Received: 1 February 2021 | Revised: 3 June 2021 | Accepted: 28 June 2021

DOI: 10.1002/cpe.6486

SPECIAL ISSUE PAPER

WILEY

Text classification using improved bidirectional transformer

Murat Tezgider[1] | Beytullah Yildiz[2] | Galip Aydin[1]

[1]Department of Computer Engineering, Faculty of Engineering, Firat University, Elazig, Turkey

[2]Department of Software Engineering, School of Engineering, Atilim University, Incek Ankara, Turkey

Correspondence
Murat Tezgider, Department of Computer Engineering, Faculty of Engineering, Firat University, Elazig 23200, Turkey.
Email: murattezgider@gmail.com

Abstract

Text data have an important place in our daily life. A huge amount of text data is generated everyday. As a result, automation becomes necessary to handle these large text data. Recently, we are witnessing important developments with the adaptation of new approaches in text processing. Attention mechanisms and transformers are emerging as methods with significant potential for text processing. In this study, we introduced a bidirectional transformer (BiTransformer) constructed using two transformer encoder blocks that utilize bidirectional position encoding to take into account the forward and backward position information of text data. We also created models to evaluate the contribution of attention mechanisms to the classification process. Four models, including long short term memory, attention, transformer, and BiTransformer, were used to conduct experiments on a large Turkish text dataset consisting of 30 categories. The effect of using pretrained embedding on models was also investigated. Experimental results show that the classification models using transformer and attention give promising results compared with classical deep learning models. We observed that the BiTransformer we proposed showed superior performance in text classification.

KEYWORDS

attention, deep learning, machine learning, text classification, text processing, transformer

1 | INTRODUCTION

We have recently come across machine learning applications in many areas of our lives. Different machine learning methods and algorithms are being developed day by day. It becomes increasingly important to gain insight and value from the enormous amount of data that we produce. Text data has an important place in this deluge of data. Considering that knowledge is transmitted between individuals and even generations through texts, the importance of learning information from text data through an automated process is much better understood.

Many machine learning applications have been proposed to process text data. Topics such as text classification, natural language understanding, machine translation, text summarization, and question answering are among the important text processing topics. Text classification has an important place as it constitutes one of the building blocks of others. Methods using deep learning models in text classification have achieved significant success in recent years.[1–3] The success of deep learning methods made it the most preferred method in text classification.[4–8] As a result, recurrent neural network (RNN) and its derivatives, long short term memory (LSTM),[9] gated recurrent unit,[10] which are deep learning models with good results in time series, have become the methods with the most applications for text processing.

The attention mechanism has recently become a more important and promising choice.[11–13] RNN models usually use the latest state obtained from the hidden units. As a result, the influence of the leading words on the context vector decreases as the number of words in the sentence grows. Besides, taking every part of the sentence equally and not being able to focus on the important words causes a critical deficiency. Moreover, the sequential nature of RNN prevents taking full advantage of parallel computing. Bahdanau et al.[14] proposed an attention mechanism to overcome such deficiencies. The general idea of the attention mechanism is to give more importance to the parts in the input data that are more effective for the result. Attention mechanism has been defined by Galassi et al.[15] as part of a neural architecture that enables dynamically highlighting the

relevant properties of input, which is typically a sequence of text data. Attention mechanisms have the advantage of parallel computing. Unlike RNN, even in long documents, the first word is looked at as attentively as the last word. The order of occurrence of words in the text has an important place in the meaning. Therefore, word order is used in calculating attention for tasks in which this sequence is important. For this, position encoding methods classified as absolute and relative are employed.[16]

Transformer was introduced by Vaswani et al. for an encoder–decoder architecture.[17] It became the cornerstone of many natural language processing models such as BERT,[18] GPT-3,[19] and DALL-E,[20] and very successful results began to emerge. Series of text inputs are converted into a fixed-size context vector in the encoder that consists of six identical layers. Each layer contains two sublayers: a multihead self-attention mechanism and a position-wise fully connected feed-forward network. A residual layer is used around each of the sublayers before the normalization layer. Given the encoding context vector, the conditional probability distribution of a target sequence is calculated by the decoder where the architecture is almost the same as the encoder except that it includes two multihead attention sublayers.

In this study, a bidirectional transformer (BiTransformer) architecture that takes into account the bidirectional word order was proposed using transformer blocks that we modified. The BiTransformer model that is equivalent to the BiLSTM structure was realized for faster and higher accuracy text processing. Roughly speaking, one transformer encoder block processes words in the text from left to right, while the other transformer encoder block processes the words from right to left. Later, the combined context vectors of transformer encoder blocks increase the classification success significantly. In addition, we conducted text classification experiments with the attention structure that we improved, inspired by the self-attention mechanism method proposed by Lin et al.[21] Approximately 90 thousand Turkish documents consisting of 30 categories were classified using four models: LSTM, LSTM + Self-Attention, Vanilla Transformer, and BiTransformer. Models were trained with different sizes of datasets to examine the effect of data size on models. Word vectors obtained from the model created with FastText using 1.5 million documents were also applied to examine the effect of pretrained word embedding on the models. We preferred FastText because it gives better results in agglutinative languages such as Turkish. The contributions of this study can be summarized as follows:

- We propose BiTransformer that took into account the bidirectional word order and showed better and faster results in text classification.
- We show that the self-attention layer that we improve increases the success of the classification models.
- We prove that self-attention and transformer-based mechanisms give more successful results in datasets containing transitive classes.
- We created a Turkish dataset consisting of 30 categories that can be used in classification processes, collected from public websites on the Internet.

Related works are given in the second section. In the third section, data and models for classification are explained. The experimental results of the study with analysis are provided in the fourth section. The last section concludes the paper with a general assessment.

2 | RELATED WORK

There are many studies in different areas related to machine learning.[22–24] Various problems related to machine learning and data have been extensively researched.[25–28] In addition, the number of studies on the processing of text data, which has an important place in the amount of data produced, also shows the importance of text processing. However, we will mainly focus on attention and transformer structures for text processing.

The attention mechanism was first introduced for the natural machine translation task by Bahdanau et al.[14] In their work, the attention mechanism was proposed to achieve the alignment between encoder and decoder. The encoder and decoder structures consisted of the encoder RNN layer that encoded the inputs into a fixed-sized vector, and the decoder RNN decoded this vector to target words. Alignment between the encoder and decoder was provided with an attention mechanism. That reduced losses due to fixed-length vectors. The model learned from all inputs is an example of soft attention. Xu et al.[29] worked on creating captions with a hard attention-based method that automatically interpreted the content of the image as a subtask of image processing. In hard attention, a part of the input is used instead of the whole input differently from soft attention. Soft attention learns from the entire inputs and provides good results but leads more computations. On the other side, using part of the input allows fewer computations in hard attention with lower results.

A hierarchical attention network for document classification was used by Yang et al.[30] They obtained the sentence vectors by applying the attention mechanism to the sentences. Rush et al.[31] conducted an abstract text summarization study using a local attention mechanism. They used a method based on local attention that generated each word of the abstract based on the input sentence. Lu et al.[32] used an image-question-answering study using a coattention model that utilized image and question interest in common. The authors argued that both image attention and question attention were equally important. They proposed a hierarchical architecture that represented the question at three different levels: word level, phrase level, and question level. Wang et al.[33] proposed a new deep learning model called coupled multilayer attentions. The proposed model consisted of multilayered attention. They used two different attentions for each sentence, one extracting aspect terms and the other extracting opinion terms.

Ying et al.[34] created a recommendation system using a two-layered attention network. They proposed a hierarchical network of two layers to find items of preference for the users. Through the attention they used on the first layer, the long-term preferences for the users from the items they had previously purchased were learned. Through the second layer, the long-term preferences learned in the first layer were combined with short-term preferences. Lin et al.[21] constructed the sentence context vectors using the self-attention mechanism. Instead of using a vector, they used a two-dimensional matrix for sentence representation. Each row of the matrix focuses on different parts of the sentence. Dos Santos et al.[35] proposed the attentive pooling method, a two-way attention mechanism consisting of CNN and RNN. They proposed a method that calculated the interaction between question and answer input pairs. Kadlec et al.[36] used an attention mechanism to select words containing answers in tasks such as answering questions. They stated that the attention mechanism worked well in the question–answering system.

Vaswani et al.[17] proposed an approach called transformer. It was shown that this architecture gained performance without sacrificing quality. Since the computation of recurrent networks is processed consecutively depending on the input order, it negatively affects the efficiency of the computation because parallelization cannot be made. Therefore, in this study, they proposed a transformer architecture that did not use recurrent connections and used self-attention to ensure alignment between input and output. They utilized absolute position encoding consisting of sinus and cosine functions in their studies for word position. After the introduction of the transformer model, we have seen many text processing applications utilizing transformers such as BERT, RoBERTa, DistilBERT.[37,38] These studies generally use pretrained models and fall into the transfer learning category. In our study, we create a BiTransformer with new approaches to improve text classification instead of fine-tuning pretraining transformer models such as BERT, RoBERTa, DistilBERT.

3 | METHODOLOGY

We use four deep learning models for the classification process, including our proposed BiTransformer. Word embedding models were created using FastText word vectors to investigate the effect of pretrained embedding on models. The configurations of the models used in this study are given in Table 1.

3.1 | Models using LSTM and attention mechanisms

The first model shown in Figure 1(A) consists of a pure LSTM layer. We use this model as a baseline model for classification. The LSTM layer contains 128 unit cells. We use a layer consisting of 30 nodes using the softmax activation function to perform the classification process.

The second model given in Figure 1(B) uses the LSTM layer with the self-attention that we adapted. The main idea of the attention mechanism is to give a higher contribution to the region that provides the maximum effect on the relationship between input and output. For example, when

TABLE 1 Text classification model configurations

Models	Description
LSTM	It is a classification model in which LSTM is used with the trainable embedding layer.
LSTM + Self-Attention	It is a classification model that uses a trainable embedding layer with LSTM and Self-Attention together.
Transformer	It is a classification model that used summation of the trainable embedding layer and positional encoding with the transformer. The transformer encoder proposed by Vaswani et al. is used.
BiTransformer (proposed)	It is a classification model composed of two concatenated transformers. One of them uses summation of the trainable embedding layer and positional encoding, and the other uses summation of the trainable embedding layer and reverse positional encoding. A modified transformer encoder block proposed by Vaswani et al. is used.
LSTM with PWE	It is a classification model in which LSTM is used with the FastText pretrained word embedding.
LSTM + Self-Attention with PWE	It is a classification model that uses FastText pretrained word embedding with LSTM and Self-Attention together.
Transformer with PWE	It is a classification model that used summation of the FastText pretrained word embedding and positional encoding with the transformer proposed by Vaswani et al.
BiTransformer with PWE (proposed)	It is a classification model composed of two concatenated transformers. One of them uses the FastText pretrained word embedding and positional encoding. The other uses summation of the FastText pretrained word embedding and reverse positional encoding part of the transformer Vaswani et al.

Abbreviations: LSTM, long short term memory; PWE, pretrained word embedding.

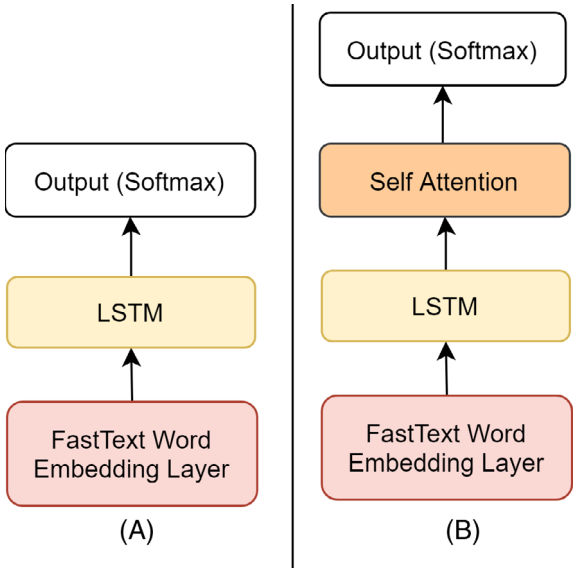

FIGURE 1 Overview of network architectures used, (A) LSTM model, (B) LSTM+ Self-Attention. LSTM, long short term memory

talking to a person, we look at the person's face more carefully and focus less on the other parts of our vision. Likewise, when reading a text, more importance is given to certain words in order to understand the text. Therefore, for text classification, paying more attention to a specific part of the text than other parts provides a significant effect. Attention mechanisms usually take three parameters as input: value, key, and query. A score is calculated using the key and query pairs, and the value parameter is weighted to generate attention with this score. Value, key, and query have the same values in self-attention mechanisms.

When building the LSTM + Self-Attention model, we were inspired by the attention mechanism proposed by Lin et al.[21] The self-attention mechanism is applied at the document level. By using word vectors fw_i, each document is converted to a document vector $(fw_1, fw_2, \ldots, fw_{300})$. Zero vectors are added for padding in documents containing less than 300 words. Since word vectors are trained in 150 dimensions, each document is represented as a two-dimensional matrix with 150×300 elements. LSTM layer is used to create the context vector of the documents. For each fw_i vector in the LSTM layer, a 128-sized hidden state $h_i = \text{LSTM}(fw_i, 128)$ is generated. Hidden states consisting of document words are represented as $H = (h_1, h_2, \ldots h_{300})$. We calculate the attentions using Formula (1), where W_1 is a weight matrix of size $S * H$ and W_2 is a weight matrix of size $S_2 * S$. S is a hyperparameter set to 128 and S_2 set to 32 refers to the number of individual components to be inferred in a document. The normalization process is performed using the softmax function. Then, a context matrix is created for the document by multiplying the obtained *Attention* score with the hidden state matrix H created for each document.

$$\text{Attention} = \text{softmax}(W_2 \tanh(W_1 H^T)). \quad (1)$$

3.2 | Models using Transformer and BiTransformer

The transformer model shown in Figure 2(A) uses an encoder block, an average pooling layer, and a fully connected neural network layer. The embedding layer can be trainable or can use pretrained word embeddings. In addition, the position vectors of each word are used to add the necessary position information for text processing. Position encoding is created using the functions of $\text{PE}(p, 2i) = \sin(p/10{,}000^{2i/d_{\text{model}}})$ and $\text{PE}(p, 2i+1) = \cos(p/10{,}000^{2i/d_{\text{model}}})$ where p is the position index and d_{model} is dimension size.

The fourth model shown in Figure 2(B) consists of two transformer encoder blocks that take into account the bidirectional word order. We proposed this structure to replace the state-of-the-art BiLSTM for text processing, where the operation between hidden layers is sequential. The model proposed in the article of vanilla transformer creates a left to right position vector with an absolute position encoding function. We use two transformer encoder blocks with a single normalization at the output. The scaled inner product given in Formula (2) is used to calculate the attention. In the first transformer encoder block, a relative positioning with indices starting from 1 to the number of words is used. For the second transformer, the indices are created in reverse for each document. Positional encoding can be defined as $\text{PositionEncoding} = \text{PE}(p)$ where p is a forward position or backward position of the words, and PE is a function to calculate a position value. In short, while the word vectors for the first transformer encoder block are given in a certain order, the word vectors for the second transformer encoder block are given in reverse order. Word position encodings are combined with the word embedding layer before the transformer encoder blocks. Thus, we obtain two-way information of text documents. We

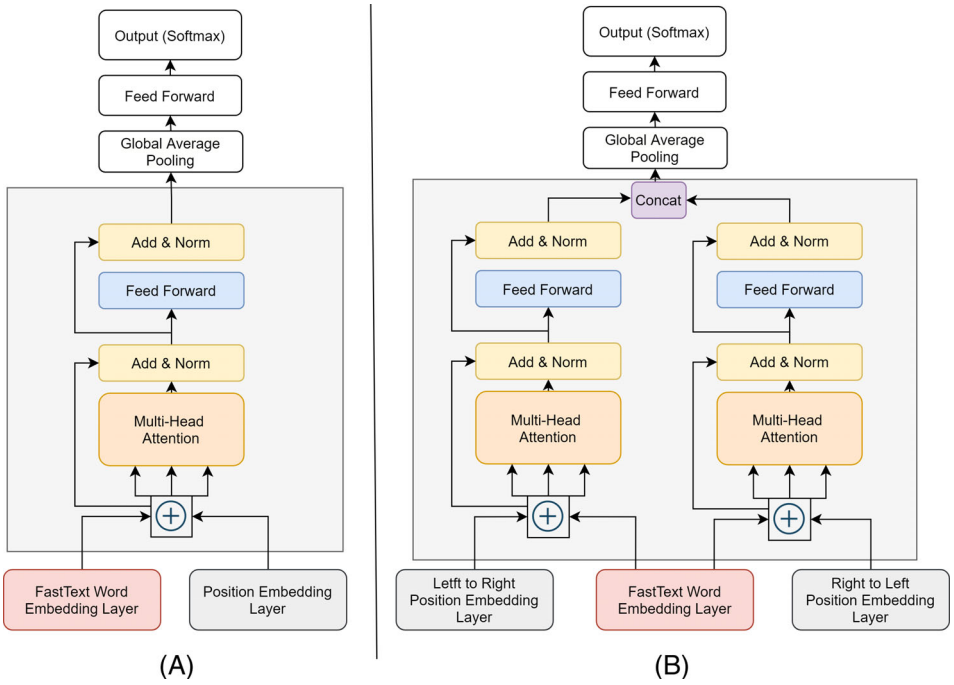

FIGURE 2 Models created using the transformer encoder block, (A) Vanilla Transformer model, (B) BiTransformer

TABLE 2 Datasets for training models

Name	Number of documents in a class	Total number of documents	Number of classes
Dataset 1	500	15,000	30
Dataset 2	1000	30,000	30
Dataset 3	2000	60,000	30
Dataset 4	3000	90,000	30

reduced the output of the transformer encoder blocks before giving them to the fully connected layer containing 30 nodes with a Softmax activation function.

$$\text{Scaled Dot Attention}(Q, K, V) = \text{softmax}\left(\frac{W_q Q W_k K^T}{\sqrt{d_k}}\right) \tag{2}$$

3.3 | Datasets

We performed text classification on our models with text data consisting of approximately 90 thousand documents obtained from public websites with 30 classes. We used 70% of the text data for training, 15% for validation, and 15% for testing. The classes of the data are mother-baby, household appliances, computer, mobile phone, electronics, real estate/construction, energy, event/organization, education, finance, clothing, food, communication, beverage, public services, cargo/transportation, personal care, media, entertainment, furniture-home textiles, kitchenware, jewelry-watches-glasses, automotive, health, insurance, sports, hygiene, tourism, transportation. Four types of datasets were created, as indicated in Table 2. In addition, pretrained word embeddings were generated using FastText with text data consisting of 1.5 million Turkish documents collected from public sites on the Internet.

Statistical information about the tokens in the training dataset is given in Table 3. Dataset 1, Dataset 2, Dataset 3, and Dataset 4 contain approximately 1.2, 2.4, 4.7, and 1.2 million tokens, respectively. Each document in all datasets contains an average of 78–80 tokens. At the same time, the SD of the number of tokens in the documents in all datasets is 51–52. The documents in the datasets contain a minimum of three tokens and a maximum of 615 tokens. 50%, 75%, 99% of documents contain up to 64, 99–100, and 300 tokens, respectively.

TABLE 3 Statistical information about the tokens in the training dataset

Name	Total number of tokens in documents	Mean of number of tokens	SD of number of tokens	Min number of tokens	25%	50%	75%	99%	Max number of tokens
Dataset 1	1,176,794	78.45	50.51	3	45	64	100	300	615
Dataset 2	2,386,683	79.56	51.87	3	45	64	100	300	615
Dataset 3	4,780,217	79.67	51.94	3	46	64	99	300	615
Dataset 4	7,166,664	79.63	52.05	3	46	64	99	300	615

TABLE 4 *F*1 scores of the text classifications

	*F*1-Score			
	Dataset 1	Dataset 2	Dataset 3	Dataset 4
LSTM (baseline)	<u>31.20</u>	<u>56.14</u>	<u>72.75</u>	<u>74.35</u>
LSTM + Self-Attention	52.06	71.76	82.26	82.85
Transformer	79.47	81.86	85.13	86.81
BiTransformer (proposed)	80.04	81.99	85.44	86.92
LSTM with PWE	80.65	83.15	85.89	85.69
LSTM + Self-Attention with PWE	83.55	85.92	86.70	88.28
Transformer with PWE	85.06	86.00	87.28	88.14
BiTransformer with PWE (proposed)	**86.29**	**86.37**	**87.83**	**88.55**

Abbreviations: LSTM, long short term memory; PWE, pretrained word embedding.
The best results are shown in bold and the baseline results are underlined.

3.4 | Training of models

All models were initially trained with a Dataset 1 with 256 batch sizes. Adam optimizer and sparse categorical crossentropy loss function were used during the training. Later, the training data was doubled, and the training was repeated. Finally, models were trained using a dataset containing 3000 documents from each class. After the training, the effect of data size on the models and the performance of the models were evaluated.

At first, all models were trained using a trainable embedding layer without using FastText pretrained word embedding. Later, models were trained by using FastText pretrained word embedding. Word embedding hyperparameters have important effects on classification success.[39,40] FastText word vectors were created using the Gensim library with the skip-gram method by setting the vector size to 150, the minimum word count to 30, the window size to 10, and the minimum number of n-grams to 2. While the deep learning models were trained, the word vectors obtained from the FastText model were used. In the FastText method, words that are not in the dictionary can be obtained by combining n-grams. In languages with many suffixes, such as Turkish, there may be many different suffixes for the root word. The n-gram method provided by FastText was used to compute the vectors of the words not included in the dictionary. In this way, using approximation vectors computed with n-grams instead of using zero vectors for the words not included in the dictionary has a positive effect on classification success.

4 | MEASUREMENT AND EVALUATION

We collected benchmark results after training the classification models. The first experiment contains the models with trainable embedding layers. It does not contain pretrained FastText word embeddings. The experimental results were collected for four datasets, which are listed in Table 4. Figure 3 shows the weighted average *F*1 scores of the classification models. It is clearly seen that our proposed BiTransformer model gives better results than other models. In cases where the training data is scarce, the success of the BiTransformer model stands out much better compared with the other models. Table 5 shows the improvements achieved with the models compared with the LSTM baseline model using the same datasets. The LSTM model alone is not sufficient. The BiTransformer model provided 48.84% better result than the LSTM and 27.98% better result than the LSTM + Self-Attention model when the data is scarce, such as in Dataset 1. Moreover, the BiTransformer model provides a 12.57% better result than the LSTM and about 4.07% better result than the LSTM + Self-Attention model when the data is abundant, such as in Dataset 4.

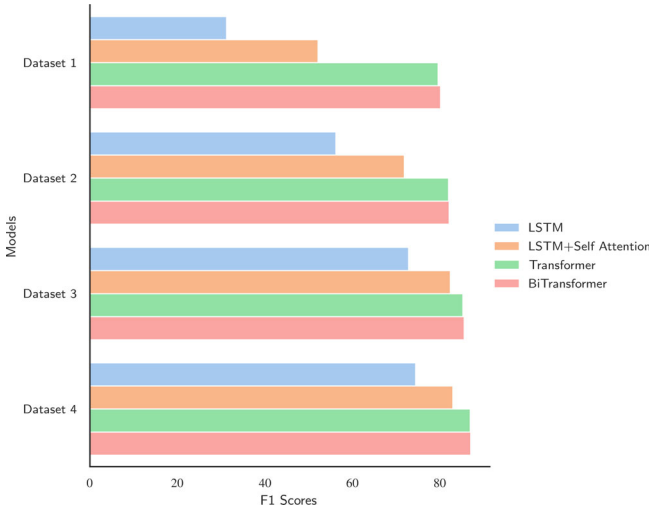

FIGURE 3 Classification of the models without pretrained word embeddings

TABLE 5 *F*1 score improvements of models over base LSTM model

	Dataset 1	Dataset 2	Dataset 3	Dataset 4
LSTM (baseline)	31.20	56.14	72.75	74.35
LSTM + Self-Attention	+20.86	+15.62	+9.51	+8.5
Transformer	+48.27	+25.72	+12.38	+12.46
BiTransformer (proposed)	+48.84	+25.85	+12.69	+12.57
LSTM with PWE	+49.45	+27.01	+13.14	+11.34
LSTM + Self-Attention with PWE	+52.35	+29.78	+13.95	**+13.93**
Transformer with PWE	**+53.86**	**+29.86**	**+14.53**	+13.79
BiTransformer with PWE (proposed)	**+55.09**	**+30.23**	**+15.08**	**+14.2**

Abbreviations: LSTM, long short term memory; PWE, pretrained word embedding.
The best results are shown in bold and the baseline results are underlined.

In the second classification experiment, we use pretrained FastText word embeddings to show the effect of word embeddings. As can be seen in Table 4, as the data size increases, the success rates of the LSTM, LSTM + Self-Attention, Transformer, and BiTransformer models without pretrained embeddings increase by 43.15%, 30.79%, 7.34%, and 6.88%, respectively. However, with using pretrained embeddings, the success rates of the LSTM, LSTM + Self-Attention, Transformer, BiTransformer increase approximately 5.04%, 4.73, 3.08%, and 2.26%, respectively. As a result, while BiTransformer still performs significantly better, the gap between the success of the BiTransformer model and the success of other models is closing as the number of data increases. This is because increasing the amount of data removes the weaknesses of other models.

Table 6 shows that pretrained embedding contributes more to the success rate when data is scarce. However, as the amount of data becomes adequate for learning, the contribution of pretrained word embedding to the success rate decreases. Pretrained word embedding leads to relatively lower performance gains for the Transformer and BiTransformer models since they have learned enough even from scarce data, unlike other models. Table 7 shows the *F*1 score improvements of the models over the baseline LSTM with the pretrained word embedding model. Our proposed BiTransformer model still gives the best results compared with other models when using pretrained embedding. However, the difference in *F*1 scores between BiTransformer and other models is narrowing as pretrained embedding contributes more to the success of other models. In other words, the use of pretrained embedding has a limited effect on the success of the BiTransformer model compared with other models. Even in this case, the BiTransformer model is 5.64% better than the LSTM model and 2.74% better than the LSTM + Self-Attention model. Figure 4 illustrates the effect of pretrained word embedding for the LSTM baseline model and our proposed BiTransformer model.

We used the coefficient of variation to measure the homogeneity of the model predictions on a class basis. The coefficient of variation was obtained by dividing the SDs of a class success by the weighted average of all classes and multiplying by 100 as given in Equation (3):

$$\text{Coefficient of variation} = \frac{\sigma * 100}{\bar{x}}. \tag{3}$$

TABLE 6 Improvements in *F*1 scores when using a pretrained word embedding compared with the model without using a pretrained word embedding

	Dataset 1	Dataset 2	Dataset 3	Dataset 4
LSTM with PWE	+49.45	+27.01	+13.14	+11.34
LSTM + Self-Attention with PWE	+31.49	+14.16	+4.44	+5.43
Transformer with PWE	+5.59	+4.14	+2.15	+1.33
BiTransformer with PWE (proposed)	**+6.25**	**+4.38**	**+2.39**	**+1.63**

Abbreviations: LSTM, long short term memory; PWE, pretrained word embedding.
The best results are shown in bold and the baseline results are underlined.

TABLE 7 The *F*1 score improvements of the models using the pretrained word embedding for classification compared with the baseline model using the pretrained word embedding

	Dataset 1	Dataset 2	Dataset 3	Dataset 4
LSTM with PWE (baseline)	80.65	83.15	85.89	85.69
LSTM + Self-Attention with PWE	+2.9	+2.77	+0.81	+2.59
Transformer with PWE	+4.41	+2.85	+1.39	+2.45
BiTransformer with PWE (proposed)	**+5.64**	**+3.22**	**+1.94**	**+2.86**

Abbreviations: LSTM, long short term memory; PWE, pretrained word embedding.
The best results are shown in bold and the baseline results are underlined.

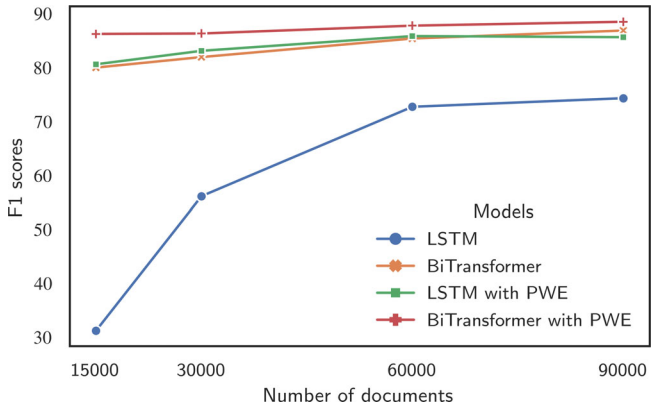

FIGURE 4 Comparing LSTM and BiTransformer models using pretrained word embedding and trainable embedding layer. LSTM, long short term memory

Since we performed a multiclass classification, the coefficient of variation of the *F*1 score was used to compare the models. A low coefficient of variation indicates that the model gives similar results for all classes. By contrast, its high value indicates that the success of the classes is far from the average performance provided by the model. A good model is expected to provide results in all classes close to the average success of the model. When the success of a model is high and the value of the coefficient of variation is low, we evaluated that model as a robust model. We thought that robust models could make better decisions in data with transitive or overlapping classes. Figure 5 shows the results of the *F*1 score coefficient of variation of the models. It is clearly seen that the coefficient of variation values of the models decrease as the data size increases, so the models become more homogeneous. Moreover, it is evident from the results that the value of the coefficient of variation of the proposed BiTransformer model is much better than the transformer models and the other models in all datasets. In short, BiTransformer is a much better classification model because it provides better predictive performance and more homogeneous results. Moreover, a robust model can also be built by using the LSTM + Self-Attention model with FastText pretrained embedding with sufficient data size. When the prediction results of models with low coefficient of variation and high success are examined, it can be seen that even classes that overlap too much with public services and education, health, and transportation are classified better.

Table 8 shows the results obtained by using the models LSTM, LSTM + Self-Attention, BiTransformer with FastText pretrained using Dataset 4. We can see that the BiTransformer model gives better results than the other models. When we look at the *F*1 scores of some classes, we see

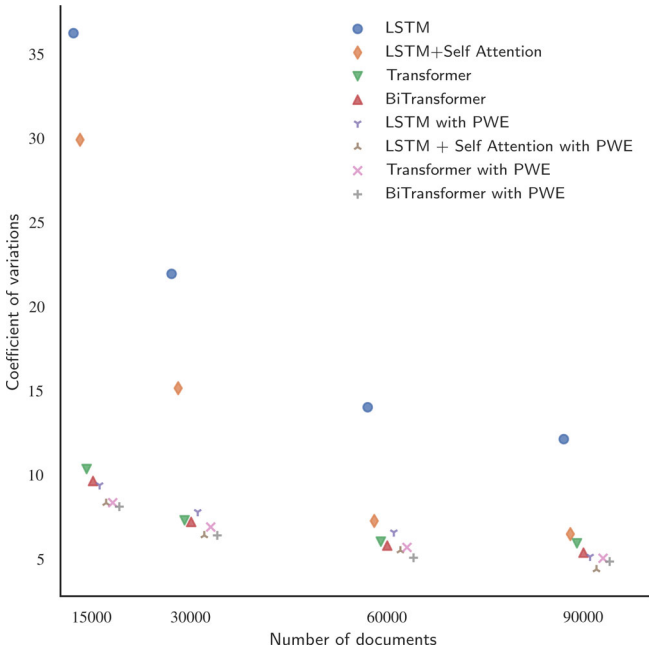

FIGURE 5 The coefficient of variation of models

TABLE 8 Result of LSTM + PWE, LSTM + Self-Attention + PWE, BiTransformer + PWE models

		LSTM + PWE			LSTM + Self-Attention + PWE			BiTransformer + PWE		
	Transitive group	Precision	Recall	*F*1-score	Precision	Recall	*F*1-score	Precision	Recall	*F*1-score
Computer	g1	75.95	73.47	74.69	86.55	79.91	83.1	84.89	81.8	83.31
Mobile phone	g1	88.85	87.17	88	88.97	83.75	86.28	90.79	89.78	90.28
Electronic	g1	84.36	80.08	82.16	85.62	88.71	87.14	89.76	81.96	85.68
Clothing	g2	77.85	89.64	83.33	82.34	90.32	86.14	84.33	90.76	87.43
Mother–baby	g2	86.67	85.99	86.33	93.59	88.74	91.1	94.2	89.22	91.64
Household appliances	g3	88.26	87.5	87.88	92.38	87.39	89.81	93.88	89.46	91.62
Small appliances	g3	90.24	92.89	91.55	90.26	93.92	92.05	93.02	92.81	92.91
Public services	g4	75.21	72.43	73.79	83.83	70.05	76.32	76.42	75.06	75.73
Transport	g4	91.94	88.72	90.3	90.33	92.57	91.43	89.65	91.67	90.65
Education	g4	86.92	81.75	84.25	81.92	84.88	83.37	78.23	84.62	81.3
Finance	g5	82.37	91.97	86.91	84.4	89.16	86.72	89.77	87.53	88.63
insurance	g5	91.52	85.42	88.36	87.95	88.74	88.34	94.33	84.53	89.16
Automotive		87.11	89.56	88.32	91.26	91.67	91.46	84.85	93.81	89.11
Communication	g6	82.39	94.35	87.97	93.22	86.71	89.85	89	91.81	90.38
Media	g6	80.08	85.31	82.61	83.94	88.29	86.06	85.14	87.75	86.42
Real estate and construction		73.36	80.92	76.95	81.26	84.2	82.71	74.43	83.06	78.51

(Continues)

TABLE 8 (Continued)

		LSTM + PWE			LSTM + Self-Attention + PWE			BiTransformer + PWE		
	Transitive group	Precision	Recall	*F*1-score	Precision	Recall	*F*1-score	Precision	Recall	*F*1-score
Energy		93.64	86.92	90.15	91.07	94.36	92.68	94.81	88.55	91.57
Event and organization		91.67	87.82	89.7	88.17	92.55	90.31	90.2	90.42	90.31
Tourism		94.42	87.12	90.62	90.64	89.41	90.02	89.7	90.97	90.33
Entertainment and places		87.23	88.85	88.03	89.81	85.36	87.53	87.42	91.03	89.19
Food		88.24	84	86.07	86.19	92.79	89.37	92.27	89.88	91.06
Drink		92.17	80.97	86.21	90.16	90.77	90.46	91.36	91.57	91.47
Cargo and shipping		92.24	84.17	88.02	88.16	93.92	90.95	84.1	92.34	88.03
Furniture-home textile		86.61	86.25	86.43	92.11	89.41	90.74	92.19	85.92	88.94
Kitchen equipment		92.79	84.43	88.41	91.16	90.54	90.85	95.8	90.21	92.92
Jewelry-watch-glasses		96.51	90.2	93.25	92.29	94.37	93.32	94.42	95.7	95.06
Health		88.71	89.07	88.89	93.58	91.89	92.73	93.24	90.7	91.95
Sports		80.22	87.25	83.59	83.41	87.16	85.24	84.84	90.79	87.71
Personal care and cosmetics		76.19	77.11	76.65	83	84.68	83.84	83.44	84.52	83.97
Hijyen		77.06	88.11	82.22	94.62	83.11	88.49	93.49	87.39	90.34
Loss				49.94			42.99			41.48
Accuracy				85.64			88.31			88.51
Macro avg		86.03	85.65	85.72	88.41	88.31	88.28	88.66	88.52	88.52
Weighted avg		85.96	85.64	85.69	88.41	88.31	88.28	88.73	88.51	88.55

Abbreviations: LSTM, long short term memory; PWE, pretrained word embedding.
The best results are shown in bold and the baseline results are underlined.

relatively low values. When we investigate the cause, we find two reasons that contribute to this result. The first is the transitivity of classes with overlapping concepts. We can see this in the confusion matrix, which we could not show here due to its size. The second reason is mislabeled data. Since we collect the dataset from the Internet and it is crowd-sourced data, we can see that there are errors in the labels. Even under these conditions, BiTransformer performs classification very well. We see improved classification performance even in transitive classes, which are shown in "Transitive Group" column in Table 8. In summary, BiTransformer is a superior model for classification tasks.

5 | CONCLUSION

Text classification is a building block for many natural language processing tasks used in everyday practice. LSTM and BiLSTM-based models were the preferred deep learning methods for building text classification models for a long time. Recently, attention mechanisms have become very popular for many applications. Transformer models that also use an attention mechanism have also had a significant impact on text processing.

We provided an improved attention mechanism to contribute to better text classification. It has been observed that the model built with the addition of the improved attention mechanism gives better performance than the model using only the LSTM structure. Using the attention mechanism with LSTM increases the success of the classification more when it is trained with fewer data compared with training it with more data. In addition, LSTM with the attention mechanism also helps to eliminate deficiencies in data. The relative increase in success decreases as the number of data increases, and the LSTM model we use as a baseline improves its learning with more data.

For a further improvement to text classification, we proposed a BiTransformer model showing very promising results to replace BiLSTM models without degrading performance due to the sequential nature of LSTM implementation. *F*1 scores show that the classification performance of

the BiTransformer model is much better than that of the LSTM models. Since the text data contains important backward and forward position information, we see that our BiTransformer model performs better than the models using LSTM with attention mechanisms and vanilla transformer models. When we consider the coefficient of variation to measure homogeneity, it is obvious that the BiTransformer performs a more consistent classification between classes. Data scarcity is a major problem in labeled data and affects the success of classification. Therefore, our BiTransformer model, which gives more successful results in situations where data is scarce, makes an important contribution to text classification.

DATA AVAILABILITY STATEMENT

Data available on request from the authors.

ORCID

Murat Tezgider https://orcid.org/0000-0002-4918-5697
Beytullah Yildiz https://orcid.org/0000-0001-7664-5145

REFERENCES

1. Russakovsky O, Deng J, Su H, et al. Imagenet large scale visual recognition challenge. *Int J Comput Vis*. 2015;115(3):211-252.
2. Szegedy C, Ioffe S, Vanhoucke V, Alemi AA. Inception-v4, inception-resnet and the impact of residual connections on learning. Paper presented at: AAAI; 2017; San Francisco, CA, USA.
3. Gu J, Wang Z, Kuen J, et al. Recent advances in convolutional neural networks. *Pattern Recogn*. 2018;77:354-377.
4. dos Santos C, Gatti M. Deep convolutional neural networks for sentiment analysis of short texts. Paper presented at: Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics; Technical Papers; 2014; Dublin, Ireland.
5. Johnson R, Zhang T. Effective use of word order for text categorization with convolutional neural networks. Paper presented at: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics; Human Language Technologies; 2015; Denver, CO, USA.
6. Hu R, Mac Namee B, Delany SJ. Active learning for text classification with reusability. *Expert Syst Appl*. 2016;45:438-449.
7. Joulin A, Grave É, Bojanowski P, Mikolov T. Bag of tricks for efficient text classification. Paper presented at: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics; Vol. 2; Short Papers; 2017; Valencia, Spain.
8. Aydin G, Hallac IR. Document Classification Using Distributed Machine Learning. *arXiv preprint arXiv:180203597*; 2018.
9. Bengio Y, Simard P, Frasconi P. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans Neural Netw*. 1994;5(2):157-166.
10. Cho K, Van Merriënboer B, Bahdanau D, Bengio Y. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:14091259*; 2014.
11. Galassi A, Lippi M, Torroni P. Attention in Natural Language Processing. *IEEE Transactions on Neural Networks and Learning Systems*. 2020;1–18. http://dx.doi.org/10.1109/tnnls.2020.3019893.
12. Parikh AP, Täckström O, Das D, Uszkoreit J. A decomposable attention model for natural language inference. *arXiv preprint arXiv:160601933*; 2016.
13. Paulus R, Xiong C, Socher R. A deep reinforced model for abstractive summarization. Paper presented at: International Conference on Learning Representations; 2018; Vancouver, Canada.
14. Bahdanau D, Cho KH, Bengio Y. Neural machine translation by jointly learning to align and translate. Paper presented at: 3rd International Conference on Learning Representations, ICLR 2015; 2015; San Diego, CA, USA.
15. Galassi A, Lippi M, Torroni P. Attention, please! a critical review of neural attention models in natural language processing. *arXiv preprint arXiv:190202181*; 2019.
16. Shaw P, Uszkoreit J, Vaswani A. Self-attention with relative position representations. Paper presented at: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies; Vol. 2 (Short Papers); 2018; New Orleans, LA, USA.
17. Vaswani A, Shazeer N, Parmar N, et al. Attention Is All You Need. *Paper presented at: 31st Conference on Neural Information Processing Systems; Long Beach, CA, USA*.. 2017.
18. Devlin J, Chang M-W, Lee K, Toutanova K. BERT: pre-training of deep bidirectional transformers for language understanding. Paper presented at: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Vol. 1 (Long and Short Papers); 2019; Minneapolis, MN, USA.
19. Brown TB, Mann B, Ryder N, et al. Language models are few-shot learners. *arXiv preprint arXiv:200514165*; 2020.
20. Ramesh A, Pavlov M, Goh G, et al. Zero-shot text-to-image generation. *arXiv preprint arXiv:210212092*; 2021.
21. Lin Z, Feng M, Santos CN, et al. A structured self-attentive sentence embedding. *arXiv preprint arXiv:170303130*; 2017.
22. Guner N, Yaldir A, Gunduz G, Çomak E, Tokat S, Iplikçi S. Predicting academically at-risk engineering students: a soft computing application. *Acta Polytech Hung*. 2014;11(5):199-216.
23. Şentaş A, Tashiev I, Küçükayvaz F, et al. Performance evaluation of support vector machine and convolutional neural network algorithms in real-time vehicle type and color classification. *Evol Intel*. 2020;13(1):83-91.
24. Tufek A, Gurbuz A, Ekuklu OF, Aktas MS. Provenance collection platform for the weather research and forecasting model. Paper presented at: 2018 14th International Conference on Semantics, Knowledge and Grids (SKG); 2018; Guangzhou, China.
25. Abeykoon V, Kamburugamuve S, Govindrarajan K, et al. Streaming machine learning algorithms with big data systems. Paper presented at: 2019 IEEE International Conference on Big Data (Big Data); 2019; Los Angeles, CA, USA.
26. Tufek A, Aktas MS. On the provenance extraction techniques from large scale log files: a case study for the numerical weather prediction models. Paper presented at: Euro-Par 2020: Parallel Processing Workshops; 2020; Warsaw, Poland
27. Onan A, Korukoğlu S, Bulut H. A multiobjective weighted voting ensemble classifier based on differential evolution algorithm for text sentiment classification. *Expert Syst Appl*. 2016;62:1-16.

28. Onan A, Korukoğlu S, Bulut H. A hybrid ensemble pruning approach based on consensus clustering and multi-objective evolutionary algorithm for sentiment classification. *Inf Process Manag*. 2017;53(4):814-833.
29. Xu K, Ba J, Kiros R, et al. Show, attend and tell: neural image caption generation with visual attention. Paper presented at: International Conference on Machine Learning; 2015; Lille, France.
30. Yang Z, Yang D, Dyer C, He X, Smola A, Hovy E. Hierarchical Attention Networks for Document Classification. *Paper presented at: The Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies; San Diego, CA, USA*. 2016. https://aclanthology.org/N16-1174.
31. Rush AM, Chopra S, Weston J. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:150900685*; 2015.
32. Lu J, Yang J, Batra D, Parikh D. Hierarchical question-image co-attention for visual question answering. Paper presented at: Advances in Neural Information Processing Systems; 2016; Barcelona, Spain.
33. Wang W, Pan SJ, Dahlmeier D, Xiao X. Coupled multi-layer attentions for co-extraction of aspect and opinion terms. Paper presented at: Thirty-First AAAI Conference on Artificial Intelligence; 2017; San Francisco, CA, USA.
34. Ying H, Zhuang F, Zhang F, et al. Sequential recommender system based on hierarchical attention network. Paper presented at: IJCAI International Joint Conference on Artificial Intelligence; 2018; Stockholm, Sweden.
35. dos Santos C, Tan M, Xiang B, Zhou B. Attentive pooling networks. *arXiv preprint arXiv:160203609*; 2016.
36. Kadlec R, Schmid M, Bajgar O, Kleindienst J. Text understanding with the attention sum reader network. Paper presented at: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics; Vol. 1; Long Papers; 2016; Berlin, Germany.
37. Shaheen Z, Wohlgenannt G, Filtz E. Large scale legal text classification using transformer models. *arXiv preprint arXiv:201012871*; 2020.
38. Chang W-C, Yu H-F, Zhong K, Yang Y, Dhillon IS. Taming pretrained transformers for extreme multi-label text classification. Paper presented at: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining; 2020; New York, NY, USA.
39. Yildiz B, Tezgider M. Learning quality improved word embedding with assessment of hyperparameters. Paper presented at: European Conference on Parallel Processing; 2019; Göttingen, Germany.
40. Yildiz B, Tezgider M. Improving word embedding quality with innovative automated approaches to hyperparameters. *Concurrency and Computation: Practice and Experience*. 2021. http://dx.doi.org/10.1002/cpe.6091.

How to cite this article: Tezgider M, Yildiz B, Aydin G. Text classification using improved bidirectional transformer. *Concurrency Computat Pract Exper*. 2021;e6486. https://doi.org/10.1002/cpe.6486