Sentiment Orientation Using Deep Learning Sequential and Bidirectional Models

Hasan J. Alyamani

Department of Information Systems, Faculty of Computing and Information Technology in Rabigh (FCITR), King Abdulaziz University, Jeddah 21589, Saudi Arabia

Summary
Sentiment Analysis has become very important field of research because posting of reviews is becoming a trend. Supervised, un-supervised and semi supervised machine learning methods done lot of work to mine this data. Feature engineering is complex and technical part of machine learning. Deep learning is a new trend, where this laborious work can be done automatically. Many researchers have done many works on Deep learning Convolutional Neural Network (CNN) and Long Shor Term Memory (LSTM) Neural Network. These requires high processing speed and memory. Here author suggested two models simple & bidirectional deep leaning, which can work on text data with normal processing speed. At end both models are compared and found bidirectional model is best, because simple model achieve 50% accuracy and bidirectional deep learning model achieve 99% accuracy on trained data while 78% accuracy on test data. But this is based on 10-epochs and 40-batch size. This accuracy can also be increased by making different attempts on epochs and batch size.
Key Word: *Deep Learning, LSTM, CNN, Machine Learning, Supervised and Un-Supervised Learning*

1. Introduction

Opinionated posting on social media and reviews web pages are increasing day by day [1]. Sentiment Analysis of these reviews/ opinions is very challenging and trending research topics and many novel sub problems has been covered [2][3][4].
Now a days, customer of the product is very conscious about the quality of any product, so before going to buy, they visit the website of product. Here all reviews from the users of the product is posted for not only customer but also for company owner [5][6]. Reading of these reviews is very time consuming and difficult. So, researchers have done lot of work to separate these reviews as negative and positive.
Work of [7] has assigned scores to opinion, feeling, or review of a person about product to find the polarity as positive or negative. Researchers has covered sentiment analysis as binary classification as “positive” and “negative” classes [8][9][10][11], [12][13]. There are two ways for such classification i.e. supervised and un-supervised machine learning methods [14][15].
Lot of the work has been done on sentiment analysis using supervised and un-supervised learning methods [16][17][18][19]. In machine learning, feature designing manually is a difficult task, imperfect and takes more time from design to validation process [20]. Learned features can be easily adopted in deep learning, that provide a framework which is very general and learnable for representing information [21]. Neural Network using multilayer approach with forward and back propagation is basis for Deep Learning. In machine learning, feature selection is done manually or using a tools, while in deep learning automatically features learned with high accuracy [24]. Long-Short-Term Memory with a conditional random field layer (BiLSTM-CRF) has been utilizing in work of [25] to improve sentiment analysis on sentence level. Deep Learning classifier has also been employing on news article [26], English movies reviews [27] to detect polarity and detection of user satisfaction [28]. Authors of paper [29] uses the deep learning technology to analyze the sentiment of the movie reviews in the IMDB dataset, and divide them into negative and positive categories. Most of the work based on sentiment analysis in deep learning uses LSTM and CNN [22][23], [24]. These are resource hunger algorithms.
This work proposed novel architectures for deep learning classifiers such as simple sequential model and bidirectional deep learning model to identify the polarity of the text. These are less computationally expensive. Performance of both models will be checked on 10-epochs to find the best model.

2. Proposed Work

Complete proposed model is depicted in Fig-1 and Fig-2. First figure is showing conversion of dataset into a form which is compatible for our models. First of all, Dependent and independent features will be extracted from dataset. One-hot encoding will be done on preprocessed independent features by using vocabulary size. This encoded vector will be the input for embedded vectors (these vectors will be defined in result section). Dependent variable will be encoded using ‘Label Encoder’. Now embedded vectors from independent variables and encoded vector from dependent variable will be split into train data and test data. At the end scaling feature will be applied on train and test data from independent variable. Now these vectors are ready for training by using model.

Manuscript received November 5, 2021
Manuscript revised November 20, 2021
https://doi.org/10.22937/IJCSNS.2021.21.11.4

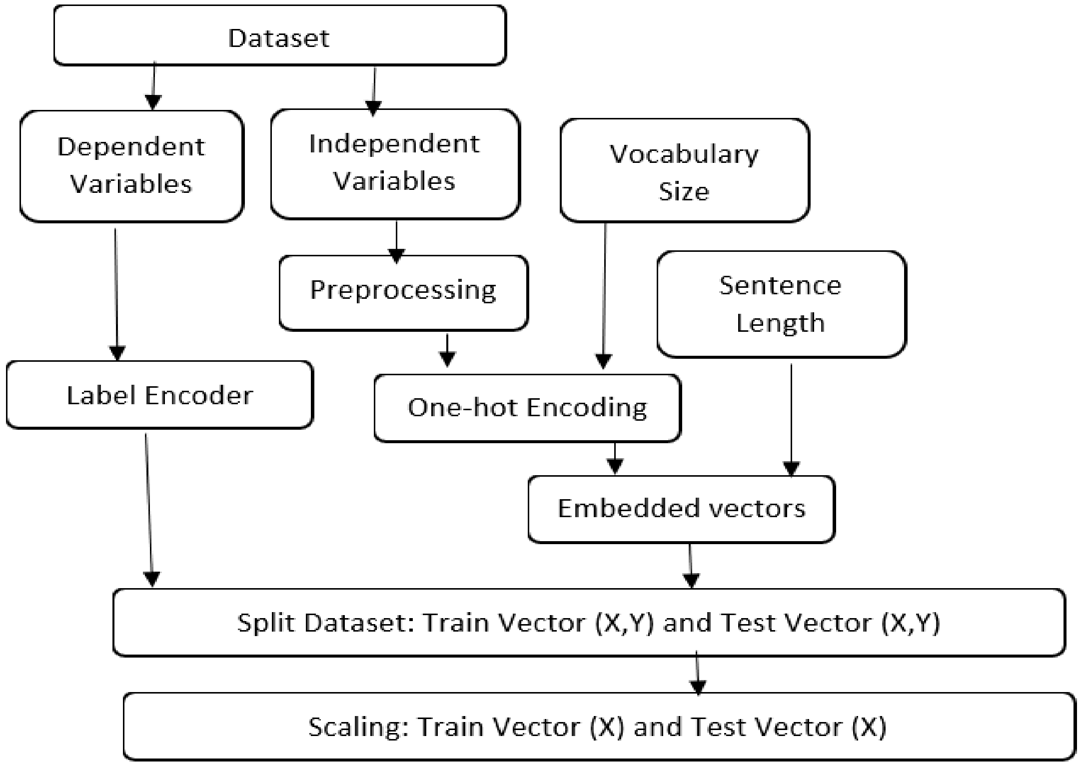

Fig-1: Preparing Input Data for Deep Learning Models

Here two models have been prepared for above vectors. Next, we will find the best model for this data. In Fig-2, M1 is simple sequential model and M2 is bidirectional model of deep learning. M1 consist of input layer containing neurons and input shape. A hidden dense layer will use 'relu' function, while output layer will use sigmoid function. Adam optimizer and 'binary cross entropy' loss function will be used for compiling the model. In result section we will describe its performance and accuracy.

Second model which is denoted as M2, bidirectional model where embedded layer will be added after inputting dense layer. Output of embedded layer will be the input for bidirectional layer. So bidirectional will be added after embedded layer. Output layer and compilation is same as in M1.

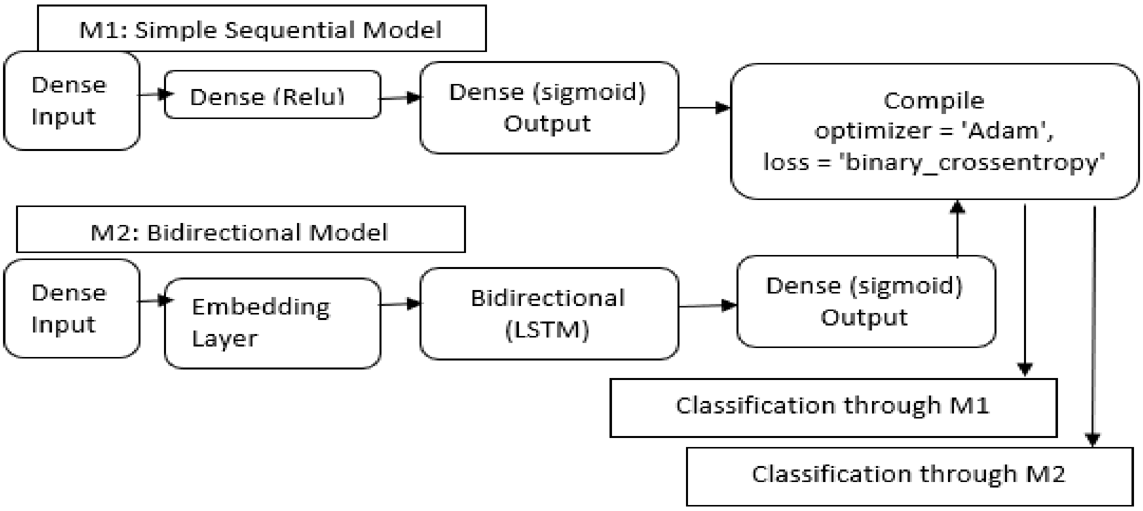

Fig-2: Proposed Models Parameters Setting

3- Results

A dataset for sentiment analysis is downloaded from [31]. It has two columns reviews and class containing 942 records. Reviews column contains the text of user reviews and class column has predefined class i.e. positive and negative. A sample consists of 10-reviews from said datasets is shown in Table-1.

Table-1: Sample of Dataset

Reviews (Independent Variable)	Class (Dependent Variable)
“wow loved this place”	“Positive”
“crust is not good”	“Negative”
“it is not tasty and the texture was just nasty”	“Negative”
“stopped by during the late may bank holiday off rick steve recommendation and loved it”	“Positive”
“the selection on the menu was great and so were the prices”	“Positive”
“now i am getting angry and i want my damn pho”	“Negative”
“honeslty it did not taste that fresh”	“Negative”
“the potatoes were not fresh and like rubber and you could tell they had been made up ahead of time being kept under a warmer”	“Negative”
“the fries were great too”	“Positive”
“a good great touch”	“Positive”

Whole dataset has been applied on M1 and M2 from Fig-2. M1 and M2 has been created in python to check the results. Here, processing of 10-reviews will be shown to check the accuracy of M1 and M2. From dataset, reviews are independent variables and class is dependent variable. As reviews are in text so, it has converted into a vector by using one-hot-encoding and vocabulary of size 500. Result of this vector for first 10-reviews is shown in Table-2.

Table-2: One-hot encoded Vector

```
[[1499, 610, 821],
 [3900, 2725],
 [1094, 37, 3237],
 [2982, 248, 3072, 104, 3062, 4075, 3380, 574, 610],
 [2709, 2426, 1881, 4870],
 [3145, 4681, 872, 4343, 1001],
 [1540, 1267, 184],
 [252, 184, 4986, 682, 455, 2189, 4392, 1336, 2191, 1309, 3179],
 [659, 1881],
 [2725, 1881, 336]]
```

Next is the process is to convert this encoded vector into embedded vector by using sentence length. Here length of sentence is used as 20-values for each review. And result of first 10-reviews is given below in Table-3.

Table-3: Embedded Vector of Size 20

```
[ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1499 610 821]
[ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3900 2725]
[ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1094 37 3237]
[ 0 0 0 0 0 0 0 0 0 0 0 2982 248 3072 104 3062 4075 3380 574 610]
[ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2709 2426 1881 4870]
[ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3145 4681 872 4343 1001]
[ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1540 1267 184]
[ 0 0 0 0 0 0 0 0 0 252 184 4986 682 455 2189 4392 1336 2191 1309 3179]
[ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 659 1881]
[ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2725 1881 336]
```

Scaling converts embedded vector into a form which can be read by deep learning models easily. Scaling process has been applied on whole vector of reviews. Here in Table-4, vector is showing the scaled values for first five reviews, because it takes very large space. Hence complete reviews have been converted into a vector which is readable for models.

Table-4: Scaling Vector

```
[ 0.         0.         0.         0.         0.        -0.03984095  -0.0552433  -0.11785471 -0.1366571  -0.17866343 -
0.22438887  -0.28749387  -0.34712616 -0.43194395 -0.52295541 -0.66088437 -0.68572486 -1.13407167
1.03841331  0.94020713]

[ 0.         0.         0.         0.         0.        -0.03984095  -0.0552433  -0.11785471 -0.1366571  -0.17866343 -
0.22438887 -0.28749387  -0.34712616 -0.43194395 -0.52295541 -0.66088437 -0.85244101  0.99697666
0.00149322  1.06744547]

[ 0.         0.         0.         0.         0.        -0.03984095  -0.0552433  -0.11785471 -0.1366571  -0.17866343 -
0.22438887 -0.28749387  -0.34712616 -0.43194395 -0.52295541 -0.66088437 -0.85244101 -1.21664113
1.39419933 -1.4795151 ]

[ 0.         0.         0.         0.         0.        -0.03984095  -0.0552433  -0.11785471 -0.1366571  -0.17866343 -
0.22438887 -0.28749387 -0.34712616 -0.43194395 -0.52295541 -0.66088437  0.23315254 -0.13000187
0.22943978  0.36544084]

[ 0.         0.         0.         0.         0.        -0.03984095  -0.0552433  -0.11785471 -0.1366571  -0.17866343 -
0.22438887  2.65903814  3.27806524  2.21068824  1.77167687  0.3996539   0.68160606  0.68371697  -
0.44763589  0.48024784]
```

Dependent variables contain 'positive' and 'negative' labels, so it has been converted into simple encoded vector by using 'Label Encoding'. Result of first 10-records is shown in Table-5. Here 1 means positive and 0 means negative.

Table-5: Label Encoding on Dependent Variable

[1, 0, 0, 1, 1, 0, 0, 0, 1, 1]

Description of M1 (Simple Sequential Deep Learning Model) from Fig-2 has been implemented in python is defined in Table-6.

Table-6: Description of Simple Deep Learning Model

Layer (type)	Output Shape	Param #
dense_12	(None, 20)	420
dense_13	(None, 20)	420
dense_14	(None, 15)	315
dense_15	(None, 1)	16
Total params: Trainable params: Non-trainable params: 0	1,171 1,171 0	

This model was compiled by using 'Adam' with 10-epochs, batch size 40. Last three epochs from this model are given in Table-7, which shows the accuracy of this model is 53% and rest of the measures is depicted in Table-8.

Table-7: Description of Last three Epochs from M1

"Epoch 8/10
11/11 [==============================] - 0s 4ms/step - loss: 0.6718 - accuracy: 0.5829 - val_loss: 0.6997 - val_accuracy: 0.5215
Epoch 9/10
11/11 [==============================] - 0s 3ms/step - loss: 0.6702 - accuracy: 0.5806 - val_loss: 0.6998 - val_accuracy: 0.5167
Epoch 10/10
11/11 [==============================] - 0s 4ms/step - loss: 0.6686 - accuracy: 0.5806 - val_loss: 0.6996 - val_accuracy: 0.5167"

Table-8: Confusion Matrix measures on M1

	precision	recall	f1-score	support
0	0.50	0.73	0.60	147
1	0.60	0.36	0.45	164
accuracy			0.53	
weighted avg	0.55	0.53	0.52	311

Description of M2 (Bidirectional Deep Learning Model) from Fig-2 has been implemented in python is defined in Table-9.

Table-9: Description of Bidirectional Deep Learning Model

Layer (type)	Output Shape	Param #
embedding_12 (Embedding)	(None, 20, 60)	300000
bidirectional_12 (Bidirectional)	(None, 200)	128800
dense_11 (Dense)	(None, 1)	201
Total params: Trainable params: Non-trainable params: 0	429,001 429,001 0	

This model was compiled by using 'Adam' with 10-epochs, batch size 40. Last three epochs from this model are given in Table-10, which shows the accuracy of 74% and rest of the measures is depicted in Table-11.

Table-10: Description of Last three Epochs from M2

"Epoch 8/10
10/10 [==============================] - 0s 47ms/step - loss: 0.0822 - accuracy: 0.9762 - val_loss: 0.7015 - val_accuracy: 0.7846
Epoch 9/10
10/10 [==============================] - 0s 47ms/step - loss: 0.0467 - accuracy: 0.9889 - val_loss: 0.7393 - val_accuracy: 0.7781
Epoch 10/10
10/10 [==============================] - 0s 41ms/step - loss: 0.0273 - accuracy: 0.9952 - val_loss: 0.9721 - val_accuracy: 0.7428"

Table-11: Confusion Matrix measures on M2

	precision	recall	f1-score	support
0	0.73	0.71	0.72	147
1	0.75	0.77	0.76	164
accuracy	0.74			
weighted avg	0.74	0.74	0.74	311

4. Conclusion

online reviews are very important for quality-related issues because negative opinions about a product on the web can change the minds of 80% of customers about their purchasing decision. These reviews can be analyzed using different machine algorithm with different type of issues and complexity. Now a days, state of art deep learning algorithm can easily apply on text dataset to find the class of review. In this paper, two deep learning models have been implemented to find the sentiment of given review as positive or negative. Different researchers have work on sentiment analysis which required high processing speed. But here two deep learning model (simple and bidirectional) have been implemented requiring less resources and then compared to find the best model. After the comparison of both models, it is measured that at first epoch accuracy of M1 on trained data is 52% and M2 is 48%. After the completion of 10-epochs, accuracy of M1 on trained data is 58% and M2 is 99%. These Accuracies on 10-epochs are given in Table-12.

Table-12: Accuracies of M1 and M2 on Trained Data

M1: Change of accuracy (trained data) through 10-epocs	M2: Change of accuracy (trained data) through 10-epocs
[0.528436005115509, 0.549763023853302, 0.578199028968811, 0.5616113543510437, 0.5687204003334045, 0.5734597444534302, 0.578199028968811, 0.5829383730888367, 0.5805687308311462, 0.5805687308311462]	[0.4817749559879303, 0.5213946104049683, 0.6814579963684082, 0.7480190396308899, 0.8922345638275146, 0.9413629174232483, 0.9714738726615906, 0.9762282371520996, 0.9889065027236938, 0.995245635509491]

Data was split into 20% training data and 80% testing data. In testing data result is measured as: at first epoch accuracy of M1 on test data is 52% and M2 is 47%. After the completion of 10-epochs, accuracy of M1 on test data is 51% and M2 is 74%. These Accuracies on 10-epochs are given in Table-13.

Table-13: Accuracies of M1 and M2 on Test Data

M1: Change of accuracy (test data) through 10-epocs	M2: Change of accuracy (test data) through 10-epocs
[0.5215311050415039, 0.4880382716655731, 0.49760764837265015, 0.5023923516273499, 0.5023923516273499, 0.5119616985321045, 0.5119616985321045, 0.5215311050415039, 0.5167464017868042, 0.5167464017868042]	[0.4726687967777252, 0.47909969091415405, 0.5787781476974487, 0.6881029009819031, 0.7427652478218079, 0.7781350612640381, 0.790996789932251, 0.7845659255981445, 0.7781350612640381, 0.7427652478218079]

As these are the results based on 10 epochs with 40 batch sizes, so this efficiency can also be increased by changing the attempts on these parameters. Rest of the parameters i.e. activations functions, vocabulary size, no of features, and sentence length by setting of padding can also be applied to increase the performance. Based on given parameters, it is concluded that Bidirectional model (M1) is best as compared to simple sequential model (M2). Performances of M1 and M2 based on loss and accuracy is shown in Fig-3 and Fig-4 respectively.

Manuscript received November 5, 2021
Manuscript revised November 20, 2021
https://doi.org/10.22937/IJCSNS.2021.21.11.4

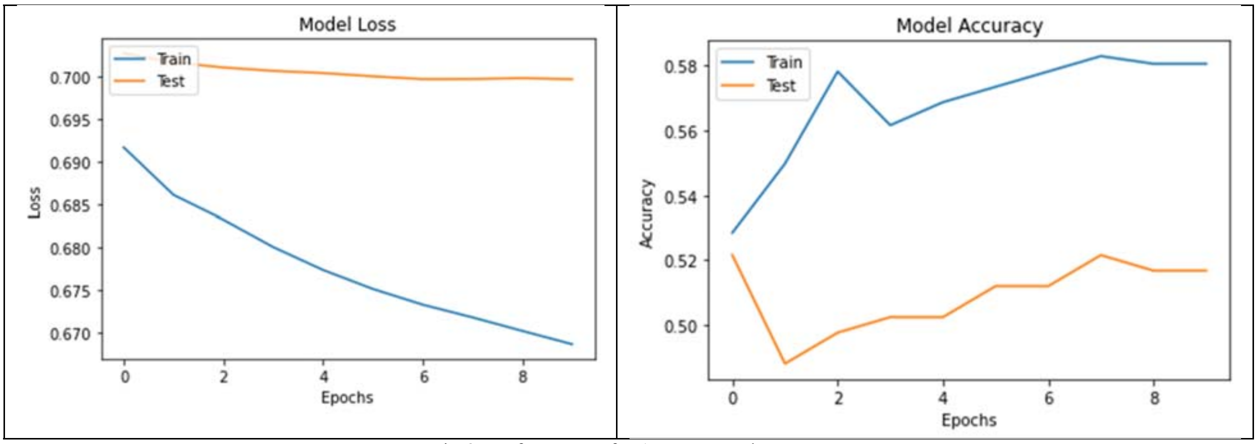

Fig-3: Performance of M1 on Loss and Accuracy

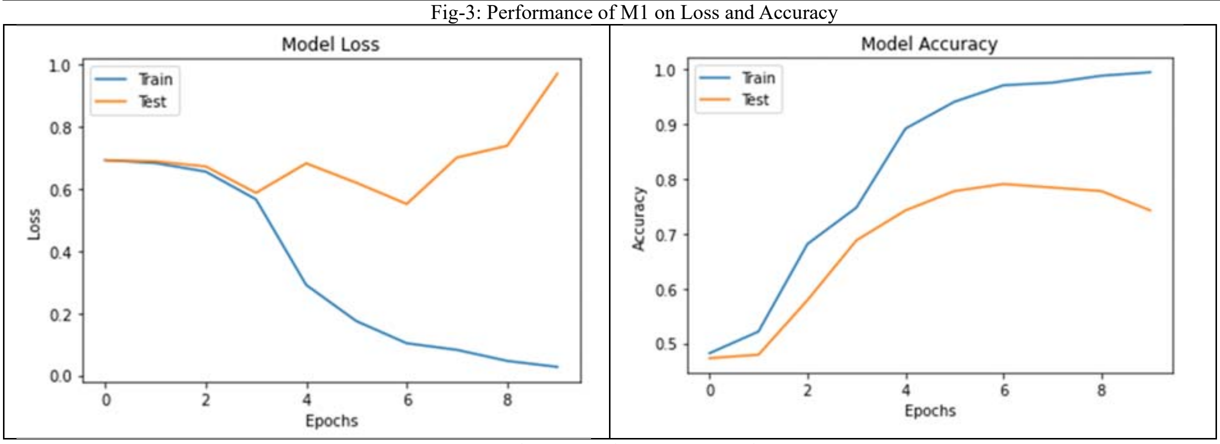

Fig-4: Performance of M2 on Loss and Accuracy

References

[1] M. Hameed, F. Tahir, and M. A. Shahzad, “Empirical comparison of sentiment analysis techniques for social media,” Int. J. Adv. Appl. Sci., vol. 5, no. 4, pp. 115–123, 2018.

[2] N. Archak, A. Ghose, and P. G. Ipeirotis, “Show me the Money ! Deriving the Pricing Power of Product,” Proc. 13th ACM SIGKDD Int. Conf. Knowl. Discov. data Min. - KDD ’07, pp. 56–65, 2007, doi: 10.1145/1281192.1281202.

[3] Y. Chen and J. Xie, “Online Consumer Review: Word-of-Mouth as a New Element of Marketing Communication Mix,” Manage. Sci., vol. 54, no. 3, pp. 477–491, 2008, doi: 10.1287/mnsc.1070.0810.

[4] I. El Alaoui, Y. Gahi, R. Messoussi, Y. Chaabi, A. Todoskoff, and A. Kobi, “A novel adaptable approach for sentiment analysis on big social data,” J. Big Data, vol. 5, no. 1, 2018, doi: 10.1186/s40537-018-0120-0.

[5] D. Gräbner, M. Zanker, G. Fliedl, and M. Fuchs, “Classification of Customer Reviews based on Sentiment Analysis,” Inf. Commun. Technol. Tour. 2012, pp. 460–470, 2012, doi: 10.1007/978-3-7091-1142-0_40.

[6] Q. Sun, J. Niu, Z. Yao, and H. Yan, “Exploring eWOM in online customer reviews: Sentiment analysis at a fine-grained level,” Eng. Appl. Artif. Intell., vol. 81, pp. 68–78, 2019, doi: 10.1016/j.engappai.2019.02.004.

[7] J. Jin, P. Ji, and R. Gu, “Identifying comparative customer requirements from product online reviews for competitor analysis,” Eng. Appl. Artif. Intell., vol. 49, pp. 61–73, 2016, doi: 10.1016/j.engappai.2015.12.005.

[8] F. e-M. K. Khan, B.B. Baharudin, A. Khan, “Mining opinion from text documents,” Adv. Res. Comput. Commun. Eng., vol. 3, no. 7, pp. 217–222.

[9] B. Pang and L. Lee, “Opinion mining and sentiment analysis,” Found. Trends Inf. Retr., vol. 2, no. 1–2, pp. 1–135, 2008, doi: 10.1561/1500000011.

[10] M. Z. Asghar, A. Khan, S. Ahmad, and F. M. Kundi, “A Review of Feature Extraction in Sentiment Analysis,” J. Basic. Appl. Sci. Res, vol. 4, no. 3, pp. 181–186, 2014.

[11] A. Kumar, K. Srinivasan, W. H. Cheng, and A. Y. Zomaya, “Hybrid context enriched deep learning model for fine-grained sentiment analysis in textual and visual semiotic modality social data,” Inf. Process. Manag., vol. 57, no. 1, 2020, doi: 10.1016/j.ipm.2019.102141.

[12] S. Wang, D. Li, Y. Wei, and H. Li, “A Feature Selection Method Based on Fisher ’s Discriminant Ratio for Text Sentiment Classification Suge,” Springer-Verlag Berlin Heidelb., pp. 88–97, 2009.

[13] A. Esmin and S. Matwin, “Hierarchical Classification Approach to Emotion Recognition in Twitter Hierarchical Classification Approach to Emotion Recognition in Twitter,” 2016, no. March, doi: 10.1109/ICMLA.2012.195.

[14] W. Y. and L. H. A. Wang S, Li D, Song X, “A feature selection method based on improved fisher’s discriminant ratio for text sentiment classification,” Expert Syst. Appl., vol. 38, no. 7, pp. 8696-8702., 2011.

[15] L. C. H. and C. H. Chen LS, “A neural network based approach for sentiment classification in the blogosphere.,” J. Informetr., vol. 5, no. 2, pp. 313–322, 2011.

[16] M. Ahmad, S. Aftab, M. S. Bashir, N. Hameed, I. Ali, and Z. Nawaz, “SVM optimization for sentiment analysis,” Int. J. Adv. Comput. Sci. Appl., vol. 9, no. 4, pp. 393–398, 2018, doi: 10.14569/IJACSA.2018.090455.

[17] M. Abbas, K. Ali Memon, and A. Aleem Jamali, “Multinomial Naive Bayes Classification Model for Sentiment Analysis,” IJCSNS Int. J. Comput. Sci. Netw. Secur., vol. 19, no. 3, p. 62, 2019, [Online]. Available: http://paper.ijcsns.org/07_book/201903/20190310.pdf.

[18] J. Khairnar and M. Kinikar, “Sentiment Analysis Based Mining and Summarizing Using SVM-MapReduce,” Int. J. Comput. Sci. Inf. Technol., vol. 5, no. 3, pp. 4081–4085, 2014.

[19] B. Liu, “Sentiment Analysis and Opinion Mining,” Synth. Lect. Hum. Lang. Technol., vol. 5, no. 1, pp. 1–167, 2012, doi: 10.2200/S00416ED1V01Y201204HLT016.

[20] V. S. D. F. X. Christopher, “Evolving Trends in Conversational Systems with Natural Language Processing,” Int. J. Comput. Intell. Informatics, vol. 8, no. 3, pp. 123–129, 2018.

[21] A. See, “Natural Language Processing with Deep Learning: Natural Language Generation,” pp. 1–39, 2019.

[22] K. Dashtipour, M. Gogate, A. Adeel, H. Larijani, and A. Hussain, “Sentiment analysis of persian movie reviews using deep learning,” Entropy, vol. 23, no. 5, 2021, doi: 10.3390/e23050596.

[23] Q. Tul et al., “Sentiment Analysis Using Deep Learning Techniques: A Review,” Int. J. Adv. Comput. Sci. Appl., vol. 8, no. 6, 2017, doi: 10.14569/ijacsa.2017.080657.

[24] N. C. Dang, M. N. Moreno-García, and F. De la Prieta, “Sentiment analysis based on deep learning: A comparative study,” Electron., vol. 9, no. 3, 2020, doi: 10.3390/electronics9030483.

[25] H. Du, X. Xu, X. Cheng, D. Wu, Y. Liu, and Z. Yu, “Aspect-Specific Sentimental Word Embedding for Sentiment Analysis of Online Reviews,” Proc. 25th Int. Conf. Companion World Wide Web, pp. 29–30, 2016, doi: 10.1145/2872518.2889403.

[26] D. Nguyen, K. Vo, D. Pham, M. Nguyen, and T. Quan, “A deep architecture for sentiment analysis of news articles,” Adv. Intell. Syst. Comput., vol. 629, pp. 129–140, 2018, doi: 10.1007/978-3-319-61911-8_12.

[27] X. Ouyang, P. Zhou, C. H. Li, and L. Liu, “Sentiment analysis using convolutional neural network,” Proc. - 15th IEEE Int. Conf. Comput. Inf. Technol. CIT 2015, 14th IEEE Int. Conf. Ubiquitous Comput. Commun. IUCC 2015, 13th IEEE Int. Conf. Dependable, Auton. Se, pp. 2359–2364, 2015, doi: 10.1109/CIT/IUCC/DASC/PICOM.2015.349.

[28] S. Liao, J. Wang, R. Yu, K. Sato, and Z. Cheng, “CNN for situations understanding based on sentiment analysis of twitter data,” Procedia Comput. Sci., vol. 111, pp. 376–381, 2017, doi: 10.1016/j.procs.2017.06.037.

[29] P. Cen, K. Zhang, and D. Zheng, “Sentiment Analysis Using Deep Learning Approach,” J. Artif. Intell., vol. 2, no. 1, pp. 17–27, 2020, doi: 10.32604/jai.2020.010132.

[30] Q. Shen, Z. Wang, and Y. Sun, “Sentiment analysis of movie reviews based on CNN-BLSTM,” IFIP Adv. Inf. Commun. Technol., vol. 510, pp. 164–171, 2017, doi: 10.1007/978-3-319-68121-4_17.

[31] “https://www.kaggle.com.” .