

Sentiment Analysis of Movie Reviews Using BERT

Gibson Nkhata

Department of Computer Science & Computer Engineering
University of Arkansas
Fayetteville, AR 72701, USA
Email: gnkhata@uark.edu

Usman Anjum, Justin Zhan

Department of Computer Science
University of Cincinnati
Cincinnati, OH 45221, USA
Email: anjumun@ucmail.uc.edu, zhanjt@ucmail.uc.edu

Abstract—Sentiment Analysis (SA) or opinion mining is analysis of emotions and opinions from any kind of text. SA helps in tracking people’s viewpoints, and it is an important factor when it comes to social media monitoring, product and brand recognition, customer satisfaction, customer loyalty, advertising and promotion’s success, and product acceptance. That is why SA is one of the active research areas in Natural Language Processing (NLP). SA is applied on data sourced from various media platforms to mine sentiment knowledge from them. Various approaches have been deployed in the literature to solve the problem. Most techniques devise complex and sophisticated frameworks in order to attain optimal accuracy. This work aims to fine-tune Bidirectional Encoder Representations from Transformers (BERT) with Bidirectional Long Short-Term Memory (BiLSTM) for movie reviews sentiment analysis and still provide better accuracy than the State-of-The-Art (SOTA) methods. The paper also shows how sentiment analysis can be applied, if someone wants to recommend a certain movie, for example, by computing overall polarity of its sentiments predicted by the model. That is, our proposed method serves as an upper-bound baseline in prediction of a predominant reaction to a movie. To compute overall polarity, a heuristic algorithm is applied to BERT-BiLSTM output vector. Our model can be extended to three-class, four-class, or any fine-grained classification, and apply overall polarity computation again. This is intended to be exploited in future work.

Index Terms—Sentiment analysis; movie reviews; BERT, bidirectional LSTM; overall polarity.

I. INTRODUCTION

Sentiment analysis aims to determine the polarity of emotions like happiness, sorrow, grief, hatred, anger and affection and opinions from text, reviews, and posts which are available in media platforms [1]. With the emergence of various social media platforms, vast amount of data are contained and various information, e.g., education, health, entertainment, etc, is shared in these online forums everyday across the globe. Therefore, there have been advances in many Natural Language Processing (NLP) tasks in conjunction with machine or deep learning techniques that automatically mine knowledge from the data sourced from these repositories [2]. As an NLP task, sentiment analysis helps in tracking people’s viewpoints. For example, it is a powerful marketing tool that enables product managers to understand customer emotions in their various marketing campaigns. It is an important factor when it comes to social media monitoring, product

and brand recognition, customer satisfaction, customer loyalty, advertising and promotion’s success, and product acceptance. Sentiment analysis is among the most popular and valuable tasks in the field of NLP [3].

Movie reviews is an important approach to gauge the performance of a particular movie. Whereas providing a numerical or stars rating to a movie quantitatively tells us about the success or failure of a movie, a collection of movie reviews is what gives us a deeper qualitative insight on different aspects of the movie. A textual movie review tells us about the strengths and weaknesses of the movie and deeper analysis of a movie review tells if the movie generally satisfies the reviewer.

Movie Reviews Sentiment Analysis is being worked on in this study because movie reviews have standard benchmark datasets, where salient and qualitative works have been published on. In fact, most of these reviews are crawled from the social media platforms [4].

Bidirectional Encoder Representations from Transformers (BERT) is a popular pre-trained language representation model and has proven to perform well on many NLP tasks like question answering, named entity recognition, and text classification [5]. BERT has been used in many works for sentiment analysis, like in [6]. However, regarding the capabilities of BERT, the performance is not satisfactory.

In this paper, BERT is fine-tuned for sentiment analysis on movie reviews and provide optimal accuracy that surpass accuracy of State-Of-The Art (SOTA) models. Our focus is on polarity classification on a 2-point scale. Polarity classification classifies a text as containing either a negative or positive sentiment.

BERT has proven to be satisfactory in many NLP downstream tasks. BERT has been used in information retrieval in [7] to build an efficient ranking model for industry use cases. The pre-trained language model was also successfully utilised in [8] for extractive summarization of text and used for question answering with satisfactory results in [9]. Yang et al. [10] efficiently applied the model in data augmentation resulting in optimal results.

Fine-tuning is a common technique for transfer learning. The target model copies all model designs with their parameters from the source model except the output layer and fine-tunes these parameters based on the target dataset.

The main benefit of fine-tuning is no need of training the entire model from scratch, and only the output layer of the target model needs to be trained. Hence, BERT is being fine-tuned in this work by coupling with Bidirectional Long Short-Term Memory (BiLSTM) and train the resulting model on movie reviews sentiment analysis benchmark datasets. BiLSTM processes input features bidirectionally [5], which helps in improving target model generalisation. Therefore, our fine-tuning approach is called BERT+BiLSTM-SA, where SA stands for Sentiment Analysis.

Finally, how results of sentiment analysis can be applied is shown. If someone wants to recommend a certain movie, for example, by computing overall polarity of its reviews sentiments predicted by the model. That is, the proposed method serves as an upper-bound baseline in prediction of the polarity of predominant reaction to a movie. To compute overall polarity, a heuristic algorithm adopted from [11] is applied to BERT-BiLSTM+SA output vector. The algorithm in their paper is applied on the output vector of three class classification on twitter dataset by LSTM, whereas our approach customises the algorithm for binary classification output vector from BERT+BiLSTM-SA.

This work is divided into two main components. First, fine-tuning BERT with BiLSTM and use the resulting model on binary sentiment polarity classification. Second, using the results of sentiment classification in computation of a predominant sentiment polarity.

1) *Our contributions:* Our contributions in this work are:

- Fine-tuning BERT by coupling with BiLSTM for polarity classification on well known benchmark datasets and achieve accuracy that beats SOTA models.
- Computing overall polarity of predicted reviews from BERT+LSTM-SA output vector.
- Comparing our experimental outcomes with results obtained from other studies, including SOTA models, on benchmark datasets.

This paper is organised as follows. Section II describes related work, Section III describes the methodology, Section IV discusses experiments and results, and last, Section V gives conclusion and talks about future work. The code for this project is available [12] to enable wider adoption.

II. RELATED WORK

A lot of work has been conducted in literature on movie reviews sentiment analysis. Starting with traditional machine learning, a step-by-step lexicon-based sentiment analysis using the R open-source software is presented in [13]. In [1], the authors implemented and compared traditional machine learning techniques like Naive Bayes (NB), K-Nearest Neighbours (KNN), and Random Forests (RF) for sentiment analysis. Their results showed that Naive Bayes was the best classifier on the task. An ensemble generative approach for various machine learning approaches on sentiment analysis is used in [3]. KNN with the help of information gain technique was also used in [14] on the task. In [15], the authors proposed training document embeddings using cosine similarity, feature

combination, and NB. KNN outperforms all other models in these works.

Deep learning approaches have also been implemented in movie reviews sentiment analysis. Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN) architectures performances were explored for semantic analysis of movie reviews in [16]. RNNs give satisfactory results, but they suffer from the problem of vanishing or exploding gradients when used with long sentences. Nonetheless, CNNs provide non-optimal accuracy on text classification. Coupled Oscillatory RNN (CoRNN), which is a time-discretization of a system of second-order ordinary differential equations, was proposed in [17] to mitigate the exploding and vanishing gradient problem, though the performance was still not convincing. Bodapati et al. [18] used LSTM on movie reviews sentiment analysis by investigating the impact of different hyper parameters like dropout, number of layers, and activation functions. Additionally, BiLSTM network for the task of text classification has also been applied via mixed objective function in [19]. BiLSTM achieved better results but at the expense of a very sophisticated architecture.

BERT has also been previously applied to sentiment analysis. BERT was used on SST-2 movie reviews benchmark for sentiment analysis in [6]. In [20], the authors used BERT for stock market sentiment analysis. BERT was also applied on target-dependent sentiment classification in [21]. However, there is still room for improvement considering their results.

Therefore, in this work, BERT is fine-tuned by coupling with BiLSTM for sentiment analysis on a 2-point scale. Afterwards, an application of sentiment analysis is shown by computing overall polarity of movie reviews, which can also be utilised in recommending a movie.

III. METHODOLOGY

Different techniques used in our work starting with description of sentiment analysis and BERT are covered in this section. Afterwards, the section explains how BERT is fine-tuned with BiLSTM, elucidates how classification is applied in our work, describes overall polarity computation, and talks about the overview of the whole work.

A. Sentiment analysis

Sentiment analysis is a sub-domain of opinion mining, which aims at the extraction of emotions and opinions of people towards a particular topic from a structured, semi-structured, or unstructured textual data [22]. In our context, the primary objective is to classify a review as carrying either a positive or negative sentiment on a 2-point scale.

B. BERT

BERT was introduced by researchers from Google [5] and focuses on pre-training deep bidirectional representations from unlabeled text by jointly conditioning on both left and right contexts in all layers of the model. As a result, BERT can be fine-tuned with an extra component on a downstream task like question answering and sentiment analysis.

There are two primary models of BERT, namely, BERT_{BASE} and BERT_{LARGE}. BERT_{BASE}, which has 12 layers, 768 hidden states, 12 attention heads, and 110M parameters is used in this work, whereas BERT_{LARGE} has almost 2 times of each of these specifications. Actually, the uncased version of BERT_{BASE} known as *bert-base-uncased*, which accepts tokens as lowercase, is adapted in this work. Because of multiple attention heads, BERT processes a sequence of input tokens in parallel, thereby improving the model generalization on the input sequence.

BERT has its own format for the input tokens. The first token of every sequence is denoted as [CLS]. The token corresponds to the last hidden layer, aggregates all the information in the input sequence, and is used for classification tasks. Sentences are packed into a single input sequence and differentiated in two ways: using a special token [SEP] to separate them and adding a learned embedding to every token identifying a sentence where it belongs to.

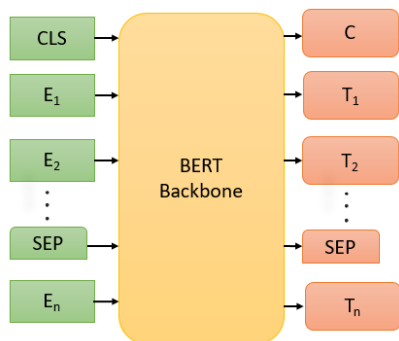


Fig. 1. Simplified diagram of BERT

Figure 1 shows a simplified diagram of BERT. E_n is an input representation of a single token constructed by summing the corresponding token, segment, and position embeddings; BERT Backbone represents main processing performed by BERT; T_n is a hidden state corresponding to token E_n ; and C is a hidden state corresponding to aggregate token CLS .

C. Fine-tuning BERT with BiLSTM

Since BERT is pretrained, there is no need of training the entire model from scratch. Hence, information is just needed to be transferred from BERT to the fine-tuning component and train the model for sentiment analysis. This saves training time of the resulting model.

Fine-tuning in this work is conducted as follows. After data preprocessing, two input layers to BERT are built, where names of the layers need to match the input values. These input values are attention masks and input ids, as shown in Figure 2. In other words, attention masks and input ids are input embeddings to the model.

The input embeddings are propagated through BERT afterwards. Dimensionality of the embeddings depends on the input sequence length, batch-size and number of units in a layer. BiLSTM is then concatenated at the very end of

BERT, and it includes a dense layer. Therefore, BiLSTM receives information from BERT and feeds it into its dense layer, which then predicts respective sentiments for the input features. BERT and BiLSTM shared same hyperparameters, and all hyperparameters are specified in Section IV under experimental settings.

The fine-tuning part of our model is illustrated in Figure 2. In the figure, *input features* are tokens in a review text, *input ids* identify a sentence that a token belongs to, and *attention masks* are binary tensors indicating the position of the padded indices to a particular sequence so that the model does not attend to them. For the attention mask, 1 is used to indicate a value that should be attended to, while 0 is used to indicate a padded value. Padding helps in making sequences have same length when sentences have variable lengths, which is common in NLP. Therefore, padded information is not part of the input and should rarely be used in model generalization.

The output from BERT has the same dimension as the input to BiLSTM. E_i and T_i mean similar items as E_n and T_n , respectively, in Figure 1. BiLSTM has only one hidden component. Finally, there is a fully connected layer (dense layer) at the end, which has output dimension of batch-size by 1, since binary classification is being worked on here. The dense layer predicts a sentiment polarity.

Weights from first layers of BERT are frozen so that our focus dwells on the last layers close to the fine-tuning component. These layers contain trainable weights, which are updated to minimize the loss during training of the model on the downstream task of sentiment analysis.

D. Classification

In this work, BERT is fine-tuned on polarity classification or binary classification of sentiment analysis. Hence, classification in this context is defined as follows. Given a movie review R , classify it as carrying either a positive sentiment or a negative sentiment.

E. Overall polarity

Overall polarity is defined as follows. Given an output vector from BERT+BiLSTM-SA containing sentiment labels of N reviews, find the dominating sentiment polarity in the vector. To compute the overall polarity of reviews, the output vector of BERT+BiLSTM-SA is fed into the heuristic algorithm. First, number of labels for each class is counted in the output vector. Then, the computation is conducted as shown in Algorithm 2. While Arasteh et al. [11] used Algorithm 1 to compute overall polarity from the results of three-class classification on twitter replies by LSTM for twitter sentiment analysis, this work adapts the algorithm to compute overall polarity from output vector of binary classification by BERT+BiLSTM-SA.

Algorithm 1 shows how overall polarity is computed from output vector of three-class classification. The overall polarity is first considered to be neutral if the proportion of neutral samples is at least higher than an empirically set threshold, which was set to be 85% of the total predictions. The reason being that most samples are generally expected to be neutral,

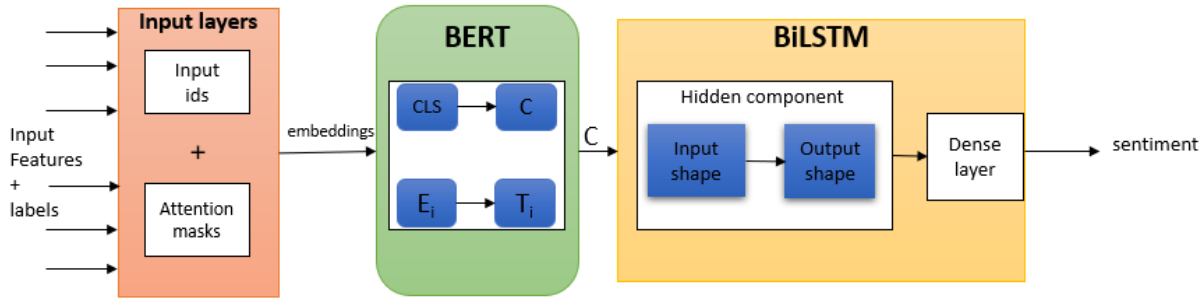


Fig. 2. Fine-tuning of the model

Algorithm 1: Computing overall polarity from three-class classification output vector as done in [11].

Result: Dominating sentiment for all reviews.

```

1 if #total neutral reviews > 85% of the total reviews
  then
2   | overall polarity ← neutral;
3 else
4   | if #total positive reviews > 1.5 × # of total
      | negative reviews then
5     | overall polarity ← positive;
6   | else if #total negative reviews > 1.5 × # of total
      | positive reviews then
7     | overall polarity ← negative;
8   | else
9     | overall polarity ← neutral;

```

for not every text can be expected to carry a positive or a negative polarity. Then, negative and positive sentiments are considered in the output vector. Again, there is usually no exclusively positive or negative text sample, that is why a positive overall sentiment is assigned if there is at least 1.5 times as many negative reviews as positive reviews, and vice versa. Meaning that the size of the dominating class must be at least higher for all the reviews to carry its polarity. Lastly, a neutral sentiment is given when the total numbers of positive and negative reviews are close to each other, implying nonexistence of dominance between the two sentiments in the reviews. All constant values in the algorithm were set depending on the various empirical observations in the experiments.

Algorithm 2 is derived from Algorithm 1 by us. Depending on various empirical observations in our experiments, a different threshold coefficient is used when multiplying. Additionally, only proportions of two sentiments are being compared here. Although there is not a neutral sentiment polarity in the output vector of our model, it is introduced for the overall polarity computation if the output of the first two conditions in Algorithm 2 is false, implying a tie between positive and negative reviews.

A naive approach to computing the overall polarity would be just counting the number of labels for each class in

BERT+BiLSTM-SA output vector and assign the overall polarity depending on majority class. However, the overall polarity computed from this approach cannot represent a good dominating majority class. For example, assume there is 102 predictions of movie reviews for one movie, and the output vector has 50 positive reviews and 52 negative reviews, the overall polarity will be negative. However, if this is applied in recommending a movie, for example, a difference of 2 is not optimal to determine the dominating polarity of all the movie reviews and conclude that the movie is not interesting. Additionally, the level of positiveness or negativity is different for every review in the datasets. As a result, the formulations in Algorithm 2 are used so that either class in the output vector must contain a bigger proportion to assign its label as the output, otherwise the overall polarity becomes neutral.

The overall polarity computation technique can be used to recommend a movie to a person disregarding subjectivity factors. First, all the reviews pertaining to the movie in question must be gathered, followed by using BERT+BiLSTM-SA to predict the associated sentiments polarities for the reviews. Then, the algorithm can be used to compute the dominating sentiment polarity to see if most people are in favor of the movie or not. Last, the movie can be recommended if the overall polarity is positive.

Algorithm 2: Computing overall polarity from binary classification output vector

Result: Dominating sentiment for all reviews.

```

1 if #total positive reviews > 1.2 × # of total negative
  reviews then
2   | overall polarity ← positive
3 else if #total negative reviews > 1.2 × # of total
  positive reviews then
4   | overall polarity ← negative
5 else
6   | overall polarity ← neutral

```

F. Overview of our work

Figure 3 portrays the overview of our work. In a nutshell, the work starts with preprocessing raw text data into features

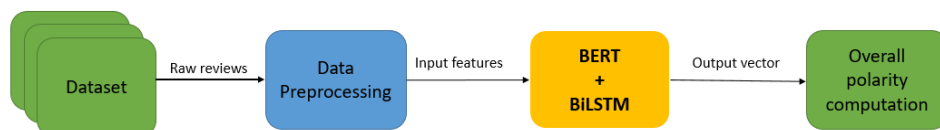


Fig. 3. Overview of our work

that can be understood by BERT and feed the features into BERT (BERT_{BASE} is the same as BERT in the diagram) and BiLSTM through the fine-tuning layer, which specifies the hyperparameters that BERT+BiLSTM should use. Lastly, an output vector from BERT+BiLSTM predictions is used to compute overall polarity.

IV. EXPERIMENTS

This section starts with an explanation of datasets that are used in the experiments followed by data preprocessing. Afterwards, it describes experimental settings, explains evaluation metrics used in experiments, and discusses experimental results.

A. Datasets

Datasets used in the experiments consist of reviews annotated for sentiment analysis on a 2-point scale. Following is the description of the datasets used and how some of them were changed to suit experimental settings. Table I shows statistics of the datasets used in the experiments.

TABLE I
STATISTICS OF THE DATASETS DIVIDED INTO TRAINING AND TEST SETS

Dataset	Train samples		Test samples	
	POSITIVE	NEGATIVE	POSITIVE	NEGATIVE
IMDb	12500	12500	12500	12500
MR	4264	4265	1067	1066
SST-2	4300	4244	886	1116
Amazon	239660	37056	59949	9231

1) *IMDb movie reviews*: Introduced in [23], IMDb movie reviews dataset is a binary sentiment analysis dataset consisting of 50,000 reviews from the Internet Movie Database (IMDb), where the name comes from. It comprises equal number of negative and positive reviews.

2) *SST-2*: The SST (Stanford Sentiment Treebank) is a corpus with fully labeled parse trees that allows for a complete analysis of the compositional effects of sentiment in language. The corpus is based on the dataset introduced in [2], and it consists of 11,855 single sentences extracted from movie reviews. It was parsed with the Stanford parser and includes a total of 215,154 unique phrases from those parse trees, each annotated by three human judges. In our work, SST-2 version of the dataset is used. SST-2 is designed for binary classification and consists of 11855 movie reviews.

3) *MR Movie Reviews*: MR Movie Reviews dataset comprises collections of movie review documents labeled with respect to their overall sentiment polarity, positive or negative, or subjective rating, for example two and a half stars, and

sentences labeled with respect to their subjectivity status, subjective or objective, or polarity. In this paper, the version introduced in [24], which consists of 5331 positive and 5331 negative processed reviews is used.

4) *Amazon Product Data dataset*: This dataset contains product reviews and metadata from Amazon, including 142.8 million reviews spanning May, 1996 through July, 2014. The dataset includes reviews, product metadata, and links. It was introduced in [4] for sentiment analysis using product review data, and in [25] to build a recommender system in collaborative filtering setting on amazon products. In our work, only video reviews are focused on. There is a total of 345896 video reviews samples. The dataset originally contained labels with scores from 1 to 5 corresponding to polarity strength variation from negative to positive. The dataset was then prepared for binary classification by replacing 1 and 2 scores with a negative label and 4 and 5 scores with a positive label, and score 3, which represents a neutral class, was discarded as in SST-2 [2].

B. Data preprocessing

The needed data preprocessing steps require to transform the input data into a format that BERT can understand. This involves carrying out two primary data preprocessing steps.

First, creating input examples using the constructor provided in the BERT library. The constructor accepts three main parameters, which are *text_a*, *text_b*, and *label*. *text_a* is the text that the model must classify, which in this case, is the collection of movie reviews without their associated labels. *text_b* is used if a model is being trained to understand the relationship between sentences, for example sentence translation and question answering. The previous scenario hardly applies in our work, so *text_b* is just left blank. *label* has labels of input features. In our case, *label* implies sentiment polarity of every movie review, which can be negative or positive. Refer to BERT original paper [5] for more details about this step and even the next step.

Last, the following preprocessing steps are conducted.

- Lowercase text, since the lowercase version of BERT_{BASE} is being used.
- Tokenize all sentences in the reviews. For example, "this is a very fantastic movie" to "this", "is", "a", "very", "fantastic", "movie".
- Break words into word pieces. That is "interesting" to "interest" and "##ing".
- Map words to indexes using a vocab file that is provided by BERT.

- Adding special tokens: [CLS] and [SEP], which are used for aggregating information of the entire review through the model and separating sentences respectively.
- Append index and segment tokens to each input to track a sentence which a specific token belongs to.

The output of the tokenizer after these steps are *input ids* and *attention masks*. These are then taken as inputs to our model in addition to the reviews labels.

C. Experimental settings

Many simulations were carried on the datasets to find optimal hyperparameters for the model. As a result, optimal results from the experiments were obtained by the following settings. 256 input sequence length (K), adam optimizer, $3e-5$ learning rate, $1e-08$ epsilon, and sparse categorical cross entropy loss. The model was trained for 10 epochs and repeated steps for each batch. These hyperparameters were cordially fine-tuned regarding both BERT and BiLSTM, and overfitting was noticed when increasing the number of epochs for the model.

D. Evaluation metrics

Accuracy was used to evaluate the performance of our model and compare it with other models. Accuracy metric is adopted because it is greatly applied in most works [6], [26]–[29]. Therefore, this adoption makes our work to be consistent with other works that are being compared against. Accuracy is defined as follows:

$$accuracy = \frac{\text{number of correct predictions}}{\text{total number of predictions}} \times 100 \quad (1)$$

E. Results

Table II presents accuracy comparisons between our model, BERT+BiLSTM-SA, and other models on all datasets. BERT+BiLSTM-SA outperforms other models on all the datasets, thereby achieving new SOTA accuracy on these benchmark datasets. The best accuracy of 98.76% accuracy is obtained on Amazon dataset.

TABLE II
ACCURACY (%) COMPARISONS OF MODELS ON BENCHMARK DATASETS FOR BINARY CLASSIFICATION

Model name	Dataset			
	IMDb-2	MR	SST-2	amazon-2
RNN-Capsule [30]	84.12	83.80	82.77	82.68
coRNN [31]	87.4	87.11	88.97	89.32
TL-CNN [31]	87.70	81.5	87.70	88.12
Modified LMU [29]	93.20	93.15	93.10	93.67
DualCL [28]	-	94.31	94.91	94.98
L Mixed [32]	95.68	95.72	-	95.81
EFL [26]	96.10	96.90	96.90	96.91
NB-weighted-BON [15] +dv-cosine	97.40	-	96.55	97.55
SMART-RoBERTa [27] Large	96.34	97.5	96.61	-
Ours	97.67	97.88	97.62	98.76

Table III shows results of ablation studies to see the impact of each component in the model. It can be seen that both BERT and BiLSTM separately give lower accuracy on the predictions compared against BERT+BiLSTM-SA. Therefore, the coupling of the two tools enhances model generalization.

TABLE III
RESULTS OF ABLATION STUDY

Model name	Dataset			
	IMDb-2	MR	SST-2	amazon-2
BiLSTM	90.42	90.5	91.12	92.18
BERT	93.81	94.29	93.55.97	94.78
BERT+BiLSTM-SA	97.67	97.88	97.62	98.76

The discussion of results is finished by talking about the overall polarity computation on all datasets by BERT+BiLSTM-SA. Table IV presents the overall polarity computed from all the datasets. **Original overall polarity** is known before input embeddings are fed into BERT+BiLSTM-SA for prediction, while **Computed overall polarity** is computed and known after BERT+BiLSTM-SA has made predictions on the reviews. The table shows that the **Computed overall polarity** is the same as the **Original overall polarity** for all the datasets. The **Original overall polarity** is calculated by counting the number of samples of each label in the input and use the result in the heuristic algorithm. That is, using original proportion of each class in the input before the model has made predictions on the reviews. The output was then used to verify the **Computed overall polarity**. The **Computed overall polarity** is computed similarly, but predictions are considered. Therefore, without loss of generality, there is confidence that the model predicts the expected sentiment on a given review and the heuristic algorithm computes accurate overall polarity from the model, regarding each dataset.

TABLE IV
OVERALL POLARITY COMPUTATION ON ALL DATASETS

Dataset	Original overall polarity	Computed overall polarity
IMDb	Neutral	Neutral
MR reviews	Neutral	Neutral
SST-2	Neutral	Neutral
Amazon	Positive	Positive

V. CONCLUSION

Sentiment analysis is an active research domain in NLP. In this work, the existing domain knowledge of sentiment analysis is extended by providing another effective way of fine-tuning BERT to improve accuracy measure on movie reviews sentiment analysis and show how to compute an overall polarity of a collection of movie reviews sentiments predicted by a model, BERT+BiLSTM-SA, for example. To fine-tune BERT, the technique of transfer learning was employed by coupling BERT with BiLSTM. BiLSTM, which had a dense layer, acted as a classifier on BERT final hidden states. The model was used for polarity classification and was experimented on

IMDb, MR, SST-2, and Amazon datasets. It was also shown that sentiment analysis can be applied, if someone wants to recommend a certain movie, for example, by computing overall polarity of its sentiments predicted by the model. That is, the proposed method serves as an upper-bound baseline in prediction of a predominant reaction to a movie. Ablation studies also show that BERT and BiLSTM separately provide non-optimal accuracy compared against BERT+BiLSTM-SA, implying coupling of the two tools is stronger for the model generalization.

To compute overall polarity, a heuristic algorithm is applied to BERT-BiLSTM+SA predictions. For all the datasets, it has been demonstrated that the original overall polarity is the same as the computed overall polarity. To the best of our knowledge, this is the first work to couple BERT with BiLSTM for sentiment classification task and use the model output vector to compute overall sentiment polarity. Our model is robust whereby it can be extended to three-class, four-class, or any fine-grained classification. This is intended to be explored in future work to prove the robustness of the model.

Future work will additionally dwell on how to effectively apply accuracy improvement techniques to transformed BERT features despite loss of semantic information in them, exploring other pre-trained language models, and how different components of a sentence contribute to its sentiment prediction since this is information that is not generally explored by current works.

REFERENCES

- [1] P. Baid, A. Gupta, and N. Chaplot, "Sentiment analysis of movie reviews using machine learning techniques," *International Journal of Computer Applications*, vol. 179, no. 7, pp. 45–49, 2017.
- [2] R. Socher, A. Perelygin, J. Wu, J. Chuang, and C. D. Manning, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the 2013 conference on empirical methods in natural language processing*, 2013, pp. 1631–1642.
- [3] G. Mesnil, T. Mikolov, M. Ranzato, and Y. Bengio, "Ensemble of generative and discriminative techniques for sentiment analysis of movie reviews," *arXiv preprint arXiv:1412.5335*, 2014.
- [4] X. Fang and J. Zhan, "Sentiment analysis using product review data," *Journal of Big Data*, vol. 2, no. 1, pp. 1–14, 2015.
- [5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.
- [6] M. Munikar, S. Shakya, and A. Shrestha, "Fine-grained sentiment classification using bert," in *2019 Artificial Intelligence for Transforming Business and Society (AITB)*, vol. 1. IEEE, 2019, pp. 1–5.
- [7] W. Guo, X. Liu, S. Wang, H. Gao, and A. Sankar, "Detext: A deep text ranking framework with bert," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020, pp. 2509–2516.
- [8] Y. Liu, "Fine-tune bert for extractive summarization," *arXiv preprint arXiv:1903.10318*, 2019.
- [9] Y. He, Z. Zhu, Y. Zhang, Q. Chen, and J. Caverlee, "Infusing disease knowledge into bert for health question answering, medical inference and disease name recognition," *arXiv preprint arXiv:2010.03746*, 2020.
- [10] W. Yang, Y. Xie, L. Tan, K. Xiong, and M. Li, "Data augmentation for bert fine-tuning in open-domain question answering," *arXiv preprint arXiv:1904.06652*, 2019.
- [11] S. T. Arasteh, M. Monajem, V. Christlein, P. Heinrich, and A. Nicolaou, "How will your tweet be received? predicting the sentiment polarity of tweet replies," in *2021 IEEE 15th International Conference on Semantic Computing (ICSC)*. IEEE, 2021, pp. 370–373.
- [12] <https://github.com/gnkhata1/Finetuning-BERT-on-Movie-Reviews-Sentiment-Analysis>, 2022, [Online; accessed March-15-2022].
- [13] M. Anandarajan, C. Hill, and T. Nolan, "Sentiment analysis of movie reviews using r," in *Practical Text Analytics*. Springer, 2019, pp. 193–220.
- [14] N. O. F. Daeli and A. Adiwijaya, "Sentiment analysis on movie reviews using information gain and k-nearest neighbor," *Journal of Data Science and Its Applications*, vol. 3, no. 1, pp. 1–7, 2020.
- [15] T. Thongtan and T. Phientrakul, "Sentiment classification using document embeddings trained with cosine similarity," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, 2019, pp. 407–414.
- [16] H. Shirani-Mehr, "Applications of deep learning to sentiment analysis of movie reviews," in *Technical report*. Stanford University, 2014.
- [17] T. K. Rusch and S. Mishra, "Coupled oscillatory recurrent neural network (cornn): An accurate and (gradient) stable architecture for learning long time dependencies," *arXiv preprint arXiv:2010.00951*, 2020.
- [18] J. D. Bodapati, N. Veeranjanyulu, and S. Shaik, "Sentiment analysis from movie reviews using lstms," *Ingenierie des Systemes d'Information*, vol. 24, no. 1, pp. 1–5, 2019.
- [19] D. Singh Sachan, M. Zaheer, and R. Salakhutdinov, "Revisiting lstm networks for semi-supervised text classification via mixed objective function," *arXiv e-prints*, pp. arXiv–2009, 2020.
- [20] M. G. Sousa, K. Sakiyama, L. de Souza Rodrigues, P. H. Moraes, and E. R. Fernandes, "Bert for stock market sentiment analysis," in *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, 2019, pp. 1597–1601.
- [21] Z. Gao, A. Feng, X. Song, and X. Wu, "Target-dependent sentiment classification with bert," *Ieee Access*, vol. 7, pp. 154 290–154 299, 2019.
- [22] T. Gadekallu, A. Soni, D. Sarkar, and L. Kuruva, "Application of sentiment analysis in movie reviews," in *Sentiment Analysis and Knowledge Discovery in Contemporary Business*. IGI global, 2019, pp. 77–90.
- [23] A. Maas, R. E. Daly, P. T. Pham, D. Huang, and A. Y. Ng, "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.
- [24] B. Pang and L. Lee, "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales," *arXiv preprint cs/0506075*, 2005.
- [25] R. He and J. McAuley, "Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering," in *proceedings of the 25th international conference on world wide web*, 2016, pp. 507–517.
- [26] S. Wang, H. Fang, M. Khabsa, H. Mao, and H. Ma, "Entailment as few-shot learner," *arXiv preprint arXiv:2104.14690*, 2021.
- [27] H. Jiang, P. He, W. Chen, X. Liu, and J. Gao, "Smart: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization," *arXiv preprint arXiv:1911.03437*, 2019.
- [28] Q. Chen, R. Zhang, Y. Zheng, and Y. Mao, "Dual contrastive learning: Text classification via label-aware data augmentation," *arXiv preprint arXiv:2201.08702*, 2022.
- [29] N. R. Chilukuri and C. EliaSmith, "Parallelizing legendre memory unit training," in *International Conference on Machine Learning*. PMLR, 2021, pp. 1898–1907.
- [30] Y. Wang, A. Sun, J. Han, Y. Liu, and X. Zhu, "Sentiment analysis by capsules," in *Proceedings of the 2018 world wide web conference*, 2018, pp. 1165–1174.
- [31] T. Semwal, P. Yenigalla, G. Mathur, and S. B. Nair, "A practitioners' guide to transfer learning for text classification using convolutional neural networks," in *Proceedings of the 2018 SIAM international conference on data mining*. SIAM, 2018, pp. 513–521.
- [32] D. S. Sachan, M. Zaheer, and R. Salakhutdinov, "Revisiting lstm networks for semi-supervised text classification via mixed objective function," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 6940–6948.