ISSN: 0974-5823 Vol. 7 No. 2 February, 2022

International Journal of Mechanical Engineering

SOLVING SENTIMENT ANALYSIS USING TRANSFORMER BASED BERT

[1]Kadiyala Swathi

Designation: Research student

Department: CSE

University: Koneru Lakshmaiah Education Foundation, Vaddeswaram, A.P., India

[2]Dr. V. Krishna Reddy

Designation: Professor

Department: CSE

University: Koneru Lakshmaiah Education Foundation, Vaddeswaram, A.P., India.

ABSTRACT

The most prevalent sequence transduction models use neural networks containing encoders and decoders. The better versions also have an attention mechanism that connects the encoder and decoder. An innovative language representation paradigm, BERT stands for bidirectional Encoder Representations from Transformers. Because it takes into account both the left and right sides of the text at every level, BERT can pre-train deep bidirectional representations from unlabeled text. For a wide variety of tasks, such as answering questions and making inferences about languages, the pre-trained BERT model may be fine-tuned to create cutting-edge models with just one extra output layer. It's easy to use, and it's backed by a tonne of data from the scientific community.

Keywords: *BERT, sentiment analysis*

I.INTRODUCTION

The text must be categorized into one of many specified sentiment categories before it can be considered sentiment categorization. It is a machine learning challenge that requires supervision. Negative and positive sentiments may be classified in binary sentiment categorization. Classification in fine-grained sentiment analysis is broken down into five categories (very negative, negative, neutral, positive, and very positive). A black-box view of a sentiment classifier model is shown in Fig 1.

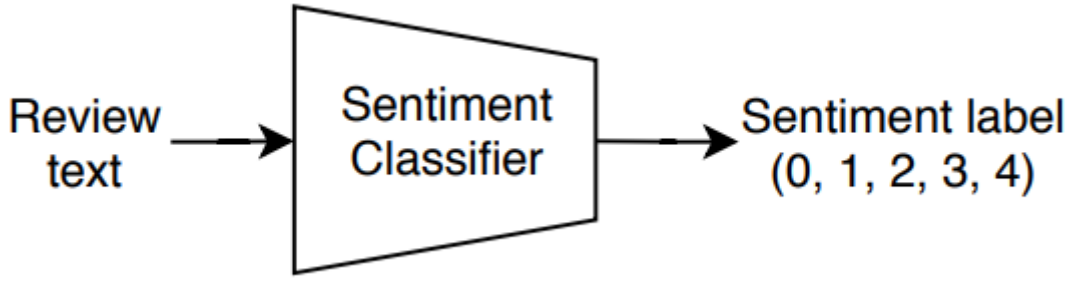


Fig. 1. The input and output of a sentiment classifier in a high-level black-box perspective.

Like any other machine learning model, the sentiment categorization model needs a fixed input, a numeric vector of size. The sequence of ASCII or Unicode characters must be encoded using a fixed-size vector to be meaningful. Numerous statistical and deep learning NLP models have been specifically designed to do this task. Advancements in NLP and other deep learning architectures have been making headlines recently. For NLP, transfer learning has not been used to its full extent, even though it has been the standard practice in computer vision (pre-training and fine-tuning). Neural language models like word vectors, paragraph vectors, and GloVe have revolutionized transfer learning in NLP [1–3]. NLP applications like sentiment analysis and voice recognition have been considerably enhanced by research from the BERT (Bidirectional Encoder Representations from Transformers) project by Google researchers [4].To train and fine-tune the BERT model for the fine-grained sentiment classification issue, this study uses the Stanford Sentiment Treebank (SST) dataset [5-6].

Language model pre-training is beneficial in a wide range of natural language processing tasks. Work at the token level requires models to provide detailed output at the token level. In contrast, work at the sentence level, such as natural language inference and paraphrasing, aims to anticipate the connections between sentences [7] holistically.

There are two approaches to reusing previously learned language representations: feature-based and finetuning. For example, Elmo [6] leverages task-specific architectures that include pre-trained representations as additional features in the feature-based technique—finetuning techniques like OpenAIGPT (Generative Pre-trained Transformer) [4] train downstream tasks by finetuning all pre-trained parameters rather than adding task-specific parameters. The pre-training goal of both methods is the same: to learn broad representations of language using unidirectional language models—both ways.

Existing tactics, we argue, restrict the potential of pretrained representations when it comes to finetuning procedures. Because most language models are unidirectional, only a few architectures may be pre-training. They also use a left-to-right architecture where each token may only pay attention to the previous tokens in its self-attention layers [9]. '[10-11'] Token-level activities, such as answering questions, may be harmed by restrictions of this kind, whereas sentence-level tasks, such as writing, may benefit [8].

Bidirectional Encoder Representations from Transformers (BERT) has been proposed in the research, which enhances fine-tuning-based methods.

BERT utilizes a "masked language model" (MLM) pre-training target for the previously mentioned unidirectionality limitation, which was inspired by the Cloze issue. After randomly masking particular tokens from the input, the model aims to estimate the masked word's original vocabulary id based on its context. Training a language model that reads from top to bottom instead of left to right is one option [9]. Transform utilizing the MLM aim to integrate both contexts in your depiction of the subject. We combine a "next sentence prediction" task with a masked language model to train text-pair representations. The following is a list of the paper's accomplishments:

- Bidirectional pre-training is needed for language representations. In contrast to Vaswani and colleagues [9], BERT leverages masked language models to pre-train deep bidirectional representations. A shallow concatenation of LMs trained in opposing directions, from right to left to left, is also used by Peters et al. [6].
- For example, we show that pre-trained representations reduce the need for many task-specific designs. The first finetuning-based representation model outperforms several task-specific designs in terms of sentence and token level performance on a wide variety of tasks. Eleven NLP tasks may be enhanced with BERT.

II.BERT

Pre-trained language models are taught on previously unannotated training data, subsequently utilized to provide context to words. Pre-trained language models take into account both the left and right sides of a word's context simultaneously [10-11]. Even though the principle is simple, it is used in various NLP applications, including sentiment analysis and question and answer systems. BERT, in contrast to previous models like Elmo [6], can extract more context information from a sequence than can train left and right separately. In order to pre-train BERT, masked language model masks are used (MLM). The purpose of MLM is to hide a random word in a phrase with a very low likelihood of occurrence. The token [MASK] is used instead of the original term when a word is masked. A transformer is used to look at what is going on around a masked word from both sides to aid with the prediction. Besides left and proper context extraction using MLM, BERT's primary goal is next-sentence prediction, distinct from earlier work.

A. Input Representation

Workpiece tokenization is the first step in the BERT model's text input processing. Tokens are generated. As a result, each of which represents a single word. Classifier token [CLS] and separation token [SEP] are two other specialized tokens inserted into the set of tokens to mark the beginning and end of a sentence, respectively. A [SEP] token will be used to demarcate the comparison of two sets of sentences using BERT. In the encoder layer, this collection of tokens is passed through three distinct embedding layers, each with the exact dimensions: the layer of token embedding, segment embedding, and position embedding [11].

B. Transformers

Sequence modeling has previously relied on the sequence-to-sequence (seq2seq) architecture [12], as well as approaches like recurrent neural networks (RNNs) [13] and long short-term memory (LSTM) [14]. Instead of using RNNs, transformers use attention mechanics [9] to choose the most significant sequence in each computation phase. The encoder converts the input into a higher-dimensional space vector and feeds the decoder the necessary keywords. With this knowledge, the decoder has an advantage since it knows which sequences and keywords contribute more to the phrase's meaning.

C. Sentence Pair Classifier Task

A pre-trained version of BERT was first developed to facilitate the finetuning of the model for a given job without making significant changes in the model and parameters. When making a model more task-specific, just one extra output layer was needed. We are looking for semantic [15] relationships between two phrases for this work. As input, the model accepts two text files and produces the relationship between the phrases. You may see how well a model understands natural language and inferences about whole phrases using tasks like these. GLUE [16] is a benchmark for evaluating natural language comprehension on models. It includes multiple tasks such as multi-genre natural language inference (MNLI) [15], the semantic textual similarity benchmark (STS-B) [17], and Microsoft research paraphrase corpus (MRPC) [18].

D. Pre-training BERT

To train BERT, we do not employ typical left-to-right or right-to-left language models, unlike Peters et al. [19] and Radford et al [20]. Instead, we used two unsupervised tasks to pre-train BERT, which are detailed in the next section.

Task #1: LM disguised herself as a woman in order to evade capture. When it comes to the strength of the model, intuition tells us that deep bidirectional models are more powerful than the more shallow concatenation of the two. Traditionally, conditional [21] language models can only be trained from left to right or right to left. This is because bidirectionalconditioning would allow each word to indirectly "see itself," and the model could trivially predict the target word in a multilayered context without any difficulty. To train a deep bidirectional representation, we first mask a random subset of the input tokens and then predict the masked tokens. This technique is referred to as a "Cloze task" in the literature, although we prefer the term "maskedLM" (MLM). The final hidden vectors corresponding to the mask tokens are sent into the output softmax, just as in a conventional LM. Every time we conduct an experiment, 15% of all WordPiece tokens are randomly hidden. Like denoising auto-encoders, we anticipate the masked words rather than recreating the whole input. As a result of fine-tuning, we end up with a model that isn't equipped for both training orientations. Using the actual[MASK] token to replace "masked" words is not usually employed. The training data generator randomly picks 15% of the token placements for prediction purposes. Only one token is chosen at random, and 80% of the time the i-th token is changed. Only a random token is utilised the rest of the time. The unaltered i-th token makes up the last 10%. To predict the original token with cross entropy loss,Ti will be used in this manner. Comparing numerous versions of this method, Appendix C.2 examines.

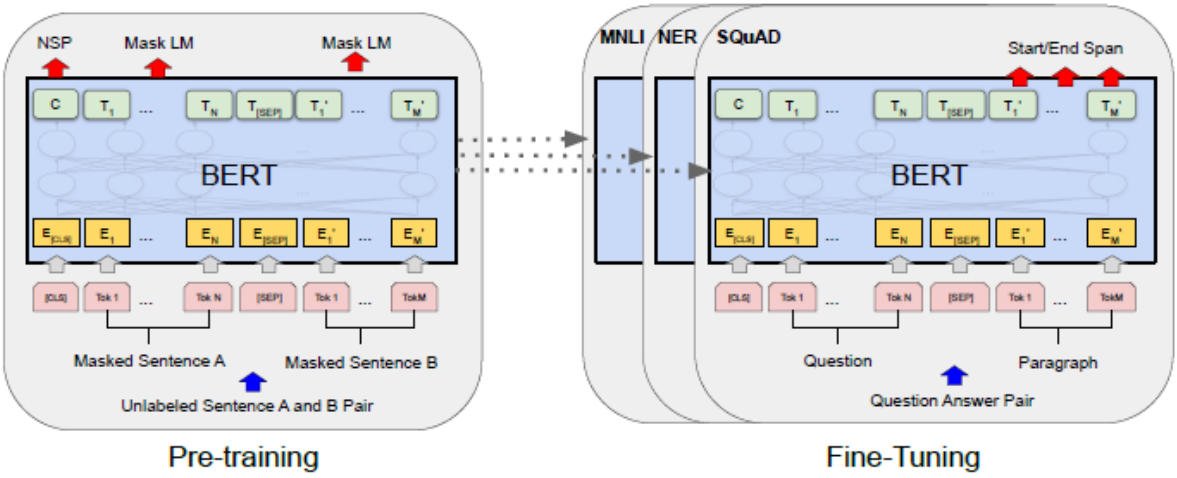


Figure 1: Detailed instructions for BERT's first training and fine-tuning. Architectures are employed for both pre-training and fine-tuning except for the output layers. Pre-trained model parameters are used to set up models for a variety of downstream activities. All of the parameters are fine-tuned during fine-tuning. Before each input example, [CLS] is a special symbol, and [SEP] is a token used to separate questions and answers.

Task #2: Foretelling of the Future Word (NSP) QuestionAnswering (QA) and Natural Language Inference (NLI) are examples of downstream activities that are not immediately captured by language modelling. In order to train a model that understands sentence dependencies, it is required to pre-train for a binarized next sentence prediction task that can be simply generated from any monolingual corpus. A random text from the corpus is used in 50 percent of pre-training examples, and an actual next phrase (marked as IsNext) is used in the other 50 percent. To be specific (labeled as NotNext). The following sentence may be predicted using C, as shown in Figure 1. (NSP). 5

The NSP task and the representation learning objectives of Jernite et al. [22] and Logeswaran and Lee [23] have a number of characteristics. BERT communicates all parameters to the end-task model parameters, while just sentence embeddings were previously sent. Preliminary training information Literature on language model pre-training is the basis of the pre-training approach. In the BooksCorpus pre-training corpus and the English Wikipedia, there are almost 800 million words. (24). (2,500M words). Wikipedia only uses text fragments from the source material. The BillionWord Benchmark's jumbled sentence-level corpus is not a suitable source for long continuous sequences[25].

E. Fine-tuning BERT

By simply swapping out the appropriate inputs and outputs in the Transformer's self-attention mechanism, Bert may represent many downstream tasks, whether they involve a single text or text pairs. Like Parikh et al [26] and Seo et al [27], BERT uses the self-attention mechanism to merge these two stages. Each job's parameters are fine-tuned end-to-end by entering the task-specific inputs and outputs into BERT. Pre-training A and B may be equated to (1) sentence pairs in paraphrase, (2) hypotheses-premise pairings in entailment, (3) question-passage couples in question answering, and(4) a degenerate text-?" At the input level, pair text classification or sequence tagging. An output layer for entailment or sentiment analysis is fed the [CLS] representation, while token representations are sent into an output layer for token tasks, such as tagging or answering questions. Pre-training is more expensive than a one-on-one fine-tuning session. In less than an hour on a Cloud TPU or a few hours on a GPU, you may duplicate the results of the study using the identical pre-trained model. 7.

III.ASPECT-BASED SENTIMENT ANALYSIS

ABSA is a far more involved process than typical text-level sentiment analysis. It focuses on recognizing the traits or features of an entity described in a text and the sentiments conveyed about each one. SemEval-2014 [28] supplied a dataset of annotated reviews on restaurants and computers, and ABSA was first offered. Until SemEval-2015 [29], no comprehensive reviews were available for the ABSA job in SemEval-2014, and the dataset for SemEval-2016 was almost unchanged from SemEval-2015. Customer reviews were analyzed as part of the SemEval-2016 ABSA job to determine how certain aspects were perceived. The purpose of SemEval-2016 is to solve the following tasks given a text review on a specific subject from the dataset (e.g., laptop, restaurant):

The purpose of aspect category classification is to identify the topic and aspect pair on which an opinion is expressed in the text. Each domain has a preset collection of topic kinds and aspects from which to choose the subject and aspect (e.g., LAPTOP, RESTAURANT, FOOD).

For each entity-aspect combination being assessed, the OTE task entails locating and extracting the linguistic phrase associated with the examined entity in the text input. The OTE is determined by the sequence's only two offsets, one at the beginning and one at the conclusion. The value "NULL" is returned if no specific entity is given**.**

The goal of sentiment polarity classification is to predict the sentiment polarity for each subject and aspect pair that the user has identified. Positive, negative, neutral, and conflict polarity are all values in the set sentiment polarity}

A. ABSA without BERT

The top entries employed SVM and conditional random field classifiers in the SemEval-2016 ABSA competitions, utilizing machine learning [30] methods. However, even though deep learning models have been demonstrated to perform well in sentiment analysis [31], deep learning submissions failed miserably in 2016. With SVM features, contextualized word representations or word lists formed by removing nouns and adjectives from datasets were often utilized [32].

B. ABSA with BERT

Since being trained on a significant text volume, BERT has shown strong performance on NLP tasks. Post-Instruction has been demonstrated to boost performance on tests like ABSA by providing extra training on Review text. In order to complete an ABSA assignment, the Post-Training paper developed ABSA as a question-and-answer issue using the "review reading comprehension" machine reading comprehension approach for reviews. A performance improvement has been seen when ABSA is solved as a classification problem for pairs of sentences using BERT by creating an auxiliary sentence.

IV.REVIEW OF LITERATURE

Much work has been done to improve NLP's ability to perform tasks like sentiment categorization. This technique has received excellent attention since massive public datasets like the IMDB movie review dataset [33] exist for binary sentiment classification. Only a few of the most important deep learning NLP algorithms for sentiment categorization are covered in this section.

To begin the process of sentiment categorization, a text is first encoded using an embedding algorithm. After tokenization and stemming, there are only a limited amount of words in the lexicon. Therefore, researchers initially tried to learn word embeddings. Mikolov et al. [1] presented the first promising language model. They used massive unlabeled texts to develop continuous semantic representations of words that could be finetuned for subsequent trials. For efficient learning of semantic word embeddings, Pennington and colleagues employed a co-occurrence matrix and trained solely on nonzero entries. Words were broken down into character n-grams by Bojanowski et al. [33] for more accessible training and a reduced vocabulary.

The next stage is to merge many word vectors into a single document vector with a set amount of words per line. The simplest method is to add up the numbers; however, this ignores the sequence of the words and results in inaccurate answers. Recursive neural networks were utilized by Tai et al. [34] to generate vector representations of sentences based on the natural language sentences' inherent tree structure. Socher et al. [35] presented a compositionality function based on tensors to improve interactions between child nodes in recursive networks. Stanford Sentiment Treebank (SST) was also established for fine-grained sentiment categorization by the researchers. Long short-term memory (LSTM) networks were used by Tai et al. [34], while convolutional neural networks were used by Kim [30] to classify sentiment.

Using any of the approaches outlined above, [36] each word in the vocabulary will have a single word embedding since there is no context. A "bank deposit" and a "riverbank" would use the same phrase to describe their respective services. Researchers have recently focused on embeddings in context. Context-sensitive characteristics were extracted from left-to-right and right-to-left LSTMs using Peters et al. [37]'s LSTM-based language model. To train deep bidirectional representations from unlabeled texts using an attention-based Transformer architecture, Devlin et al. [4] proposed BERT (Bidirectional Encoder Representations from Transformers). Because it does not rely on sequential or recurrent connections, its design achieves cutting-edge performance on many NLP applications. It allows for a high degree of parallelism.

BERT is the first model for representing both bidirectional and unsupervised language. Before BERT, various pre-trained language models used bidirectional unsupervised learning. Another is Elmo [37], which deals with word representations based on context. Word representations are trained from left to right and right to left using a Recurrent Neural Network (RNN) termed Long Short Term Memory (LSTM) [38]. BERT does not use LSTM but instead uses transformers [39] which are attention-based strategies that do not depend on repetition for the properties of word context.

A. Unsupervised Feature-based Approaches

Non-neural [28] and neural [1] approaches have been used for decades to learn generally applicable representations of words. Modern NLP systems rely on pre-trained word embeddings, which provide considerable advantages over embeddings that must be learned from scratch. For pretraining word embedding vectors, language modeling and correct/incorrect word discrimination targets have been utilized ([29, 30].).

In the past, these methods have been used to finer granularities, such as embedding sentences [31] or embedding paragraphs. Objectives have previously been used to rate candidate following sentences [32] in training sentence representations. Denoising autoencoder-derived aims [40] or left-to-right creation of future sentence words given a representation of the previous sentence

Elmo and its predecessor [6] extend conventional word embedding research into a new path. The language models used to extract context-sensitive traits are left-to-right and right-to-left. Each token's contextual representations are made up of its left and correct representations combined.

ELMoadvances several major NLP benchmarks [6], including question answering [33], sentiment analysis[41-44], and named entity recognition] using context-sensitive word embeddings in combination with task-specific architectures. Melamud and colleagues used LSTMs to challenge participants to anticipate a word from both the left and correct contexts to develop contextual representations. In contrast to Elmo's technique, which is heavily bidirectional, this one relies on features. Fedus et al. [45] suggest using the cloze task to improve text generation models' robustness.

B. Unsupervised Finetuning Approaches

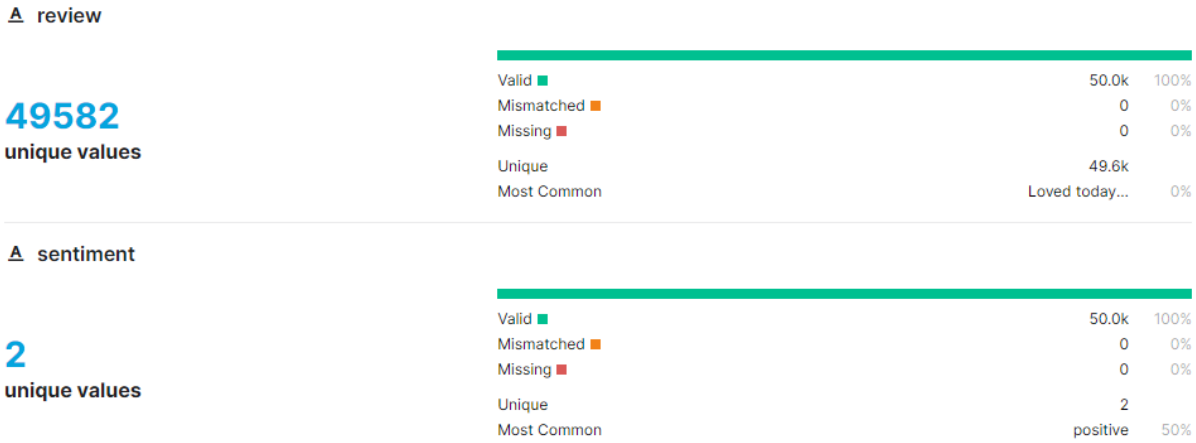
The initial step in this technique uses solely pre-trained word embedding parameters from unlabeled text, similar to the approach based on features. Pre-trained and finetuned encoders that provide contextual token representations for supervised downstream tasks have recently been developed. These methods benefit from requiring just a few parameters to be learned from the beginning. OpenAI GPT [46] previously achieved state-of-the-art results on several GLUE problems because of this advantage. Pre-training these models has used left-to-right language modeling and auto-encoder aims.

C. Transfer Learning from Supervised Data

Natural language inference [47] and machine translation have also shown successful transfer from supervised tasks with massive datasets. Researchers in computer vision have found that transfer learning from large pre-trained models may be a powerful tool to get the most out of huge pre-trained models.

V.DATASET

The IMDB Dataset from Kaggle contains more than 50,000 movie reviews for NLP. There are two columns in the CSV file: review and sentiment. A review may be either positively or negatively polarised in terms of its polarity. A binary classification challenge arises in the supervised learning context.



A. Data preprocessing

How to generate appropriate BERT input embeddings is a key component of data prepossessing, though. Using the functions in the following code block, you may convert a review into one of three embedding formats and prepare inputs for use by the model during training and testing. (1). Sequences may be as long as 500 characters.

B. Modelling

We use BERT BASE as the pre-trained model to develop a state-of-the-art NLP model for sentiment analysis. We add a layer with 768 ReLU activation units and a dropout of 0.1, which is completely linked. It's also possible to use the TensorFlow 1-BERT Tutorial's two softmax functions as an output layer.

C. Model training

In order to minimise the categorical crossentropy loss, we utilise Adam optimizer with a learning rate of 2e-5 and these hyperparameters are same to TensorFlow 1-BERT Tutorial. The advantages of improved model performance exceed the

computational costs, even if the fine-tuning step of model training takes time. In Colab Pro, using a single GPU, the model training takes around 47 minutes each epoch. Awesome! Just after a single period of training, the nlp model attained a 94% accuracy rate [47].

VI.RESULTS AND CONCLUSION

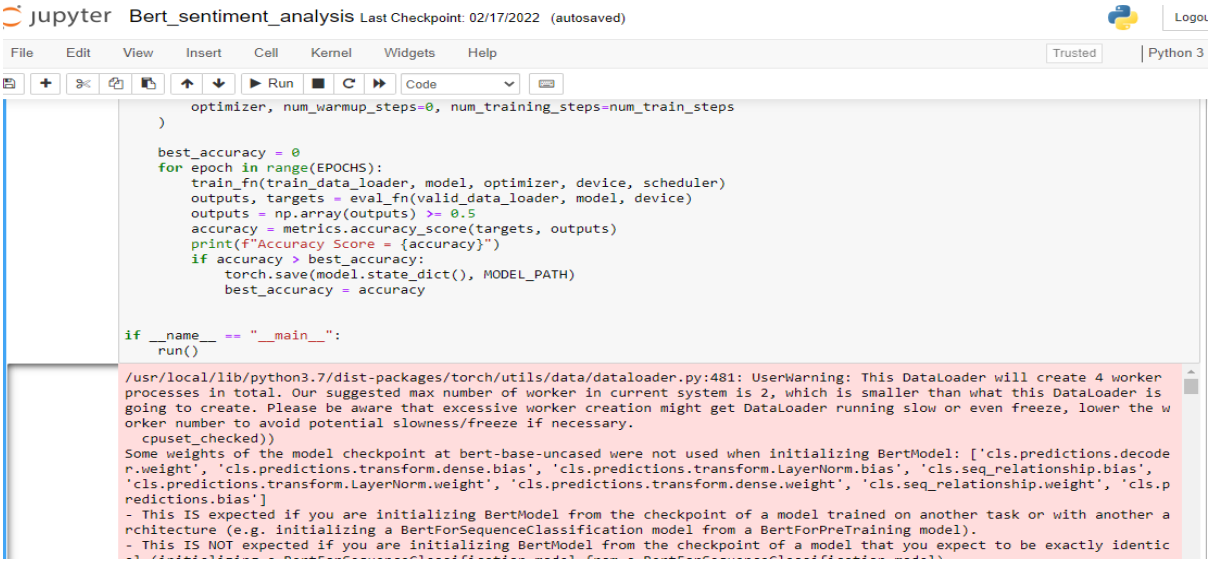
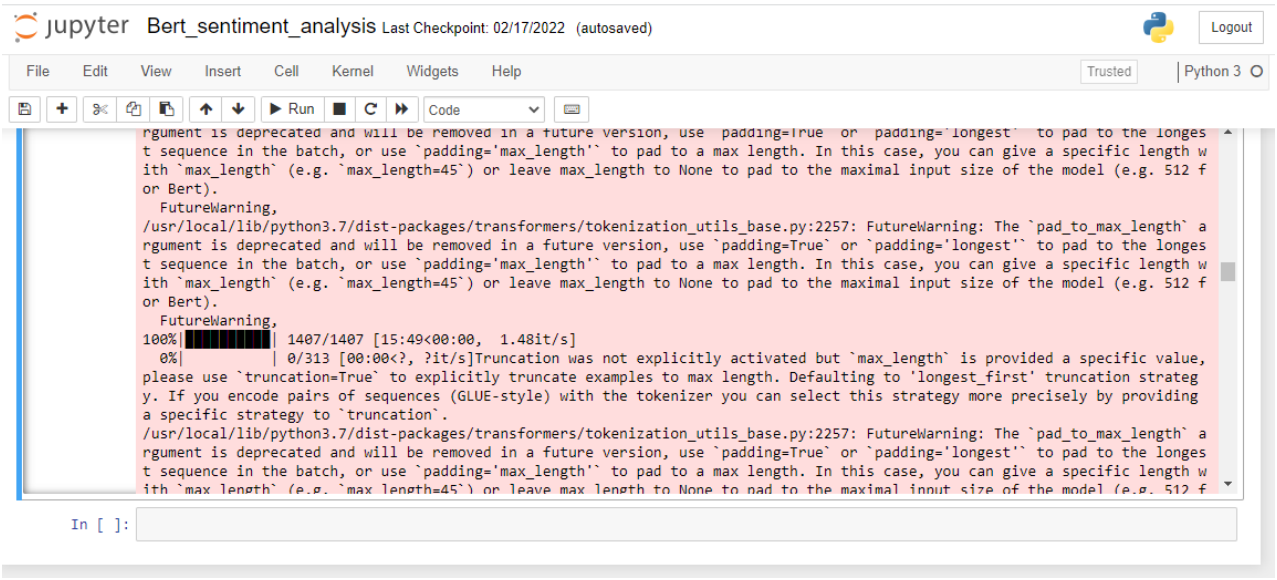
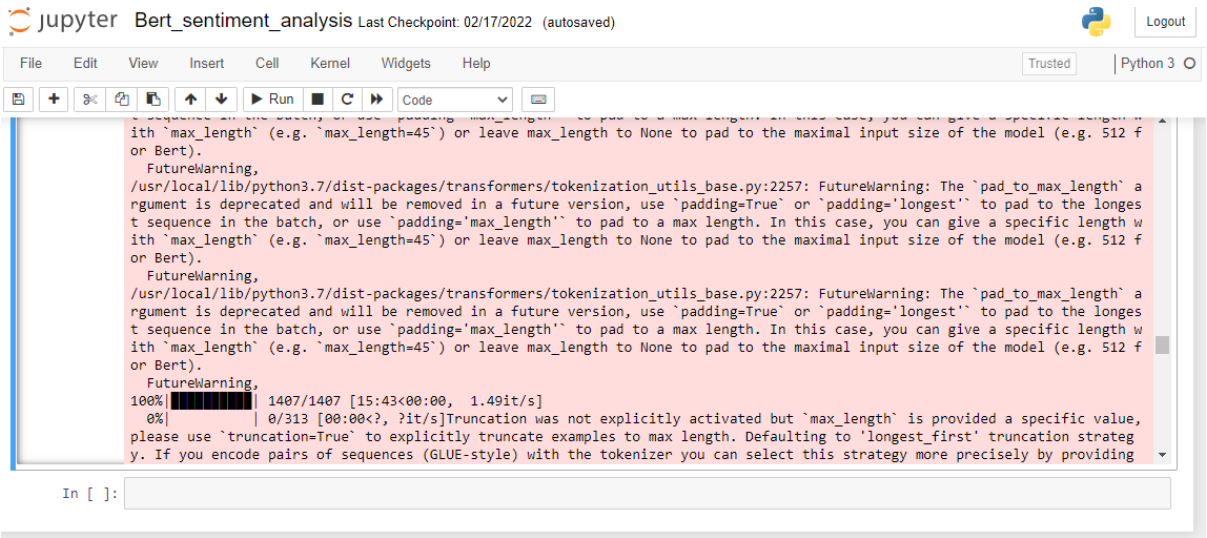
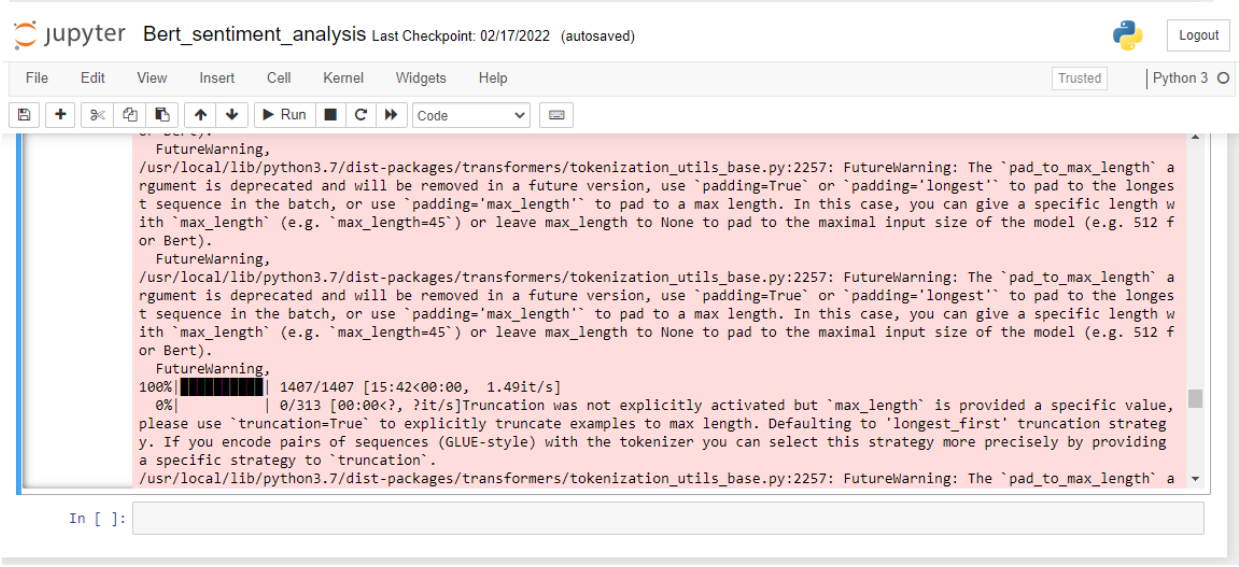
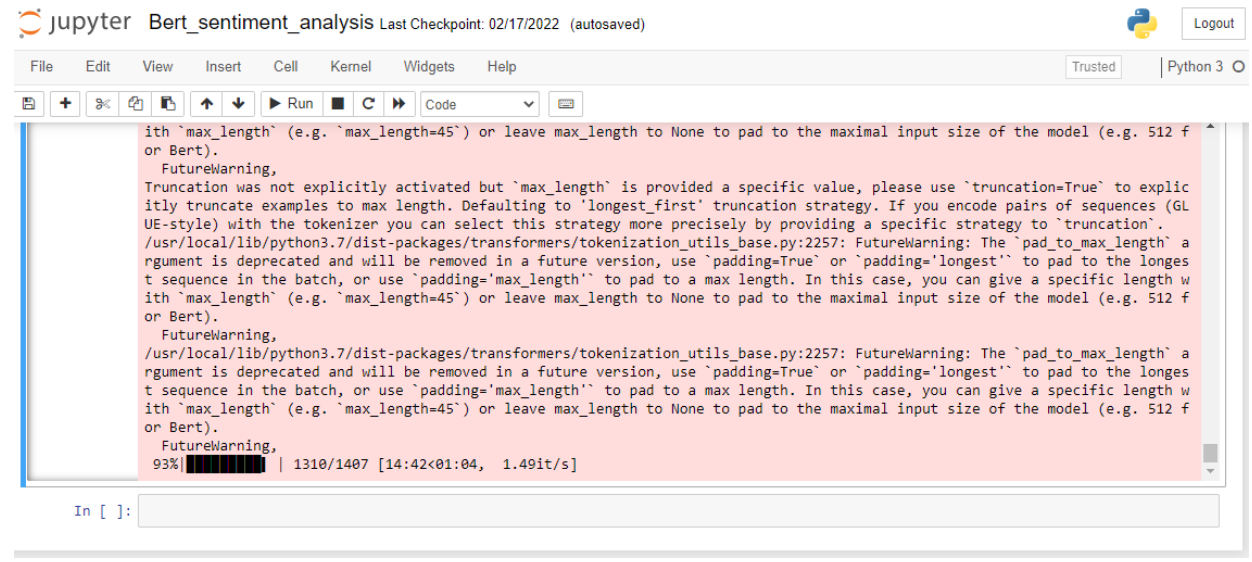


Fig3:Adding more no of epochs increases the better accuracy of BERT









VII. CONCLUSION

To do the fine-grained sentimental analysis on the SST dataset, we used a BERT model that had already been trained. Recursive, convolutional, and convolutional neural networks were all beaten by our model's simple downstream architecture. Our findings indicate how deep contextual language models like BERT enable NLP's transfer learning abilities.

We tried finetuning the model for the sentiment classifier to establish a relationship between an aspect and a text and make the model learn when a contextual representation indicated a sentiment context. Our combination model, which uses just one phrase

pair classification model, can identify both aspect and sentiment. Compared to earlier state-of-the-art results for sentiment categorization based on aspects, a combined model exceeds.

REFERENCES

[1]. Mikolov, T. K. Chen, K. Corrado, G. S. and Dean, J. (2013). "Efficient estimation of word representations in vector space," CoRR, vol. abs/1301.3781, 2013. [

[2]. Le, Q. and Mikolov, T. (2014). "Distributed representations of sentences and documents," in International Conference on Machine Learning, 2014, pp. 1188–1196.

[3]. Pennington, J. Socher, R. and Manning, C. (2014). "GloVe: Global vectors for word representation," in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532–1543.

[4]. Devlin, J. Chang, M.-W. Lee, K. and Toutanova, K. (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding," in NAACLHLT, 2018.

[5]. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and Polosukhin, A. (2017). "Attention is all you need," in Advances in Neural Information Processing Systems, 2017, pp. 5998–6008

[6]. Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. (2018). Deep contextualized word representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

[7]. Adina Williams, Nikita Nangia, and Samuel Bowman. (2018). A broad-coverage challenge corpus for sentence understanding through inference. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

[8]. William B. Dolan and Chris Brockett. (2005). Automatically constructing a corpus of sentential paraphrases. In Proceedings of the Third International Workshop on Paraphrasing (IWP2005).

[9]. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and IlliaPolosukhin. (2017). Attention is all you need.

[10]. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

[11]. Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, ukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, HidetoKazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation.

[12]. Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. (2014). Sequence to sequence learning with neural networks. In Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, NIPS'14, pages 3104–3112, Cambridge, MA, USA. MIT Press.

[13]. Alex Graves. (2013). Generating sequences with recurrent neural networks. CoRR, abs/1308.0850

[14]. Sepp Hochreiter and Jurgen Schmidhuber. (1997). Long ¨ short-term memory. Neural Comput., 9(8):1735– 1780.

[15]. Alexis Conneau, DouweKiela, Holger Schwenk, Lo¨ıcBarrault, and Antoine Bordes. (2017). Supervised learning of universal sentence representations from natural language inference data. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics

[16]. Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. (2018). GLUE: A multi-task benchmark and analysis platform for natural language understanding. In Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Net works for NLP, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

[17]. Daniel Cer, Mona Diab, EnekoAgirre, Inigo Lopez- ˜ Gazpio, and Lucia Specia. (2017). SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

[18]. William B. Dolan and Chris Brockett. (2005). Automatically constructing a corpus of sentential paraphrases. In Proceedings of the Third International Workshop on Paraphrasing (IWP2005).

[19]. Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018a. Deep contextualized word representations. In NAACL

[20]. Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. (2018). Improving language understanding with unsupervised learning. Technical report, OpenAI.

[21]. Wilson L Taylor. (1953). Cloze procedure: A new tool for measuring readability. Journalism Bulletin, 30(4):415–433.

[22]. YacineJernite, Samuel R. Bowman, and David Sontag. (2017). Discourse-based objectives for fast unsupervised sentence representation learning. CoRR, abs/1705.00557.

[23]. LajanugenLogeswaran and Honglak Lee. (2018). An efficient framework for learning sentence representations. In International Conference on Learning Representations.

[24]. Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In Proceedings of the IEEE international conference on computer vision, pages 19–27
[25]. CiprianChelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. (2013). One billion word benchmark for measuring progress in statistical language modeling. arXiv preprint arXiv:1312.3005
[26]. Ankur P Parikh, Oscar Tackstr ¨ om, Dipanjan Das, and ¨ Jakob Uszkoreit. (2016). A decomposable attention model for natural language inference. In EMNLP.
[27]. MinjoonSeo, Aniruddha Kembhavi, Ali Farhadi, and HannanehHajishirzi. (2017). Bidirectional attention flow for machine comprehension. In ICLR.
[28]. Maria Pontiki, DimitriosGalanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. (2014). Semeval-2014 task 4: Aspect based sentiment analysis. Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014), pages 27–35
[29]. Thorsten Joachims. (1998). Text categorization with support vector machines: Learning with many relevant features. In Proceedings of the 10th European Conference on Machine Learning, ECML'98, pages 137–142, Berlin, Heidelberg. Springer-Verlag
[30]. Yoon Kim. (2014). Convolutional neural networks for sentence classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1746–1751, Doha, Qatar. Association for Computational Linguistics
[31]. Jeffrey Pennington, Richard Socher, and Christopher Manning. (2014). Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1532–1543, Doha, Qatar. Association for Computational Linguistics
[32]. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C Potts, A. L. (2011). "Learning word vectors for sentiment analysis," in Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, June 2011, pp. 142–150.
[33]. Bojanowski, E. Grave, A. Joulin, and Mikolov, P. (2017). "Enriching word vectors with subword information," Transactions of the Association for Computational Linguistics, vol. 5, pp. 135–146, 2017.
[34]. Tai, K. S. Socher, R. and Manning, C. D. (2015). "Improved semantic representations from tree-structured long short-term memory networks," in Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 2015, pp. 1556–1566
[35]. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, R. and Potts, C. (2013). "Recursive deep models for semantic compositionality over a sentiment treebank," in Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2013, pp. 1631–1642
[36]. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, M. E. and Zettlemoyer, L. (2018). "Deep contextualized word representations," arXiv preprint arXiv:1802.05365, 2018
[37]. Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. (1992). Class-based n-gram models of natural language. Computational linguistics, 18(4):467–479.
[38]. Andriy Mnih and Geoffrey E Hinton. (2009). A scalable hierarchical distributed language model. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, Advances in Neural Information Processing Systems 21, pages 1081–1088. Curran Associates, Inc.
[39]. Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In Advances in Neural Information Processing Systems 26, pages 3111–3119. Curran Associates, Inc.
[40]. Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. (2015). Skip-thought vectors. In Advances in neural information processing systems, pages 3294–3302.
[41]. LajanugenLogeswaran and Honglak Lee. (2018). An efficient framework for learning sentence representations. In International Conference on Learning Representations
[42]. Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. (2016). Squad: 100,000+ questions for machine comprehension of text. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 2383–2392
[43]. Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In Proceedings of the 2013 conference on empirical methods in natural language processing, pages 1631–1642.
[44]. Oren Melamud, Jacob Goldberger, and Ido Dagan. (2016). context2vec: Learning generic context embedding with bidirectional LSTM. In CoNLL.
[45]. William Fedus, Ian Goodfellow, and Andrew M Dai. (2018). Maskgan: Better text generation via filling in the .arXiv preprint arXiv:1801.07736.
[46]. Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. (2018). Improving language understanding with unsupervised learning. Technical report, OpenAI.
[47]. Alexis Conneau, DouweKiela, Holger Schwenk, Lo¨ıcBarrault, and Antoine Bordes. (2017). Supervised learning of universal sentence representations from natural language inference data. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.
[48]. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. FeiFei. (2009). ImageNet: A Large-Scale Hierarchical Image Database. In CVPR09.