Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.033647

Article

Text Simplification Using Transformer and BERT

Sarah Alissa[1,*] and Mike Wald[2]

[1]College of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University,
Dammam, Saudi Arabia
[2]School of Electronics and Computer Science, University of Southampton, Southampton, United Kingdom
*Corresponding Author: Sarah Alissa. Email: saalissa@iau.edu.sa

Received: 23 June 2022; Accepted: 06 February 2023

Abstract: Reading and writing are the main interaction methods with web content. Text simplification tools are helpful for people with cognitive impairments, new language learners, and children as they might find difficulties in understanding the complex web content. Text simplification is the process of changing complex text into more readable and understandable text. The recent approaches to text simplification adopted the machine translation concept to learn simplification rules from a parallel corpus of complex and simple sentences. In this paper, we propose two models based on the transformer which is an encoder-decoder structure that achieves state-of-the-art (SOTA) results in machine translation. The training process for our model includes three steps: preprocessing the data using a subword tokenizer, training the model and optimizing it using the Adam optimizer, then using the model to decode the output. The first model uses the transformer only and the second model uses and integrates the Bidirectional Encoder Representations from Transformer (BERT) as encoder to enhance the training time and results. The performance of the proposed model using the transformer was evaluated using the Bilingual Evaluation Understudy score (BLEU) and recorded (53.78) on the WikiSmall dataset. On the other hand, the experiment on the second model which is integrated with BERT shows that the validation loss decreased very fast compared with the model without the BERT. However, the BLEU score was small (44.54), which could be due to the size of the dataset so the model was overfitting and unable to generalize well. Therefore, in the future, the second model could involve experimenting with a larger dataset such as the WikiLarge. In addition, more analysis has been done on the model's results and the used dataset using different evaluation metrics to understand their performance.

Keywords: Text simplification; neural machine translation; transformer

1 Introduction

Reading and writing are the main interaction methods with web content. Text simplification tools are helpful for people with cognitive impairments such as aphasia, dyslexia and autism [1,2], new language learners, and children as they might find difficulties in understanding complex web content. In addition, it could improve the preprocessing step for Natural Language Processing (NLP) tasks including parsers, summarizers, machine translators, and semantic role labelers [3,2]. Making text simplification automated is a challenging task, and it has been studied since the late 90s and is still an active research area. Although the development in NLP, statistical, and machine translation has improved the simplification tasks, the current results have not yet reached a satisfactory level that could be integrated into a system [4].

Text simplification is the process of changing the complex text into more readable and understandable text. It could be based on a lexical level that substitutes difficult words with more common synonyms, rule-based which simplifies using the knowledge-base of human-created rules, or using machine translation which learns simplification rules from a parallel corpus. Most recent research adopted the machine translation idea into sentence simplification. The recent development in machine learning for natural language processing introduced unsupervised pre-trained models such as the Bidirectional Encoder Representations from Transformer (BERT) [5], which is trained on a general-purpose language understanding corpus and was able to break several records for natural language processing (NLP) tasks that it was not trained specifically on.

The main objective of this paper is to enhance automatic text simplification by maximizing the BLEU score using neural machine translation and enhancing the training time and output result by using transfer learning. So we aim to answer the following research questions:

- What is the impact of using the Transformer architecture in text simplification with the WikiSmall dataset?
- What is the impact of using pre-trained model with the Transformer in text simplification?

To answer these questions, we aim to use the Transformer [6] architecture which is used in neural machine translation tasks. The Transformer will convert the complex text into a simple text so that simplification will be done on the content and the structure of the sentence. In addition, we aim to use the pre-trained language model BERT to enhance training simplification by applying the concept of transfer learning. As BERT is an encoder built from the Transformer architecture, it could be integrated and used for the simplification task.

2 Related Work

Text simplification aims to modify the complex sentence structure or content to make it simpler and understandable to various groups of people. Modification of the sentence could include substituting the complex word with a simpler one, adding or explaining, removing unnecessary words, shortening, splitting long sentences, and dropping or merging sentences [7]. Text simplification is ongoing research and is mainly divided into three approaches: simplification based on the lexical level, rule-based simplification, and simplification using machine translation. Lexical simplification aims to simplify the complex sentence by replacing the difficult, uncommon words with alternative, simpler or most used words with the same meaning [8]. Rule-based simplification requires a manual definition of the rules for the simplification of the syntactic structure and a dictionary of predefined words [7]. The above approaches lack the simplification on both content and structure and require human involvement. Simplification using machine translation means using an automated translator system to translate complex text into simple text and aims to simplify both the sentence content and structure. Previous research has used one approach or has combined multiple approaches.

2.1 Text Simplification Using Phrase-Based Machine Translation

The Phrase-based statistical machine translation was dominant for machine translation tasks [9]. It considers translation as a machine learning task where an algorithm is trained on a parallel corpus and then the learned model can translate new sentences that it was not trained on. In phrase-based MT, a phrase table is constructed containing phrase pairs mapped to their translation with their probabilities using the parallel corpus [10].

Text simplification using machine translation was first used with the Phrase-Based Statistical Machine Translation (PBMT) system. The Phrase-Based Statistical Machine Translation with re-ranking (PBMT-R) model proposed by [10] was trained on a parallel corpus of simple and complex sentence pairs and learned to generate several best-simplified sentences which were then re-ranked depending on the dissimilarity between them and the complex sentence. Next, the Hybrid model was proposed by [3] which also used a phrase-based machine translation (PBMT) to learn to substitute the complex words in the sentence with simpler ones and change the sentence structure. Furthermore, the models are integrated with a language model to learn to split the sentence or delete part of it to improve the fluency of the sentence and the grammar.

2.2 Text Simplification Using Neural Machine Translation

Neural Machine Translation was then introduced as an approach that can overcome the weaknesses of the Phrase-based machine translation system [11]. The phrase-based MT uses a phrase table that contains each phrase and its translation with their probabilities calculated using the frequency of how many times this phrase was translated to that phrase. Also, it requires three systems: a translation model, a language model, and a reordering model. The neural machine translation uses a single, large neural network to learn the translation by mapping the source input to the target output [11] without the need for predefined rules as it can capture the simplification rules by itself [7].

The recent advances in machine translation using neural networks encourages adapting it in text simplification to learn simplification rewrite operations from complex and simple sentence pairs and achieve encouraging results. This could be done by considering the complex text as a language that will be translated to another language which is the simple text. Zhang et al. [12] proposed two models; (DRESS) which stands for Deep Reinforcement Sentence Simplification, and (DRESS-LS) which integrates lexical simplification. Both models used Neural machine translation to learn the simplification rule from a parallel corpus. They use a neural encoder-decoder model that is implemented using Long Short-Term Memory (LSTM) recurrent neural networks. Also, they try to enhance how the model learns to simplify by using reinforcement learning; the model is rewarded if the generated sentence is simple, fluent, and preserves the meaning.

Following this [13] tried to improve the simplification with neural machine translation by augmenting the model architecture with a memory to remember the dependencies between the words when the sentence is long or complicated. They proposed two models implemented using (LSTM) and Neural Semantic Encoders (NSE), the (LSTMLSTM-B) model consists of an attention-based encoder and a decoder both using (LSTM) layers. The second model (NSELSTM-B) consists of an encoder implemented using (NSE) layer and a decoder with (LSTM) layer.

Afterwards, when the Transformer was published it used simpler architecture with feedforward neural networks instead of the recurrent neural network, and its ability to simultaneously process the input sequence which allows for training the model faster. The Transformer consists of an encoder and a decoder, the encoder maps the sequence input (X1; . . . ; Xn) to a sequence of representations vectors

(V1; . . . ; Vn), and all the inputs are processed in parallel, then the decoder generates a sequence output (Y1; . . . ; Yn) using the representation vector. The output is generated one item at a time [6].

Zhao et al. [14] proposed a model using the Transformer architecture for machine translation, but they integrated it with a database of human-created paraphrase simplification rules. The model size was 4 layers of encoders and decoders, 5 attention layers and the dimension of the vector produced is 300 as they initialize it with GloVe, a pre-trained vector representation of 300 dimensions. Their models were trained on two datasets, the WikiLarge and the Newsela dataset, and they optimized them using the Adagrad [15] optimizer to update the network weights during training. Omelianchuk et al. [16] introduced a simplification model by tagging using transformer-based encoders, which follow the edit-based approach not the machine translation approach. Their model was used with the WikiSmall dataset but not evaluated with BLEU.

This paper differs from the work of Zhao et al. [14], in that it will use different datasets and training details such as the optimizer and the number of layers and attention head and hidden units, and will integrate BERT.

2.3 The BERT Model

The BERT model is a language representation model that refers to Bidirectional Encoder Representations from Transformer. It is an encoder from the Transformer that is pre-trained on a large corpus which is the BooksCorpus [17] and the English Wikipedia. This model was fine-tuned and used in different natural language processing (NLP) tasks that it was not trained specifically on and achieved state-of-the-art results, such as the task of natural language inference and the task of question answering [5]. One of the distinguishing features of BERT is that it can be used to generate contextualized word embeddings. Contextualized word embeddings produce different word representation vectors depending on the context of the sentence and the word position [5].

Qiang et al. [8] proposed a lexical simplification approach using the BERT model. They exploit the masked language model task that BERT was trained on to mask the complex word on a sentence, then the BERT is used to get the probability distribution of the words that are equivalent to the masked word. This paper aims to use BERT model in text simplification using monolingual machine translation, not lexical simplification.

3 Experimental Setup

3.1 Datasets

Mainly in text simplification, three datasets are used in experiments, the WikiSmall, the WikiLarge which are available in the project [12] and the Newsela datasets [13]. Due to time and hardware limitations, only the WikiSmall dataset was used. The WikiSmall dataset is broadly used as a benchmark in text simplification and it contains parallel sentence pairs (complex and simple) collected automatically from the Main English Wikipedia and Simple English Wikipedia [12]. The simple English Wikipedia was written using easy words and short sentences to target audiences that are children or new English learners [18]. There are about 89,042 sentences for training, 205 for development and 100 for testing. Table 1 shows a sample of the complex text and its simpler form from the WikiSmall dataset.

3.2 Training Details

This paper aims to use two models; the first is the Transformer model and the second is the Transformer with integrating BERT model. The training process includes three steps: preprocessing the data, training the model, and using the model to decode the output. The training process for the model integrated with BERT is the same, but it will be explained separately.

Table 1: Sample of WikiSmall complex and simple text

Complex text	Simple text
Genetic engineering has expanded the genes available to breeders to utilize in creating desired germlines for new crops.	New plants were created with genetic engineering.
Wikipedia is free content that anyone can edit and distribute.	Wikipedia is free content that anyone may change.
A material such as gold, which is chemically inert at normal scales, can serve as a potent chemical catalyst at nanoscales.	A material such as gold, which does not react with other chemicals at normal scales, can be a powerful chemical catalyst at nanoscales.

3.2.1 Pre-processing

First, to train the Transformer model, the input should be tokenized and indexed. Following the Annotated Transformer by OpenNMT [19], the English tokenizer from SpaCy [20] was used and two lists of vocabularies were created for both complex and simple datasets generated from WikiSmall by setting the minimum appearance of a word to 2. Table 2 shows the number of vocabularies in each list.

Table 2: Number of vocabularies with SpaCy tokenizer

Vocabulary list	Number of words
Complex vocabulary	52691
Simple vocabulary	44823

After training using this tokenizer, an out-of-vocabulary (OOV) problem was encountered, meaning the words outside the vocabulary list will have resulted in the unknown symbol <UNK>. Therefore, the subword tokenizer [21] was used which split the sentence into subword units. This tokenizer can learn to compound and generate new words out of the training vocabulary list. The implementation that was used for the subword tokenizer is the open-source SentencePiece [22]. It implements a two segmentation algorithm byte pair encoding (BPE) [21] and unigram language model [23] and both algorithms are used and it was noticed that the unigram encoding produced a better result by a small difference from the byte pair encoding. This is the same finding as [23] that the unigram language model achieved the best BLEU score, and therefore it was used in this project. The byte pair encoding differs from the unigram language model in that it is based on deterministic replacement while the unigram language model is based on probabilistic replacement which makes it more flexible so that it could generate different segmentations with their probabilities [23].

Tables 3 and 4 show the number of vocabularies for both the complex and simple vocabularies for byte pair encoding (BPE) and unigram language model respectively. The two of them used a model trained on the raw training dataset for tokenization with 30K number of merge operations for BPE and 30K final vocabulary size for unigram. Both were generated using SentencePiece on the

WikiSmall dataset. It is noticed in Tables 3 and 4 that the number of vocabularies is reduced with the subword tokenizer compared to the SpaCy tokenizer in Table 2 as many words could be generated from combining sub words units and this increased time and space efficiency [21].

Table 3: Number of vocabularies with byte pair encoding (BPE)

Vocabulary list	Number of words
Complex vocabulary	28726
Simple vocabulary	28314

Table 4: Number of vocabularies with unigram encoding

Vocabulary list	Number of words
Complex vocabulary	30020
Simple vocabulary	29566

3.2.2 Training the Model (Transformer Only)

Two different architectures were used to train the model; one is similar to the base transformer called (TF512) with N = 6, the number of an identical stack of encoder layers and an identical stack of decoder layers, the word embeddings dimension d_model = 512, and the number of parallel attention layers (head) h = 8. The second model called (TF300-G) initialized the encoder and decoder word embedding with GloVe [24] vector of 300 dimensions and 840B tokens, following previous research such as [13,12] and [14]. Therefore, the word embedding is set to d_model = 300 dimensions. This model has N = 6 stack of encoder and decoder layers, and the attention head was set to H = 12. The number of attention heads needs to be dividable by the dimension size. Also, we tried the same model size initialized randomly without initialization using Glove and called it (TF300).

For both models, training was optimized using the Adam optimizer [25]. Following the Transformer [6] training optimization, the exponential decay rates were $\beta_1 = 0.9, \beta_1 = 0.98,$ and $\varepsilon = 10^{-9}$. The learning rate was varied during the training, it was initialized with $lr = 0$, and increased linearly over the first (4000) warmup steps, and then decreased over the next steps.

The batch size used is 4096 and for the regularization dropout rates of 0.15 and 0.2 are used. Different hyperparameter values were experimented with for the learning rate, the batch size, the dropout rate, the warmup steps, the number of heads, the word embedding size and the number of stacks before fixing them as seen in Table 5.

3.2.3 Decode the Output

Beam search was used which keeps K optimal words at each decoding step, and then it returns the word with the highest score [26]. Different beam sizes were used and compared, then fixed with the beam size that produced the higher BLEU score which was 5 during training. After the model finished training, it was tried with a beam size higher and lower than 5, and the one that produced the higher BLEU score was selected.

Table 5: Training hyperparameter values

Model	Number of heads	Number of stacks	Word embedding size	Learning rate initialization	Batch size	Dropout rate	Warmup steps
TF512	8	6	512	0	4096	0.15	4000
TF300-G	12	6	300 (GloVe)	0	4096	0.2	4000
TF300	12	6	300	0	4096	0.2	4000

3.2.4 Training Using BERT

This project aims to apply the transfer learning concept in text simplification by integrating the pre-trained model BERT into the Transformer as the Transformer consists of an encoder and a decoder, and the BERT is a pre-trained encoder from the Transformer. Therefore, the (BERT-FX) model was created using the BERT encoder with fixed weights to extract fixed features from the sentences and trained the decoder only to map these extracted features with the target output. The BERT tokenizer was used to convert the sentence into tokens and the tokens to indexes using WordPiece [11]. The vocabulary size for both complex and simple sentences is 28997 which is the vocab size for the BERT tokenizer. The batch size for the model (BERT-FX) is 4096. Here the GPU used is from Google Colab which has a larger memory than the local GPU.

The (BERT-FX) model was trained and compared with a model that uses the Transformer only. Both models had the same size, d_model = 768, h = 8, N = 6 and the batch size = 4096. They were trained on 40 epochs and it is seen from Fig. 1 (the left box) that there is a difference in the maximum BLUE score between the two models; the TF768 was able to reach a score over 50, while the (BERT-FX) maximum score was about 44. In contrast, the loss value in Fig. 1 (the right box) decreased fast with BERT-FX, but this did not improve the BLEU score. The expected reason for this result could be that the BERT-FX model is overfitting as it is pre-trained so it was able to memorize the data; that is why the validation loss decreased quickly, but when used on new data the test set was not able to perform as well as the model without BERT. To overcome the overfitting problem, we tried to reduce the model size to h = 1 and N = 1, while the d_model could not be changed as it is the size of the BERT output, but still, the loss decreased fast. Next, we tried to increase the dropout to 0.5 which means that in each layer where the dropout is used 50 percent of the neurons randomly selected are dropped, which helps the model to generalize well. The loss decreased slower but the BLEU score did not improve.

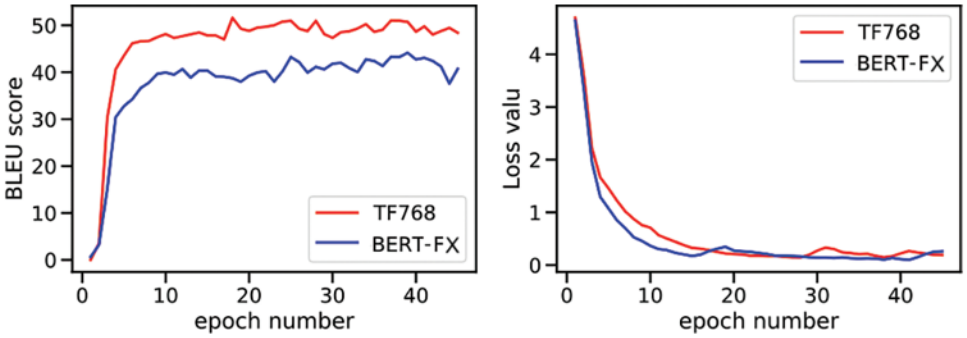

Figure 1: Training comparison between the BERT-FX model and TF768

3.3 Hardware and Schedule

The training was done on NVidia RTX2070 GPU. It was sufficient for the Transformer models, however for the BERT models, increasing the batch size could not fit in the memory, therefore Google Colab was used. Training time depends on the model, and the model took approximately 20 to 25 min to complete one epoch.

3.4 Evaluation

The system output was evaluated using an automatic metric called (BLEU) Bilingual Evaluation Understudy [27] which is used widely in text simplification research such as [12] and [13]. The BLEU measures the degree of similarity between the generated output and the standard references [12], by counting the n-gram matches which is the sequence of n words that appear on both the output and the target [13]. The score is calculated at the corpus level on the test dataset; its value is between the range 0 to 100 and the higher the score the better the simplification. The BLEU score has a high correlation with human judgment [27].

Also, the readability metric Flesch–Kincaid Grade Level (FKGL) [28] is used to evaluate the readability of the system output. The formula used to calculate the readability metric is in Eq. (1).

$$0.39\left(\frac{\text{total words}}{\text{total sentences}}\right) + 11.8\left(\frac{\text{total syllables}}{\text{total words}}\right) - 15.59 \tag{1}$$

The lower the score, the more readable the text. However, the problem with the FKGL metric is that it considers the sentence's characteristics such as the length of the word and the length of the sentence and does not take into account the grammar and the adequacy of the sentence, which leads to having a more readable score with shorter sentences although they might be grammatically incorrect and do not convey the same meaning [10].

In addition, the Translation Error Rate (TER) [29] is used following [12], which is commonly used in evaluating the quality of the sentences generated by a machine translation system. The TER was used to calculate the number of edits that are required to change the generated output by the system into the target output with the type of edit such as substituting, adding, deleting, and shifting.

SARI [30] is a common metric that is used in text simplification, which compares the generated output by the system against the source and the target sentences; by comparing with the source sentence, it ensures that the output differs from the source and not copying a lot, which BLEU does not evaluate. However this metric is not used in this project as it needs multiple references which is not applicable for the WikiSmall as it has one reference.

4 Experimental Results

This section presents the results of the automatic evaluation metrics that were used to evaluate the output of the systems, including (BLEU) the Bilingual Evaluation Understudy, (FKGL) the Flesch–Kincaid Grade Level, and (TER) the Translation Error Rate. The tables below show the results of our models; the model (TF512) which has the dimension of the input and the output d_model = 512 and beam size = 4, the model (TF300) which has d_model = 300 and beam size = 6, the model (TF300-G) which has d_model = 300, beam size = 5 and use Glove vector, the model (TF768) which has d_model = 768, beam size = 5 and the (BERT-FX) model which is integrated with BERT. The tables also present the results of the previous models for comparison, including phrase-based machine translation models such as the Hybrid [3] and the PBMT-R [10] models and the neural machine translation models such as LSTMLSTM-B, NSELSTM-B [13], DRESS, and DRESS-LS [12].

4.1 Result with BLEU

The correctness of the simplification was measured by evaluating the BLEU score which could be seen in Table 6. It shows that the Hybrid model still achieved the state of the art (SOTA) result with a score of (53.94) for models that used a phrase-based statistical machine translation. Among the neural machine translation models which are LSTMLSTM-B, NSELSTM-B, DRESS, DRESS-LS and the proposed models, the TF512 model achieves a better score than the previous highest score which is the NSELSTM-B with (53.42) score. TF512 model scores 53.78 which is close to the SOTA result. In addition, the source sentences are also tested using BLEU against the target and got a score of (49.07).

Table 7 illustrates the percentage of the number of sentences that were copied by the proposed models and not simplified. As seen from the table, about 30% were copied for the three models TF512, TF300 and TF300-G, while the lowest is the BERT-FX with 8%, and the highest is the TF768 with 43%.

Table 6: Models evaluation result using BLEU (Bilingual Evaluation Understudy)

Model	BLEU
Source	49.07
Hybrid	**53.94**
PBMT-R	46.31
NSELSTM-B	**53.42**
LSTMLSTM-B	50.53
DRESS	34.53
DRESS-LS	36.32
TF512	**53.78**
TF300	52.73
TF300-G	52.32
TF768	51.62
BERT-FX	44.54

Table 7: The percentage of the number of sentences that are not simplified by the proposed models

Model	Not simplified
TF512	31%
TF300	32%
TF300-G	32%
TF768	43%
BERT-FX	8%

4.2 Result with FKGL

Table 8 shows the results of the readability metric (FKGL) Flesch–Kincaid Grade Level. As seen from the table, the neural machine translation model DRESS has an output with the lowest score which is (7.48), then the DRESS-LS which has a close score to DRESS (7.55), followed by the Hybrid. The PBMT-R model and the reference got a close score of (11.42) and (11.08) respectively, while the proposed models were the least readable on the FKGL metric. Also, the average sentence length and the average word length is measured to compare them with the readability metric. It is seen from Table 8 that DRESS and DRESS-LS also got the shortest average sentence length (16.35) and (16.53) although they got the longest average word length (4.54) and (4.56). In contrast, the Hybrid and PBMT-R which are phrase-based machine translation models obtain the shortest average word length (4.35) and (4.48) after the neural MT model BERT-FX (4.33), however, they obtain the longest average sentence length (25.9) and (27.16). For the other proposed models of this project, the average sentence length and word length are in the middle as well as the reference average lengths. The score for LSTMLSTM-B and NSELSTM-B models were not calculated as their output was not published.

Table 8: Models evaluation results using the readability metric FKGL (Flesch–Kincaid Grade Level)

Model	FKGL	Average sent. len	Average word. len
Source	14.06	27.5	4.61
Reference	11.08	22.7	4.48
Hybrid	9.20	25.9	**4.35**
PBMT-R	11.42	27.16	4.48
LSTMLSTM-B	–	–	–
NSELSTM-B	–	–	–
DRESS	**7.48**	**16.35**	4.54
DRESS-LS	7.55	16.53	4.56
TF512	11.80	22.9	4.50
TF300	11.95	23.17	4.50
TF300-G	11.64	22.8	4.49
TF768	12.52	23.62	4.50
BERT-FX	11.58	22.25	**4.33**

4.3 Results with TER

The Translation Error Rates (TER) were analyzed, to measure the average number of operations required for a system to change its output sentences to the target sentences. The tool used is the Perl implementation by [29] which defined the measurement.

The scores results are shown in Table 9 with the average number for each type of edit; the Hybrid model got (41.10) which is the lowest TER score, followed by PBMT-R with a (44.86) score. The DRESS obtained the highest score (80.26), while the proposed models' scores were in the middle between those systems. In addition, the source sentences were also tested, and they scored (43.10).

Moreover, the TER was used to calculate the number of edits that were used by the models to change the source (complex) sentences to the models output. Table 10 shows that the DRESS model obtained the highest score, and the PBMT-R model got the lowest one. For the LSTMLSTM-B and NSELSTM-B models, their results were not reported here as their output was not published.

Table 9: Translation Error Rate (TER) results for the models output compared to the target reference

Models	TER	Ins	Del	Sub	Shft
Source	43.10	137	580	377	106
Hybrid	41.10	205	494	282	100
PBMT-R	44.86	154	559	404	115
LSTMLSTM-B	–	–	–	–	–
NSELSTM-B	–	–	–	–	–
DRESS	80.26	804	115	331	76
DRESS-LS	77.59	784	117	326	72
TF512	48.42	372	346	322	81
TF300	47.46	335	339	358	81
TF300-G	47.03	335	301	364	85
TF768	45.56	260	410	375	90
BERT-FX	55.85	399	407	414	92

Table 10: Translation Error Rate (TER) results for the source sentences compared to the models output

Models	TER	Ins	Del	Sub	Shft
Hybrid	18.42	63	217	184	49
PBMT-R	8.62	38	76	120	6
LSTMLSTM-B	–	–	–	–	–
NSELSTM-B	–	–	–	–	–
DRESS	44.54	2	1134	100	4
DRESS-LS	43.35	2	1112	90	3
TF512	22.8	24	493	99	19
TF300	19.93	17	456	76	6
TF300-G	21.51	11	488	89	11
TF768	13.86	10	303	66	7
BERT-FX	28.84	75	510	203	15

To understand the output results better, the TER score and FKGL were calculated for the training and development datasets that were used to train the proposed models. Table 11 shows the Flesch–Kincaid Grade Level (FKGL) score for the training and development datasets for both the complex and simple sentences. The complex sentences have close scores (12.29) and (12.87) for the training and development respectively. Similarly, the simple sentences have adjacent scores, the train set has (9.99) and the development set has (10.21). Table 12 shows the number of edits for each type required to transfer the complex sentence into the simple sentence for the training dataset and the development dataset.

Table 11: The Flesch–Kincaid Grade Level (FKGL) score for the training and development dataset

Dataset	FKGL	Average sent. len	Average word. len
Train complex	12.29	24.25	4.52
Train simple	9.99	20.33	4.36
Development complex	12.87	24.94	4.49
Development simple	10.21	19.60	4.37

Table 12: The Translation Error Rate (TER) score for the training and development dataset, the score is for the number of operation needed to convert the complex dataset to simple dataset

Dataset	TER	Ins	Del	Sub	Shft
Train complex	44.32	132,738	485,373	271,363	77,186
Development complex	46.87	273	1400	652	216

5 Discussion

5.1 BLEU Score

As seen from Table 6, the BLEU score is presented which measured the correctness of the simplification and how close the simplification result was to the target output. The Hybrid model got the highest score which is the state-of-the-art result (SOTA) for all the simplification models. However, the Hybrid model is implemented using phrase-based machine translation, therefore for the neural machine translation models, the proposed model TF512 obtained the highest score (53.78) exceeding the previous highest score for NSELSTM-B with (53.42). Also, the other proposed models TF300 and TF300-G got comparable results with the previous highest scores, which are (52.73) and (52.32). The difference between those two models is the (TF300-G) initialized with GloVe word embedding [24] and it was noticed that it did not improve the result but just increased the model size and the training time to load the GloVe vector. Also, the model (TF512) which has a larger size (512) got a better score than size (300), but when increased to size (768) as the models (TF768) it did not improve which got 51.62. The model (BERT-FX) which is integrated with BERT did not improve the simplification and got a score of (44.54).

An issue with the BLEU score is that it does not measure how the generated output differs from the source sentence, so the number of sentences that were copied as the source sentence without simplification is calculated. Table 7 shows that more than 30% of the test dataset were copied in the proposed models' output which got a BLEU score over 50. The proposed models achieved high BLEU scores although about 30 percent of the sentences were copied, therefore, the BLEU score for the source sentences were calculated as seen from Table 6 and got (49.07) and it is higher than some of the simplification systems outputs. This high score for the source sentences could be reasonable, as simplification does not always make large changes to the sentence; it might substitute one word or shorten the sentence. This means that the copied sentences could get some score that could increase the total score of the output. As a result, other measurements were used to further evaluate the output results.

5.2 *FKGL Score*

The readability metric Flesch–Kincaid Grade Level (FKGL) is measured and is presented in Table 8. As the smaller the score, the better the readability, so the models DRESS and DRESS-LS got good readability compared to the other systems. The proposed models did not perform well as they had the highest score, which could be due to the fact that more than 30 percent of the sentences were not simplified, and it is seen that the model (BERT-FX) which has only 8 percent of the sentences that were not simplified got the lowest score among the proposed models. The Hybrid model which got the highest BLEU score got a score in the middle and is better than the target reference. However, the FKGL considers the length of the word and the length of the sentence, and so a model that got a high readability score may not preserve the meaning or has grammatical errors [10].

Therefore, the average length of the words and the average length of the sentences were measured to have a better analysis of the FKGL results which could be seen in Table 8. The models DRESS and DRESS-LS are the best at generating sentences with shorter lengths with an average of (16.35) and (16.53) number of words in a sentence respectively. This could be the reason for having a good FKGL score, however, their words selection was not the shortest among the other models. The Hybrid was the best at selecting shorter words with an average of (4.35) characters in a word, after the BERT-FX models with an average of (4.33). This was followed by PBMT-R with an average of (4.48) which is the same as the Reference average word length. On the other hand, the Hybrid and PBMT-R generated an average sentence length that was not as good as they have the longest average sentence length. For the DRESS and DRESS-LS which use reinforcement learning, they were trained by rewarding the system and encouraging it to apply other operations such as deleting more than copying, hence they may generate a shorter sentence. This also is shown in Table 10; the DRESS and DRESS-LS models perform the deletion operation about double the number of times when compared to the proposed models and more than that for the Hybrid and PBMT-R models. For the Hybrid and PBMTR models which use a phrase-based machine translation system, it might be that the words with shorter lengths have assigned higher probability, so the models were trained to select the shortest word.

For the proposed models' results, they are the closest to the Reference in terms of both the average sentence length and the average word length. This could be because the system is trained to produce an output that follows the reference pattern. The LSTMLSTM-B and NSELSTM-B output is not published, so it could not be added to the comparison. However, the shorter the sentence length or the word length does not always imply better simplification as stated by the guidelines of the Simple Wikipedia, "simpler does not mean short" [10]. Therefore, more analysis is done to investigate the simplification output and whether it requires a lot of edits to change it to the target, and how much is changed from the source to the simplified output and whether this change enhances it or makes it close to the target output.

5.3 *TER Score*

The Translation Error rate (TER) shown in Tables 9 and 10, indicates that the phrase-based machine translation models including the Hybrid and PBMT-R which had the lowest TER score, require the least operation changes to be similar to the target. In contrast, the DRESS and DRESS-LS have the highest TER score which implies that they need a lot of work and changes to reach the target despite them performing the highest average of operation; but it might be not the correct one. As seen in Tables 9 and 10, they perform deletion more, while they need to increase the insertion operation. Zhang et al. [12] found similar performance regarding the high number of deletion operations performed by the DRESS and DRESS-LS and the small number performed by the PBMT-R model, although their result was in the Newsela dataset, not the WikiSmall. The proposed models' results were also in the middle; they performed few insertions, shifting operations, and more deletions but still need to delete more, possibly because of the not simplified sentences. In general, all the models perform a high average of deletion, however, they need to learn insertion and substitution more as well.

Therefore, the training and development datasets that the models were trained on, are further analyzed using the Flesch–Kincaid Grade Level (FKGL) and Translation Error Rate (TER). It is seen in Table 11 that the models were trained to reach a readability score of about (10) with an average sentence length of about (20) and an average word length (3.365). From Table 12, it is seen the rewrite operation with the highest average is the deletion which could be the reason the models learned to use it more.

5.4 Sample Output

Table 13 shows one sample from the WikiSmall dataset test set; it presents the source sentence with the target sentence and all the models' simplification output. The proposed model TF512 was able to perform rewrite operations such as deletion and substitution.

Table 13: Sample output for all the models

Source sentence	It was founded in the 14th century by Genoese colonists, who employed large numbers of workmen (Calfats) in repairing ships.
Target sentence	Calafat was started in the 14th century by Genoese colonists.
Hybrid	It was founded in the 14th century by genoese colonists, who used large numbers of workmen (calfats) in repairing ships.
PBMT-R	It was founded in the 14th century by Genoese colonists workmen (Calfats) in repairing ships.
DRESS	It was founded in the 14th century by Genoese colonists.
DRESS-LS	It was founded in the 14th century by Genoese colonists.
TF512	Calafat was started in the 14th century by Genoese colonists.
TF300	Calfats were started in the 14th century by Genoese colonists.
TF300-G	Calfat was started in the 14th century by Genoese colonists.
TF768	Calfat was started in the 14th century by Genoese colonists.
BERT-FX	Calafat was founded in the 14th century by Genoese colonists.

6 Conclusion

To answer the paper' research questions we used the Transformer, the latest neural machine translation architecture that achieved state-of-the-art results in machine translation with the text simplification task using the WikiSmall dataset and tried to integrate the pre-trained model BERT in text simplification using a machine translation approach. The BERT model also achieved state-of-the-art results in a different task. Therefore, two architectures were used, the first architecture used the Transformer only, and the second architecture was integrated with BERT.

The findings of using these models are, text simplification could be enhanced using the Transformer, as one of the proposed models (TF512) obtains a BLEU score (53.78) that exceeds the previous highest score for neural machine translation model for (NESLSTM) which got (53.42), but it did not exceed the SOTA result for the Hybrid model (53.94) which use phrase-based machine translation. Unfortunately, the proposed models did not decrease the FKGL score nor the TER score, due to the fact that 30% of the sentences were not simplified. In addition, the experiment shows that integrating BERT increased the training time as the model size should be increased to use the output of the BERT and encoding the input using BERT took a longer time. Nevertheless, the scores did not improve with integrating BERT as the maximum BLEU score reached is (44.54) and this could be due to the overfitting problem, as the validation loss was decreasing faster with the model (BERT-FX) compared to the model (TF768) which has the same size but without BERT. So the model maybe was memorizing the data instead of generalizing and by this, the BLEU score did not increase.

7 Future Work

One of the problems found with the proposed models is copying a lot of the sentences without simplification; it was found later was that Google's neural machine translation [11] used coverage penalty in their beam search algorithm to encourage the model to translate (cover) all the words in the sentence, which could help to simplify all sentence. However, due to time limitations, this was not tried but it could further be investigated; also analyzing the sentences that were not simplified such as what operation they require, the length of the sentence, and the average length of the words could help to learn the reason why the model was not able to simplify them and how to improve it.

Another problem was with integrating the BERT model where it overfits during training; the small size of the dataset could be the reason, therefore using a larger dataset in the future such as the WikiLarge could produce a better result. Also, the training dataset could have an impact on the way the model learns, so producing a better dataset could enhance the training. Lastly, implementing a program that uses the simplification model as a suggestion simplification tool could be future work.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] L. Martin, É. V. de la Clergerie, B. Sagot and A. Bordes, "Controllable sentence simplification," in *Proc. of LREC, 2020—12th Int. Conf. on Language Resources and Evaluation*, Marseille, France, pp. 4689–4698, 2020.

[2] H. Guo, R. Pasunuru and M. Bansal, “Dynamic multi-level multi-task learning for sentence simplification,” in *Proc. of COLING, 2018—27th Int. Conf. on Computational Linguistics*, Santa Fe, New-Mexico, USA, pp. 462–476, 2018.

[3] S. Narayan and C. Gardent, “Hybrid simplification using deep semantics and machine translation,” in *Proc. of the Conf. 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014*, Baltimore, Maryland, USA, vol. 1, pp. 435–445, 2014.

[4] S. S. Al-Thanyyan and A. M. Azmi, “Automated text simplification: A survey,” *ACM Computing Surveys*, vol. 54, no. 2, pp. 43:1–43:36, 2021.

[5] J. Devlin, M. W. Chang, K. Lee and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *Proc. of the Conf. NAACL HLT 2019—2019 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Minneapolis, MN, USA, pp. 4171–4186, 2019.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones *et al.,* “Attention is all you need,” in *Proc. of 31st Conf. on Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, USA, pp. 5999–6009, 2017.

[7] T. Wang, P. Chen, J. Rochford and J. Qiang, “Text simplification using neural machine translation,” in *Proc. of AAAI, 2016—30th AAAI Conf. on Artificial Intelligence*, Phoenix, Arizona, USA, pp. 4270–4271, 2016.

[8] J. Qiang, Y. Li, Y. Zhu, Y. Yuan and X. Wu, “Lexical simplification with pretrained encoders,” in *Proc. of AAAI, 2020—34th AAAI Conf. on Artificial Intelligence*, New York, NY, USA, pp. 8649–8656, 2020.

[9] A. Lopez, “Statistical machine translation,” *ACM Comput. Surv.*, vol. 40, no. 3, pp. 1–49, 2008. https://doi.org/10.1145/1380584.1380586

[10] S. Wubben, E. Krahmer and A. Van den Bosch, “Sentence simplification by monolingual machine translation,” in *Proc. of the 50th Annual Meeting of the Association for Computational Linguistics*, Jeju Island, Korea, pp. 1015–1024, 2012.

[11] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi *et al.,* “Google’s neural machine translation system: Bridging the gap between human and machine translation,” arXiv, 2016. [Online]. Available: http://arxiv.org/abs/1609.08144

[12] X. Zhang and M. Lapata, “Sentence simplification with deep reinforcement learning,” in *Proc. of EMNLP, 2017—Conf. on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark, pp. 584–594, 2017.

[13] T. Vu, B. Hu, T. Munkhdalai and H. Yu, “Sentence simplification with memory-augmented neural networks,” in *Proc. of NAACL HLT, 2018—2018 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, New Orleans, LA, USA, vol. 2, pp. 79–85, 2018.

[14] S. Zhao, R. Meng, D. He, S. Andi and P. Bambang, “Integrating transformer and paraphrase rules for sentence simplification,” in *Proc. of EMNLP, 2018— The 2018 Conf. on Empirical Methods in Natural Language Processing*, Brussels, Belgium, pp. 3164–3173, 2018.

[15] A. Lydia and S. Francis, “Adagrad-An optimizer for stochastic gradient descent,” *International Journal of Information and Computing Science*, vol. 6, no. 5, pp. 566–568, 2019.

[16] K. Omelianchuk, V. Raheja and O. Skurzhanskyi, “Text simplification by tagging,” in *Proc. of the 16th Workshop on Innovative Use of NLP for Building Educational Applications, BEA, 2021—Held in Conjunction with the 16th Conf. of the European Chapter of the Association of Computational Linguistics, EACL 2021*, Online, pp. 11–25, 2021.

[17] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun *et al.,* “Aligning books and movies: Towards story-like visual explanations by watching movies and reading books,” in *Proc. of the IEEE Int. Conf. on Computer Vision*, Santiago, Chile, vol. 2015, pp. 19–27, 2015.

[18] Z. Zhu, D. Bernhard and I. Gurevych, “A monolingual tree-based translation model for sentence simplification,” in *Proc. of Coling 2010—23rd Int. Conf. on Computational Linguistics*, Beijing, China, vol. 2, pp. 1353–1361, 2010.

[19] G. Klein, Y. Kim, Y. Deng, J. Senellart and A. M. Rush, “OpenNMT: Open-source toolkit for neural machine translation,” in *Proc. of ACL, 2017—55th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, Vancouver, Canada, pp. 67–72, 2017.

[20] M. Honnibal and I. Montani, “spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing,” 2017. To appear. https://doi.org/10.5281/zenodo.1212303

[21] R. Sennrich, B. Haddow and A. Birch, “Edinburgh research explorer neural machine translation of rare words with subword units neural machine translation of rare words with subword units,” in *Proc. of the 54th Annual Meeting of the Association for Computational Linguistics*, Berlin, Germany, pp. 1715–1725, 2016.

[22] T. Kudo and J. Richardson, “SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing,” in *Proc. of EMNLP, 2018—Conf. on Empirical Methods in Natural Language Processing: System Demonstrations*, Brussels, Belgium, pp. 66–71, 2018, https://doi.org/10.18653/v1/d18-2012

[23] T. Kudo, “Subword regularization: Improving neural network translation models with multiple subword candidates,” in *Proc. of ACL, 2018—56th Annual Meeting of the Association for Computational Linguistics*, Melbourne, Australia, vol. 1, pp. 66–75, 2018.

[24] J. Pennington, R. Socher and C. Manning, “GloVe: Global vectors for word representation,” in *Proc. of the 2014 Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, pp. 1532–1543, 2014.

[25] D. P. Kingma and J. L. Ba, “Adam: A method for stochastic optimization,” in *Proc. of the 3rd Int. Conf. on Learning Representations, ICLR, 2015—Conf. Track Proceedings*, San Diego, CA, USA, pp. 1–15, 2015.

[26] I. Kulikov, A. H. Miller, K. Cho and J. Weston, “Importance of search and evaluation strategies in neural dialogue modeling,” in *Proc. of INLG, 2019—12th Int. Conf. on Natural Language Generation*, Tokyo, Japan, pp. 76–87, 2019.

[27] K. Papineni, S. Roukos, T. Ward and W. -J. Zhu, “BLEU: A method for automatic evaluation of machine translation,” in *Proc. of the 40th Annual Meeting on Association for Computational Linguistics*, Philadelphia, PA, USA, pp. 311–318, 2002.

[28] J. P. J. Kincaid, R. F. R. L. Rogers Jr., B. S. B. Chissom and R. P. Fishburne Jr., “Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel (No. RBR-8-75). Naval technical training command millington TN research branch,” *Naval Technical Training Command Millington TN Research Branch*, 1975. https://doi.org/10.21236/ADA006655

[29] M. Snover, B. Dorr, R. Schwartz, L. Micciulla and J. Makhoul, “A study of translation edit rate with targeted human annotation,” in *Proc. of AMTA 2006—The 7th Conf. of the Association for Machine Translation of the Americas: Visions for the Future of Machine Translation*, Cambridge, MA, USA, pp. 223–231, 2006.

[30] W. Xu, C. Napoles, E. Pavlick, Q. Chen and C. Callison-Burch, “Optimizing statistical machine translation for text simplification,” *Transactions of the Association for Computational Linguistics*, vol. 4, pp. 401–415, 2016.