Efficient Transformer Based Sentiment Classification Models

Leeja Mathew[1*], Bindu V R[2]
[1*]Research Scholar, School of Computer Sciences, Mahatma Gandhi University, Kerala, India
2Professor& Dean, School of Computer Sciences, Mahatma Gandhi University, Kerala, India
* corresponding author's e-mail : leejarejibpc@gmail.com

Keywords: Attention, Deep Learning, Pre-Trained Models, Sentiment Analysis, Transformers, Simple Transformers.

Received: August 8, 2022

Recently, transformer models have gained significance as a state-of-the art technique for sentiment prediction based on text. Attention mechanism of transformer model speeds up the training process by allowing modelling of dependencies without regard to their distance in the input or output sequences. There are two types of transformer models – transformer base models and transformer large models. Since the implementation of large transformer models need better hardware and more training time, we propose new simpler models or weak learners with lower training time for sentiment classification in this work. These models enhance the speed of performance without compromising the classification accuracy. The proposed Efficient Transformer-based Sentiment Classification (ETSC) models are built by setting configuration of large models as minimum, shuffling dataset randomly and experimenting with various percentages of training data. Early stopping and smaller batch size in training techniques improve the accuracy of the proposed model. The proposed models exhibit promising performance in comparison with existing transformer-based sentiment classification models in terms of speed and accuracy.

Povzetek: V članku je opisan nov način uporabe klasifikatorjev za jezikovne pretvornike.

1 Introduction

Sentiment Analysis (SA) exists in the category of text mining, which uses natural language processing (NLP) and related computer technology to extract or classify emotions in text [1]. Sentiment analysis has a very wide range of applications in customer industry where user preference for different commodities can be obtained through the analysis of product reviews to help companies adjust their sales strategies and make suitable decisions, in analyzing hot topics which attract users on social media [2]. Sentiment analysis methods can be categorized into machine learning methods, deep learning methods and transfer learning methods[3]. In addition to these, novel language embedding models named transformers are used for sentiment classification.

Transformer is the best performing pre-trained model (PTM) which consists of encoder and decoder through an attention mechanism[4]. BERT (Bidirectional Encoder Representations from Transformers)[5] is the most popular pre-trained model and is the base of all transformer models. It follows a complex deep neural network architecture. This model is designed to pre-train deep bidirectional representations from unlabeled text on both left and right contexts in all layers. As a result, this model can be fine-tuned and used for a wide range of tasks such as question answering, language inference, sentiment analysis etc. Our proposed model is based on aspect-based sentiment analysis (ABSA). RoBERTa is such a PTM in ABSA [6]. BERT model has memory constrains and need longer training time such as 16 cloud TPU, 4 days; a modified lighter version of BERT named, ALBERT [7], is also included in our experiment. However, implementation of large transformer models is very complex. Introducing Simple Transformers is the solution to this problem.

Simple Transformer models or lighter versions of transformer models can be built for NLP tasks such as Text Classification, Named Entity Recognition, Question Answering, Language Modelling etc. [8]. The procedure includes initializing a task-specific model, training the model, evaluating the model and testing the model. Our proposed models are built using this concept, by modifying the default structure of transformer-large models BERT-large, RoBERTa-large and ALBERT-large; and is grouped under a single name Efficient Transformer-based Sentiment Classification (ETSC) models.

Transformers and pre-trained models have an important role in NLP[9]. These models are accessible to researchers and end-users. A pre-trained model is a model that was trained on a large benchmark dataset to solve a problem of similar down sampled dataset which reduces the computational cost of training time [10]. The following are some of the state-of-the art works done in sentiment analysis based on transformer models.

Kant Neel et al. [11] have done sentiment analysis for multi-emotion in NLP with valuable use cases on real world. Kumar Varun et al.[12] conducted a study on different transformer based pre-trained models and provide an effective way to condition these models for data augmentation. Xu Hu et al.[13] analyze the hidden

representations in BERT model in aspect-based sentiment analysis.

Munikar M et al. [14] demonstrate the effectiveness of transfer learning in NLP using BERT model by building a simple architecture with a dropout regularization and softmax layers on top of the model. Zhao Mengjie et al. [15] present an efficient method for pre-trained language models by applying new masking scheme. Naseem Usman et al.[16] developed a transformer-based model for sentiment analysis using twitter data by combining BERT embedding with earlier pre-trained models. Kaiser Lukasz Mieczyslaw et al. [17] introduce a method to extend sequence models using discrete latent variables in the decoding phase of language translation model.

Tang Tiancheng et al. [18] investigate and deal with the problem of the unbalanced distribution of emotion in sentiment analysis using BERT model. Truşcă, M.M.et al. [19] extend the state-of-the-art hybrid approach for aspect-based sentiment analysis by replacing non-contextual word embedding with deep contextual word embeddings and add hierarchical attention layer for increasing accuracy.

Wang Chenguang et al. [20] explore effective architecture for language model by including additional LSTM layer for improving computation efficiency. Farahani Mehrdad et al. [21] propose a monolingual BERT for Persian language which outperforms all other multilingual models.

Cheng Xingyi et al. [22] propose a semi supervised method for the aspect-term sentiment analysis problem by using the Variational Autoencoder based on Transformer. Biesialska Katarzyna et al. [23] introduced a universal multilingual sentiment classifier based on Polish and German languages. Voita Elena et al. [24] evaluate machine translation by pruning heads using a method based on stochastic gate and a differentiable relaxation of the L0 regularizer.

Hoang M et al. [25] analyse aspect-based sentiment analysis using BERT model in SemEval 2015 and SemEval 2016 dataset. The authors showed the potential of using contextual word representation using BERT.

Xu Q et al.[26] predict sentiment polarity based on aspect term instead of considering entire sentence polarity. Instead of sequence models, they propose a multi-attention network which employs a transformer encoder which reduces training time. In this paper, the authors modify only multi-head attention mechanism of transformer model for their aspect level sentiment analysis. These literatures are summed up into Table 1.

Table 1. Summary of related works result

Authors	Models	Methods	Accuracy
Munikar M et al. [14]	BERT-base BERT-large	BERT with dropout reguralization	94.0 94.7
Zhao Mengjie et al. [15]	BERT-base RoBERTa-base DistillBERT	Alternative masking	93.3 94.0 91.6
Naseem Usman et al.[16]	Transformers	Various embedding	94.6
Tang Tiancheng et al. [18]	BERT-base	Data augmentation method	77.3
Truşcă, M.M. et al. [19]	BERT-base	Deep contextual word embedding and hierarchical attention	89.2
Wang Chenguang et al. [20]	BERT	Adding LSTM layer for embedding	53.82
Cheng Xingyi et al. [22]	Transformer	Adding Variational autoencoder	78.34
Hoang M et al. [25]	BERT	Aspect based-sentence level	79.9
Xu Q et al.[26]	Transformer	Multi attention network	85.87

We have in an earlier work proposed a model named TSC for an efficient sentiment classification using Transformer models, choosing only transformer base models for classification [27]. Our proposed models based on modified form of transformer large models reduce training time in comparison to TSC model.

The rest of the paper is organized into four sections. In section 2, the proposed method is explained. In section 3, the results are presented and analyzed. Finally, section 4 includes conclusion.

2 The proposed classification method using simple transformer models

Our proposed ETSC model is a modified form of transformer large models – BERT-large, RoBERTa-large and ALBERT-large with reduced number of attention heads and hidden layers described as follows.

2.1 The transformer models

2.1.1 BERT

BERT stands for Bidirectional Encoder Representations from Transformers which is a new language representation model [5]. Unlike recent language representation models [28], BERT is designed in bidirectional context to pretrain deep bidirectional representations from unlabelled text in all layers. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer named fully connected classification layer, to create new models. BERT is conceptually simple and empirically powerful. Pre-training in BERT is done in two ways. i) Randomly masked with [MASK] token. This method is known as MLM (Masked Language Model). For example, consider the Sentence: Who was Jim Henson? Jim Henson was a puppeteer. After Masking: [CLS] Who was Jim Henson? [SEP] Jim [MASK] was a puppeteer. [SEP]. ii) Next Sentence Prediction: - In this method, the model receives pairs of sentences as input and learns to predict whether the second sentence from this pair is the successive

sentence in the original document. The sentences are arranged by taking 50% of the inputs as a pair in which the second sentence is the subsequent sentence in the original document and 50% as random sentence from the corpus during training time. The supposition is that the random sentence will be disconnected from the first sentence. Preprocessing in BERT means converting our dataset into BERT model understandable form. It is done by 3 steps- i) tokenization ii) attention masks and iii) padding. Tokenization means separating each word of a given text and adding special tokens like [CLS], [SEP], [PAD], [UNK]. Padding means placing 0s as pad token and 1 as real token to a fixed length token. This is known as attention mask. The implementation process is described in the following steps:

(i) Load Bert Tokenizer from BERT model

(ii) Tokenize sentence

(iii) Extract each token's token id from BERT vocabulary of about 30000 words

(iv) Encode the given sentence with max length which can be set accordingly. Add special tokens [CLS], [SEP] and attention mask with padding and store in a tensor.

BERT model had been trained based on BOOKCORPUS and English WIKIPEDIA. This model performs a wide range of tasks, such as question answering and language inference, without substantial task specific architecture modifications. It obtains new state-of-the-art results on eleven natural language processing tasks [5], GLUE with 80.5% (7.7%-point absolute improvement), MultiNLI accuracy to 86.7% (4.6% absolute improvement), SQuAD v1.1 question answering Test F1 with 93.2 (1.5-point absolute improvement) and SQuAD v2.0 Test F1 with 83.1 (5.1-point absolute improvement). BERT model is mainly of two types-BERT-base and BERT-large. The architecture is described in Table 2.

2.1.2 RoBERTa

RoBERTa stands for Robustly Optimized BERT Pretraining Approach. RoBERTa is developed based on BERT which share lots of configurations. Reserved Token: BERT uses [CLS] and [SEP] as starting token and separator token respectively while RoBERTa uses <s> and </s> to covert sentences. Size of subword: BERT has around 30k subwords while RoBERTa has around 50k subwords. RoBERTa performs better than BERT [29] by applying 1) bigger training data (16G vs 161G), 2) dynamic masking pattern and 3) replacing the next sentence prediction training objective. BERT masks training data once for MLM objective while RoBERTa duplicates training data 10 times and masks those data differently. Larger mini batches and learning rate are chosen at training time. This leads to better performance in downstream task. This model had taken longer training time based on large datasets named BookCorpus (16G), CC-NEWS (76G), OpenWebText (38G) and Stories (31G) [25]. Thus, the entire data set is about 160 GB of text. This model attains state-of-the-art results on SQuAD, GLUE and RACE. Like BERT, RoBERTa consists of mainly two model architectures described in Table 2.

2.1.3 ALBERT

ALBERT is a LITE BERT consisting of self-supervised learning of language representation. This model overcomes limitation of BERT which takes longer training time. This is implemented by using parameter reduction techniques. A self-supervised loss is added that focuses on modeling inter-sentence coherence [30] which is used instead of next sentence prediction in BERT. This model achieves new state-of-the-art results on the GLUE, RACE, and SQuAD benchmarks. The model architecture is described in Table 2.

Table 2: Model architecture

Model Name	Number of Encoders/ Hidden Layers	Hidden Size	Attention Heads	Total Parameters(in millions)
BERT-base	12	768	12	110
BERT-large	24	1024	16	340
RoBERTa-base	12	768	12	125
RoBERTa-large	24	1024	16	355
ALBERT-base	12	768	12	11
ALBERT-large	24	1024	16	17

2.2 The proposed model

The proposed architecture is illustrated in Fig.1. Here we are using simple transformers which is a modified version of transformer.

2.2.1. Embedding

First, Input is given to the model as input sequence from the corpus by tokenizing. The input sequence is fed into an embedding layer. The embedding layer is classified into three- token embedding, segment embedding and position embedding. In token embedding, the input sequence is split into tokens/words. Each token is converted into token-id which is like a lookup table to seize a learned vector representation of each word. Since neural networks learn

through numbers, each word maps to a vector with continuous values to represent that word. Here 768 high dimensional axes vectors are used to represent each word. Length of input sequence is varying. So, it is possible to enter combination of sentences according to the maximum length. In segment embedding, each token is placed in the corresponding sentence / segment. In position embedding, some information about the position is added into the input embedding. This is done with the following equations.

$$PE(POS, 2J) = \sin\left(\frac{POS}{10000^{\frac{2J}{D}}}\right) \quad (1)$$

$$PE(POS, 2J+1) = \cos\left(\frac{POS}{1000^{\frac{2j}{D}}}\right) \quad (2)$$

Where J is the index of word vector, POS indicates position and D is the dimension.

Here, for even index on input vector, sin function is used and for every odd index on the input vector, cos function is used. Then these vectors are added to their corresponding input embedding. This leads to successful relationship on the position of each vector [9]. The linear properties of the above sine and cos functions can easily learn to attend relative position. In the architecture, x1, x2, x3 and x4 are such vectors which are inputs to the simple transformer encoder.

2.2.2 Simple transformer encoder

The proposed Simple Transformer Encoder consists of two sections like transformer – multi head attention with reduced number compared to transformer encoder and feed forward network.

The attention mechanism empowers the transformers to have extremely long-term memory. A transformer model can "attend" or "focus" on words that are relevant to the generated word by the model. This ability is achieved by learning in training time through back propagation. This overcomes Recurrent Neural Network's (RNN's), Gated Recurrent Unit's (GRU's) and Long Short-Term Memory's (LSTM's) sequence problem of generating words. All the attention of tokens is done in parallel and generate multiple attention heads which discard the token's similarity to itself.

If attention is for single word, then it is named self-attention and we have sequence of words as input so that needs multi head attention. Multi-headed attention in the encoder applies a specific attention mechanism called self-attention which allows the models to associate each word in the input, to other words. Self-attention mechanism is illustrated in Fig. 2.

Step by step procedure of attention

1. Multiply the embedding vector(input) with Query (WQ), key (WK) and value (WV) in order to get corresponding token's query(Q), key(K) and value(V) vectors.

2. Find the relationships of query vector with other words by calculating the dot product of query vector and key vector, ie. Q.K

3. Scale the above quantity by dividing the square root of the dimensionality of key vector which leads to more stable gradients.

4. Perform softmax operation for finding relevant words.

5. Multiply softmax value with the value vector(V) for eliminating irrelevant words. Sum of these new vectors will be the attention of the first token.

Equation (3) describes the above steps

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{D}}\right)V \quad (3)$$

Concatenating these attention heads will calculate the attention weights for the input and produces an output vector with encoded information on how each word should keep relationship among other words. Then a residual connection is set by adding multi-headed attention output vector to the original positional input embedding. The output of the residual connection again passes through a layer normalization. The attention layer of transformer model is very powerful compared to RNN/LSTM, which is restricted to looking at the tokens to the left, by looking at all the other tokens at the same time.

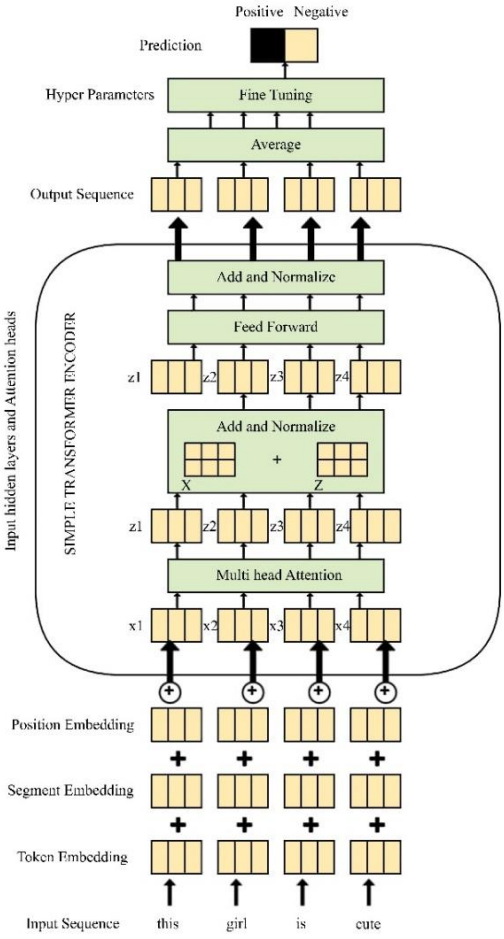

Figure 1: Proposed model architecture

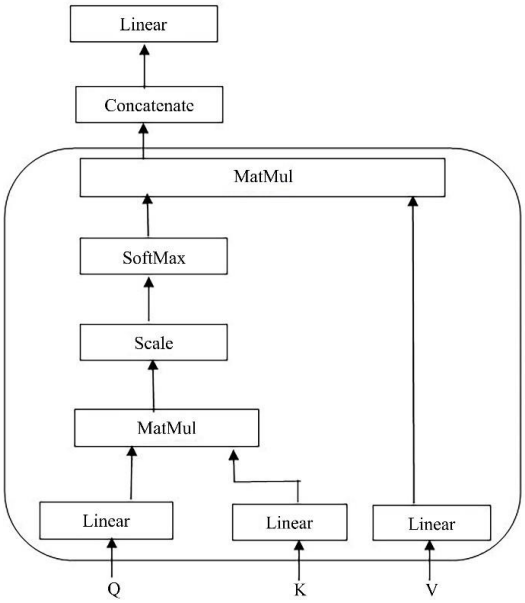

Figure 2: Attention mechanism

The architecture configuration of our proposed ETSC models using simple transformers by modifying the architecture of existing transformer models - BERT-large, RoBERTa-large and ALBERT-large is described in Table 3.

In the proposed architecture, we build different models by setting hidden layer (H) as 2 and attention head (A) with values 2, 4 and 6. We obtained best result by setting H and A as 2.

2.2.3 Fine Tuning

Fully connected classification layer is the output of simple transformer model. Using this layer of transformer model, our down sampling dataset is trained and fine tuning is performed for getting better classification accuracy. Label probabilities are calculated using softmax according to equation (4).

$$P = softmax(CW^T) \quad (4)$$

where C is special [CLS] token for classification and W is fine tuning parameters. Here batch size, maxlen, number of training epochs and learning rate are set as hyper parameters for fine tuning our model.

Table 3: Architecture configuration

Model Name	Existing Architecture		Proposed Architecture	
	Hidden Layers	Attention Heads	Hidden Layers	Attention Heads
BERT-large	24	16	2	2,4,6
RoBERTa-large	24	16	2	2,4,6
ALBERT-large	24	16	2	2,4,6

3 Results and discussion

3.1 Experimental setup

The model is implemented in Python programming language in google colaboratory by setting GPU based execution. The experimental setup is as explained below.

The loaded dataset is shuffled randomly and split for training and testing. In training time small number of rows are selected in each batch and the value is set as 6. The configuration of model such as method; is set as grid and metric as ‘train-losses. Training epochs are set as 2 and 4. Other parameters are chosen such as seed value as 4, learning rate as 1e-5, numbers of hidden layers as 2, number of attention heads as 2 to 6, maximum sequence length as 128, evaluation batch size as 2. Early stopping is implemented in order to avoid overfitting. Features like ‘mcc’, 'tp', 'tn', 'fp', 'fn', 'auroc', 'auprc', 'eval_loss' and ‘time’ is generated at each step. The periodic execution evaluation is recorded and the best model is saved after processing. The next section shows the detailed result.

3.2 Results dataset

IMDb movie review is a benchmark dataset from Kaggle.com with 50000 samples which contains 25000 positive and 25000 negative reviews taken for analysis.

This section summarizes experimental results achieved by our model using IMDb dataset. We built 104 different models with lower configuration of hidden layers and attention heads having good accuracy. We chose configuration for hidden layer as 2 and different configuration for attention head as 2, 4 and 6. Training data is set by splitting 10%, 20%, 30%, 40%, 60% and 80% of actual data for getting different models. Random state is set as 45 and 200. Fig. 3 to Fig. 8 show accuracy wise and time wise comparison of our built in BERT-large-weak learners, RoBERTa-large weak learners and ALBERT-large-weak learners. Fig. 9 and Fig.10 show the accuracy wise and time wise comparison chart of our overall built models. Separate results for some combinations are shown in Fig. 11, Fig.12, Fig.13 and Fig.14. Fig. 15 shows the results of Roberta-large weak learners with different random state. Fig. 16 shows our best model.

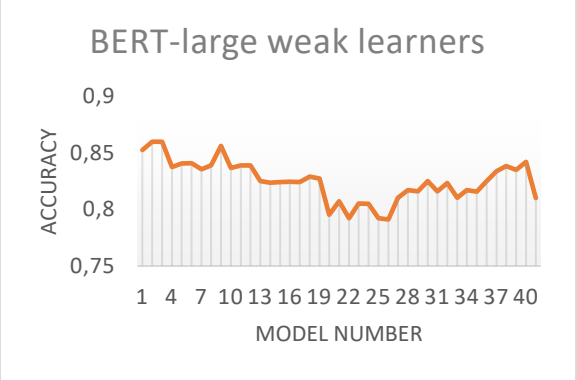

Figure 3: Accuracy wise comparison of BERT models

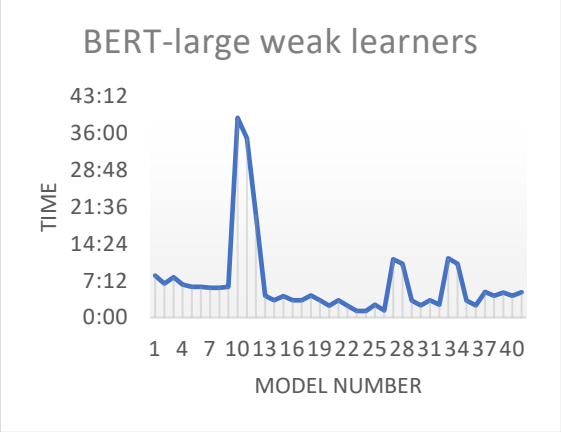

Figure 4: Time wise comparison of BERT models

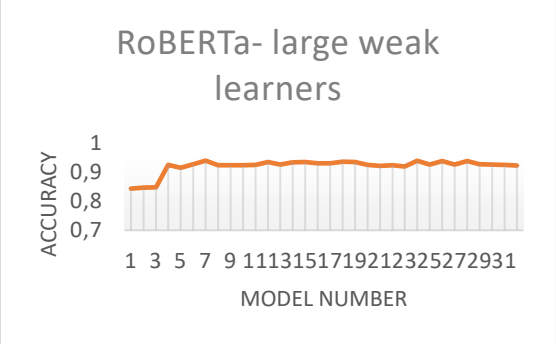

Figure 5: Accuracy wise comparison of RoBERTa models

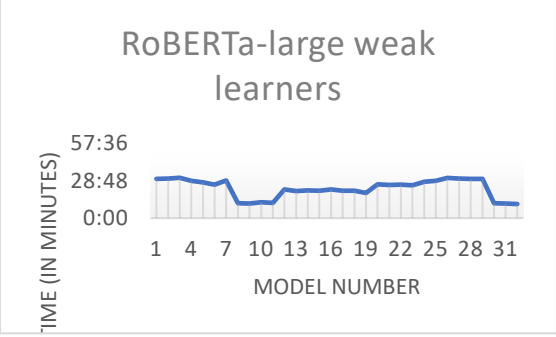

Figure 6: Time wise comparison of RoBERTa models

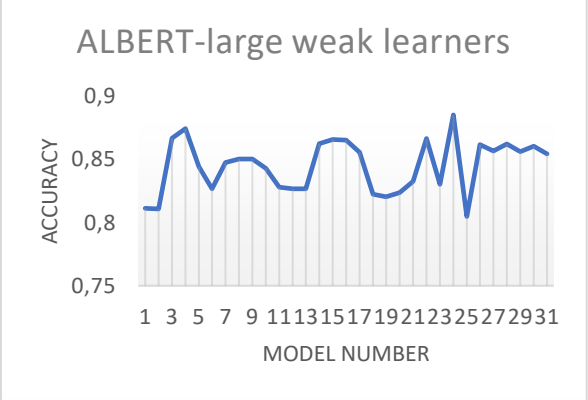

Figure 7: Accuracy wise comparison of ALBERT models

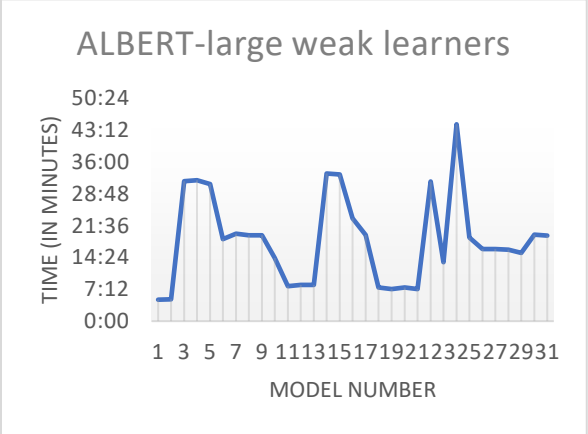

Figure 8: Time wise comparison of ALBERT models

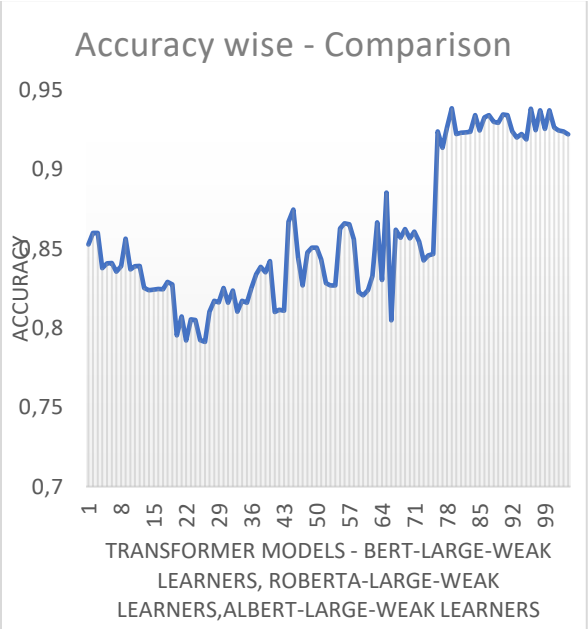

Figure 9: Accuracy wise comparison of built models

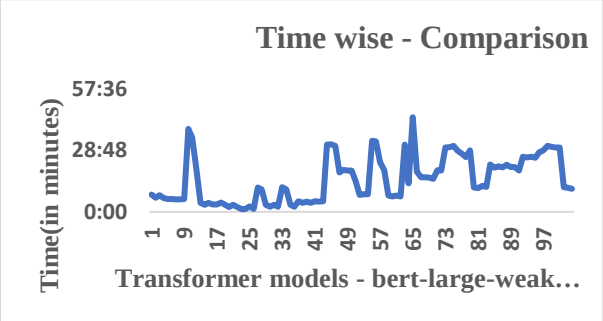

Figure 10 Training time wise comparison of built models

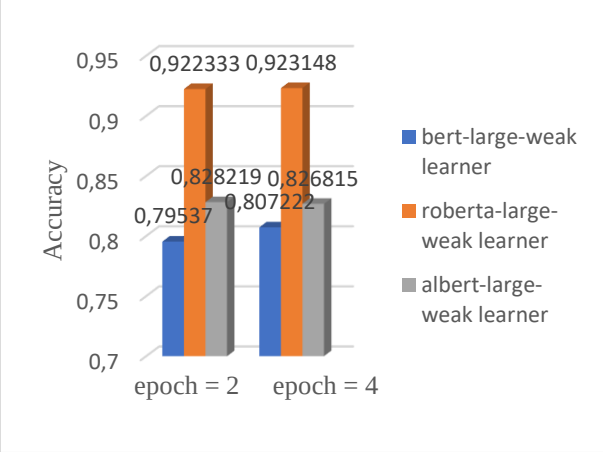

Figure 11: Hidden layer=2 att. head=2 training 10%

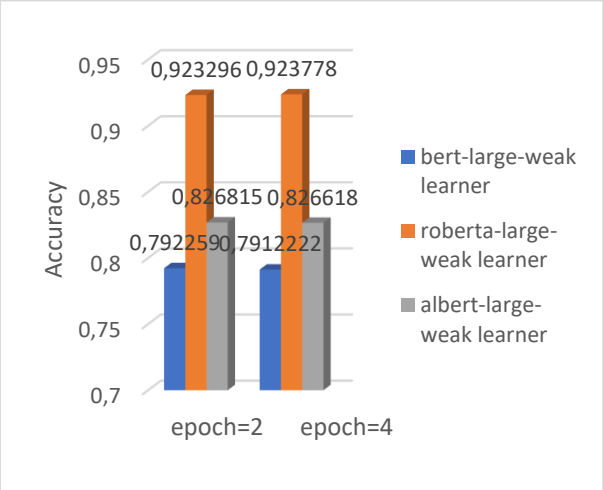

Figure 12: Hidden layer=2 att. head=6 training 10%

Table 4 shows the result of the best model among 104 models using modified architecture by setting hidden layers and attention heads as 2, number of epochs as 2 & 4 and training data as 60%.

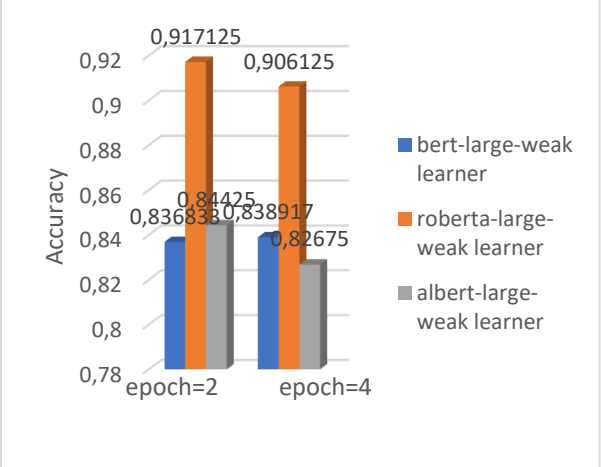

Figure 13: Hidden layer =2 attention head =6 training 60%

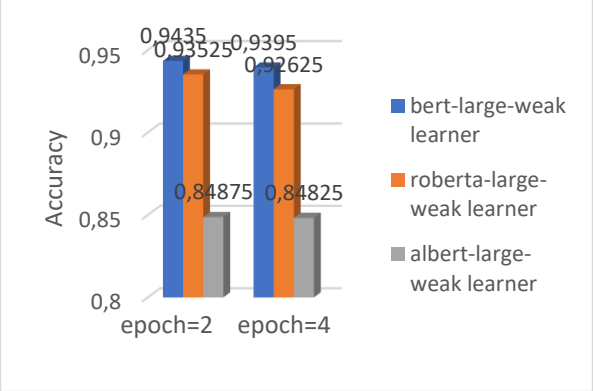

Fig. 14 hidden layer=2 attention head=2 training 80%

Table 4: Result of best model

Model Name	Model Configuration		Epoch	Eval_loss	Accuracy	Time (In minutes)
	Hidd. Lay.	Attention Heads				
BERT-large - weak learner	2	2	2	0.396619	0.837667	06:27
BERT-large-weak learner	2	2	4	0.394821	0.84075	06:01

RoBERTa-large-weak learner	2	2	2	0.250963	0.947875	48:44
RoBERTa-large-weak learner	2	2	4	0.422717	0.91375	19:57
ALBERT-large-weak learner	2	2	2	0.410716	0.885417	44:31
ALBERT-large-weak learner	2	2	4	0.513458	0.804917	18:53

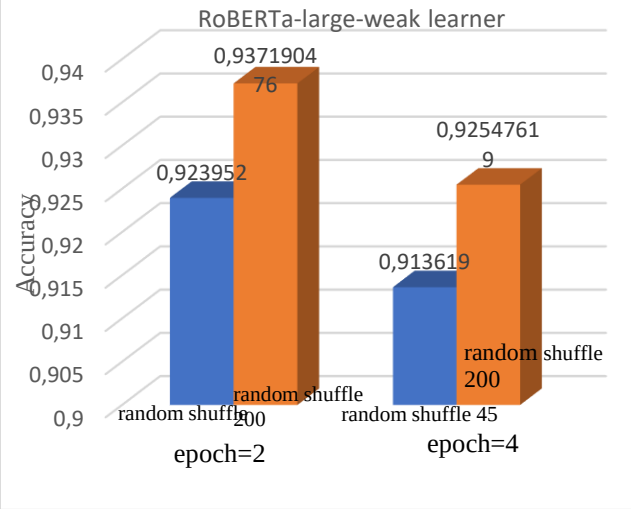

Figure 15: Hidden layer =2 att. head =2 training = 30% with random shuffle

The comparison results of existing transformer-based sentiment classification (TSC) model and Efficient transformer-based sentiment classification (ETSC) in accuracy and training time based on IMDb dataset are shown in Fig.17 and Fig.18 respectively. The comparison results are also described in Table 5.

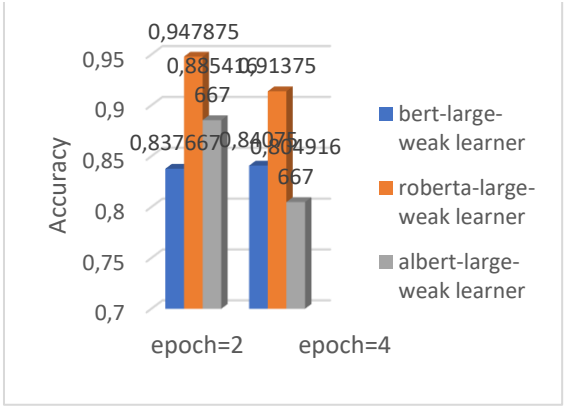

Figure 16: Best model- hidden layer=2 att. head=2 training 60%

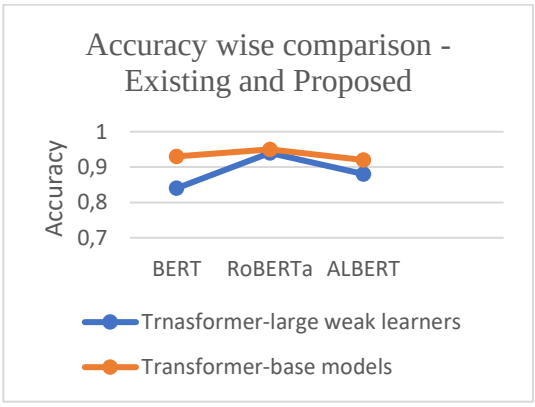

Figure 17: Comparison chart of existing and proposed (accuracy)

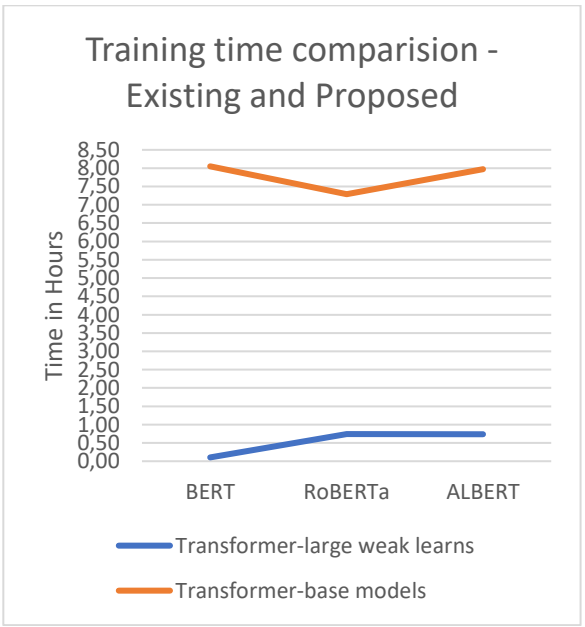

Figure 18: Comparison chart of existing and proposed model(training time)

Table 5; Comparison with previous model

Model Name	Method	Accuracy (%)	Training time(hrs)
TSC [27]	BERT-base	93.83	08:05:00
	RoBERTa-base	95.62	07:29:00
	ALBERT-base	92.99	07:37:00
	DistillBERT-base	92.96	04:20:00
ETSC[present work]	BERT-large-weak learner	84.07	00:06:01
	RoBERTa-large-weak learner	94.78	00:48:44
	ALBERT-large-weak learner	88.57	00:44:31

Table 6: Summary of parameters

Name	**Details**
Optimizer	Adam
Learning rate	le-5
Method	Grid
Metric	Minimixing Training Loss
Max_Seq_Length	128
Train_batch_size	6
Eval_batch_size	2

Table 6 shows the parameters description used in this work.In the proposed models (ETSC), we got 0.08% more accuracy in RoBERTa-large-weak learner than the model BERT-large with dropout regularization; 0.78% more accuracy than alternative masking - RoBERTa-base; 40.96% more accuracy than adding LSTM layer for embedding; 8.91% more accuracy than multi attention network method as illustrated in Table 1.

4 Conclusion

We have built feasible and faster model architectures named ETSC for sentiment classification using simplified versions of transformer models - BERT-large, RoBERTa-large and ALBERT-large with hidden layers as 2 and attention heads as 2, 4 and 6 named as BERT-large-weak learners, RoBERTa-large-weak learners and ALBERT-large-weak learners. We are choosing different training samples - 10%, 20%, 30%, 40%, 60% and 80%; with same configuration for building weak learners. The selection of a smaller number of rows in each batch of training improves our model. The early stopping method for regularization included in our model avoids overfitting in training step. Best model among 104 models is identified by setting model configurations such as hidden layer and attention head as 2, split training data as 60% and epoch as 2 and 4. Both RoBERTa-large-weak learner and ALBERT-large-weak learner got better accuracy in epoch 2 while BERT-large-weak learner achieves better results in epoch 4. RoBERTa-large-weak learner model outperforms with 94.78% accuracy in 2nd epoch, BERT-large-weak learner secures 84.07% in 4th epoch and ALBERT-large-weak learner secures 88.54% accuracy in 2nd epoch. This model outperforms existing TSC model with reduced training time and nearer accuracy. Existing BERT-base model has taken training time as 8.05 hours while proposed BERT-large weak learner has taken 0.1045 hours. Similarly, RoBERTa-base model has taken 7.29 hours of training time while proposed RoBERTa-large weak learner has taken 0.74133hours. The ALBERT-base model has taken 7.97 hours for training and proposed ALBERT-large-weak learner has taken 0.7385. Building a generalized hybrid model for sentiment classification is our future scope.

References

1. Zhang L, Wang S, Liu B. Deep learning for sentiment analysis: A survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. 2018 Jul;8(4): e1253. 10.1002/widm.1253
2. Al-Otaibi ST, Al-Rasheed AA. A Review and Comparative Analysis of Sentiment Analysis Techniques. Informatica. 2022 Jul 29;46(6). https://doi.org/10.31449/inf.v46i6.3991
3. Liu R, Shi Y, Ji C, Jia M. A survey of sentiment analysis based on transfer learning. IEEE Access. 2019 Jun 26; 7:85401-12. https://doi.org/10.1109/access.2019.2925059
4. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I. Attention is all you need. Advances in neural information processing systems. 2017;30. https://doi.org/10.48550/arXiv.1706.03762
5. Devlin J, Chang MW, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805. 2018 Oct 11. https://doi.org/10.48550/arXiv.1810.04805
6. Dai J, Yan H, Sun T, Liu P, Qiu X. Does syntax matter? a strong baseline for aspect-based sentiment analysis with roberta. arXiv preprint arXiv:2104.04986. 2021 Apr 11.

https://doi.org/10.18653/v1/2021.naacl-main.146

7. "XLNet, RoBERTa, ALBERT models for Natural Language Processing (NLP)." https://iq.opengenus.org/advanced-nlp-models/ (accessed Oct. 30, 2021).
8. "Binary Classification - Simple Transformers." https://simpletransformers.ai/docs/binary-classification/ (accessed Oct. 30, 2021).
9. Wolf T, Debut L, Sanh V, Chaumond J, Delangue C, Moi A, Cistac P, Rault T, Louf R, Funtowicz M, Davison J. Transformers: State-of-the-art natural language processing. In Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations 2020 Oct (pp. 38-45). https://doi.org/10.18653/v1/2020.emnlp-demos.6
10. "How do pre-trained models work?. …and why you should use them more often | by Dipam Vasani | Towards Data Science." https://towardsdatascience.com/how-do-pretrained-models-work-11fe2f64eaa2 (accessed Oct. 30, 2021).
11. Kant N, Puri R, Yakovenko N, Catanzaro B. Practical text classification with large pre-trained language models. arXiv preprint arXiv:1812.01207. 2018 Dec 4. https://doi.org/10.48550/arXiv.1812.01207
12. Kumar V, Choudhary A, Cho E. Data augmentation using pre-trained transformer models. arXiv preprint arXiv:2003.02245. 2020 Mar 4. https://doi.org/10.48550/arXiv.2003.02245
13. Xu H, Shu L, Yu PS, Liu B. Understanding pre-trained bert for aspect-based sentiment analysis. arXiv preprint arXiv:2011.00169. 2020 Oct 31. https://doi.org/10.18653/v1/2020.coling-main.21
14. Munikar M, Shakya S, Shrestha A. Fine-grained sentiment classification using BERT. In2019 Artificial Intelligence for Transforming Business and Society (AITB) 2019 Nov 5 (Vol. 1, pp. 1-5). IEEE. https://doi.org/10.1109/aitb48515.2019.8947435
15. Zhao M, Lin T, Mi F, Jaggi M, Schütze H. Masking as an efficient alternative to finetuning for pretrained language models. arXiv preprint arXiv:2004.12406. 2020 Apr 26. https://doi.org/10.18653/v1/2020.emnlp-main.174
16. Naseem U, Razzak I, Musial K, Imran M. Transformer based deep intelligent contextual embedding for twitter sentiment analysis. Future Generation Computer Systems. 2020 Dec 1; 113:58-69. https://doi.org/10.1016/j.future.2020.06.050
17. Kaiser L, Bengio S, Roy A, Vaswani A, Parmar N, Uszkoreit J, Shazeer N. Fast decoding in sequence models using discrete latent variables. InInternational Conference on Machine Learning 2018 Jul 3 (pp. 2390-2399). PMLR. https://doi.org/10.48550/arXiv.1803.03382
18. Tang T, Tang X, Yuan T. Fine-tuning BERT for multi-label sentiment analysis in unbalanced code-switching text. IEEE Access. 2020 Oct 12; 8:193248-56. https://doi.org/10.1109/access.2020.3030468
19. Trușcă, M.M., Wassenberg, D., Frasincar, F. and Dekker R., 2020, June. A hybrid approach for aspect-based sentiment analysis using deep contextual word embeddings and hierarchical attention. In International conference on web engineering (pp. 365-380). Springer, Cham. https://doi.org/10.1007/978-3-030-50578-3_25
20. Wang C, Li M, Smola AJ. Language models with transformers. arXiv preprint arXiv:1904.09408. 2019 Apr 20. https://doi.org/10.48550/arXiv.1904.09408
21. Farahani M, Gharachorloo M, Farahani M, Manthouri M. Parsbert: Transformer-based model for persian language understanding. Neural Processing Letters. 2021 Dec;53(6):3831-47. https://doi.org/10.1007/s11063-021-10528-4
22. Cheng X, Xu W, Wang T, Chu W. Variational semi-supervised aspect-term sentiment analysis via transformer. arXiv preprint arXiv:1810.10437. 2018 Oct 24. https://doi.org/10.18653/v1/k19-1090
23. Biesialska K, Biesialska M, Rybinski H. Sentiment analysis with contextual embeddings and self-attention. InInternational Symposium on Methodologies for Intelligent Systems 2020 Sep 23 (pp. 32-41). Springer, Cham. https://doi.org/10.1007/978-3-030-59491-6_4
24. Voita E, Talbot D, Moiseev F, Sennrich R, Titov I. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. arXiv preprint arXiv:1905.09418. 2019 May 23. https://doi.org/10.18653/v1/p19-1580
25. Hoang M, Bihorac OA, Rouces J. Aspect-based sentiment analysis using bert. In Proceedings of the 22nd nordic conference on computational linguistics 2019 (pp. 187-196). https://aclanthology.org/W19-6120.pdf
26. Xu Q, Zhu L, Dai T, Yan C. Aspect-based sentiment classification with multi-attention network. Neurocomputing. 2020 May 7; 388:135-43. https://doi.org/10.1016/j.neucom.2020.01.024
27. Mathew L, Bindu V R. Efficient classification techniques in sentiment analysis using transformers. International Conference on Innovative Computing and Communications 2022 (pp. 849-862). Springer, Singapore. https://doi.org/10.1007/978-981-16-2594-7_69
28. Ruder S, Peters ME, Swayamdipta S, Wolf T. Transfer learning in natural language processing. In Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: Tutorials 2019 Jun (pp. 15-18). https://doi.org/10.18653/v1/n19-5004
29. Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, Levy O, Lewis M, Zettlemoyer L, Stoyanov V. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692. 2019 Jul 26. https://doi.org/10.48550/arXiv.1907.11692
30. Lan Z, Chen M, Goodman S, Gimpel K, Sharma P, Soricut R. Albert: A lite bert for self-supervised learning of language representations. arXiv preprint arXiv:1909.11942. 2019 Sep 26. https://doi.org/10.48550/arXiv.1909.11942