them to the softmax layer and only the unaffected unit can serve to calculate gradients. The proportion of dropped units is a hyper-parameter in experiment configuration. The result of the classification is the output of the softmax layer following the pooling layer. This layer using the regularized features computes the probability distribution of the input over all the labels where in our case we only deal with the binary labelling task. The following depicts the softmax process.

$$P(y = j|x) = softmax_j(x^T w + b) \quad = \frac{e^{x^T w_j + b_j}}{\sum_{k=1}^{K} e^{x^T w_k + b_k}} \tag{3}$$

Here w_k and b_k denotes the weight and bias parameters in the classifiers for the k-th class emotion.

E. Training Adaboost Model and Merging Prediction

Adaboost is an algorithm which can combine weak classifiers into a strong classifier. In this research we use this mechanism to find the appropriate weights for the classifiers adjusted to different N-grams. The classifiers map sentences to a space of {+1,-1}, where +1 stands for positivity and -1 indicates negativity. Due to the application of Adaboost, we need to get the statistics of the weak classifiers results on the training samples and adjust the weights of the training samples and classifiers to attain the eventually used strong classifier. The networks ahead of the Adaboost Merging part are regularly trained by using back propagation. The Adaboost training procedure is as follows:

1. Initialize the distributions $\mathcal{D}^1$ for the training samples equally,$\mathcal{D}^i$ denotes the ith distribution of the training samples.:

For any sample index i, $\mathcal{D}^1_i = \frac{1}{\#training\_samples}$

2. In one training epoch t:

Three network classifiers are trained consecutively by back propagation, so the following process are applied to all the classifiers in one epoch.

1) **weak classifiers statistics estimation:** After the classifiers are trained and output prediction labels,the architecture saves the statistics of their classification over the samples, and then gets the errors on the weak classifier $G_m(x)$,

$$e^t_m = \sum_i \mathcal{D}^t_i \boldsymbol{I}(G_m(x) \neq y(x)) \tag{4}$$

2) **weights adjustment:** Every time a weak classifier is trained, the architecture uses the classification errors to alter the distribution over the training set and calculates the error index and weight of the weak classifier.

classifier weights calculation:

$$a(m) = \frac{1}{2} \ln \frac{1 - e_m^t}{e_m^t} \quad (5)$$

distribution adjustment:

$$\mathcal{D}_i^{t+1} = \frac{\mathcal{D}_i^t exp(-a(m)y(x)G_m(x))}{\mathcal{D}^t} \quad (6)$$

3) **boosted validation:** After training the classifiers, we perform element-wise multiplication of the outputs and the weights, getting the final predicted class the sentence belongs to. We use the learned weights a to do the boosted validation by the following equations (i is the classifier index, a specifies the ensemble of the classifiers' weights and l is the output labels of the classifiers):

$$L(s) = \sum_i a(i) * l(i) \quad (7)$$

IV. Experimental Study

In this section, we conduct experiments to evaluate the qualities of the proposed model against other traditional sentimental analysis methods. The experiments tries to answer 1) whether the proposed model have a better prediction accuracy or not compared with the state-of-the-art techniques, and 2) how the weight adjustment will affect the prediction accuracy and what the best strategy to adjust these parameters is.

A. Data Collection

We tested our model on two commonly used datasets for sentiment analysis, i.e,, **Movie Review**[2] and **IMDB**[3]. Both datasets contain movie reviews in positive/negative classes. The detailed statistics for the corpus in Table I. As to the training and testing set, we follow the configuration adopted in [23], which selects 90% of a random permutation of the dataset to train the model while using the rest 10% data to evaluate the trained model and repeats this method 10 times for the 10 folds. In this research, we apply this strategy to both datasets.

TABLE I
Statistics of Dataset

Dataset	Movie Review	IMDB
#size	10662	50000
#average sentence length	20	225
Positive/negative ratio	0.5/0.5	0.5/0.5

B. Experiment Setup

To conduct experimental study, we need to firstly set some hyperparameters. For the filters length, we follow the setting in [23] and also choose filters of window size as 3, 4, 5 respectively and each kind of filter contains 100 instances. For the dropout rate, we also set it as 0.5 and the cross validation is conducted with the training set selected randomly from permutation of the original dataset. We also apply Adadelta [30] to strengthen our gradients calculation at a reasonable rate.

Furthermore, initializing the parameters of the neural network is a must in getting satisfactory experimental results due to the neural model's vulnerability to local optima. To this end, we use the public **Google News word2vec**[4] word embeddings which were trained on 100 billion words from Google News for input matrix generation. The vectors have 300 dimensions. For the words not in the Google embeddings vocabulary, they are randomly initialized in this research.

C. Baselines

For the MR Dataset, we refer to the following methods in [31], [13], [14], [27], [23]:

1. Bag-of-Words [31]: This method utilizes a logistic regression model with BoW feature from documents to predict its polarity.

2. Vote by lexicon [31]: This method employs each word's polarity score to vote the if the whole statement is positive or negative.

3. Rule-Based Reversal [31]: This method uses pre-defined rules to detect the whole sentence's polarity.

4. Tree-Based CRF [31]: This method employs conditional random fields with hidden variables to classify sentiment polarities.

5. RAE [13]: This method proposes a recursive autoencoder with the pre-trained word vectors learned from Wikipedia. The vector space representations for multi-word phrases are constructed and aggregated to form a sentence-level classification.

6. MV-RNN [14]: The method of matrix-vector recursive neural network is designed with parse trees. The model assigns a vector and a matrix to every node in the syntactic parse tree of the sentence: the vector captures the semantic meaning of the words, while the matrix measures the node's interrelation to neighbouring words or phrases.

7. AEBPA [27]: This is an auto-encoder based bagging architecture for sentiment analysis, which uses stacked autoencoders to reduce the dimension of the representations and cuts down generalization errors by bagging.

8. CNN-Static [23]: A convolutional neural network for sentence classification in which 3 different filters are employed the generated features are directly combined together.

For the IMDB Dataset, we refer to the following methods revealed in [27], [23], [32]:

1. LDA [32]: The topic modelling method considers a statement as mixtures of latent topics and then uses the topics to predict the statement's polarity

2. LSA [33]: This method uses latent semantic analysis for sentiment analysis.

3. MAAS Semantic [32]: This method is a probabilistic model to optimise a objective function which only contains semantic information.

4. MAAS Full [34]: A probabilistic model which optimizes a objective function containing both semantic and sentiment information.

[2]http://www.cs.cornell.edu/people/pabo/movie-review-data/

[3]http://ai.stanford.edu/ amaas/data/sentiment/

[4]https://code.google.com/p/word2vec/

5. AEBPA [27]: This is an auto-encoder based bagging architecture for sentiment analysis, which uses stacked autoencoders to reduce dimension of the representations and cut down generalization errors by bagging.

6. CNN-Static [23]: A convolutional neural network for sentence classification in which 3 different filters are employed the generated features are directly combined together.

D. Evaluation Metrics

The accuracy of this model follows the accuracy of prediction defined [35]:

$$Accuracy = \frac{\sum_{i=1}^{n} 1 \times (l_i == y_i)}{\#test\ instances} \quad (8)$$

Here l_i stands for the label our architecture predicts for the sentence input and y_i refers to the original class it belongs to.

E. Results and Discussion

In Tables II and III, we present the results of the proposed model with comparison to other baseline approaches. These results suggest that the proposed boosted model can yield improved result for sentiment information extraction in the texts and leverage the contribution of the different features to form a better classifier. It is also found that the deep learning models do present better achievement on average than linguistic oriented methods using lexical and/or semantic information conveyed by the sentences. Furthermore, the CNN architecture proves its capacity in exploiting the information by sequentially scanning the combination of words and keeping the sequential information intact at pool operation which bridges the space of the information at both ends of the sentence.

TABLE II
PERFORMANCE SUMMARY

Approach	Movie Review
Bag-of-Word	76.4%
Voting by lexicon	63.1%
Rule-Based Reversal	62.9%
Tree-Based CRF	77.3%
RAE	77.7%
MV-RNN	79.0%
AEBPA	78.8%
CNN-static	81.0%
Our boosted model	81.2%

TABLE III
PERFORMANCE SUMMARY

Approach	IMDB
LDA	67.4%
LSA	83.9%
MAAS Semantic	87.3%
MAAS Full	87.4%
AEBPA	87.7%
CNN-static	89.2%
Our boosted model	89.4%

Furthermore, we also present the importance of weights adjustment for the boosted CNN model. The following Table IV shows some learned weights when some mediocre and the best performance on test come out on IMDB. We extract the weights from different training epoch after witnessing the test performance growth. From the weights learned in the table, it is discovered the ascending trend of 4-gram and 5-gram classifiers' weights when the whole architecture escalates to its best. The 4-word and 5-word length phrases seem to play a vital role in the contribution to the polarity while the 3-word length phrases' weights are generally less than the other two features. As such it is believed that the 3-word phrases are not that strong to generalize the information embedded in the relatively long sentence of IMDB Dataset and combination of more words to lead to a more accurate prediction.

TABLE IV
WEIGHTS LEARNED ON IMDB DATASET

Performance	3-gram weight	4-gram weight	5-gram weight
84.8%	0.58	0.11	0.31
86.7%	0.40	0.08	0.52
87.3%	0.07	0.54	0.39
89.2%	0.15	0.39	0.46
89.4%	0.17	0.44	0.39

V. CONCLUSION AND FUTURE WORK

In this research, we proposed a boosted CNN model to deal with the problem of sentiment analysis. Different from previous work which integrates different features generated from different filters together, the proposed model separates the features by parsing the documents from different filters, and then boosts the classifiers trained on these representations and modulates them to reach high performance for binary classification task. The experimental study has shown its promising potential.

Although the model gains improvements over compared methods, to better investigate the role of N-gram features in the neural network, we plan to integrate this weighting system into the classifier itself. Moreover, inspired by the work by DCNN [17], we wish to make more than one convolutional layer to process the data and view the features as different channels of the original information. Furthermore, [23] utilized non-static word embeddings where word embeddings were optimized for the task at hand and demonstrated great performance improvement. We will try to incorporate this setup in the future work.

ACKNOWLEDGEMENTS

This work was partially supported by the National Natural Science Foundation of China (No. 61332018), the National Department Public Benefit Research Foundation of China (No. 201510209), and the Fundamental Research Funds for the Central Universities.

REFERENCES

[1] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Foundations and Trends in Information Retrieval*, vol. 2, no. 1-2, pp. 1–135, 2008.

[2] R. Dong, M. P. O'Mahony, M. Schaal, K. McCarthy, and B. Smyth, "Sentimental product recommendation," in *Proceedings of 7th ACM Conference on Recommender Systems*, 2013, pp. 411–414.

[3] X. Zhou, X. Tao, J. Yong, and Z. Yang, "Sentiment analysis on tweets for social events," in *Proceedings of 17th IEEE International Conference on Computer Supported Cooperative Work in Design*, 2013, pp. 557–562.

[4] A. Tumasjan, T. O. Sprenger, P. G. Sandner, and I. M. Welpe, "Predicting elections with twitter: What 140 characters reveal about political sentiment," in *Proceedings of 4th International Conference on Weblogs and Social Media*, 2010.

[5] K. Kim and J. Lee, "Sentiment visualization and classification via semi-supervised nonlinear dimensionality reduction," *Pattern Recognition*, vol. 47, no. 2, pp. 758–768, 2014.

[6] E. Boiy and M. Moens, "A machine learning approach to sentiment analysis in multilingual web texts," *Information Retrieval*, vol. 12, no. 5, pp. 526–558, 2009.

[7] S.-M. Kim and E. H. Hovy, "Automatic identification of pro and con reasons in online reviews," in *Proceedings of 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, 2006.

[8] T. Wilson, J. Wiebe, and P. Hoffmann, "Recognizing contextual polarity in phrase-level sentiment analysis," in *Proceedings of 2005 Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, 2005.

[9] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, 2003.

[10] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of 27th Annual Conference on Neural Information Processing Systems*, 2013, pp. 3111–3119.

[11] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," *Cs224n Project Report*, 2009.

[12] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? sentiment classification using machine learning techniques," in *Proceedings of 2002 Conference on Empirical Methods in Natural Language Processing*, 2002, pp. 79–86.

[13] R. Socher, J. Pennington, E. H. Huang, A. Y. Ng, and C. D. Manning, "Semi-supervised recursive autoencoders for predicting sentiment distributions," in *Proceedings of 2011 Conference on Empirical Methods in Natural Language Processing*, 2011, pp. 151–161.

[14] R. Socher, B. Huval, C. D. Manning, and A. Y. Ng, "Semantic compositionality through recursive matrix-vector spaces," in *Proceedings of 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 2012, pp. 1201–1211.

[15] F. A. Gers, J. Schmidhuber, and F. A. Cummins, "Learning to forget: Continual prediction with LSTM," *Neural Computation*, vol. 12, no. 10, pp. 2451–2471, 2000.

[16] Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil, "Learning semantic representations using convolutional neural networks for web search," in *Proceedings of 23rd International World Wide Web Conference*, 2014, pp. 373–374.

[17] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," in *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics*, 2014, pp. 655–665.

[18] B. Hu, Z. Lu, H. Li, and Q. Chen, "Convolutional neural network architectures for matching natural language sentences," *CoRR*, vol. abs/1503.03244, 2015.

[19] Y. Shen, W. Rong, Z. Sun, Y. Ouyang, and Z. Xiong, "Question/answer matching for CQA system via combining lexical and sequential information," in *Proceedings of 29th AAAI Conference on Artificial Intelligence*, 2015, pp. 275–281.

[20] M. Denil, A. Demiraj, N. Kalchbrenner, P. Blunsom, and N. de Freitas, "Modelling, visualising and summarising documents with a single convolutional neural network," *CoRR*, vol. abs/1406.3830, 2014.

[21] C. N. dos Santos and M. Gatti, "Deep convolutional neural networks for sentiment analysis of short texts," in *Proceedings of 25th International Conference on Computational Linguistics*, 2014, pp. 69–78.

[22] A. Severyn and A. Moschitti, "Unitn: Training deep convolutional neural network for twitter sentiment classification," in *Proceedings of 9th International Workshop on Semantic Evaluation*, 2015, pp. 464–469.

[23] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of 2014 Conference on Empirical Methods in Natural Language Processing*, 2014, pp. 1746–1751.

[24] Y. Freund, "An adaptive version of the boost by majority algorithm," *Machine Learning*, vol. 43, no. 3, pp. 293–318, 2001.

[25] Y. Chen, J. You, M. Chu, Y. Zhao, and J. Wang, "Identifying language origin of person names with n-grams of different units," in *Proceedings of 2006 IEEE International Conference on Acoustics Speech and Signal Processing*, 2006, pp. 729–732.

[26] D. Stojanovski, G. Strezoski, G. Madjarov, and I. Dimitrovski, "Twitter sentiment analysis using deep convolutional neural network," in *Proceedings of 10th International Conference on Hybrid Artificial Intelligent Systems*, 2015, pp. 726–737.

[27] W. Rong, Y. Nie, Y. Ouyang, B. Peng, and Z. Xiong, "Auto-encoder based bagging architecture for sentiment analysis," *Journal of Visual Languages and Computing*, vol. 25, no. 6, pp. 840–849, 2014.

[28] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. P. Kuksa, "Natural language processing (almost) from scratch," *Journal of Machine Learning Research*, vol. 12, pp. 2493–2537, 2011.

[29] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[30] M. D. Zeiler, "ADADELTA: an adaptive learning rate method," *CoRR*, vol. abs/1212.5701, 2012.

[31] T. Nakagawa, K. Inui, and S. Kurohashi, "Dependency tree-based sentiment classification using crfs with hidden variables," in *Proceedings of 2010 Conference of the North American Chapter of the Association of Computational Linguistics*, 2010, pp. 786–794.

[32] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings of 49th Annual Meeting of the Association for Computational Linguistics*, 2011, pp. 142–150.

[33] S. T. Dumais, "Latent semantic analysis," *Annual Review of Information Science and Technology*, vol. 38, no. 1, pp. 188–230, 2004.

[34] P. Massa and P. Avesani, "Trust-aware recommender systems," in *Proceedings of 2007 ACM Conference on Recommender Systems*, 2007, pp. 17–24.

[35] E. Bauer and R. Kohavi, "An empirical comparison of voting classification algorithms: Bagging, boosting, and variants," *Machine Learning*, vol. 36, no. 1-2, pp. 105–139, 1999.