İSTANBUL TECHNICAL UNIVERSITY
FACULTY OF COMPUTER AND INFORMATICS

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

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 REVIEWS

(SUMMARY)

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. Sentiment analysis can be performed on different levels, document level, sentence level, and aspect level. Sentiment analysis generally begins with text preprocessing then continues with feature extraction finally ends with applying different learning algorithms. This research presents the findings derived from experiments of sentiment analysis of movie reviews on the document level and aims to provide comparative results of conventional machine learning approaches and deep learning approaches.

The dataset chosen for this study contains an evenly distributed total of 50000 positive and negative reviews collected from IMDB. Reviews were in raw form in the dataset therefore some of the text preprocessing steps were applied. For noise removal, HTML tags, punctuations, and URLS were cleaned. For normalization constraints fixed, sentiment labels are encoded and the entire training set is lowercased. Removing stop words is discussed and its effect is evaluated in the results. Whole review text tokenize to construct feature vectors.

Before the training of conventional models, feature extraction methods bag of words (BOW) and term frequency–inverse document frequency were selected and applied separately for each model to extract feature vectors. The first proposed conventional model for sentiment analysis is Logistic Regression which predicts the class by calculating the weights in a multiple linear function. Then Support Vector Machine which finds a hyperplane vector to separate feature vectors, is trained to predict the sentiment. After that, the Multinomial Naïve Bayes model which works based on multinomial distributions of term frequencies, was selected to be another candidate model for comparison. Finally, the Random Forest model, which classifies the sentiment using a group of decision trees, is trained on the dataset. Results obtained from these four conventional models compared with each other and also with neural network models. The effect of different feature extraction methods and removing stop words are observed.

For the neural network classifiers three different approaches are selected; Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Bidirectional Encoder Representation for Transformer (BERT). Global vectors for word representation (GloVe) are used in LSTM and GRU for feature extraction. To utilize GloVe, text inputs are converted to integer sequences by tokenizing and padding. These sequences are used with pre-trained GloVe

embeddings to form an embedding matrix. These embeddings matrix is used in embedding layers of LSTM and GRU models. In the LSTM model embedding layer is followed by an LSTM layer with a size of 300, then a fully connected layer size of 100, and finally, a classification layer consists of just one node for binary classification. Similar to LSTM, In the GRU model embedding layer is followed by a GRU layer with a size of 150, and then a classification layer consists of just one node for binary classification. The loss function is selected as binary cross entropy and the Adam optimizer is used in both LSTM and GRU models. DistilBERT was selected as the final approach in neural network models, which is a faster, smaller, distilled version of BERT. To implement DistilBERT for sentiment classification, text inputs tokenized with WordPiece, padded and truncated to turn them into sequences. These sequences are used for input to the DistilBERT model. A classification layer with sigmoid activations is used to handle the output of the DistilBERT model. Finally, the model was fine-tuned for 3 epochs using the Adam optimizer with a learning rate of 10^{-5} and loss function selected as binary cross entropy.

After the completion of the training step, both conventional and neural network models are compared with each other according to the accuracies they get on the test set. Results show that the TF-IDF feature extraction method gives better scores on conventional models except the Random Forest Model. Removing stop words did not improve the performance of conventional models at a valuable level. Logistic Regression achieved the best accuracy score of %90.25 without stop word removal and %90.36 with stop word removal. Among neural network models, DistilBERT performed the best results on both validation and training sets with accuracies of 91,1% and 89,54% in order. LSTM and GRU models could achieve around 85% accuracy on the validation set however they could not perform well on the test set due to the inability of GloVe embeddings to cover raw text data.

FİLM İNCELEMELERİNİN DUYGUSAL ANALİZİ İÇİN GELENEKSEL VE SİNİR AĞLARI SINIFLANDIRICILARININ KARŞILAŞTIRILMASI

(ÖZET)

Sosyal medyanın da etkisiyle insanların internette duygu ve düşüncelerini paylaşması artık standart bir davranış haline geldi. Bu düşünceleri analiz edebilmek ve yazarın duygusal tonunun pozitif, negatif veya nötr olduğunu belirleyebilmek için duygu analizi, Doğal Dil İşleme alanında popüler bir konu haline geldi. Duygu analizi belge düzeyinde, cümle düzeyinde ve bakış açısı düzeyinde olmak üzere üç farklı seviyede uygulanabilir. Duygu analizi genellikle metin ön işleme ile başlayıp ardından öznitelik çıkarma ile devam eder ve son olarak farklı öğrenme algoritmalarının uygulanmasıyla sona erer. Bu araştırma, film incelemelerinin belge düzeyinde duygu analizi için yapılan deneylerden elde edilen bulguları sunmakta ve geleneksel makine öğrenimi yaklaşımları ile derin öğrenme yaklaşımlarının karşılaştırmalı sonuçlarını vermeyi amaçlamaktadır.

Bu çalışma için seçilen veri seti, IMDB'den toplanmış, eşit sayıda olumlu ve olumsuz olmak üzere toplam 50000 film incelemesinden oluşmaktadır. Veri setindeki incelemeler ham halde olduğundan ötürü bazı metin ön işleme adımlarının uygulanması tercih edildi. Metinlerdeki gürültülü yapının giderilmesi için HTML etiketleri, noktalama işaretleri ve URL'ler temizlendi. Metini normal forma getirmek için kısaltmalar düzeltilip tüm eğitim seti küçük harfli metinlere dönüştürüldükten sonra, incelemenin duygusunu belirten etiketler sayısal olarak kodlandı. Duraksama kelimelerinin kaldırılıp kaldırılmayacağı tartışıldı etkisi değerlendirildi. Öznitelik vektörlerini oluşturmak için inceleme metinleri jetonlaştırıldı.

Geleneksel modellerin eğitimine başlamadan önce, her model için ayrı ayrı olmak üzere sözcük torbası (BOW) ve terim frekansı-ters belge frekansı (TF-IDF) öznitelik çıkarma yöntemleri uygulandı. Duygu analizi için önerilen ilk geleneksel model, çoklu doğrusal bir fonksiyondaki ağırlıkları hesaplayarak duygunun sınıfını tahmin eden Lojistik Regresyon modeli oldu. Ardından, öznitelik vektörlerini ayırmak için bir hiper düzlem vektörü bulan Destek Vektör Makinesi, duyguyu tahmin etmek için eğitildi. Daha sonra, multinominal terim frekansı dağılımlarına göre çalışan Multinomial Naïve Bayes modeli, karşılaştırma için başka bir aday model olarak seçildi. Son olarak, bir dizi karar ağacı kullanarak duyguyu sınıflandıran Rastgele Orman modeli ilgili veri setinde eğitildi. Bu dört geleneksel modelin elde ettiği sonuçlar, birbirleriyle ve sinir ağları modelleri ile karşılaştırıldı. Farklı öznitelik çıkarma yöntemlerinin ve duraksama kelimelerinin kaldırılmasının etkisi deneyler boyunca gözlemlendi.

Sinir ağı sınıflandırıcıları için üç farklı yaklaşım seçildi: Uzun Kısa Vadeli Bellek (LSTM), Geçitli Tekrarlayan Unite (GRU) ve Dönüşüm için Çift Yönlü Kodlayıcı Temsili (BERT). Kelime temsili için küresel vektörler (GloVe), öznitelik çıkartmak için LSTM ve GRU'da kullanıldı. GloVe'u kullanmak için, metin girdileri, jetonlaştırıldı ve aynı uzunluklara tamamlanarak sayı dizilerine dönüştürüldü. Bu diziler, bir gömme matrisi oluşturmak için önceden eğitilmiş GloVe gömmeleri ile kullanıldı. Bu gömme matrisi, LSTM ve GRU modellerinin gömme katmanlarında kullanıldı. Denylerde kullanılan LSTM modelinde, gömme katmanını 300 hücreli bir LSTM katmanı, ardından 100 hücreli tam bağlantılı bir katman ve son olarak, ikili sınıflandırma için yalnızca bir düğümden oluşan bir sınıflandırma katmanı takip eder. LSTM'ye benzer şekilde, GRU modelinde gömme katmanını 150 hücreli bir GRU katmanı ve ardından ikili sınıflandırma için yalnızca bir düğümden oluşan bir sınıflandırma katmanı takip eder. Yitim fonksiyonu olarak ikili çapraz entropi seçildi ve hem LSTM hem de GRU modellerinde Adam optimize edici kullanıldı. BERT'in daha hızlı, daha küçük, damıtılmış bir versiyonu olan DistilBERT, sinir ağı modellerinde son yaklaşım olarak seçildi. Duygu sınıflandırması için DistilBERT'i uygulamaya metin girdilerinin WordPiece ile jetonlaştırılması ile başlandı ve bu girdiler sabit uzunlukta olması için doldurulup ve kısaltılarak sayı dizilerine dönüştürüldü. Bu diziler, DistilBERT modelinin girişi için kullanılıp modelinin çıktısını işlemek için sigmoid işlevli bir sınıflandırma katmanı kullanıldı. Son olarak model, 10^{-5} öğrenme oranı, ikili çapraz entropi yitim fonksiyonu ve Adam optimizasyonu ile 3 dönem boyunca ince ayar yapılarak veri setine uygun hale getirildi.

Eğitim aşaması tamamlandıktan sonra, geleneksel ve sinir ağı modelleri, test setindeki başarılarına göre birbirleriyle karşılaştırıldı. Sonuçlar, TF-IDF öznitelik çıkarma yönteminin, Rastgele Orman Modeli hariç diğer geleneksel modellerde daha iyi puanlar verdiğini gösteriyor. Duraksama kelimelerini kaldırmak, geleneksel modellerin performansını anlamlı ölçüde geliştirmedi. Lojistik Regresyon, duraksama kelimelerini kaldırmadan %90.25 kaldırarak ise %90.36 ile en iyi doğruluk puanına ulaştı. Sinir ağı modelleri arasında, DistilBERT hem doğrulama hem de eğitim setlerinde sırasıyla %91,1 ve %89,54 doğruluk ile en iyi sonuçları verdi. LSTM ve GRU modelleri, doğrulama setinde yaklaşık %85 doğruluk elde edebildi, ancak GloVe'un ham metin halinde gelen içeriği tam olarak kapsayamaması nedeniyle test setinde iyi performans gösteremedi.

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 research and show 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 classifies whether the overall sentiment of that document is positive or negative.
- Sentence Level: First, sentence level analysis determines whether each sentence has expressed an opinion or not, and then it focuses on the polarity of opinion.
- Aspect Level: Aspect level analysis directly focuses on the opinion itself and needs to consider both the opinion and target of that opinion. This level performs a finer analysis and it is more challenging than document and sentence level analysis.

This research presents the findings derived from experiments of sentiment analysis of movie reviews on the document level and aims to provide comparative results of conventional machine learning approaches and deep learning approaches. The data utilized for this study 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, and removing stop words then it continues with feature extraction methods like bag of words(BOW), term frequency-inverse document frequency (TF-IDF), word embeddings finally ends with applying different learning algorithms like logistic regression, decision trees or neural networks for classification[3]. While the feature extraction methods BOW and TF-IDF are used widely due to their simplicity and efficiency they can struggle to deal with negation and long-range word ordering. Therefore, neural network approaches can be followed by giving pre-trained GloVe and Word2Vec embeddings as input to the Long Short-Term Memory(LSTM) model, and for comparison, 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 a 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 lighter form DistilBERT[7] can achieve high accuracies on sentiment classification[8].

This research combines four distinct conventional methods consisting of Logistic Regression, Support Vector Machine, Multinomial Naïve Bayes, and Random Forest with two distinct feature extraction methods Bag of Words and Term Frequency-Inverse Document Frequency. In addition to conventional methods, LSTM, GRU models with GloVe embeddings, and DistilBERT models were selected as neural 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, analyzes different levels of sentiment analysis, shows its usage areas, and mentions the latest developments in that area [1]. Ahuja et al. analyze 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 provide 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 present detailed explanations for machine learning techniques applied to solve that problem. Moreover, their study compares the results of different n-grams for feature extraction [9]. Madasu and Sivasankar evaluate the impact of feature extraction methods TF-IDF and doc2vec on the 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 are followed to solve sentiment analysis problems. Barry's study shows the impact of different word embeddings on LSTM by comparing them with his baseline approaches Multinomial Naïve Bayes and Support Vector Machines. His results show the success of LSTM for sentiment analysis but do 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 the GRU model can achieve up to %80 accuracy in 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 a wide range of NLP tasks such as sentiment analysis [6]. Sanh et al. present a faster and smaller version of BERT which has almost the same capability called DistilBERT [7]. Joshy and Sundar compare BERT with its versions DistilBERT and RoBERTa for sentiment analysis tasks. In their results DistilBERT and RoBERTa give close performance however BERT archives higher accuracies [8]. Studies [16,17] present detailed approaches to implementing CNNs and LSTMs for solving sentiment analysis problems in movie reviews. Sachin et al. compare GRU, LSTM along with the Bi-GRU, and Bi-LSTM 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 an 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 they contain a lot of contractions, HTML tags, and punctuations. However, to be able to have a true unseen, real-life form of test data, the dataset was partitioned (80:20) to train and test sets before applying any preprocessing methods.

For text preprocessing, the following steps are considered:

- Noise removal: Because the dataset is raw, it contains a lot of HTML tags, punctuations, and URLs, these have been cleared from the training set.
- Normalization: Review texts include a lot of constraints like “he’s”, “isn’t”. These constraints normalized as “he is”, “is not”. Also, the entire training set is lowercased. Sentiment labels are 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 are used to obtain feature vectors. The bag of word method vectorizes text input by counting the number of occurrences of unigrams, bigrams, etc. [9]. TF-IDF method calculates the 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 are to be evaluated and trained with both BOW and TF-IDF feature vectors to achieve detailed comparison. Moreover, these algorithms are 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 sentiment by calculating the weights or coefficients and estimates multiple linear function where S is the probability of the presence of 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 a quite effective method of conventional classification methods which finds a hyperplane 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 that the thinking words are independent of 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 the 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

The 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 the most votes [3]. For this study number of trees was selected as 100 and the classification criteria selected as Gini Impurity.

3.2 Neural Network Models

For the pretraining process of neural network models, pre-trained word embeddings are used to form feature vectors. LSTM and GRU models trained with GloVe (Global vectors for word representation) which trained on 1.9 million unique words [14]. GloVe embeddings are applied to LSTM and GRU models in four steps by following Barry's study [4]. Text inputs are converted to integer sequences and padded into a length of the longest sequence. Then, to use pre-trained word embedding vectors an embedding matrix is constructed. Finally, the embedding matrix is used to feed the neural network with an embedding layer. On the other hand, for the 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 a 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 networks (RNN) developed to overcome gradient problems that can occur when training traditional RNNs [16]. The LSTM model uses input, forget, and output gates to regulate memory cells (LSTM units) and saves long-term dependencies efficiently with these gates [17]. LSTM architecture can be formulated as where x_t is the input word, h_{t-1} is the past hidden state, c is a 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 LSTM following layered structure is used:

- Embedding Layer: GloVe embeddings are used with a size of 300 so it is used as the output dimension of the embedding layer, number of unique words (vocabulary size) is provided as the input dimension. Input length is used as the 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 are set as untrainable because of it is pre-trained.
- LSTM Layer: The optimal layer size was selected as 300 and a dropout value of 40% was applied to prevent overfitting.
- Fully Connected Layer (Dense): Layer with 100 units that applies linear transformation, and connects each input to every node. It has been observed to increase performance when it is added before the output layer.
- Classification Layer (Dense): The output layer consists of one node for binary classification. The layer activation function is chosen as Sigmoid which converts the output value of the network to the probability of review classes.

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

3.2.2 Gated Recurrent Unit

Gated Recurrent Unit (GRU) is a version of RNNs that solves vanishing gradient problems similar to LSTM and controls the movement of information in the unit without the 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 the same parameters as in the LSTM model's implementation.
- GRU Layer: The optimal layer size was selected as 150 and a dropout value of 40% was applied to prevent overfitting.
- Classification Layer (Dense): Applied using the same parameters as in the LSTM model's implementation.

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

3.2.3 DistilBERT

BERT is a pretrained general purpose NLP model that uses Masked Language Modelling (MLP) and Next Sentence Prediction (NSP) methods in its pretraining phase, and can applied to task specific datasets [8]. Batra et al. explain MLP and NSP as, in MLP method model tries to predict randomly masked words according to non-masked words in text [20]. In the NSP method, during the training of the model half of the inputs are paired as (sentence, subsequent sentence) while the other half are 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 the BERT model for specific classification tasks, like less training time with few epochs, ease of application, and flexible data requirements.

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

4 Experimental Environment and Design

For the experimental environment, Anaconda's Python (3.9.18) Distribution was used with Jupyter Notebook. Pandas and NumPy libraries are used for basic data operations. Matplotlib library is 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 was selected and used with cudatoolkit and cuDNN. All of the experiments were performed on a computer with 16 GB of RAM, a Ryzen 7 5800H CPU, and a 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 considerably except the RF model. The Logistic Regression model performed the best result of %90.25 without stop word removal and %90.36 with stop word removal among conventional models nevertheless RF and MNB models ranked last with still satisfying accuracies of around 84%.

Neural network classifiers are trained with different variations of parameters multiple times to find optimal results. Furthermore, stop words are never removed for training neural networks. Figure 5.1 represents the final results of neural network models. Interpreting the results, DisilBERT obtained the best performance with an accuracy score of 89,54%. Although the LSTM model performed 84,93% 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 in covering 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 tokenizes text into smaller pieces than words.

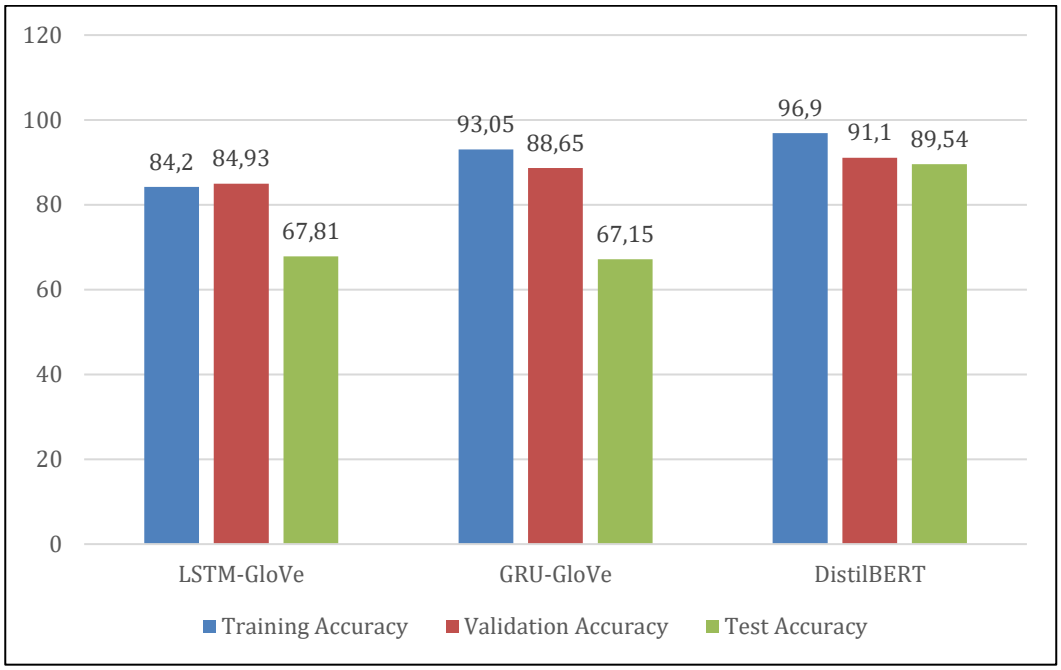

Figure 5.1: Results obtained from neural network models

For comprehensive comparisons of both conventional and neural network 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.

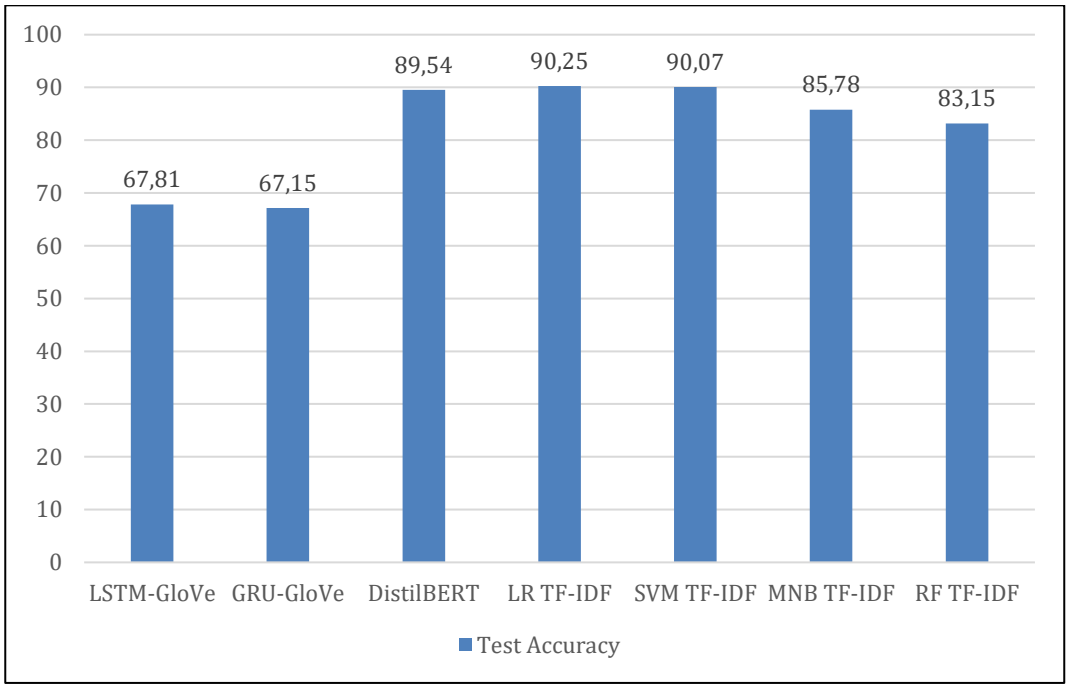

Figure 1.3: Overall accuracy comparison of proposed models

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

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

In general terms, results obtained from this study present a 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 problems 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 LSTM, GRU, and DistilBERT are applied. Therefore, extending these models with improvements may allow them to achieve higher precisions on predictions. The key point is adapting appropriate preprocessing and feature extraction methods to appropriate models.

6 Conclusion and Future Work

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

This comparative study of sentiment analysis methods revealed several key insights. Text preprocessing has different effects on different models. Stop word removal did not provide 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 to 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 using pre-trained word embeddings, task specific embeddings can be trained to feed LSTM and GRU layers. To achieve better comparison different pre-trained word embeddings like fastText, Word2Vec, and ELMo can be used along with the GloVe. It is also possible to adapt Bi-LSTM, Bi-GRU, variations of CNNs, and a 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