Sentiment Analysis of Yelp Reviews

Yoon Tae Kim
Department of Computer Science and Engineering
York University
Toronto, ON, M3J1P3, Canada
kimx3966@cse.yorku.ca

Md Tahmid Rahman Laskar
Department of Computer Science and Engineering
York University
Toronto, ON, M3J1P3, Canada
tahmedge@cse.yorku.ca

ABSTRACT

In this paper, six different algorithms were applied for sentiment analysis to identify the polarity of reviews in Yelp review dataset. Three of these algorithms are neural network based which outperformed all the non-neural network based algorithms based on accuracy. We applied both binary classification and ternary classification for sentiment analysis in yelp reviews. For binary classification, we applied all the six algorithms to determine whether the polarity of the text is positive or negative. For ternary classification, we applied five algorithms to identify whether the sentiment of the text is positive or negative or neutral. It's found that the accuracy of our models for binary classification is much better than for ternary classification. For the binary classification, BLSTM performs the best with an accuracy of 91.41%. For the ternary classification, GRU performs the best with an accuracy of 76.08%. We have also found that Keras word embedding technique outperformed the GloVe word embedding technique for all the neural network based algorithms in terms of accuracy.

Keywords

Yelp, Reviews, Sentiment Analysis, Neural Network, LSTM, GRU.

1. INTRODUCTION

User opinion regarding products, brands or services plays a vital role for the success of a business. It helps the companies to get key insights and take decisions based on user feedback. With the advancement of technology, now the users can give reviews about different products in online websites. So, effective mining of these reviews is very important for the businesses to improve their products.

In a world where a huge amount of text data related to user opinions are generated every day, it is very useful for businesses to use these data for commercial applications like marketing analysis, product feedback, product reviews, public relations, etc. But it is not possible to manually analyze the sentiment of these myriad of texts. So, these text data are required to be mined by an automated process.

Sentiment analysis is the process of computationally identifying and categorizing opinions expressed in a text. These opinions can be related to detecting the polarity (positive or negative opinion) or identifying the subject (the thing that is being talked about) of a text. Various natural language processing or text analysis techniques are applied for sentiment analysis.

Many organizations (Amazon, Yelp, IMDB, etc.) are making the dataset of user reviews related to their products or services public. These datasets are used by researchers to build various new models to improve the sentiment analysis techniques. With the recent advances in deep learning, the ability of algorithms to analyze the texts has been improved significantly.

In this paper, we have used Yelp[1] review dataset to apply various sentiment analysis algorithms and compare their performances. We have applied six algorithms which are: a) Multinomial Naïve Bayes, b) Random Forest, c) Support Vector Machine (SVM), d) Long Short-Term Memory Model (LSTM), e) Bidirectional Long Short-Term Memory Model (BLSTM), and f) Gated Recurrent Unit (GRU). Three of these approaches are neural network based and the other three are non-neural network based. We applied these algorithms for two types of classification tasks on yelp review dataset: a) Binary (Positive or Negative) and b) Ternary (Positive, Negative or Neutral).

2. RELATED WORK

Significant research has been conducted in recent years in the field of sentiment analysis. Most of the research in this domain were related to either rule-based approach or machine learning based approach.

Prabowo and Thelwall [13] proposed a hybrid approach for sentiment analysis by combining rule-based classification with supervised learning and machine learning. They applied this hybrid approach in movie reviews, product reviews and myspace comments and found that hybrid approach could improve classification effectiveness. Kouloumpis and et al. [5] investigated the influence of linguistic features for detecting the sentiment of twitter messages and found that parts-of-speech features were not useful for sentiment analysis in micro blogging domain.

Hutto and Gilbert [4] proposed a new rule-based model for sentiment analysis and found better F1 accuracy than the benchmarks used in their study. But they did not compare the performance of their proposed model with neural network based approaches. Liu et al. [8] applied sentiment analysis techniques to find out the most helpful reviews for a given product.

Basiri et al. [2] exploited reviewers review histories to impact sentiment analysis and found that their proposed solution performed better than different machine learning based algorithms though they did not compare the performance of their approach with neural network based approaches.

Sharma and Dey [14] demonstrated that feature selection methods could improve sentiment classification but the performance depends on the feature that was selected. They observed that gain ratio performed the best among the feature selection methods. Agarwal et al. [1] introduced POS specific prior polarity features and explored the use of tree kernel to obviate the need of tedious feature engineering. Their proposed method performed 4% better

[1] https://www.yelp.com/dataset

than the state-of-the-art unigram-based approach for both binary and ternary sentiment classification.

Mass et al. [9] presented a model that used a mix of supervised and unsupervised techniques to learn word vectors in order to capture semantic term and sentiment content. Their model outperformed many sentiment classification techniques.

Wilson et al. [18] presented a new approach for phrase-level sentiment analysis where they first determined whether the sentiment was polar or neutral and then disambiguated the polarity of the polar expressions.

Lin and He [6] proposed an unsupervised probabilistic modeling framework based on Latent Dirichlet Allocation (LDA), called joint sentiment-topic model (JST), which could detect sentiment and topic simultaneously from text. Melville et al. [10] developed an effective framework for incorporating lexical knowledge in supervised learning for text categorization which was applicable to any text classification task when some relevant background information is available.

Paltoglou and Thelwall [11] found that variants of the classic TF-IDF scheme adapted to sentiment analysis provided significant increase in accuracy. Tang et al. [15] provided an overview of the successful deep learning approaches for sentiment analysis and addressed their advantages and limitations.

Pang et al. [12] applied three different machine learning classifiers (Naïve Bayes, Support Vector Machine, Maximum Entropy Classification) for document level sentiment analysis and found that their methods outperformed all the human-selected-unigram baselines in experimental evaluation. Turney [16] introduced an unsupervised algorithm for sentiment classification based on average semantic orientations of the phrases using the PMI score. Their approach performed well in reviews from different domains but for movie reviews the performance was bad.

Turney and Littman [17] introduced a method for inferring the semantic orientation of a word from its statistical association with a set of positive and negative paradigm words. They used the pointwise mutual information (PMI) and latent semantic analysis (LSA) for word association. The LSA approach performed more accurately in classifying semantic orientation.

3. METHODOLOGY

We applied six different algorithms for sentiment analysis in Yelp review dataset. Three of them are neural network based and three of them are non-neural network based. A brief description of each of our algorithms are given below.

3.1 Multinomial Naïve Bayes

Multinomial Naïve Bayes is a simple classification method based on Bayes rule. It relies on simple bag of words representation of documents or texts.

For a document or text *d*, and class *c*, Naïve Bayes predicts the probability of the class *c* for text *d* with the following conditional probability:

$$P(c\,|\,d) = \frac{P(d\,|\,c)P(c)}{P(d)} \tag{1}$$

We can write it as:

$$c_{MAP} = \underset{c\in C}{\operatorname{argmax}}\, P(c\,|\,d) \tag{2}$$

$$= \underset{c\in C}{\operatorname{argmax}}\, \frac{P(d\,|\,c)P(c)}{P(d)} \tag{3}$$

$$= \underset{c\in C}{\operatorname{argmax}}\, P(d\,|\,c)P(c) \tag{4}$$

Here, in (2), MAP is maximum a posteriori, which indicates the most likely class. In (3), Bayes rule is applied. The denominator is dropped in (4).

If we represent document *d* as features *x1, x2, …… xn,* then we can write:

$$\begin{aligned} c_{MAP} &= \underset{c\in C}{\operatorname{argmax}}\, P(d\,|\,c)P(c) \\ &= \underset{c\in C}{\operatorname{argmax}}\, P(x_1, x_2, \ldots, x_n\,|\,c)P(c) \end{aligned} \tag{5}$$

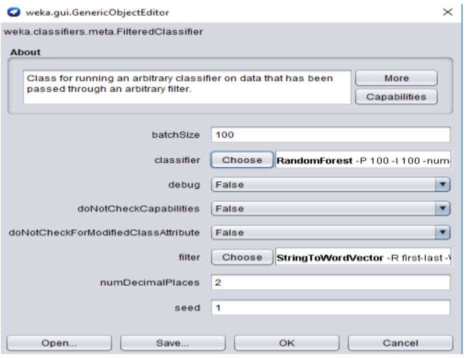
Finally, after applying Naïve Bayes rule, we get:

$$c_{NB} = \underset{c\in C}{\operatorname{argmax}}\, P(c_j)\prod_{x\in X} P(x\,|\,c) \tag{6}$$

For Naïve Bayes Multinomial, we integrated the TF-IDF weighting to generate the list of words. We implemented the Multinomial Naïve Bayes in Python using the Scikit-learn[2] library.

3.2 Random Forest

Random Forest is an ensemble model which builds multiple decision trees to avoid overfitting to a training dataset. Notably, the model is robust to noises and outliers which possibly deteriorate the classification performance. The Yelp review dataset includes noise data, because each user has different standard in rating. For this reason, the Random Forest model is a good candidate solution for this task. To test the performance of Radom Forest classifier, we used 'FilteredClassifier' in Weka with Random Forest model and 'StringToWordVector' filter. We converted our dataset to '.arff' file format to make it readable by WEKA. An example of Random Forest in Weka using 'FilteredClassifier' is shown on Figure 1.



weka.gui.GenericObjectEditor

weka.classifiers.meta.FilteredClassifier

About

Class for running an arbitrary classifier on data that has been passed through an arbitrary filter.

batchSize 100

classifier **RandomForest** -P 100 -I 100 -num

debug False

doNotCheckCapabilities False

doNotCheckForModifiedClassAttribute False

filter **StringToWordVector** -R first-last -

numDecimalPlaces 2

seed 1

Figure 1. Random Forest using 'FilteredClassifier' in Weka.

[2] https://scikit-learn.org/stable/

3.3 Support Vector Machine

Support Vector Machine (SVM) is a supervised learning algorithm which utilizes large-margin technique. SVM has shown great performance in classification tasks. Especially, the SVM models are able to efficiently handle a non-linear classification problem using the kernel function, which transforms the low dimensional input data to relatively higher dimensional spaces. Furthermore, the models can efficiently handle high dimensional feature vectors, such as 'Bag of Words' feature vectors. In this regard, the SVM has the great potential as a solution for the sentiment classification task. We used 'FilteredClassifier' from Weka with the SVM model along with the 'StringToWordVector' filter.

3.4 Long Short-Term Memory Model (LSTM)

Long short term memory model (LSTM) units are units of Recurrent neural network (RNN). LSTM networks are used for classification or prediction based on time series data. It can deal with exploding and vanishing gradient problems. One of the major problems of RNN is the long term dependency [3]. LSTM can avoid such long term dependencies. A common LSTM unit is composed of a cell, input gate, output gate and a forget gate. In Figure 2, a simple LSTM network is shown.

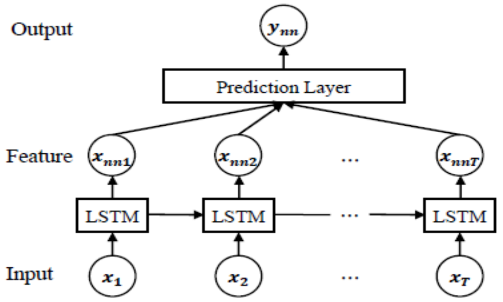


Figure 2. LSTM Network.

For our experiment, we used one LSTM layer with 100 hidden nodes, a dropout rate of 0.2, 'adam' as the optimizer, the sigmoid function as the activation function and 'early stopping' during training. We implemented the LSTM architecture in Python using Keras[3] library. We also used Keras embedding to generate word vector of for the embedding layer. The optimal model is selected based on best performance on the development set.

3.5 Bidirectional Long Short-Term Memory Model (BLSTM)

With bidirectional long short-term memory model (BLSTM), the output layer can get information from past (backward) and future (forward) states simultaneously. BLSTM was introduced to increase the amount of input information to the network. In the standard RNN network, future input information cannot be reached from the current state. But using BLSTM, future input information can be reached from the current state. It also doesn't require the input data to be fixed. The structure of a BLSTM network is shown on Figure 3.

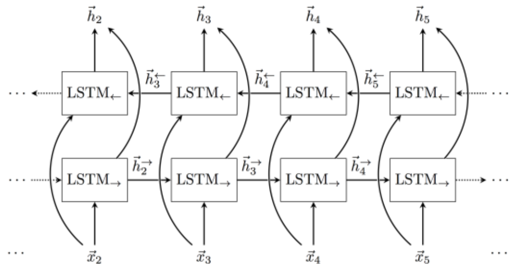


Figure 3. BLSTM structure.

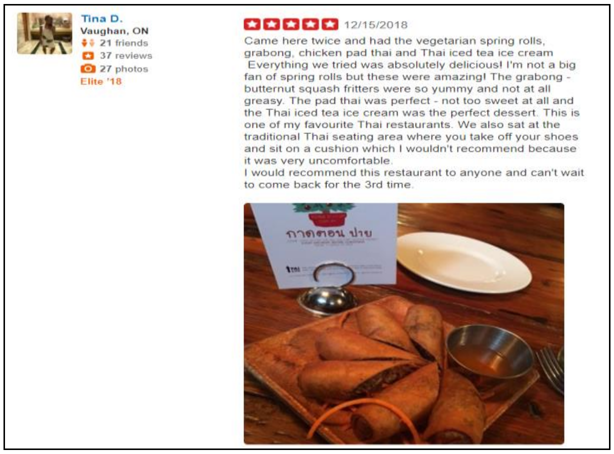
In this paper, we used almost similar architecture like LSTM for BLSTM. Only difference is that for BLSTM, we used only 50 hidden nodes whereas for LSTM we used 100 hidden nodes.

3.6 Gated Recurrent Unit (GRU)

Gated Recurrent Unit (GRU) is a simplified version of Long Short-Term Memory (LSTM) model. Even though GRU has fewer parameters, the model is able to efficiently capture long term dependencies between sequences. Therefore, GRU is comparable to LSTM in terms of performance and computational efficiency [7]. In this regard, GRU model can be used as one of the solutions in the classification task. We implemented the deep neural architecture using Keras library. The architecture includes 3 layers: 1) Word embedding layer, 2) GRU layer, 3) Fully connected layer with activation function. Input data was 'Bag of Words' vectors from reviews. For activation function in fully connected layer, we used sigmoid function for both binary and ternary classifications. We kept the optimal parameters based on the best performance in the development set. Then, the optimal model was used for evaluation on the test set. Dropout rate was 0.2.

4. DATASETS

We used the Yelp review dataset to see the performance of our algorithms for the task of sentiment analysis. Yelp is a website which allows people to share their reviews about local businesses. It contains 5,996,996 reviews. Each review includes different attributes such as text, rating (1 to 5 star), pictures, etc. Example of a review at Yelp website is shown on Figure 4.



Tina D.
Vaughan, ON
21 friends
37 reviews
27 photos
Elite '18

★★★★★ 12/15/2018

Came here twice and had the vegetarian spring rolls, grabong, chicken pad thai and Thai iced tea ice cream
Everything we tried was absolutely delicious! I'm not a big fan of spring rolls but these were amazing! The grabong - butternut squash fritters were so yummy and not at all greasy. The pad thai was perfect - not too sweet at all and the Thai iced tea ice cream was the perfect dessert. This is one of my favourite Thai restaurants. We also sat at the traditional Thai seating area where you take off your shoes and sit on a cushion which I wouldn't recommend because it was very uncomfortable.
I would recommend this restaurant to anyone and can't wait to come back for the 3rd time.

Figure 4. A review at Yelp Website.

We used Yelp dataset for two types of classifications. One is binary (positive or negative), another is ternary (positive, negative or neutral).

[3] https://keras.io/

For binary classification, we considered 1 to 2 star reviews as negative and 4 to 5 star reviews as positive. We did not consider 3-star reviews for binary classification because in the review dataset of Yelp, it is often difficult to identify whether the 3-star review indicating a positive review or a negative review.

Some examples of 3-star reviews are stated below:

i) 3-star review with positive sentiment: *"We frequent John's for the chicken parmesan. They also have amazing collard greens and beans. They have AMAZING coconut cream pie. Don't miss it. My only complaint about John's is that they won't let me order from the kids menu. I don't eat much so a full serving is always too much food for me."*

ii) 3-star review with negative sentiment: *"Overrated and overpriced!"*

So, while pre-processing the data, we ignored the 3-star reviews for binary classification because the sentiment of a 3-star review depends on reviewers. But for ternary classification, we considered 1 to 2 star reviews as negative, 3 star reviews as neutral, and 4 to 5 star reviews as positive.

We randomly selected 80000 reviews for binary classification. The dataset is balanced with each star contains 20000 reviews. For ternary classification, we concatenated 20000 reviews of 3-star rating with the 80000 reviews that we selected for binary classification. So, in total we used 100000 reviews for ternary classification. For the neural network based algorithms, we used 60% of the data for training, 20% for validation, and 20% for testing. For non-neural network based algorithms, we used 80% data for training with and rest of the data for testing. The dataset was preprocessed by removing the stop words, punctuation marks, and numerical values. We also converted all the letters to lowercase.

5. PERFORMANCE EVALUATION

We evaluated the performance of all of our approaches (except SVM) for both binary (positive or negative) and ternary (positive or negative or neutral) classifications. The SVM model was only applied for binary classification. We also evaluated the effect of word embedding techniques on our neural network based approaches. We describe the results in the following sections.

5.1 Classification Results

5.1.1 Multinomial Naïve Bayes

We evaluated the performance of Multinomial Naïve Bayes for both binary and ternary classifications. The result is stated below:

Classification Type	Accuracy
Binary	87.5%
Ternary	70.2%

Table 1: Result of Multinomial Naïve Bayes

Multinomial Naïve Bayes performed the worst among non-neural network based approaches in terms of accuracy. It also showed the worst performance among all the six algorithms used in this paper. Though the computation time of Naïve Bayes is very efficient.

5.1.2 Random Forest

For both binary and ternary classifications, Random Forest showed the second best performance among non-deep learning based models.

Classification Type	Accuracy
Binary	88.11%
Ternary	71.35%

Table 2: Result of Random Forest

5.1.3 Support Vector Machine (SVM)

For binary classification, SVM showed the best performance among the three non-deep learning based models, and the fourth best performance among all six models. SVM was applied only for binary classification.

Classification Type	Accuracy
Binary	89.10%

Table 3: Result of Support Vector Machine (SVM)

5.1.4 LSTM

LSTM performs better than all the non-neural network based approach in terms of both binary and ternary classifications. The accuracy of LSTM for both binary and ternary classifications was very close to our best models. For both binary and ternary classifications, LSTM performed as the second best. The result of LSTM is stated below.

Classification	Accuracy
Binary	91.37%
Ternary	75.78%

Table 4: Result of LSTM

5.1.5 BLSTM

BLSTM performs the best in terms of binary classification among all the six approaches that we applied in this paper. Though it performed slightly below LSTM for ternary classification. In terms of accuracy for ternary classification, BLSTM ranked third.

Classification	Accuracy
Binary	91.41%
Ternary	75.28%

Table 5: Result of BLSTM

5.1.6 GRU

GRU performs the best among all the approaches for ternary classification with an accuracy of 76.08%. But it gave similar performance like LSTM and very slightly below than BLSTM in terms of accuracy for binary classification.

Classification	Accuracy
Binary	91.37%
Ternary	76.08%

Table 6: Result of GRU

5.2 Discussions

The overall result of our models in terms of binary classification is shown on Figure 5.

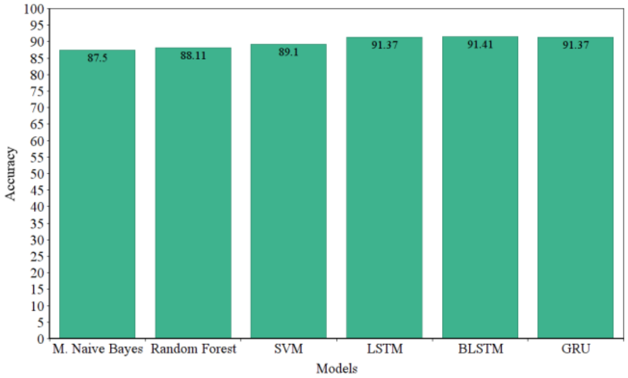


Figure 5. Accuracy (%) of each model in Binary Classification.

For binary classification, we can see that all the neural network based approaches provided more than 90% accuracy. BLSTM performs the best with 91.41% accuracy and Multinomial Naïve Bayes performs the worst with 87.5% accuracy.

Among the non-neural network based approaches, SVM performs the best with 89.1% accuracy.

We did not use SVM for ternary classification. Among the five other models, GRU performs the best for ternary classification with 76.08% accuracy. Among the non-neural network based approaches, Random Forest performs the best for ternary classification with 71.35% accuracy.

The overall result of our five models for ternary classification is shown on Figure 6.

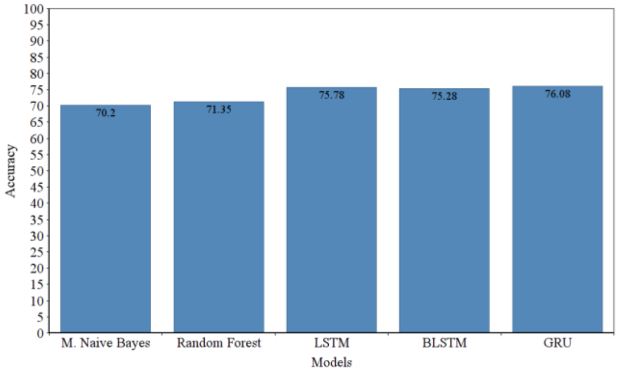


Figure 6. Accuracy (%) of each model in Ternary Classification.

5.3 Effects of Word Embedding

We also observed the result of our neural network based approaches by changing the word embedding techniques. We found that the use of different word embedding techniques have effect on our results.

Here, we compared the performance of Keras embedding with GloVe embedding[4]. It is to be noted that GloVe is a pre-trained word embedding model. But Keras embedding learns word embedding during the training process.

The result based on different word embedding techniques for binary classification is shown on Figure 7.

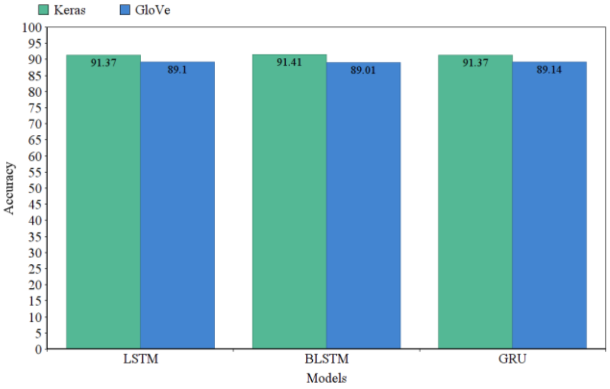


Figure 7. Effects of Word Embedding in terms of Accuracy (%) for Binary Classification.

The result based on different word embedding techniques for ternary classification is shown on Figure 8.

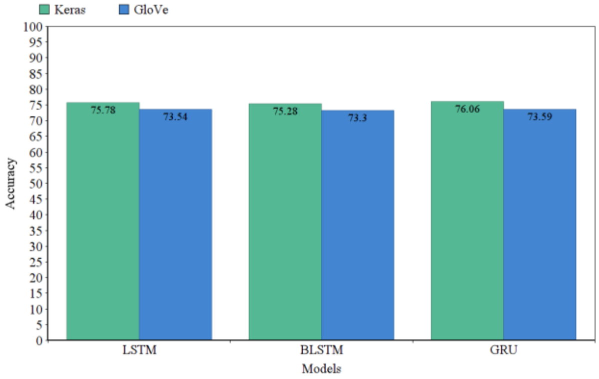


Figure 8. Effects of Word Embedding in terms of Accuracy (%) for Ternary Classification.

From both Figure 7 and 8, we can see that the Keras Embedding outperformed GloVe Embedding in terms of accuracy.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we applied six different algorithms for sentiment analysis in Yelp review dataset. We found that all the neural network based approaches outperformed the non-neural network based approaches. We implemented all the neural network based algorithms in Python using Keras library. The Multinomial Naïve Bayes was also implemented in Python using Scikit-learn library. The Random Forest and SVM were implemented in Weka. For Binary classification, BLSTM performed the best with 91.41% accuracy. For Ternary classification, GRU performed the best with 76.06% accuracy. We also evaluated the effect of word embedding techniques on our neural network based approaches and found that Keras Embedding outperformed GloVe embedding. In future, these models can be evaluated on more datasets. Though the accuracy for multiclass classification is still not that good with none of the models could predict with more than 80% accuracy. So, future research can be done on how to improve the classification accuracy for multiclass classification.

[4] https://nlp.stanford.edu/projects/glove/

7. ACKNOWLEDGMENTS

We would like to thank Professor Aijun An for teaching us different data mining algorithms and techniques in Fall 2018 which helped us to develop this project.

8. REFERENCES

[1] Agarwal, A., Xie, B., Vovsha, I., Rambow, O., & Passonneau, R. (2011). Sentiment analysis of Twitter data. In Proceedings of the Workshop on Languages in Social Media , LSM '11, ACL, 30-38.

[2] Basiri, M., Ghasem-Aghae, N., & Naghsh-Nilchi, A. (2014). Exploiting reviewers' comment histories for sentiment analysis. Journal of Information Science, 40(3), 313-328.

[3] Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks, 157-166.

[4] Hutto, C., & Gilbert, E. (2014). VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text. In International AAAI Conference on Weblogs and Social Media, AAAI, 216–225.

[5] Kouloumpis, E., Wilson, T. & Moore, J., (2011). Twitter Sentiment Analysis: The Good the Bad and the OMG! In Proceedings of the Fifth International Conference on Weblogs and Social Media, 538-541.

[6] Lin, C., & He, Y. (2009). Joint sentiment/topic model for sentiment analysis. In Proceedings of the 18th ACM conference on Information and knowledge management, CIKM '09, ACM, 375-384.

[7] Liu, J., Wu, C., & Wang, J. (2018). Gated recurrent units based neural network for time heterogeneous feedback recommendation. Information Sciences—Informatics and Computer Science, Intelligent Systems, Applications: An International Journal, 50-65.r

[8] Liu, Y., Huang, X., An, A., & Yu, X. (2008). Modeling and Predicting the Helpfulness of Online Reviews, in Eighth IEEE International Conference on Data Mining, Pisa, 443-452.

[9] Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011). Learning word vectors for sentiment analysis. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistic, ACL, 142-150.

[10] Melville, P., Gryc, W., & Lawrence R. D. (2009). Sentiment analysis of blogs by combining lexical knowledge with text classification. In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '09, ACM, 1275-1284.

[11] Paltoglou, G., & Thelwall, M. (2010). A study of information retrieval weighting schemes for sentiment analysis. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, ACL '10. ACL, 1386-1395.

[12] Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up?: sentiment classification using machine learning techniques. In Proceedings of the ACL-02 conference on Empirical methods in natural language processing, EMNLP '02, ACL, 79-86.

[13] Prabowo, R., & Thelwall, M. (2009). Sentiment analysis: A combined approach. Journal Of Informetrics, 3(2), 143-157.

[14] Sharma, A., & Dey, S. (2012). A comparative study of feature selection and machine learning techniques for sentiment analysis. Proceedings of the 2012 ACM Research in Applied Computation Symposium, RACS '12, 1-7.

[15] Tang, D., Qin, B., & Liu, T. (2015). Deep learning for sentiment analysis: successful approaches and future challenges. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 292-303.

[16] Turney, P. D. (2002). Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, ACL '02, ACL, 417-424.

[17] Turney, P. D., & Littman, M. L. (2003). Measuring praise and criticism: Inference of semantic orientation from association. ACM Transactions on Information Systems (TOIS), 21(4), 315-346.

[18] Wilson, T., Wiebe, J., & Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. In Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, HLT '05, ACL, 347-354.