İSTANBUL TECHNICAL UNIVERSITY
FACULTY OF COMPUTER AND INFORMATICS

COMPARISON OF CONVENTIONAL AND NEURAL NETWORK CLASSIFIERS FOR SENTIMENT ANALYSIS OF MOVIE REVIWS

Graduation Project Final Report

Melih Kağan Özçelik
150160050

Department: Computer Engineering
Division: Computer Engineering

Advisor: Prof. Dr. Şule Gündüz Öğüdücü

January 2024

Statement of Authenticity

I/we hereby declare that in this study

1. all the content influenced by external references is cited clearly and in detail,
2. and all the remaining sections, especially the theoretical studies and implemented software/hardware that constitute the fundamental essence of this study is originated from my/our own authenticity.

İstanbul, 15.01.2024

Melih Kağan Özçelik

COMPARISON OF CONVENTIONAL AND NEURAL NETWORK CLASSIFIERS FOR SENTIMENT ANALYSIS OF MOVIE REVIWS

(SUMMARY)

FİLM İNCELEMELERİNİN DUYGUSAL ANALİZİ İÇİN GELENEKSEL VE SİNİR AĞLARI SINIFLANDIRICILARININ KARŞILAŞTIRILMASI

(ÖZET)

Contents

1 Introduction and Project Summary

In today's web world, it is almost impossible to imagine the existence of an application that does not contain comments and reviews. Sharing emotions and thoughts on the Internet is now a standard behavior for people with the influence of social media. To be able to "mine "these thoughts and decide whether the emotional tone of the writer is positive, negative or neutral, sentiment analysis has become a popular subject in Natural Language Processing. Numerous application areas of sentiment analysis like predicting sales performance, box-office revenues, stock market even election results clarify the strong motivations for researches and shows the necessity of automated systems for this task [1].

According to Liu, sentiment analysis can be performed on different levels[1]:
Document Level: Assuming that each document expresses opinions on a single topic, document level analysis classify wheter a overall sentiment of that document is positive or negative.
Sentence Level: First, sentence level analysis determines whether each sentece has expressed an opinion or not, then it focuses the polarity of opinion.
Aspect Level: Aspect level analysis directly focuses the opinion itself and needs to consider both opinion and target of that opinion. This level performs a finer analysis and it is more challenging from document and sentece level analysis.

This research presents the findings derived from experiments of sentiment analysis of movie reviews on document level and aims to provide comparative results of conventional machine learning approaches and deep learning approaches. The data utilized for this study is contains 50000 movie reviews collected from IMDB and introduced by Maas et al[2].

Sentiment analysis generally begins with text preprocessing that includes normalization, tokenization, removing stop words then it continiues with feature extreaction methods like bag of words(BOW), term frequency-inverse document frequency (TF-IDF), word embeddings finally ends with appliying different learning algorithms like logistic regression, decision trees or neural networks for classification[3]. While the feature extraction methods BOW and TF-IDF used widely due to their simplicty and efficiency they can struggle dealing negation and long-range word ordering. Therefore, neurel network approaches can be follow giving pre-trained GloVe and Word2Vec embeddings as input to Long Short-Term Memory(LTSM) model and for comparasion implementations of conventional methods like Multinomial Naïve Bayes , Support Vector Machines can used with BOW and IT-IDF[4]. Another possible neural network approach Gated Recurrent Units(GRU), which has simpler structure, can also utilize pre-trained word embedings for sentiment classification[5]. Alternative to these methods, a more recent way for language understanding, Bidirectional Encoder Representation for Transformer (BERT) presented by Google Research[6] and its distilled, smaller, faster and ligther form DistilBERT[7] can achive high accuries on sentiment classification[8].

This research combines four disctint conventional methods consist of Logistic Regression, Support Vector Machine, Multinomial Naïve Bayes and Random Forest with two disctint feature extraction methods Bag of Words and Term Frequency-Inverse Document Frequency. In additon to conventional methods, LTSM, GRU models with GloVe ebmeddings and DistilBERT model selected as neurel network approaches to present comparative results of sentiment analysis in hotel reviews.

2 Literature Survey

For the field sentiment analysis, Liu provides a comprehensive introduction, analyze different levels of sentiment analysis, shows its usage areas and mentions latest developments in that area [1]. Ahuja et al. analyzes the impact of the feature extraction techniques BOW and TF-IDF by using conventional methods like logistic regression, Naïve Bayes, random forest and support vector machine also provides a road map for applying conventional models. Their results showed that TF-IDF method gives slightly better performance for sentiment analysis [3]. Pang et al. examine the sentiment analysis problem comprehensively and presents detailed explanations for machine learning techniques applied to solve that problem. Moreover, their study compares the results different n-grams for feature extraction [9]. Madasu and Sivasankar evaluate the impact of feature extraction methods TF-IDF and doc2vec to performances of conventional models like logistic regression, support vector machines and Bernoulli Naïve Bayes [10]. Pennington et al. propose a model that produces a word vector space, GloVe, which can be used for feature extraction in text inputs [14].

With the frequent use of neural networks different approaches followed to solve sentiment analysis problem. Barry's study shows the impact of different word embeddings on LTSM's by comparing them his baseline approaches Multinomial Naïve Bayes and Support Vector Machines. His results show success of LTSM's for sentiment analysis but does not fail the MNB and SVM [4]. Zouzo and Azami conducted a study on sentiment analysis of movie reviews and compared the performances of CNN, GRU and their combined models. Their study shows that GRU model can achieve up to %80 accuracies predicting sentiments of reviews [5]. Devlin et al. introduced a new language model called BERT and opened a new way for transfer learning. BERT can be successfully fine-tuned for wide range of NLP tasks such as sentiment analysis [6]. Sanh et al. present faster and smaller version of BERT which has almost the same capability called DistilBERT [7]. Joshy and Sundar compares BERT with it is versions DistilBERT and RoBERTa for sentiment analysis task. In their results DistilBERT and RoBERTa gives close performance however BERT archives higher accuracies [8]. Studies [16,17] presents detailed approaches to implement CNN's and LTSM's for solving sentiment analysis problem on movie reviews. Sachin et al. compares GRU, LTSM along with the Bi-GRU and Bi-LTSM according to their performances on sentiment analysis of movie reviews. Their results show each model predicts the sentiment around %70 accuracy.

3 Developed Approach and System Model

As can be seen in [3], before applying any feature extraction and classification method, describing the dataset and using appropriate text preprocessing techniques are essential. The dataset chosen for this study contains evenly distributed total of 50000 positive and negative reviews. Reviews were collected from IMDB considering only polarized reviews and spreading over different movies [2]. Reviews are in raw form in the dataset therefore it contains a lot of contractions, HTML tags and punctuations. However, to be able to have true unseen, real life form of test data, dataset partitioned (80:20) to train and test sets before applying any preprocessing methods.

For text preprocessing these following steps considered:

- Noise removal: Because the dataset is raw, it contains a lot of HTML tags, punctuations and URLs, these have been cleared from training set.
- Normalization: Review texts includes a lot of constraints like “he’s”, “isn’t”. These constraints normalized as “he is”, “is not”. Also, entire training set lowercased. Sentiment labels encoded as 1-0 instead of “positive”- “negative”.
- Tokenization: Whole review text broken into word tokens to construct feature vectors.
- Removing Stop Words: Stop words left as is because removing them did not improve the accuracies of models significantly.

3.1 Conventional Models

At the pretraining processes of conventional models, feature extraction methods bag of words and term frequency-inverse document frequency is used to obtain feature vectors. Bag of word method vectorizes text input as counting the number of occurrences of unigrams, bigrams etc. [9]. TF-IDF method calculates weight of the term in a document by multiplying term frequency and inverse document frequency to extract feature vectors: [10].

$$TF = \frac{Number\ of\ times\ the\ term\ appers\ in\ document}{Total\ number\ of\ terms\ in\ that\ document}$$

$$IDF = \log_e \frac{Total\ Number\ of\ documents}{Total\ numver\ of\ documents\ that\ includes\ the\ term}$$

All the conventional classification algorithms to be evaluated trained with both BOW and TF-IDF feature vectors to achieve detailed comparison. Moreover, these algorithms trained with and without removing stop words to see the effect.

3.1.1 Logistic Regression

One of the popular models of Generalized Linear Models, Logistic regression also called Maximum Entropy [3], predicts the class of a sentiment by calculating the weights or coefficients and estimates a multiple linear function where S is the probability of presence the feature, b’s are the coefficients and M’s are the input features [11]:

$$\mathrm{LR(S)} = \mathrm{b0} + \mathrm{b1M1} + \mathrm{b2M2} + \mathrm{b3M3} \ldots . \mathrm{bkMk} \ldots \ldots$$

3.1.2 Support Vector Machine

Support Vector Machine is quite effective method of conventional classification methods which finds a hyper plane vector to separate feature vectors from each other and maximize the margin of that separation [9]. For hyperplane vector $\vec{w}$, letting $c_j \in \{-1, 1\}$ be the correct class of document d_j solution can be formulated as [9]:

$$\vec{w} := \sum_j \alpha_j c_j \vec{d_j}, \alpha_j \geq 0,$$

3.1.3 Multinomial Naïve Bayes

Even with the assumption of the thinking words are independent from each other, Naïve Bayes can still achieve good results for sentiment analysis [4]. Multinomial Naïve Bayes (MNB) which is a probabilistic approach similar to Naïve Bayes works based on multinomial distributions of term frequencies and it can be considered as a suitable method for sentiment analysis [12]. MNB can be formulated, where n is number of terms in a document, 1 and |V| are smoothing constants, for document d and its class c as [13]:

$$P(c|d) \propto \mathrm{P(c)} \prod_{1 \leq k \leq n} P(\, t_k \mid \mathrm{c})$$

$$P(\, t_k \,|c\,) = \frac{(number\ of\ time\ term\ t_k\ occurs\ in\ d\ of\ class\ c\,) + 1}{(total\ number\ of\ terms\ in\ d\ of\ class\ c) + |V|}$$

$$P(c) = \frac{number\ of d\ of\ class\ c}{total\ number\ of\ d}$$

3.1.4 Random Forest

Random Forest algorithm works with group of decision trees where each tree votes for class then it classifies the class by selecting the winner with most votes [3]. For this study number of trees selected as 100 and classification criteria selected as Gini Impurity.

3.2 Neural Network Models

For the pretraining process of neural network models, pre-trained word embeddings used to form feature vectors. LTSM and GRU models trained with GloVe(Global vectors for word representation) which trained on 1.9 million unique words[14]. GloVe embeddings applied to LTSM and GRU models in four steps by following the Barry's study [4]. Text inputs converted to integer sequences and padded into length of the longest sequence. Then, to use pre-trained word embedding vectors an embedding matrix constructed. Finally, the embedding matrix is used to feed neural network with a embedding layer. On the other hand, for BERT model, texts are tokenized using WordPiece and BERT vocabulary size of 30000 [6,7], encoded sequences padded into maximum length and prepared for being input to the model. WordPiece tokenization starts with small vocabulary size but it applies iterative merging and handles out-of-vocabulary words efficiently [15].

3.2.1 Long Short-Term Memory

Long short-term memory is a version of recurrent neural network (RNN) developed to overcome gradient problems that can occur when training traditional RNN's [16]. LTSM model uses input, forget and output gates to regulate memory cell (LTSM unit) and saves long-term dependencies efficiently with these gates [17]. LTSM architecture can be formulated as where x_t is input word, h_{t-1} is past hidden state, c is memory cell, W's and U's are weight matrices [4,16]:

$$Input\ gate: i_t = \sigma(W^{(i)}x_t) + U^{(i)}h_{t-1}$$
$$Forget\ gate: f_t = \sigma(W^{(f)}x_t) + U^{(f)}h_{t-1}$$
$$Output\ gate: o_t = \sigma(W^{(o)}x_t) + U^{(o)}h_{t-1}$$
$$New\ memory\ cell: \widetilde{c_t} = \tanh(W^{(c)}x_t) + U^{(c)}h_{t-1}$$
$$Final\ memory\ cell: c_t = f_t * c_{t-1} + i_t * \widetilde{c_t}$$
$$h_t = o_t * \tanh(c_t)$$

For the implementation of LTSM following layered structure is used:

- Embedding Layer: GloVe embeddings is used with size 300 so it is used as output dimension of embedding layer, number of unique words (vocabulary size) provide as input dimension. Input length is used as length of the longest sequence which other sequences padded to that length. Embedding matrix that created in pretraining processes used for weights. The parameters of this layer set as untrainable because of it is pre-trained.
- LTSM Layer: Optimal layer size selected as 300 and dropout value of 40% applied to prevent overfitting.
- Fully Connected Layer (Dense): Layer with 100 units that applies linear transformation, connects each input to every node. It has been observed to increase performance when it added before output layer.
- Classification Layer (Dense): The output layer consists of one node for binary classification. Layer activation function chosen as Sigmoid that converts output value of network to probability of reviews classes.

Adam optimizer used to optimize model and loss function selected as binary cross entropy. %20 of the training data used as validation data to observe the model and find appropriate hyperparameters. Number of epochs set to 9 during training and early stopping mechanism used according to validation accuracy.

3.2.2 Gated Recurrent Unit

Gated Recurrent Unit (GRU) is a version of RNN's that solving vanishing gradient problems similar to LTSM, controls the movement of information in the unit without usage of extra defined memory cells [18]. Generally, GRU can be formulated as where x_t is input vector, h_t is output vector, z_t is update vector, σ and tanh are activation functions [19]:

$$z_t = \sigma(W_z.[h_{t-1}, x_t])$$
$$r_t = \sigma(W_r.[h_{t-1}, x_t])$$
$$\widetilde{h_t} = tanh(W.[r_t * h_{t-1}, x_t])$$
$$h_t = (1 - z_t)^{\circ} h_{t-1} + z_t * \widetilde{h_t}$$

For the implementation of GRU following layered structure is used:

- Embedding Layer: Applied using same parameters as in LTSM model's implementation.
- GRU Layer: Optimal layer size selected as 150 and dropout value of 40% applied to prevent overfitting.
- Classification Layer (Dense): Applied using same parameters as in LTSM model's implementation.

Adam optimizer used to optimize model and loss function selected as binary cross entropy. %20 of the training data used as validation data to observe the model and find appropriate hyperparameters. Number of epochs set to 9 during training and early stopping mechanism used according to validation accuracy.

3.2.3 DistilBERT

BERT is a pretrained general purpose NLP model which uses Masked Language Modelling (MLP) and Next Sentence Prediction (NSP) methods in its pretraining phase, can applied to task specific datasets [8]. Batra et al. explains MLP and NSP as, in MLP method model tries to predict randomly masked words according to non-masked words in text [20]. In NSP method, during the training of model half of the inputs paired as (sentence, subsequent sentence) while the other half paired as (sentence, random sentence from the corpus), with this setup model learns to predict whether the second sentence in the pair is the subsequent sentence or not. Furthermore, they mention the advantages of fine-tuning BERT model for specific classification tasks, like less training time with few epochs, ease of application, flexible data requirements.

For the implementation of DistilBERT, text inputs encoded with WordPiece tokenizer which is the default tokenization for DistilBERT. Then input sequences padded and truncated for maximum length of 192. Dense layer with sigmoid activation is used for classification layer. %10 of the training data used as validation data to observe the model. Finally, the model fine-tuned for 3 epochs using Adam optimizer with learning rate of 1e-5 and loss function selected as binary cross entropy.

4 Experimental Environment and Design

For experimental environment Anaconda's Python (3.9.18) Distribution used with Jupyter notebook. Pandas and NumPy libraries used for basic data operations. Matplotlib library used for basic visualizations of dataset. Texthero [21] library is used for data preprocessing Scikit-learn [22] is used for the evaluation metrics and implementations of Logistic Regression, Support Vector Machine, Random Forest and Multinomial Naïve Bayes models. Keras [13] is used to implement neural network models. To utilize GPU on model training TensorFlow's GPU distribution selected and used with cudatoolkit and cuDNN. All of the experiments performed on a computer with 16 GB of RAM, Ryzen 7 5800H CPU and GeForce RTX 3060 Laptop GPU.

5 Comparative Evaluation and Discussion

Conventional methods Logistic Regression (LR), Support Vector Machine (SVM), Multinomial Naïve Bayes (MNB) and Random Forest (RF) models trained for times with the combination of feature extraction methods and with or without stop word removal. Table 5.1 summarizes the final result achieved on the test data.

Table 5.1: Comparison of feature extraction methods and effect of stop words

Model / Feature Extraction	Accuracies Without stop word removal	Acc. With stop word removal
LR / BOW	88.99	88.84
LR / TF-IDF	90.25	90.36
SVM / BOW	87.01	87.06
SVM / TF-IDF	90.07	90.13
MNB / BOW	84.3	84.85
MNB / TF-IDF	85.78	85.57
RF / BOW	85.63	87.26
RF / TF-IDF	83.15	85.25

As can be seen in the results TF-IDF reaches higher accuracies except Random Forest Model. Removing stop words does not improve the performance results except RF model. Logistic Regression model performed the best result of 90,25% among conventional models nevertheless RF and MNB models ranked last with still satisfying accuracies around 84%.

Neural network classifiers trained with different variations of parameters multiple times to find optimal result. Furthermore, stop words never removed for training neural networks. Figure 5.1 represents the final results of neural network models. Interpreting the results, DisilBERT obtained best performance with accuracy score of 89,54%. Although LTSM model performed around 84,93% of accuracy score on validation data, it remains at 67,81% accuracy on testing data. The same situation is also observed in the GRU model. GRU model performed only 67,15% accuracy on test data while reaching 88,65% on validation data. Considering that validation data is split from preprocessed training data, it can be concluded that this situation is due to the GloVe vocabulary is not successful to cover the raw text data compared to preprocessed text. However, BERT is not affected by this situation due to its ability to understand contextual information also BERT tokenize text into smaller pieces than words.

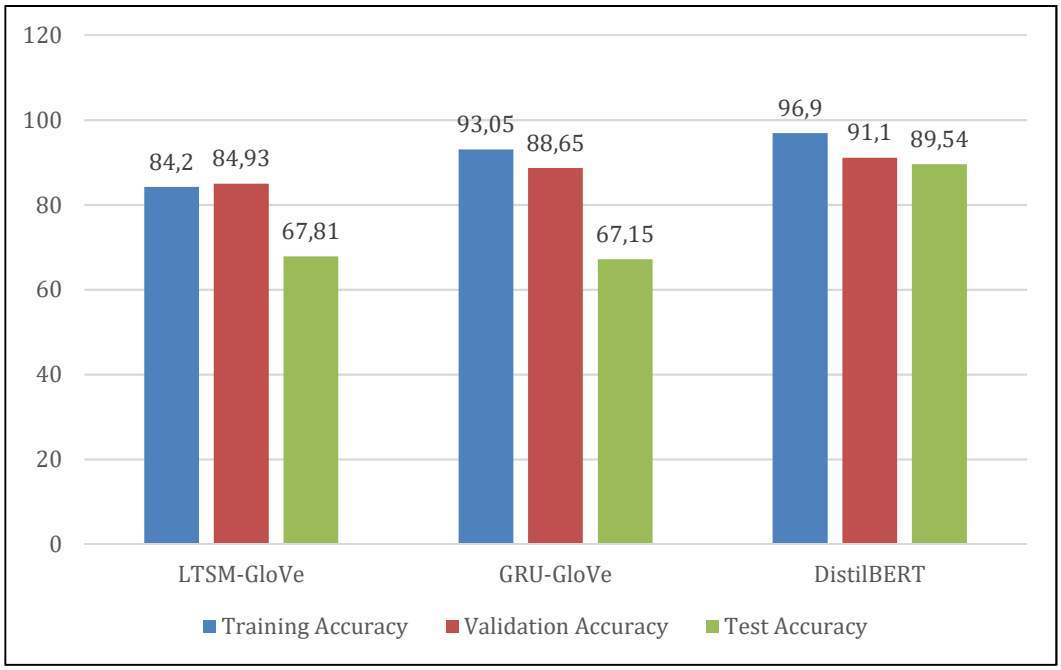

Figure 5.1: Results obtained from neural network models

For comprehensive comparisons of both conventional and neural networks models, test accuracies selected from conventional models with TF-IDF without stop word removal and combined to test accuracies of neural network models. Figure 5.2 visualizes the overall comparison of the models applied throughout this study.

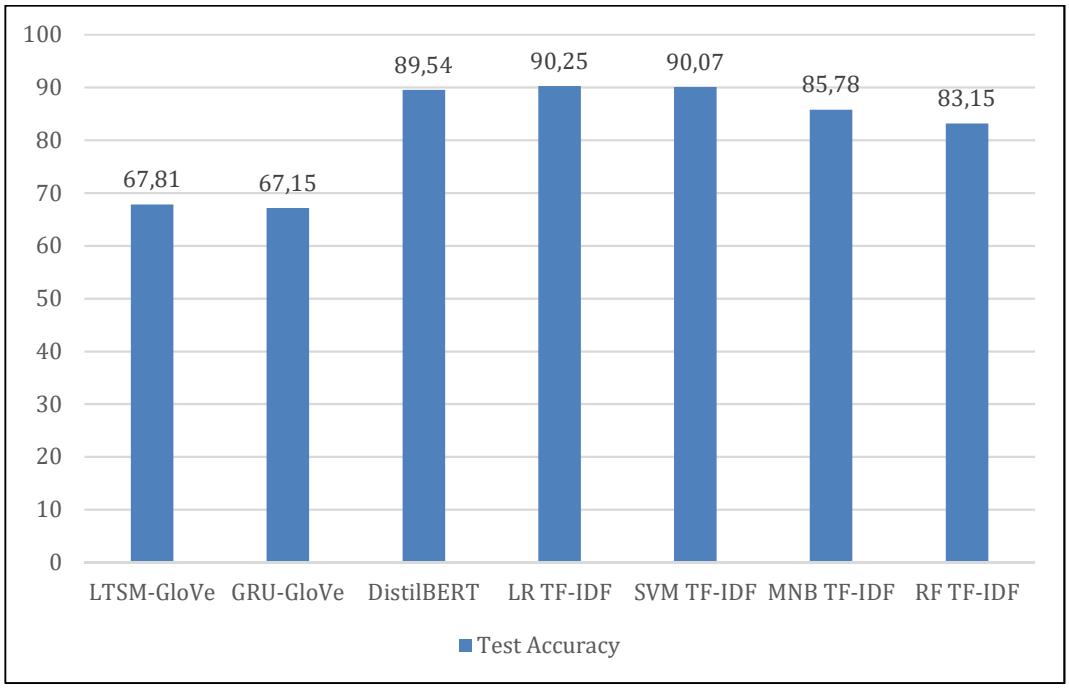

Figure 1.3: Overall accuracy comparison of proposed models

As can be observed from Figure 1.3, LR with TF-IDF takes first place with accuracy of 90,25% and followed by DistilBERT and SVM with TF-IDF with quite close accuracies. LTSM and GRU models with GloVe were performed weak in terms of accuracy compared to others due to feature extraction mechanism. Considering the training process takes at least

100 times more in neural networks for this study, conventional methods LR and SVM is fairly successful for binary sentiment analysis.

In general terms, results obtained from this study presents detailed comparison of conventional and neural networks models as well as the detailed comparison of feature extraction methods. Results showed that to overcome sentiment analysis problem fast and low time cost solutions like conventional methods still can be applied in some cases. However, it should be mentioned that in this study basic versions of LTSM, GRU and DistilBERT are applied. Therefore, extending these models with improvements may allow to achieve higher precisions on predictions. The key point is adapting appropriate preprocessing and feature extraction methods to appropriate model.

6 Conclusion and Future Work

In this study conventional and neural network approaches used for sentiment analysis analyzed and compared according to their performances. Firstly, text preprocessing steps evaluated and removing stop words discussed. Then LR, SVM, MNB and RF models selected as conventional models and implemented with two different feature extraction technique BOW and TF-IDF. Finally, LTSM and GRU models implemented with GloVe embeddings and DistilBERT model implemented with WordPiece tokenizer. Results obtained from all of these models used for comprehensive comparison from the perspective of solving sentiment analysis problem.

This comparative study of sentiment analysis methods revealed several key insights. Text preprocessing has different effects to different models. Stop word removal did not provided considerable benefit. Among text vectorization techniques, TF-IDF generally outperformed bag-of-words (BOW) in conventional algorithms. While deep learning techniques like LSTMs and GRUs using GloVe embeddings struggled on the test set, DistilBERT achieved superior performance. Classic machine learning approaches provide surprisingly effective performance for basic sentiment analysis. These findings suggest that, depending on the specific context and resource constraints, both machine and deep learning methods can be viable options for sentiment analysis.

Several improvements can be applied the proposed methodologies in this study. Instead of using bag of words, bag of n-grams can be applied as one of the feature extraction techniques. Instead of use pretrained word embeddings, task specific embeddings can be trained to fed LTSM and GRU layers. To achieve better comparison different pre-trained word embeddings like fastText, Word2Vec, ELMo can be used along with the GloVe. It is also possible to adapt Bi-LTSM, Bi-GRU, variations of CNN’s and combination of those models.

7 References

[1] B. Liu, "Sentiment analysis and opinion mining", Synthesis Lectures on Human Language Technologies, vol.5, no.1, p. 1-167, 2012. https://doi.org/10.2200/s00416ed1v01y201204hlt016
[2] 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 the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, 2011, pp. 142–150.
[3] R. Ahuja, A. Chug, S. Kohli, S. Gupta, & P. Ahuja, "The impact of features extraction on the sentiment analysis", Procedia Computer Science, vol. 152, p. 341-348, 2019. https://doi.org/10.1016/j.procs.2019.05.008
[4] J. Barry, 'Sentiment Analysis of Online Reviews Using Bag-of-Words and LSTM Approaches', in Irish Conference on Artificial Intelligence and Cognitive Science, 2017.
[5] A. Zouzou and I. Azami, "Text sentiment analysis with CNN & GRU model using glove", 2021 Fifth International Conference on Intelligent Computing in Data Sciences (ICDS), 2021. https://doi.org/10.1109/icds53782.2021.9626715
[6] J. Devlin, M. Chang, K. Lee, & K. Toutanova, "Bert: pre-training of deep bidirectional transformers for language understanding", 2018. https://doi.org/10.48550/arxiv.1810.04805
[7] V. Sanh, L. Debut, J. Chaumond, & T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter", 2019. https://doi.org/10.48550/arxiv.1910.01108
[8] A. Joshy and S. Sundar, "Analyzing the performance of sentiment analysis using bert, distilbert, and roberta", 2022 IEEE International Power and Renewable Energy Conference (IPRECON), 2022. https://doi.org/10.1109/iprecon55716.2022.10059542
[9] B. Pang, L. Lee, & S. Vaithyanathan, "Thumbs up? sentiment classification using machine learning techniques", 2002. https://doi.org/10.48550/arxiv.cs/0205070
[10] A. Madasu and E. Sivasankar, "A study of feature extraction techniques for sentiment analysis", 2019. https://doi.org/10.48550/arxiv.1906.01573
[11] A. Prabhat and V. Khullar, 'Sentiment classification on big data using Naïve bayes and logistic regression', in 2017 International Conference on Computer Communication and Informatics (ICCCI), 2017, pp. 1–5.
[12] P. P. M. Surya, L. V. Seetha, and B. Subbulakshmi, 'Analysis of user emotions and opinion using Multinomial Naive Bayes Classifier', in 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), 2019, pp. 410–415.
[13] G. Singh, B. Kumar, L. Gaur, and A. Tyagi, 'Comparison between Multinomial and Bernoulli Naïve Bayes for Text Classification', in 2019 International Conference on Automation, Computational and Technology Management (ICACTM), 2019, pp. 593–596.
[14] J. Pennington, R. Socher, & C. Manning, "Glove: global vectors for word representation", Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014. https://doi.org/10.3115/v1/d14-1162
[15] M. Schuster and K. Nakajima, 'Japanese and Korean voice search', in 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2012, pp. 5149–5152.
[16] G. Murthy, S. Allu, B. Andhavarapu, & M. Bagadi, "Text based sentiment analysis using lstm", International Journal of Engineering Research And, vol. V9, no. 05, 2020. https://doi.org/10.17577/ijertv9is050290

[17] A. Rehman, A. Malik, B. Raza, & W. Ali, "A hybrid cnn-lstm model for improving accuracy of movie reviews sentiment analysis", Multimedia Tools and Applications, vol. 78, no. 18, p. 26597-26613, 2019. https://doi.org/10.1007/s11042-019-07788-7
[18] S. Sachin, A. Tripathi, N. Mahajan, S. Aggarwal, & P. Nagrath, "Sentiment analysis using gated recurrent neural networks", SN Computer Science, vol. 1, no. 2, 2020. https://doi.org/10.1007/s42979-020-0076-y
[19] Y. Santur, "Sentiment analysis based on gated recurrent unit", 2019 International Artificial Intelligence and Data Processing Symposium (IDAP), 2019. https://doi.org/10.1109/idap.2019.8875985
[20] H. Batra, N. Punn, S. Sonbhadra, & S. Agarwal, "Bert-based sentiment analysis: a software engineering perspective", Lecture Notes in Computer Science, p. 138-148, 2021. https://doi.org/10.1007/978-3-030-86472-9_13
[21] "Texthero · Text preprocessing, representation and visualization from zero to hero." https://texthero.org/ (accessed Jan 18, 2023)
[22] F. Pedregosa et al., 'Scikit-learn: Machine Learning in Python', Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.
[23] F. Chollet and Others, 'Keras', 2015. [Online]. Available: https://keras.io