

Fine-tuning BERT with Bidirectional LSTM for Fine-grained Movie Reviews Sentiment Analysis

Gibson Nkhata, Susan Gauch

Department of Computer Science & Computer Engineering
University of Arkansas
Fayetteville, AR 72701, USA
Email: gnkhata@uark.edu, sgauch@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) is instrumental in understanding people’s viewpoints, facilitating social media monitoring, recognizing products and brands, and gauging customer satisfaction. Consequently, SA has evolved into an active research domain within Natural Language Processing (NLP). Many approaches outlined in the literature devise intricate frameworks aimed at achieving high accuracy, focusing exclusively on either binary sentiment classification or fine-grained sentiment classification. In this paper, our objective is to fine-tune the pre-trained BERT model with Bidirectional LSTM (BiLSTM) to enhance both binary and fine-grained SA specifically for movie reviews. Our approach involves conducting sentiment classification for each review, followed by computing the overall sentiment polarity across all reviews. We present our findings on binary classification as well as fine-grained classification utilizing benchmark datasets. Additionally, we implement and assess two accuracy improvement techniques, Synthetic Minority Oversampling Technique (SMOTE) and NLP Augmenter (NLPAUG), to bolster the model’s generalization in fine-grained sentiment classification. Finally, a heuristic algorithm is employed to calculate the overall polarity of predicted reviews from the BERT+BiLSTM output vector. Our approach performs comparably with state-of-the-art (SOTA) techniques in both classifications. For instance, in binary classification, we achieve 97.67% accuracy, surpassing the leading SOTA model, NB-weighted-BON+dv-cosine, by 0.27% on the renowned IMDb dataset. Conversely, for five-class classification on SST-5, while the top SOTA model, RoBERTa+large+Self-explaining, attains 55.5% accuracy, our model achieves 59.48% accuracy, surpassing the BERT-large baseline by 3.6%.

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

I. INTRODUCTION

This paper builds upon our prior research [1] into sentiment analysis (SA) of movie reviews using Bidirectional Encoder Representations from Transformers (BERT) [2] and computing overall polarity within the scope of binary sentiment classification. Still, SA aims to discern the polarity of emotions (e.g., happiness, sorrow, grief, hatred, anger, and affection) and opinions derived from text, reviews, and posts across various media platforms [3]. It plays a crucial role in gauging public sentiment, serving as a potent marketing tool for comprehending customer emotions across diverse marketing campaigns. SA significantly contributes to social media monitoring, brand

recognition, customer satisfaction, loyalty, advertising effectiveness, and product acceptance. Consequently, SA stands as one of the most sought-after and impactful tasks within Natural Language Processing (NLP) [4]. It encompasses polarity classification, involving binary categorization, and fine-grained classification, encompassing multi-scale sentiment distribution.

Movie reviews serve as a crucial means to evaluate the performance of a film. While assigning a numerical or star rating offers a quantitative assessment of a movie’s success or failure, a collection of movie reviews offers a qualitative exploration of various aspects within the film. Textual movie reviews provide insights into strengths and weaknesses of a movie, allowing for a deeper analysis that gauges the overall satisfaction of the reviewer. This study focuses on SA of movie reviews due to the availability of standardized benchmark datasets and the existence of significant qualitative works within this domain, as highlighted in publications such as [5].

BERT stands as a renowned pre-trained language representation model, demonstrating strong performance across various NLP tasks such as named entity recognition, question answering, and text classification [2]. Its versatility extends to information retrieval, where it has been leveraged to construct efficient ranking models for industry-specific applications [6]. Additionally, its adaptability is evident in applications like extractive summarization of text, as successfully demonstrated in [7], and in question answering tasks, where it yielded satisfactory results as seen in [8]. The model’s efficacy was further highlighted in data augmentation strategies, leading to superior outcomes, as exemplified in [9]. While BERT has found primary use in SA [10], its accuracy on certain datasets remains a challenge.

Ambiguity in NLP, particularly SA, arises due to the complexity of language. Words and phrases often carry multiple meanings or interpretations based on context, making it challenging to accurately discern sentiment. This challenge is evident in instances where words might have different connotations depending on their context within a sentence or across diverse texts. BERT addresses this issue by leveraging its bidirectional context understanding [2]. Unlike earlier models that processed language in one direction, BERT comprehends words based on both preceding and succeeding

Extend version of Sentiment Analysis of Movie Reviews Using BERT, presented at The Fifteenth International Conference on Information, Process, and Knowledge Management, eKNOW23, Venice, Italy, 2023.

words, allowing it to capture a more nuanced understanding of context. Therefore, BERT can better grasp the intricate meanings and resolve ambiguities present in natural language, thereby improving SA accuracy.

Bidirectional Long Short-Term Memory (BiLSTM) networks are also very popular for text classification in NLP [11]. BiLSTM is beneficial in SA due to its ability to capture contextual information from both past and future inputs in long sequences [12]. In SA, context is crucial, words derive their meaning not just from preceding words but also from the subsequent ones. BiLSTMs excel in capturing this bidirectional context by processing sequences in two directions simultaneously: forward (from the beginning to the end of a sequence) and backward (from the end to the beginning). This capability enables BiLSTMs to model more nuanced and complex dependencies in text, leading to improved SA performance compared to unidirectional models.

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 [2]. The main benefit of fine-tuning is no need to train the entire model from scratch, reducing the training time of the target model. Hence, we are fine-tuning BERT by coupling BiLSTM and training the model on movie reviews SA benchmarks. BERT, with its attention mechanisms and bidirectional context understanding, captures rich contextual information [2]. Combining it with a BiLSTM enhances this capability further. The ability of the BiLSTM to retain long-range dependencies complements contextual understanding of BERT, leading to a more nuanced comprehension of sentiment in text. The efficacy of this amalgamation is corroborated by conducting an ablation study in our experiments.

In our approach, we derive an overall polarity from the output vector generated by BERT+BiLSTM, employing a heuristic algorithm adapted from [13]. This algorithm is tailored uniquely in our study, addressing the specifics of the output vectors derived from binary, three-class, four-class, or five-class sentiment classifications. As a result, our work implements four distinct iterations of the algorithm, each corresponding to one of the four different sentiment classification tasks undertaken in this research.

Previous studies have predominantly focused on either binary sentiment classification or fine-grained SA, rarely combining both aspects. This paper addresses this gap by presenting an approach that fine-tunes BERT specifically for SA on movie reviews. Our objective is to conduct a comparative analysis encompassing both binary and fine-grained sentiment classifications. Through the integration of BERT and BiLSTM architecture, our fine-tuning methodology caters to both binary and fine-grained sentiment classification tasks. Notably, our approach surpasses state-of-the-art (SOTA) models in accuracy using our best-performing method. To address the challenge of class imbalance in fine-grained classification, we implement oversampling and data augmentation techniques on the respective datasets before feeding the data into the model classifier.

The main contributions in this work are as follows:

- Fine-tune BERT by coupling it with BiLSTM for both binary and fine-grained sentiment classification on well-known benchmark datasets and achieve accuracy that surpasses SOTA models.
- Refine techniques to enhance the model's accuracy in fine-grained sentiment classification.
- Compute the overall sentiment polarity of predicted reviews based on the output vector from BERT+BiLSTM.
- Compare and evaluate our experimental outcomes against the results obtained from other studies, including those from SOTA models, using benchmark datasets.

This paper is organized as follows: Section II describes the related work, Section III details the methodology, Section IV discusses the experiments and results, and finally, Section V presents the conclusion and discusses future work.

II. RELATED WORK

This section covers relevant literature concerning SA, the intersection of deep learning and SA, particularly focusing on movie reviews SA, and the role of BERT in SA.

A. SA

SA within NLP remains an active area of research. [14] introduced a step-by-step lexicon-based SA method using the R open-source software. The study conducted polarity classification on 1,000 movie reviews from the IMDb dataset, achieving an accuracy of 81.30% by evaluating built-in lexicons.

In a contrasting approach, [3] explored traditional machine learning techniques, i.e., Naive Bayes (NB), K-Nearest Neighbours (KNN), and Random Forests, for SA on IMDb movie reviews. Their findings highlighted NB as the top performer, achieving an accuracy of 81.45%.

A different route was taken in [4], deploying an ensemble generative technique across multiple machine learning approaches for movie reviews SA, achieving an accuracy of 90.57%. Conversely, [15] focused on the Cornell movie review dataset, solely utilizing KNN with the information gain technique, yielding an accuracy of 90.8%. These studies underscore the effectiveness of KNN in traditional machine-learning methods for movie reviews SA.

A novel approach by training document embeddings using cosine similarity and feature combination with NB weighted bag of n -grams was proposed in [16]. Their comparison between training document embeddings using cosine similarity and dot product favored cosine similarity, achieving an accuracy of 91.42% on the IMDb dataset. In addition, [17] applied mixed objective function for binary classification on SA on IMDb benchmark. Their approach reported an error rate of 9.95%. The aforementioned models targeted polarity or binary sentiment classification only. Nevertheless, both binary classification and fine-grained classifications on SA were implemented in the following two studies. [18] used transfer learning, while [19], utilised entailment and few-shot learning. Both studies used IMDb, SST-2, and MR benchmarks

for binary classification, and Yelp and SST-5 for fine-grained classification. Average accuracy of 87.57% is reported in [18] on binary datasets, while [19] has reported 88.16%. For fine-grained classification, they reported average accuracy of 52.65% and 54.65%, respectively.

In our work, we adopt both polarity and fine-grained classifications from [19] but use deep learning techniques and the BERT pre-trained language model. We also adopt transfer learning from [18].

B. Deep learning

This section explores the realm of deep learning applied to both the general SA task and specifically to movie reviews SA.

1) *Deep Learning on SA*: Deep learning stands as a SOTA technique for many NLP tasks, including SA. In a study by [20], Character-level Convolutional Neural Networks (CCNNs) were explored for text classification using Yelp and Amazon benchmarks. These CCNNs were compared against bag-of-words, n -grams, their TF-IDF variants, word-based CNNs, and Recurrent Neural Networks (RNNs). Reported error rates were 7.82% and 6.93% on the Yelp and Amazon benchmarks, respectively. Notably, on Yelp, the n -grams model outperformed the CCNNs with a 6.36% error rate. The study highlighted several influencing factors on model performance, including dataset size, text curation, and the alphabet used, specifically distinguishing between uppercase and lowercase letters.

In a separate study, [21] introduced a unique approach using CNNs with meta-networks to learn context-sensitive convolutional filters for text processing. Applying this approach on Yelp, they achieved a lower error rate of 4.89% compared to the former CCNNs approach. However, deeper networks pose increased computational complexity, impacting practical applications. Addressing this, [22] proposed shallow word-based Deep Pyramid CNNs (DPCNN) for text categorization. They studied deepening word-level CNNs to capture comprehensive text representations without significantly increasing computational costs. Evaluating on Yelp and Amazon datasets, their method achieved error rates of 7.88% and 7.92%, respectively.

2) *Deep Learning on Movie Reviews SA*: To commence, [23] explored the performances of RNNs and CNNs architectures for SA of movie reviews. They utilized pre-defined 300-dimensional vectors from word2vec [24] instead of training word vectors along with other parameters using samples. The study indicated that CNNs outperformed RNNs, achieving the best accuracy of 46.4% on the SST dataset. It was concluded that basic RNNs were inefficient in capturing the structural and contextual properties of sentences. Basic RNNs encounter issues such as vanishing or exploding gradients, leading to model underfitting and overfitting when networks become very deep. Addressing this challenge, [25] proposed Coupled Oscillatory RNN (CoRNN), a time-discretization of a system of second-order ordinary differential equations, which mitigated the exploding and vanishing gradient prob-

lem. CoRNN achieved 87.4% accuracy on IMDb by precisely bounding the gradients of hidden states.

LSTMs also contribute to mitigating these problems. [26] applied LSTM on movie review SA, exploring different hyperparameters like dropout, number of layers, and activation functions. Their LSTM configuration, including embedding, LSTM layer, dense layer, 0.5 dropout, and 100 LSTM units, achieved an accuracy of 88.46% on IMDb. Although LSTMs handle longer sequences efficiently, whether the incorporated gates in the LSTM architecture offer sufficient generalization or additional data training is required remains unclear [11]. To address this, [17] applied a BiLSTM network using a mixed objective function for text classification, employing both supervised and unsupervised approaches. Their study showcased that a simple BiLSTM model using maximum likelihood training delivered competitive performance in polarity classification, reporting a 6.07% error rate.

Although BiLSTM displayed superior results compared to other deep learning methods, room for performance enhancement remains at a 6.07% error rate. Hence, in our work, BiLSTM is adopted to further improve performance in this task.

C. BERT

This section delves into works that utilize BERT for SA and specifically for analyzing movie reviews.

1) *BERT and SA*: BERT stands out as a renowned SOTA language model for its exceptional performance across various NLP tasks. For instance, in the realm of SA, [27] delved into attention mechanisms and pre-trained hidden representations of BERT for Aspect-Based SA (ABSA). Their analysis revealed BERT's utilization of minimal self-attention heads to encode contextual words, such as prepositions or pronouns indicating an aspect, and opinion words associated with aspects. Conversely, [28] explored the potential of contextualized embeddings from BERT in an end-to-end ABSA task, focusing on integrating BERT embeddings with various neural models. Their findings showcased the impressive performance of BERT when combined with models like Gated Recurrent Units (GRU).

In a similar vein, [29] leveraged the pre-trained BERT model for target-dependent sentiment classification, examining its context-aware representation for potential improvements in ABSA. Their study revealed that coupling BERT with complex neural networks previously effective with embedding representations did not add substantial value to ABSA.

Other investigations of BERT involve transfer learning approaches. [30] fine-tuned a pre-trained BERT model for ABSA by transforming ABSA into a sentence-pair classification task, achieving an impressive 92.8% accuracy on the SentiHood dataset. Meanwhile, [31] explored BERT for fine-tuning on Review Reading Comprehension (RRC) and ABSA tasks, generating a ReviewRC dataset from a benchmark for ABSA. Their novel post-training fine-tuning approach on BERT achieved an accuracy of 90.47%.

These studies collectively showcase effective fine-tuning techniques with BERT, particularly in ABSA tasks. In our work, inspired by the coupling technique used in [28], we opt to couple BERT with BiLSTM for the movie reviews SA task.

2) *BERT and Movie Reviews SA*: In the realm of movie reviews, [32] employed BERT to transform words into contextualized word embeddings. They fine-tuned BERT's parameters using the IMDb movie reviews corpus through Inductive Conformal Prediction (ICP), achieving an accuracy of 92.28%. In contrast, [33] pursued a different approach, comparing BERT against SentiWordNet [34], logistic regression, and LSTM for Movie Reviews SA on the IMDb dataset. Their study aimed to ascertain the relative efficacy of the four SA algorithms, highlighting the undeniable superiority of the pre-trained advanced supervised BERT model in text sentiment classification. BERT notably outperformed other models, achieving an accuracy of 92.31%. Notably, both studies focused on binary classification.

Conversely, [10] employed BERT for both binary and fine-grained classifications on SST-2 and SST-5 datasets, respectively. Their model showcased superior performance compared to other deep learning-based models, such as CNN and RNN, achieving an accuracy of 93.7% on SST-2 and 55.5% on SST-5 tasks.

It is evident that deep learning techniques have proven to be the most accurate approaches for SA. Transfer learning, particularly fine-tuning BERT, consistently yields outstanding results. However, despite the reported advancements and the capabilities of BERT, there remains significant potential to further enhance the pre-trained language model's performance in this field. Furthermore, many studies have predominantly focused on either polarity sentiment classification or fine-grained sentiment classification, overlooking an exploration into how the categorical scope of sentiment polarities affects model robustness. Additionally, most researchers have not evaluated their approaches across a wide spectrum of available benchmark datasets for SA. For instance, results are often reported exclusively for IMDb or SST variants.

Therefore, our current work aims to fine-tune BERT by coupling it with BiLSTM for both polarity classification and fine-grained sentiment classification, given that these techniques have demonstrated superior performance in the existing literature. Leveraging transfer learning insights from [18], we extend previous methodologies by computing the overall polarity of sentiments, as demonstrated in [13]. While a prior study computed overall polarity based on a single output vector derived from Twitter replies using LSTM for three-class classification, we intend to compute this using the output vector obtained from BERT coupled with BiLSTM, tailored to each specific classification task. Our experimental evaluations will encompass diverse benchmark datasets as shown in Section IV.

III. METHODOLOGY

This section discusses various techniques employed in this study, encompassing SA, BERT, BiLSTM, fine-tuning BERT

with BiLSTM, different classification tasks, techniques for performance enhancement, computation of overall polarity, and an overview of the entire study.

A. SA

SA is a sub-domain of opinion mining, focusing on extracting emotions and opinions regarding a specific topic from structured, semi-structured, or unstructured textual data [35]. It can be approached either as a binary or fine-grained sentiment classification task. In binary classification, machine learning models categorize text into positive or negative sentiment categories. In contrast, fine-grained classification involves utilizing more than two sentiment classes, enabling the computation of multi-scale sentiment distribution. In our context, we investigate both classifications. Our aim is to explore the robustness and consistency of the model in generalization across varying categorical scopes of sentiment polarities.

B. Model Architecture

The model architecture consists of BERT, BiLSTM and a fully connected dense layer.

1) *BERT*: BERT was introduced by researchers from Google [2]. BERT primarily focuses on pre-training deep bidirectional representations from unlabeled text by jointly considering both left and right contexts across all layers of the model. Consequently, BERT can be fine-tuned with a single additional layer for various downstream tasks such as SA, question answering, and more. Pre-training of BERT involved two unsupervised tasks.

a) *Masked Language Modeling*: For this task, 15% of the tokens within the input sequence undergo random masking. Subsequently, the complete input sequence is processed through a deep bidirectional transformer encoder, and the output softmax layer is designed to predict the masked words [2].

b) *Next Sentence Prediction*: BERT establishes a relationship between two input sentences, denoted as sentence *A* and sentence *B*, by predicting whether these sentences logically follow each other within a specific monolingual corpus. During training, 50% of the inputs consist of sentence pairs where the second sentence is the immediate subsequent sentence in the source document. Conversely, the remaining 50% involve pairs where a random sentence from the corpus is chosen as the second sentence [10].

BERT processes a sequence of input tokens simultaneously due to its multiple attention heads, and the model is primarily available in two sub-models: BERT_{BASE} and BERT_{LARGE}. In this work, we utilize BERT_{BASE}, which comprises 12 layers, 768 hidden states, 12 attention heads, and 110M parameters. In contrast, BERT_{LARGE} features approximately twice the specifications of BERT_{BASE}. Specifically, the uncased version of BERT_{BASE}, referred to as *bert-base-uncased*, which processes input tokens in lowercase, is employed in this study.

In BERT, there is a specific format for input tokens. The initial token of each sequence is labeled as [CLS]. This token corresponds to the final hidden layer, gathering and

aggregating all the information within the input sequence, particularly for classification tasks. To distinguish between sentences within a single input sequence, two methods are employed: the use of a special token, [SEP], for separation and the addition of a learned embedding to each token, thereby identifying the sentence to which it belongs.

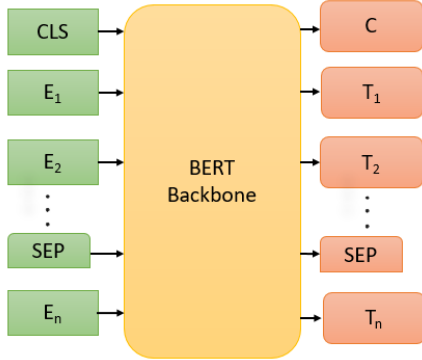


Fig. 1. Simplified diagram of BERT

Figure 1 illustrates a simplified diagram of BERT. E_n denotes the input representation of a single token, generated by summing the respective token, segment, and position embeddings. The *BERT Backbone* symbolizes the primary processing carried out by BERT, while T_n represents the hidden state corresponding to token E_n . Additionally, C represents the hidden state corresponding to the aggregated token [CLS]. Consequently, we utilize C as the input for the fine-tuning components in sentiment classification.

2) *BiLSTM*: BiLSTM is an LSTM variant that processes input features in both forward and backward directions [12]. This bidirectional characteristic provides BiLSTM an advantage in effectively capturing higher-level sentiment representations from the BERT hidden state C . Additionally, BiLSTM inherits the advantageous features of LSTM, including the ability to retain long-distance temporal dependencies and avoiding the issues of vanishing or exploding gradients during backpropagation through time. The enhancement of the performance of the model by supplementing it with the BiLSTM module, as demonstrated in experiments, outperformed the usage of solely BERT with a dense layer.

3) *Dense Layer*: The hidden state from BiLSTM feeds into a dense layer. This layer, based on the BiLSTM output, generates a higher-level feature set that is readily distinguishable for the targeted number of classes. Finally, a softmax layer is added atop the dense layer to yield the predicted probability distribution for the target classes.

C. Fine-tuning BERT with BiLSTM

Because BERT is pre-trained, there is no necessity to train the entire model from the beginning [2]. Consequently, we simply transfer knowledge from BERT to the added fine-tuning layer and train this layer for SA.

In this study, the fine-tuning process operates as follows. After data pre-processing, three layers are established, one

utilizing BERT and the subsequent layers involving BiLSTM and dense networks. The data pre-processing phase generates two input values, known as *attention masks* and *input ids*. These serve as the input embeddings to the model.

The input embeddings are then passed through the BERT module. The dimensionality of these embeddings is contingent on various factors, including the input sequence length, batch-size, and the number of units in BERT's hidden state.

Subsequently, the BiLSTM assimilates information from BERT and forwards it to the dense layer, which predicts the corresponding classes for the input features through the succeeding softmax layer. BERT and BiLSTM share the same hyperparameters, detailed in Section IV under experimental settings.

The fine-tuning process is depicted in Figure 2. Within the figure, *Input features* represent tokens in a review, *Input ids* symbolize an input sequence, and *Attention masks* are binary tensors that highlight the position of the padded indices in a particular sequence to exclude them from attention. These masks use binary values 1 to indicate positions that require attention and 0 for padded values. Padding ensures uniform sequence lengths for input data sentences, a common practice in NLP. Consequently, the padded information is not considered part of the input and has minimal impact on the model's generalization.

The output from BERT matches the dimensionality of the input to BiLSTM, set at 768 and represented by C . Only C is transmitted to BiLSTM. The BiLSTM layer includes a single hidden component, followed by a dense layer that receives the hidden state H from BiLSTM.

D. Classification

In this study, BERT is fine-tuned for both binary (polarity) sentiment classification and fine-grained SA. The fine-grained classification aspect encompasses three distinct tasks: three-class (3-point scale), four-class (4-point scale), and five-class (5-point scale) classifications.

1) *Polarity classification*: In this context, polarity classification entails classifying a movie review R as either conveying a positive or a negative sentiment polarity. This fundamental task is often a cornerstone of SA, as it primarily focuses on discerning between positive and negative sentiments within a text [4].

Additionally, to assess the robustness of the model in handling varying categorical scopes of sentiment polarities, we further perform fine-grained sentiment classification. This involves utilizing different classification scales to gauge how the polarity of an individual review and the overall polarity of a collection of reviews change as the classification becomes more detailed. For instance, if a review is classified as negative in a binary classification dataset, we aim to understand how this polarity aligns when labels are expanded to include highly negative, negative, neutral, positive, and highly positive categories. We extend this analysis to overall sentiment polarity as well. Consequently, we conduct the following fine-grained classification tasks.

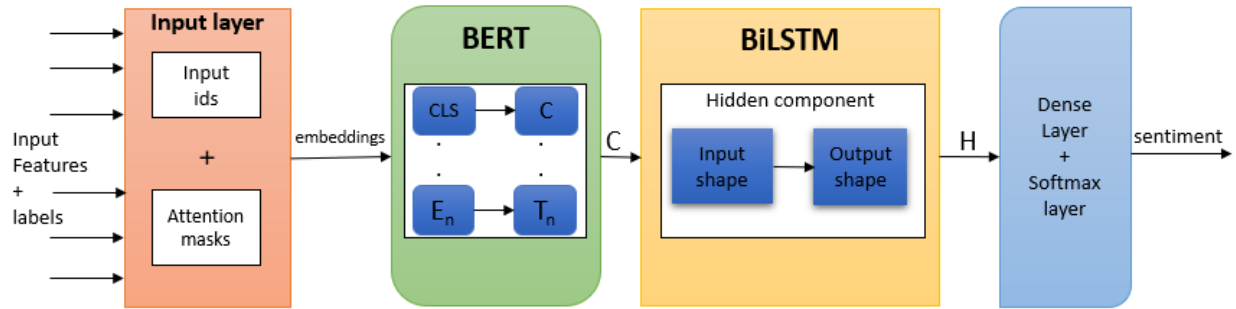


Fig. 2. Fine-tuning part of BERT with BiLSTM

2) *Three-class classification*: This expands upon binary classification by introducing a neutral class to account for instances where reviewers might not distinctly assign a positive or negative sentiment to a movie due to ambiguous sentiments or a lack of clear preference [13]. Sometimes, the review might contain a balance of positive and negative words, leading to ambiguity in sentiment. Hence, a neutral polarity is introduced to accommodate such cases. In the context of three-class classification, the task is defined as follows: given a movie review R , classify it as carrying a negative, neutral, or positive sentiment polarity.

Alternatively, the output vector from binary classification can be directly adapted into three-class classification without altering the label scope in the training data or restarting the training process for the three-class task. By employing a sigmoid activation function or a softmax layer that assigns varying confidence levels to negative and positive classes, a neutral class can be incorporated using a delta value, δ . In this approach, the actual output label can be replaced by a neutral label if the discrepancy between the probabilities of the original two classes is smaller than δ . However, this method necessitates a careful definition of δ to ensure meaningful and accurate outcomes.

3) *Four-class classification*: This task specifically targets the IMDb dataset. To extend the binary IMDb classification to four classes, we adopt a hierarchical approach using binary tree splitting. This technique, initially introduced in [36], leverages binary segmentation to identify homogeneous nodes within a tree structure. In our scenario, negative reviews are divided into highly negative and negative, while positive reviews are categorized into positive and highly positive, as illustrated in Figure 3. Here, D represents the dataset, while N , P , HN , and HP symbolize Negative, Positive, Highly Negative, and Highly Positive reviews, respectively. A detailed explanation of the binary tree splitting method's application is provided in Section IV within the IMDb dataset analysis. Consequently, the four-class classification is defined as follows: given a movie review R , classify it based on whether it carries a highly negative, negative, positive, or highly positive sentiment polarity.

4) *Five-class classification*: Five classes are used here as in [10] and [37]. While [37] used the output vector obtained from

this classification to estimate the distribution of data examples across the five classes, we use the vector to find the overall sentiment polarity of all the predictions. Five-class task is defined as follows. Given a movie review R , classify it as whether carrying a highly negative, negative, neutral, positive, or highly positive sentiment polarity.

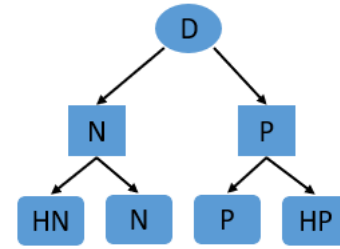


Fig. 3. Binary Tree Splitting

E. Accuracy Improvement Approaches

We utilized distinct data oversampling and augmentation techniques individually to improve the accuracy of the model for fine-grained sentiment classification.

1) *Oversampling*: The Synthetic Minority Over-sampling Technique (SMOTE) was introduced in [38] to address imbalanced datasets problem and enhance model performance. As SMOTE primarily operates with numerical input data, we initially converted reviews from the minority classes into their corresponding numerical features. Subsequently, we used these features as input for SMOTE, which conducted oversampling to balance the dataset.

2) *Augmentation*: Data augmentation serves to address class imbalance and maximize information extraction from limited resources [39]. We employed NLP Augmenter (NLPAUG), a technique leveraging operations based on abstractive summarization and synonym replacement driven by the proximity of word embedding vectors. Our experiments found this technique successful, and we depict it in Eq. (1):

$$V_{\text{AUG}} = F(V_{\text{IN}}) \quad (1)$$

Here, V_{AUG} represents the output matrix of augmented sentences, where F symbolizes the abstractive summarization augmentation function applied to the input matrix V_{IN} containing raw text data. Both matrices contain n vectors, corresponding to the misrepresented (minority) classes within the dataset. Furthermore, the input vectors comprise randomly sampled data examples from specific misrepresented classes, establishing a one-to-one mapping between input and output vectors.

After applying these techniques, we initially combined the output from SMOTE with the original input data and proceeded to train the model using the SST-5 benchmark dataset. Subsequently, we performed a similar procedure using V_{AUG} .

F. Overall Sentiment Polarity

We define the overall sentiment polarity as follows: Given an output vector V from BERT+BiLSTM containing sentiment labels for N reviews, we compute the dominant sentiment polarity within the vector. To derive the overall sentiment polarity of the reviews, we input the output vector of BERT+BiLSTM into a heuristic algorithm. In this process, BERT+BiLSTM initially predicts the sentiment category for each review, aggregating the results in an output vector. The occurrences of each class label within the output vector are tallied, and the result is passed to the heuristic algorithm to determine the predominant sentiment polarity for all the reviews collectively.

The algorithm has been modified to compute the overall sentiment polarity from the output vectors of binary, four-class, and five-class classifications. The functionality of the algorithm is dependent on the number of classes in the output vector, necessitating the derivation of three variants of the algorithm for these different classification tasks.

Algorithm 1 outlines the computation of the overall polarity from the output vector of the three-class classification. Initially, if the proportion of neutral reviews exceeds a threshold, set at 85%, the overall polarity is designated as neutral. This threshold acknowledges that a majority of reviews might tend towards a neutral sentiment rather than distinctly positive or negative.

Next, the algorithm considers the distribution of negative and positive sentiments in the output vector. Typically, individual reviews seldom express exclusively positive or negative sentiments. Therefore, if the number of negative reviews exceeds positive reviews by at least 1.5 times, or vice versa, the dominant sentiment determines the overall polarity. This criterion ensures that a sentiment must significantly outweigh the other to influence the overall polarity.

Finally, when the counts of positive and negative reviews are nearly equal, indicating a lack of a clear dominance between positive and negative sentiments, the overall polarity is categorized as neutral once more. This scenario implies a balanced representation of both sentiments across the reviews.

To compute the overall polarity from the binary classification task, Algorithm 2 is derived from Algorithm 1 with modifications to account for the absence of a neutral polarity in binary classification. While Algorithm 1 considers the

Algorithm 1: Overall sentiment polarity computation from three-class classification output vector.

Result: Dominating sentiment polarity 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;
```

Algorithm 2: Overall sentiment polarity computation from binary classification output vector.

Result: Dominating sentiment polarity 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
```

presence of a neutral sentiment, binary classification does not include this category. Therefore, in Algorithm 2, the logic is adjusted to directly assess the quantities of positive and negative reviews. However, in cases where neither category dominates the sentiment output (i.e., quantities of positive and negative reviews are approximately similar), a neutral sentiment polarity is introduced for the overall polarity computation, representing a tie between positive and negative sentiments.

In our formulations, we employed variable coefficients, namely 1.2 and 1.5, to ascertain the majority gap for decision-making within the algorithms, for comparisons within the algorithms.

Algorithm 2 serves as the foundation for Algorithm 3, enabling the computation of overall polarity from the four-class output vector of BERT+BiLSTM. This algorithm operates hierarchically, taking into account the binary tree splitting illustrated in Figure 3. The hierarchical comparison starts with base classes, progressing to sub-classes within a base class that holds the majority of samples. The process involves aggregating the sample counts of all fine-grained classes under each super class. For instance, the total for the negative super class is derived from the highly negative and negative sub-class totals.

Three distinct scenarios are considered within the algorithm.

Algorithm 3: Overall sentiment polarity computation from four-class classification output vector.

Result: Dominating sentiment polarity for all reviews.

```

1 if # (highly negative reviews + negative reviews) > 1.2
  × # of (positive reviews + highly positive reviews)
  then
2   if #highly negative reviews > 1.5 × # of negative
     reviews then
3     | overall polarity ← highly negative
4   else
5     | overall polarity ← negative
6 else if # (positive reviews + highly positive reviews)
  > 1.2 × # of (highly negative reviews + negative
  reviews) then
7   if #highly positive reviews > 1.5 × # of positive
     reviews then
8     | overall polarity ← highly positive
9   else
10    | overall polarity ← positive
11 else
12  | overall polarity ← neutral

```

In the first case, if the total number of negative reviews across super classes exceeds the total number of positive reviews by at least 1.2 times, and the count of highly negative sub-class reviews is at least 1.5 times that of the negative sub-class, a highly negative overall polarity is assigned. Conversely, if the positive reviews surpass the negative reviews by 1.2 times, and the highly positive sub-class reviews dominate by 1.5 times over the positive sub-class, an overall positive polarity is assigned. Finally, if no significant dominance is observed between the total numbers of negative and positive reviews within the base classes, a neutral overall polarity is assigned.

For base classes, the threshold of 1.2 often works best to determine the dominating class. However, in the more finely-grained sub-classes, such as highly positive and highly negative, a threshold of 1.5 tends to perform better. This distinction is due to the increased granularity of the sub-classes, requiring a higher sample size of the dominating subclass to designate its label as the overall polarity.

For five-class classification, Algorithm 4 extends from Algorithm 3 with an additional initial step. Initially, the overall polarity is evaluated as neutral if the proportion of neutral reviews exceeds a designated threshold, typically set at 85%. Following this initial consideration, the subsequent steps in Algorithm 3 are maintained without alteration.

A simplistic method for determining overall polarity involves tallying the count of labels for each class within the BERT+BiLSTM output vector and assigning overall polarity based on the majority class. However, this basic approach may not adequately represent the dominant sentiment. Consider binary classification, if the output vector consists of 49 positive reviews and 51 negative reviews, the overall polarity will be negative. Yet, a difference of 2 is not conclusive enough to

Algorithm 4: Overall sentiment polarity computation from five-class classification output vector.

Result: Dominating sentiment polarity for all reviews.

```

1 if #total neutral reviews > 85% of the total reviews
  then
2   | overall polarity ← neutral
3 else
4   if (#highly negative reviews + #negative reviews)
     > 1.2 × # of (positive reviews + highly positive
     reviews) then
5     | if #highly negative reviews > #1.5 × negative
       reviews then
6       | overall polarity ← highly negative
7     else
8       | overall polarity ← negative
9   else if # (positive reviews + highly positive
     reviews) > 1.2 × # of (highly negative reviews +
     negative reviews) then
10    | if #high positive reviews > 1.5 × # of positive
       reviews then
11    | overall polarity ← highly positive
12    else
13    | overall polarity ← positive
14  else
15  | overall polarity ← neutral

```

determine the prevailing sentiment across all reviews.

Furthermore, the levels of positivity or negativity vary among reviews in the datasets. To address this, we employ specific formulations in the respective algorithms. These formulations require a significantly higher majority within a class in the output vector to assign its label as the overall sentiment polarity; otherwise, the overall sentiment polarity defaults to neutral.

Additionally, we refined the coefficients to 1.2 and 1.5 in the algorithms based on various observations of the model's behavior across different scenarios. These empirically determined values significantly improved the accuracy of overall polarity computation, closely reflecting the original sentiment polarity of input features.

G. Overview of the Work

Illustrated in Figure 4, the overview of our work encapsulates several key steps. Initially, raw text data undergoes pre-processing to transform it into features compatible with BERT. Subsequently, these features are fed into BERT+BiLSTM through the fine-tuning layer, which configures the necessary hyperparameters for BERT+BiLSTM. Finally, the output vector from BERT+BiLSTM predictions is utilized to compute the overall sentiment polarity.

IV. EXPERIMENTS

This section provides an insight into the datasets utilized in this study, followed by an in-depth exploration of the data pre-processing techniques. Subsequently, it delves into the experi-

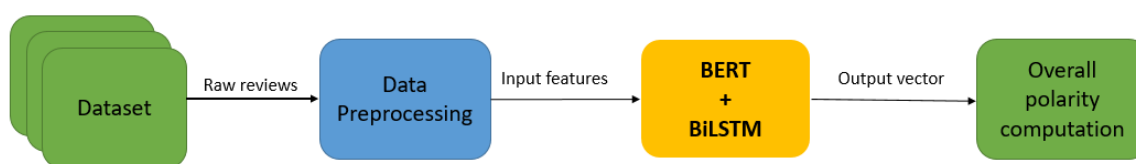


Fig. 4. Overview of our work

mental settings, elucidates the evaluation metrics employed in the experiments, and culminates in a comprehensive discussion of the experimental findings.

A. Datasets

The datasets employed within this study revolve around movie reviews meticulously annotated for SA across a diverse array of scales, specifically, 2-point, 3-point, 4-point, and 5-point gradations. The subsequent elucidation encapsulates the detailed descriptions of these datasets.

1) *IMDb movie reviews*: The IMDb movie reviews dataset [40], stands as a widely embraced binary SA collection, encompassing a colossal 50,000 reviews sourced from the Internet Movie Database (IMDb). This assortment impeccably balances negativity and positivity, presenting an equal split between negative and positive reviews. Within this study, the dataset assumes a pivotal role in binary classification tasks, further expanding its utility to encompass three-class and four-class classification endeavors. Comprising three fundamental columns (reviews, sentiment score, and label), the dataset's sentiment scores range from integers 1 to 10, excluding the neutral ground of 5 and 6. Reviews carrying scores between 1 to 4 are distinctly labeled as negative, while those within the 7 to 10 spectrum are unequivocally regarded as positive.

In extending the dataset towards a four-class classification paradigm, our approach leveraged the concept of binary-tree splitting, visually represented in Figure 3. Each review within the dataset boasts eight distinct sentiment scores, owing to the absence of scores 5 and 6. Our classification strategy involves categorizing reviews bearing scores 1 and 2 as highly negative, while scores 3 and 4 are tagged as negative. Conversely, scores 7 and 8 merit a positive label, and ratings of 9 and 10 earn the designation of highly positive. For the three-class model, our breakdown was as follows: scores 1, 2, and 3 corresponded to negative sentiment, scores 4 and 7 indicated a neutral stance, and ratings of 8, 9, and 10 were indicative of positive sentiment. We refer to these modified datasets as IMDb-2, IMDb-3, and IMDb-4, representing the binary, three-class, and four-class renditions of the original dataset, respectively.

2) *SST*: The Stanford Sentiment Treebank (SST) stands as a punctiliously curated corpus adorned with fully labeled parse trees, offering a profound exploration into the intricate nuances of sentiment's compositional impact on language. Originating from the dataset introduced in [41], SST comprises a collection of 11,855 individual sentences meticulously extracted from movie reviews. These sentences underwent parsing via the Stanford parser, resulting in an extensive repository of 215,154

unique phrases diligently annotated by three human judges. Each phrase within this dataset bears a label denoting its sentiment, falling within the spectrum of negative, somewhat negative, neutral, somewhat positive, or positive, equating to the highly negative, negative, neutral, positive, and highly positive labels used within our annotations.

The SST dataset manifests in two distinct versions: SST-5, known as SST fine-grained, employs five labels to characterize sentiment nuances, while SST-2, termed SST binary, simplifies the classification into two primary labels, negative and positive. Within SST-2, negative sentiments encompass judgments labeled as negative or somewhat negative, while positive sentiments entail those marked as somewhat positive or positive. Notably, neutral reviews find no place within SST-2. In our present study, we harness SST-2 for binary classification tasks and SST-5 for more intricate five-class classification endeavors, thus delving deeper into the subtleties of SA within the corpus.

3) *MR Movie Reviews*: The MR movie reviews dataset encompasses a wealth of movie review documents carefully labeled based on their overarching sentiment polarity, whether they lean positively or negatively, or subjective rating, such as nuanced assessments like two and a half stars. Additionally, it contains sentences categorized according to their subjectivity status, distinguishing between subjective or objective content, and polarity.

Within the scope of this paper, we specifically employ the version introduced in [42], a rigorously curated subset comprising 5,331 positive and 5,331 negative processed reviews. In our experiments, we exclusively harness the MR movie reviews dataset for a singular purpose: the binary classification task. This dataset serves as the cornerstone of our experiments, focusing solely on the polarity aspect of SA.

4) *Amazon Product Data dataset*: This dataset is an expansive repository housing product reviews and metadata sourced from Amazon, encompassing a staggering 142.8 million reviews dating from May 1996 to July 2014. Its comprehensive scope spans reviews, product metadata, and interlinkages. Initially introduced in [43] as a resource for SA utilizing product review data, it was further utilized in [44] to construct a recommender system operating within a collaborative filtering framework tailored specifically for Amazon products.

Within the context of our study, our focal point centers solely on video reviews within this extensive dataset. Initially, the dataset featured labels graded from scores 1 to 5, signifying a spectrum of polarity strength ranging from highly negative to highly positive sentiments. To streamline our analysis toward

binary classification, we undertook a transformation: scores 1 and 2 were consolidated into a negative label, while scores 4 and 5 were amalgamated into a positive label. Simultaneously, score 3, representing a neutral stance, was omitted, akin to the approach adopted in SST-2 [41].

For the sake of clarity and specificity within our research, we distinguish between Amazon-2 and Amazon-5, representing the binary and five-class versions of the dataset, respectively. Table I provides a comprehensive statistical overview of the datasets employed in our experiments, where categorical labels like H POS (Highly Positive), POS (Positive), NEG (Negative), and H NEG (Highly Negative) elucidate distinct sentiment categories.

B. Data pre-processing

The data pre-processing step was commenced by eliminating empty reviews across all datasets, ensuring a clean and consistent starting point for our analysis. Subsequently, the focus shifted towards implementing recommended data pre-processing measures aimed at translating the raw input data into a format compatible with BERT's comprehension. This undertaking encompassed two pivotal steps integral to the transformation process.

Initially, we generate input examples utilizing the BERT-provided constructor, which operates based on three key parameters: *text_a*, *text_b*, and *label*. Here, *text_a* encapsulates the body of text we aim to classify, specifically, the collection of movie reviews sans their associated labels. The *text_b* parameter, primarily utilized in scenarios involving sentence relationships such as translation or question answering, remains intentionally left blank given its minimal relevance to our current research focus. Meanwhile, *label* serves as the container for the sentiment polarity labels associated with each movie review, representing an essential component of the input features.

For a deeper understanding of this foundational step, additional insights can be gleaned from the original BERT paper [2]. Subsequently, we progress through a series of text pre-processing steps integral to our methodology.

- Lowercase all 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.
- Add 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 that a specific token belongs to.

Following these processes, the tokenizer yields crucial outputs: *input ids* and *attention masks*. These outputs serve as pivotal components, subsequently utilized as inputs alongside

the review labels for our model. These elements collectively form the foundational input data fed into our analysis framework.

C. Experimental settings

We conducted fine-tuning on *bert-base-uncased*, a variant of BERT designed to process lowercase tokens. While initializing most of BERT's layers from the model checkpoint, some layers required new initialization. Throughout the training process, we deliberately froze the initial layers containing BERT weights, allowing the primary focus to rest on the latter layers housing the fine-tuning components. These latter layers, holding trainable weights, were continuously updated to minimize loss during the training phase, specifically tailored to the downstream task of SA.

Numerous simulations were executed across the datasets to optimize the hyperparameters of the model. From our exhaustive experiments, the most optimal results emerged with specific hyperparameters. For binary classification, the model exhibited exceptional performance using the Adam optimizer [45], a batch size of 32, a learning rate of 3e-5, an epsilon value of 1e-08, a maximum sequence length of 128, and employing binary cross-entropy loss. Training was conducted over 10 epochs, iterating through each batch.

Conversely, in all fine-grained classifications, a batch size of 64, a learning rate of 1e-4, a maximum sequence length of 256, a decay of 1e-5, alongside the Adam optimizer and sparse categorical cross-entropy loss, facilitated optimal outcomes. This specific version underwent training for 15 epochs, repeating steps for batches. Notably, we observed instances of overfitting when attempting to increase the number of epochs for these respective models.

To facilitate broader adoption and enable replication of our work, the code for this project is readily available [46], ensuring accessibility for interested parties.

D. Evaluation Metrics

In line with the evaluation metrics adopted in [10], our approach also employs accuracy as a performance measure to assess the efficacy of our model in comparison to other models. Accuracy, in its essence, is straightforwardly defined as:

$$accuracy = \frac{\text{number of correct predictions}}{\text{total number of predictions}} \times 100 \quad (2)$$

E. Results and Discussion

We delineate accuracy comparisons across various datasets and classification tasks in our study. Table II illustrates binary classification results, encompassing a comparison between our model and others. Subsequently, in Table III, we present outcomes for three-class and four-class classifications, exclusively on the IMDb dataset. Last, Table IV delves into five-class classification on SST-5 and Amazon-5 datasets. Across these tables, our model consistently surpasses all others in performance across diverse classification tasks on every

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

Dataset	Train samples					Test samples				
	H POS	POS	NEUTRAL	NEG	H NEG	H POS	POS	NEUTRAL	NEG	H NEG
IMDb-2	-	12500	-	12500	-	-	12500	-	12500	-
MR	-	4264	-	4265	-	-	1067	-	1066	-
SST-2	-	4300	-	4244	-	-	886	-	1116	-
Amazon-2	-	239660	-	37056	-	-	59949	-	9231	-
IMDb-3	-	18227	4816	14958	-	-	4556	1204	3739	-
IMDb-4	11471	8530	-	8234	11767	2867	2132	-	2058	2941
SST-5	1482	2489	1794	2512	1208	370	622	448	628	302
Amazon-5	182000	57688	27767	15168	21863	45500	14421	6941	3791	5465

dataset, solidifying its position as the new benchmark with SOTA accuracy levels.

The analysis of our results reveals a distinct trend: our model excels notably in binary classification tasks. However, a discernible pattern emerges as we move towards finer classification scales, evident across the IMDb, SST, and Amazon datasets showcased in Table II, Table III, and Table IV. As the classification granularity increases, the accuracy of our model experiences a consistent decline.

This decline in accuracy can be attributed to several factors. The primary reason lies in the heightened complexity that arises with an augmented number of classes within a dataset. As the distinctions between classes become more intricate, the task of accurate classification becomes increasingly challenging for the model.

Another contributing factor is the distribution of samples across these classes. Despite the proliferation of classes, the number of samples per class remains constrained by the original dataset size. Consequently, with the expansion of classes, each sentiment category possesses a reduced number of examples available for effective model learning. This scarcity of samples within each sentiment category hampers the ability of the model to learn distinct patterns for accurate predictions, particularly as the categorical spectrum of sentiment polarities widens within a dataset.

Through our experimental observations, a striking finding emerges: the model exhibits robustness and consistency, showcasing resilience even amidst alterations in the categorical scope of sentiment polarities within a specific dataset. To illustrate, we delve into a review snippet sourced from the IMDb benchmark. *“This show is great. Not only is ‘Haruhai Suzumiya’ a very well written anime show, it also reflects things like Philosophy, Science Fiction and a little religion. It’s hilarious at some points and ‘cute’ (for lack of a better term) at others. Actually this may be effect to my lack of experience with Japanese anime shows, but it is one of the best of its genre I have seen. I mainly have to give credit to the writers. I haven’t seen such brilliant scopes of imagination in a television show since the original Star Trek. I hope the writers continue to add strange new characters and give more insight on the already great characters that have been added”*. In the context of binary classification within IMDb-2, the model confidently designates the text as *Positive*. Similarly, in the three-class

classification of IMDb-3, it maintains this classification. However, when exploring the four-class classification in IMDb-4, the model assigns a *Highly Positive* label. This classification aligns impeccably with the sentiment score embedded within the review text, a rating of 9 on a scale of 1 to 10 for goodness.

This astute adaptability demonstrates the proficiency of the model in discerning and leveraging nuanced semantic cues embedded within a dataset. It dynamically adjusts its classification strategy based on the chosen categorical scope, effectively capturing and reflecting the varying degrees of sentiment granularity present within the dataset.

Our experimental findings, showcased in Table V using the SST-5 dataset, include results concerning oversampling with SMOTE and augmentation using NLP AUG. Two distinct approaches are highlighted: BERT+BiLSTM+SMOTE, utilizing SMOTE in conjunction with BERT and BiLSTM for SA, and BERT+BiLSTM+NLP AUG, employing NLP AUG for augmentation.

Surprisingly, the inclusion of SMOTE in our model exhibits no discernible impact on accuracy. Remarkably, the accuracy of a simpler model, BERT+BiLSTM, employing solely BERT and BiLSTM without any accuracy enhancement techniques, surpasses that of BERT+BiLSTM+SMOTE. This implies that the oversampling technique fails to impart semantic understanding to the model from the augmented data it produces.

Conversely, BERT+BiLSTM+NLP AUG showcases a performance boost, elevating the accuracy of the model from 58.44% to 60.48%. The rationale behind these observations lies in the input transformation process. SMOTE operates on text inputs that have been transformed into BERT features. This transformation introduces noise and disrupts the efficacy of SMOTE, as some semantic nuances are lost during this process. In contrast, NLP AUG operates directly on raw text data, facilitating an easier extraction of semantic information during the learning process. This direct access to raw text enables NLP AUG to enhance the performance of the model by leveraging the inherent semantics present in the data.

To encapsulate the discussion of results, we focus on the comprehensive computation of overall sentiment polarity across all datasets facilitated by our model, as detailed in Table VI. Here, *OP* signifies Overall Polarity. We distinguish between *Original OP*, known prior to input embeddings entering BERT+BiLSTM, and *Computed OP*, derived after the model

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

Model name	Dataset			
	IMDb-2	MR	SST-2	Amazon-2
RNN-Capsule [47]	84.12	83.80	82.77	82.68
coRNN [18]	87.4	87.11	88.97	89.32
TL-CNN [18]	87.70	81.5	87.70	88.12
Modified LMU [48]	93.20	93.15	93.10	93.67
DualCL [49]	-	94.31	94.91	94.98
L Mixed [50]	95.68	95.72	-	95.81
EFL [19]	96.10	96.90	96.90	96.91
NB-weighted-BON+dv-cosine [16]	97.40	-	96.55	97.55
SMART-RoBERTa Large [51]	96.34	97.5	96.61	-
Ours	97.67	97.88	97.62	98.76

TABLE III
ACCURACY (%) COMPARISONS FOR THREE AND FOUR CLASS CLASSIFICATION ON IMDD

Model name	Dataset	
	IMDb-3	IMDb-4
CNN-RNF-LSTM [52]	73.71	63.78
DPCNN [22]	76.24	66.17
BERT-large [10]	77.21	66.87
Ours	81.87	70.75

TABLE IV
ACCURACY (%) COMPARISONS OF MODELS ON BENCHMARK DATASETS FOR FIVE CLASS CLASSIFICATION

Model name	Dataset	
	SST-5	Amazon-5
CNN+word2vec [23]	46.4	48.85
TL-CNN [18]	47.2	58.1
DRNN [53]	-	64.43
BERT-large [10]	55.5	65.83
BCN+Suffix+BiLSTM-Tied+Cove [54]	56.2	65.92
RoBERTa+large+Self-explaining [55]	59.10	-
Ours	60.48	69.68

TABLE V
ACCURACY (%) OF OUR MODEL WITH ACCURACY IMPROVEMENT TECHNIQUES ON SST-5

Classification task	Accuracy
BERT+BiLSTM	58.44
BERT+BiLSTM+SMOTE	58.36
BERT+BiLSTM+NLAUG	60.48

predicts sentiments for reviews.

Remarkably, the table demonstrates an intriguing consistency: *Computed OP* aligns precisely with *Original OP* across all datasets. *Original OP* was initially computed by tallying the sample counts for each label within the input features and then applying a specific heuristic algorithm tailored to the classification task. This initial computation was utilized as a benchmark to verify the accuracy of *Computed OP*.

Furthermore, the table also highlights a consistent *Computed OP* across different versions of the same dataset. This uniformity across dataset variations reinforces our confidence in the

TABLE VI
OVERALL POLARITY COMPUTATION ON ALL THE DATASETS

Dataset	Original OP	Computed OP ^b
IMDb-2	Neutral	Neutral
IMDb-3	Neutral	Neutral
IMDb-4	Neutral	Neutral
MR reviews	Neutral	Neutral
SST-2	Neutral	Neutral
SST-5	Neutral	Neutral
Amazon-2	Positive	Positive
Amazon-5	Positive	Positive

ability of the heuristic algorithm to accurately compute the expected overall polarity from the BERT+BiLSTM output vector. This confidence persists despite variations in the categorical scope of sentiment polarities within the movie reviews across different dataset iterations. This robustness ensures that the heuristic algorithm consistently delivers expected and accurate overall polarity computations, emphasizing its reliability and generalizability across diverse datasets.

V. CONCLUSION

In this section, we consolidate our efforts by offering a succinct summary of our work, highlighting the contributions we have made to the domain's knowledge, and outlining prospective avenues for future exploration based on the insights gleaned from our observations.

A. Conclusion

In this endeavor, we have expanded the existing domain knowledge of SA through our primary contributions, showcasing significant advancements:

First, we have adeptly fine-tuned BERT, a pivotal aspect enhancing accuracy within movie reviews SA. Leveraging transfer learning, we coupled BERT with BiLSTM, utilizing BERT-generated word embeddings as inputs for BiLSTM. This amalgamation was instrumental for polarity and fine-grained classification tasks, spanning three-class, four-class, and five-class categorizations across various datasets, i.e., IMDb, MR, SST, and Amazon. Notably, our model consistently outperformed prior works across all classification tasks and datasets.

Moreover, to augment the accuracy of our model for five-class classification, we delved into the impact of employing SMOTE and NLPAUG on SST-5, a notably challenging fine-grained classification benchmark. Intriguingly, SMOTE led to a decrease in accuracy from 58.44% to 58.36%, while NLPAUG remarkably boosted accuracy to 60.48%.

Second, we introduced a heuristic algorithm tailored to the BERT-BiLSTM output vector, dynamically adapting to the specific classification task at hand. Demonstrating its reliability, we confirmed that the original overall polarity aligned perfectly with the computed overall polarity across all datasets. Furthermore, variations within dataset versions exhibited consistent computed overall polarity.

This work marks a pioneering effort, coupling BERT with BiLSTM and applying the resulting model across diverse sentiment classification tasks and benchmark datasets. Notably, it is the first to utilize the output vector of a model for computing overall sentiment polarity. Our exploration not only enhances understanding regarding review polarity but also sheds light on the nuanced shifts in review and overall polarities as classification granularity intensifies. These combined contributions significantly advance the understanding and application of SA within diverse contexts.

B. Future Work

Moving forward, our proposed future endeavors revolve around addressing key challenges in the realm of SA, aiming to enhance model performance and extract deeper insights. One significant area of focus involves exploring strategies to effectively apply accuracy improvement techniques to transformed BERT features, despite the inherent loss of semantic information during their transformation. This endeavor seeks to overcome the limitations posed by the semantic information loss, potentially revolutionizing the effectiveness of these techniques.

Additionally, our exploration will delve into the nuanced contributions of different sentence components to sentiment prediction. This unexplored facet holds immense potential, as the intricate interplay between sentence elements often remains untapped by current methodologies. By dissecting these contributions, we aim to unravel hidden layers of information critical to sentiment prediction, thereby enriching the understanding of sentence structures and their impact on SA.

Furthermore, our future roadmap includes an extensive exploration of alternative pre-trained language models beyond BERT, such as RoBERTa and GPT. This exploration aims to broaden the horizons of our analysis, leveraging the unique capabilities and architectures of these models to potentially enhance SA outcomes. Diversifying our approach by incorporating these SOTA models could unlock new dimensions and avenues for deeper exploration within the field of SA.

ACKNOWLEDGMENT

We would like to express our gratitude to anonymous reviewers for their valuable insights and comments. We also

acknowledge the Data Analytics that are Robust and Trusted (DART) project under the National Science Foundation (NSF) for financially supporting this work. We also extend our gratitude to all individuals who contributed to this work.

REFERENCES

- [1] G. Nkhata, U. Anjun, and J. Zhan, "Sentiment analysis of movie reviews using bert." The Fifteenth International Conference on Information, Process, and Knowledge Management, eKNOW23, 2023.
- [2] 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.
- [3] 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.
- [4] 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.
- [5] Z. Bingyu and N. Arefyev, "The document vectors using cosine similarity revisited," *arXiv preprint arXiv:2205.13357*, 2022.
- [6] W. Guo, X. Liu, S. Wang, H. Gao, A. Sankar, Z. Yang, Q. Guo, L. Zhang, B. Long, B.-C. Chen *et al.*, "Detext: A deep text ranking framework with bert," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020, pp. 2509–2516.
- [7] Y. Liu, "Fine-tune bert for extractive summarization," *arXiv preprint arXiv:1903.10318*, 2019.
- [8] 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.
- [9] W. Yang, Y. Xie, L. Tan, K. Xiong, M. Li, and J. Lin, "Data augmentation for bert fine-tuning in open-domain question answering," *arXiv preprint arXiv:1904.06652*, 2019.
- [10] 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.
- [11] S. Siami-Namini, N. Tavakoli, and A. S. Namin, "The performance of lstm and bilstm in forecasting time series," in *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 2019, pp. 3285–3292.
- [12] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural networks*, vol. 18, no. 5-6, pp. 602–610, 2005.
- [13] S. T. Arasteh, M. Monajem, V. Christlein, P. Heinrich, A. Nicolaou, H. N. Boldaji, M. Lotfinia, and S. Evert, "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.
- [14] M. Anandarajan, C. Hill, and T. Nolan, "Sentiment analysis of movie reviews using r," in *Practical Text Analytics*. Springer, 2019, pp. 193–220.
- [15] 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.
- [16] T. Thongtan and T. Phienthrakul, "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.
- [17] 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.
- [18] 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.
- [19] S. Wang, H. Fang, M. Khabsa, H. Mao, and H. Ma, "Entailment as few-shot learner," *arXiv preprint arXiv:2104.14690*, 2021.
- [20] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," *Advances in neural information processing systems*, vol. 28, 2015.
- [21] D. Shen, M. R. Min, Y. Li, and L. Carin, "Learning context-sensitive convolutional filters for text processing," *arXiv preprint arXiv:1709.08294*, 2017.

- [22] R. Johnson and T. Zhang, "Deep pyramid convolutional neural networks for text categorization," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017, pp. 562–570.
- [23] H. Shirani-Mehr, "Applications of deep learning to sentiment analysis of movie reviews," in *Technical report*. Stanford University, 2014.
- [24] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.
- [25] 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.
- [26] J. D. Bodapati, N. Veeranjanyulu, and S. Shaik, "Sentiment analysis from movie reviews using lstms," *Ingenierie des Systemes d'Information*, vol. 24, no. 1, 2019.
- [27] H. Xu, L. Shu, P. S. Yu, and B. Liu, "Understanding pre-trained bert for aspect-based sentiment analysis," *arXiv preprint arXiv:2011.00169*, 2020.
- [28] X. Li, L. Bing, W. Zhang, and W. Lam, "Exploiting bert for end-to-end aspect-based sentiment analysis," *arXiv preprint arXiv:1910.00883*, 2019.
- [29] 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.
- [30] C. Sun, L. Huang, and X. Qiu, "Utilizing bert for aspect-based sentiment analysis via constructing auxiliary sentence," *arXiv preprint arXiv:1903.09588*, 2019.
- [31] H. Xu, B. Liu, L. Shu, and P. S. Yu, "Bert post-training for review reading comprehension and aspect-based sentiment analysis," *arXiv preprint arXiv:1904.02232*, 2019.
- [32] L. Maltoudoglou, A. Paisios, and H. Papadopoulos, "Bert-based conformal predictor for sentiment analysis," in *Conformal and Probabilistic Prediction and Applications*. PMLR, 2020, pp. 269–284.
- [33] S. Alaparthi and M. Mishra, "Bert: A sentiment analysis odyssey," *Journal of Marketing Analytics*, vol. 9, no. 2, pp. 118–126, 2021.
- [34] S. Baccianella, A. Esuli, F. Sebastiani *et al.*, "Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining," in *Lrec*, vol. 10, no. 2010, 2010, pp. 2200–2204.
- [35] 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.
- [36] F. Mola and R. Siciliano, "A two-stage predictive splitting algorithm in binary segmentation," in *Computational statistics*. Springer, 1992, pp. 179–184.
- [37] S. Rosenthal, N. Farra, and P. Nakov, "Semeval-2017 task 4: Sentiment analysis in twitter," in *Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017)*, 2017, pp. 502–518.
- [38] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.
- [39] G. Rizos, K. Hemker, and B. Schuller, "Augment to prevent: short-text data augmentation in deep learning for hate-speech classification," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 991–1000.
- [40] A. 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.
- [41] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "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.
- [42] B. Pang and L. Lee, "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales," *arXiv preprint cs/0506075*, 2005.
- [43] X. Fang and J. Zhan, "Sentiment analysis using product review data," *Journal of Big Data*, vol. 2, no. 1, pp. 1–14, 2015.
- [44] 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.
- [45] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
- [46] <https://github.com/gnkhatal/Finetuning-BERT-on-Movie-Reviews-Sentiment-Analysis>, 2022, [Online; accessed June-15-2023].
- [47] 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.
- [48] N. R. Chilukuri and C. Eliasmith, "Parallelizing legendre memory unit training," in *International Conference on Machine Learning*. PMLR, 2021, pp. 1898–1907.
- [49] 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.
- [50] 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.
- [51] H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and T. Zhao, "Smart: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization," *arXiv preprint arXiv:1911.03437*, 2019.
- [52] Y. Yang, "Convolutional neural networks with recurrent neural filters," in *Proceedings of Empirical Methods in Natural Language Processing*, 2018.
- [53] B. Wang, "Disconnected recurrent neural networks for text categorization," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 2311–2320.
- [54] S. Brahma, "Improved sentence modeling using suffix bidirectional lstm," *arXiv preprint arXiv:1805.07340*, 2018.
- [55] Z. Sun, C. Fan, Q. Han, X. Sun, Y. Meng, F. Wu, and J. Li, "Self-explaining structures improve nlp models," *arXiv preprint arXiv:2012.01786*, 2020.