BERT-based Text Simplification Approach to Reduce Linguistic Complexity of Bangla Language

Nahid Hossain, Adil Ahnaf
Computer Science and Engineering
United International University
Dhaka, Bangladesh
nahid@cse.uiu.ac.bd, aahnaf151054@bscse.uiu.ac.bd

***Abstract*—The text simplification approach simplifies the linguistic complexity of a particular language so that the grammar and structure of a language are greatly simplified to read and understand while preserving the information and underlying meaning. Despite being spoken globally and having a rich history of Bangla literature, there is no work has been done in the Bangla language on this important topic. The work has been done to increase the number of Bangla literature readers and save Bangla historical writings from becoming extinct. We have also collected and used an extensive corpus consisting of 1,52,230 sentences along with a lexicon consisting of 22,580 complex-simple unique word pairs, which are mapped manually. This paper has presented two text simplification models based on Long Short Term Memory (LSTM) and Bidirectional Encoder Representations from Transformers (BERT). However, the proposed model based on BERT shows a satisfactory accuracy rate of 95.3%.**

***Index Terms*—Bangla, Text Simplification, Linguistic Complexity, LSTM, BERT**

I. Introduction

Nowadays, we barely show our interest in literature for the propagation of technologies; however, we often are fascinated to read old literature, i.e., old novels, but we hardly could gain anything out of it due to lack of linguistic knowledge or enough understanding of archaic words. As a result, those masterpiece pieces of literature remain unknown and neglected. The swiftness of any language never halts and always moving forward. Bangla (also known as Bengali) is the 6th most spoken language with nearly 228 million native speakers and around 27 million people who speak it as a second language around the world [1] [2].

Bangla language has a rich literary history with numerous famous poets, writers, composers, and philosophers, including Nobel prize winner Rabindranath Tagore, Michael Madhusudan Dutt, Kazi Nazrul Islam, Shirshendu Mukhopadhyay, Bankim Chandra Chattopadhyay, Sarat Chandra Chattopadhyay, and many more [3]. However, many works, including poems and novels of these famous poets and writers, are highly complicated to read and understand due to their linguistic complexity and use of uncommon and deprecated lexical resources. For example, although, "তাহার দীর্ঘ ঘোর কৃষ্ণ কেশরাশি দিয়া জল ঝরিতছে। (Tāhāra dīrgha ghōra kr̥ṣṇa kēśarāśi diẏā jala jharitēchē.)" and "তার লম্বা ঘন কালো চুল থেকে পানি ঝরছে। (Tāra lambā ghana kālō cula thēkē pāni jharachē.)" both are Bangla sentences with the exact same meaning, the second sentence is a lot easier to read and understand. The example sentence is from the novel 'Krishnakanter Will' by the famous novelist Bankim Chandra Chattopadhyay. It would be immensely pitiful if these resources get extinct over time due to the lack of readers. This motivates us to develop the model to make these poems, novels, and articles more readable and easily understandable by reducing the linguistic complexity.

Bangla is a mixed language of Bangladesh and Eastern India and belongs to the Indo-Aryan language family, which is a part of the Indo-European group of languages. Bangla is much difficult to learn and understand than other languages such as Danish or French because of its arduous grammar patterns, difficult pronunciation, and complex vocabulary. Since it is a rapidly growing language, it is important to build a system that can automatically generate simple comprehensive text of highly complicated texts, i.e., poems and novels, for the sake of better understanding and bring simplicity for education to a large measure of people. According to our study, several works have been done for Bangla text summarization and other parts of natural language processing. However, despite having high importance, no attempt is yet taken to simplify linguistic complexity in Bangla texts. Here, we mean to simplify the complex meaning lying in a sentence rather than recapitulating the texts.

This paper proposed a system that simplifies text for the Bangla language without changing the meaning or shortening it. Our system can take a text either manually or from an uploaded text file and measure each sentence and find the difficulty level, and finally, it converts it into a simplified version where necessary. In the beginning, in order to develop the Bangla text simplification model, we have implemented a model with LSTM. However, it shows several limitations and inaccuracies. Therefore, later, we implemented another model with BERT, which shows significantly higher performance than the model with LSTM. The notable contributions of the project is:

- Developing the first text simplification approach in Bangla language.
- Developing two models using the cutting-edge technology (LSTM and BERT) to compare which model perform

978-1-6654-3305-1/21/$31.00 ©2021 IEEE

better for Bangla language.
- Developed a sufficiently large monolingual corpus of archaic ad linguistically complex Bangla texts.

This paper is organised in the following manner: In section II, related works have been discussed. Section III demonstrates our proposed method, including dataset, mapping words, preprocessing, model architecture, and methodology. In section IV, experimental results and analysis of the system are described. Finally, we conclude the paper by bringing up the limitations, including future works on the text simplification system in section V.

II. Related Works

As we have mentioned earlier, there is no work has been done on text simplification to reduce the linguistic complexity of the Bangla language. However, there are quite a few works that have been done on English and other languages. We have taken the help of the following papers for a better understanding of text simplification methodologies. Julia Suter et al. [4] proposed a rule-based text simplification model by the University of Zurich for the German language in 2016. Every rule works on each character and adjusts the output layout. In the same year, Tong Wang et al. [5] described an study of the LSTM-based model for text simplification. This study shows several operational rules such as sorting, reversing, replacing sentence pairs, etc. They describe how RNN and LSTM work with experiments. Their model can separately make operations on sorting, reversing, or replacement. To simplify, it needs to use a combination of all three processes. In 2017, Yaoyuan Zhang et al. [6] projected a seq2seq neural model that takes part in the original sentence and preprocesses it into a vector of integers. They used a bi-directional RNN algorithm, and gated recurrent unit cells are used to show output. In the same year, Shuming Ma and Xu Sun [7] developed a model for text summarization and text simplification. Their goal was to improve the semantic relevance between the source text and generated simplified text based on a neural network model. As a dataset, they used Large Scale Chinese Short Text Summarization (LCSTS). In this model, the encoder represents source text, and the decoder represents the generated text. They claim that their model is better than the state-of-the-art systems. In the same year, Laurens et al. [8] introduced a dataset to aid medical text simplification research. Moreover, they use the dataset to train a MT model and compare it to a model trained on the text simplification dataset. Daiki Nishihara et al. [9] proposed a controllable text simplification with lexical constraint loss in 2019. To train the model, they used a publicly available dataset. The model implemented on s2s+grade, which adds Term Frequency-Inverse Document Frequency-based word weighting to the loss function. In 2020, Takumi Maruyama et al. [10] developed a text simplification model for the Japanese language when the resources are extremely low. They used a unidirectional model including GPT37 with an article from Japanese Wikipedia for pre-training. Robert Mihai et al [11] proposed a sequence-to-sequence models for automated text simplification based on paraphrasing. According to the authors, their best model achieved a high BLEU score based on universal transformer architecture.

III. Proposed Method

This section has described the detailed methodology to develop the proposed text simplification approach to reduce the linguistic complexity of the Bangla language.

A. Corpus and Lexicon

To create a monolingual corpus of archaic and linguistically complex texts, data are collected as plain text from several online sources mentioned in Table I.

Table I: Source of data.

SL	Source Name	URL
1	Pratilipi	https://bengali.pratilipi.com
2	Storymirror	https://storymirror.com/read/bengali/story
3	Fussilatbd	https://www.fussilatbd.com/Story-Book.php
4	Banglalovestory	https://www.banglalovestory.in
5	Bangla Amader	https://banglaamader.com/bangla-golpo

Our corpus consists of 72 novels, poems, short stories, and articles from famous poet and writer Rabindranath Tagore, Kazi Nazrul Islam, Michael Madhusudan Dutt, Sarat Chandra Chattopadhyay, Shirshendu Mukhopadhyay, Baren Gangopadhyay, Buddhodeb Bashu, Shahidulla Kayser, Bankim Chandra Chattopadhyay, and Uttam Ghosh. After extracting plain texts from these sources, we find 1,52,230 sentences (1,033,623 words). A portion of the short story 'Bajra' by Baren Gangopadhyay in our corpus has been shown below.

"পঞ্চাশ বছর আগে হলে দেখা যেত এই বজরাখানারই গমক কত! ঝাড়লণ্ঠনের নিচে ফরাস বিছিয়ে সারেঙ্গীর উপর ছড় টানতো বড়ে মিঞা হারমোনিয়মের বেলো ছেড়ে দিয়ে ফাঁক বুঝে আতরমাখা পান তুলে নিত প্রেমচাঁদ আর রুপোর থালা সামনে মেলে ধরে রূপসী বাঈজীর মাতাল করা গান শুনতে শুনতে মেজকর্তা চেঁচিয়ে উঠতেন কেয়াবাত কেয়াবাত ফুলবাই মেরি জান। হাঁ হে ওস্তাদ, অমন মিইয়ে পড়ছ কেন? সরাব টরাব ছোঁও না, শেষটায় বেলাইন পাকড়ে বসলে-আঁ?"

The text portion above shows how complicated the literature can be to read and understand. On the other hand, we have created a lexicon of complex-simple word pairs by manually mapped 22,580 unique words from our corpus. It is a combined effort by volunteers and ourselves. After constructing the lexicon, we spent an ample amount of time proofreading and making sure there are no spelling errors in the lexicon. This lexicon has been used to replace a complex lexeme with the corresponding mapped simple lexeme. Table II shows few samples of complex-simple word pairs from the lexicon and Table III summarizes the corpus and lexicon sizes.

Table II: Samples of complex-simple word pairs

Complex	Simple
জ্বালাইয়া (Jālā'iẏā)	জ্বালিয়ে (Jāliẏē)
ফলস্বরূপ (Phalasarūpa)	ফলে (Phalē)
পুষ্ট (Puṣṭa)	পাকা (Pākā)
তাহাদের (Tāhādēra)	তাদের (Tādēra)
জলদ (Jalada)	মেঘ (Mēgha)
সরসী (Sarasī)	দীঘি (Dīghi)

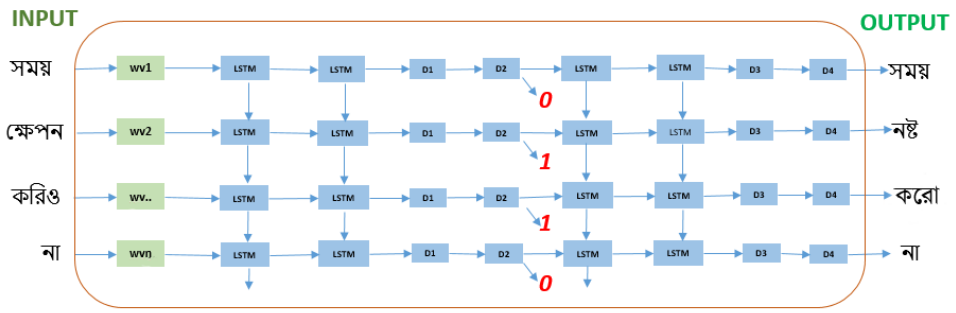

Figure 1: Pre-processing of LSTM.

Table III: Corpus and Lexicon.

Dataset	Number of Record
Corpus	1,52,230 sentences or 1,033,623 words
Lexicon(pair of words)	22,580 words

B. Data Preprocessing

Data preprocessing plays a vital role in our model. That ensures the quality of the model. Bangla language is a multi-byte language, and it has complex grammatical rules. Therefore, preprocessing is an essential task before implementing the model. The following steps have been followed before model preparation:

- Numeric Values: To fit the dataset into our model, we represented every unique word of the dataset into a unique numeric value. On the other hand, for creating training samples, a list is created for each sentence of the dataset, which contains three features, the first one is the original input sentence, the second one is the corresponding numeric value of each word in the sentence, and the final one is the auxiliary output that indicates the difference between input and output. An example is shown Table IV.

Table IV: Example of numeric values.

1.	Input:	[ইসকুল এ যাইবো ।]
2.	Output:	[150 85 76 52]
3.	Auxiliary Output:	[1 0 1 0]

- Generator: There is some additional preparation to be done before using the data in the model. The main purpose of the generator is to padding the sentences into a fixed length. In this work, all sentences of the dataset are set to be the word length of 15. For sentences containing less than 15 words, the generator place zero before them to make the length consistent. The reason for placing zero on the left side of the phrase is significant. While using data in the model, weighty data needs to be closer to the output layer to avoid vanishing gradient problems. Each sample made in a generator will later go to the text model, which will then provide a 100 size 'word2vec' [12] for each word. Vectors provided by the text model are being kept in an array. Figure 2 shows an example sentence for generator.

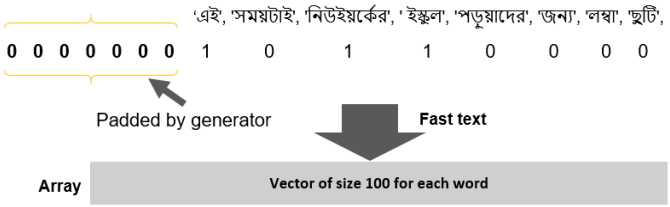

Figure 2: Example sentence for generator

- 1-Hot Vector: The numeric value assigned for each word is now converted in a 1-hot vector. A one-hot vector is a 1 × N matrix (vector) used to distinguish each word in a vocabulary from every other word in the vocabulary in natural language processing. The vector consists of 0s in all cells except for a single 1 in a cell used uniquely to identify the word. All vectors then being kept in the array. An example has been shown in Figure 3.

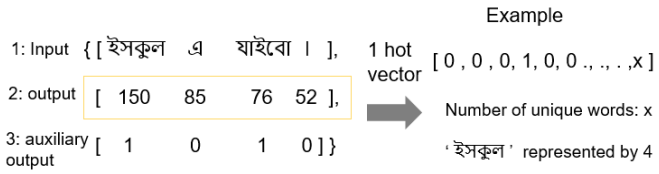

Figure 3: Example of 1-Hot Vector.

C. Model Preparation

For the text simplification problem, LSTM [13] is popular to use. LSTM is a deep learning architecture that uses an artificial recurrent neural network (RNN) [14]. We implemented the LSTM model and compared it with our proposed model, which is based on BERT [15] technique. In the beginning, we have implemented a unidirectional LSTM for pre-training our model. It can process not only single data points but also entire data sequences—for example, tasks like unsegmented, connected handwriting recognition [16]. LSTMs were created to solve the problem of vanishing gradients that can occur while training conventional RNNs. Figure 1 demonstrates the procedure. After giving input, the words are passed into a

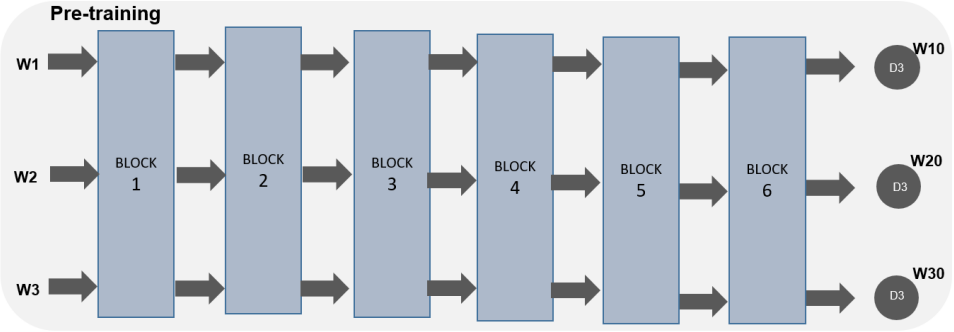

Figure 4: Pre-processing of BERT.

fast text model, and we get vector output. This module helps us to train word embedding from a training corpus, as well as get word vectors for words that are not in our vocabulary. Then each word passes through two-layer of LSTM and two dense layers. Meanwhile, we get an auxiliary vector which is a representation of each sentence that we mentioned earlier. Two dense layers are using for matrix-vector multiplication. Figure 5 shows the dense layers. Here, the output is equals to activation (dot(input, kernel) + bias), which means the dot product of input and kernel matrix. Each neuron of the dense layer receives input from all neurons of the previous layer.

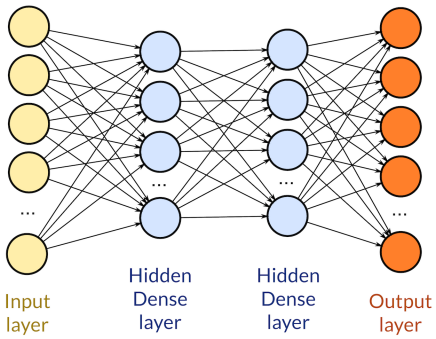

Figure 5: Dense layer.

To improve model performance, we apply fine-tuning. Fine-tuning helps boost the accuracy of a new neural network model by incorporating data from an existing neural network and using it as an initialisation point, which saves time and resources during the training phase. We apply Fine-tuning to our model that we mentioned above in Figure 1. Instead of taking the last two dense layers d3 and d4, we also added two dense layers d5 and d6 now.

Finally, we implemented our proposed model with BERT. It has been used for understanding the context of words. It makes the model learn the conceptual relations with other words, which can be left and right. For example, two sentences are 'nine to five' and 'a quarter to five', the meaning of these two 'to's are not the same, which may be obvious to humans but not for machines. Instead of using LSTM and dense layers, we have used BERT for our proposed system.

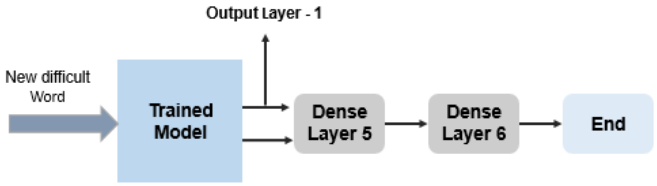

Figure 6: Fine-tuning of our trained model.

There are two parts to implementing BERT, which are pre-training and fine-tuning. This preprocessing predict an easier word for each complex word. In BERT, the transformer model is divided into two parts- an Encoder and a decoder. The encoder reads input text, where the decoder produces a prediction of appropriate text. We show six blocks in diagram Figure 4 and later on fine-tuning to incorporate missing data from the previous model. For fine-tuning, we need to use a trainer, which allows us to train the model.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, we have demonstrated the experimental result and performance analysis of our model. The experiment has been performed on Kaggle Notebook [17] that provides 5 GB disk space, 13 GB RAM and NVIDIA Tesla P100 GPU with 16 GB memory. To develop the Bangla text simplification model, first, we have implemented the model with LSTM. However, we have figured out some drawbacks of this model. The major drawback of this model is that it does not recognize longer sequences properly. That is why we have proposed another model with BERT. Sample input and output of the model have been shown in Table V.

Each model has two parts pre-training and fine-tuning. Figure 7 shows the performances of both models. For the LSTM model, the model achieved 82.8% accuracy of pre-training and achieved 86.7% accuracy on fine-tuning. On the other hand, the proposed BERT model shows 90.6% accuracy on pre-training and 95.3% accuracy after fine-tuning.

Table V: Sample input and output of the model.

Sample Input	Sample Output
মহেন্দ্র কহিলেন “ঢের কাল ক্ষেপণ হইয়াছে। এবার খানসামা কে বলিয়া ভোজনের আয়োজন করা হোক” এহেন কথা শুনিয়া মেজোবউ অশ্রুপাত করিয়া নীরবে চলিয়া গেলেন। (Mahēndra kahilēna “ḍhēra kāla kṣēpaṇa ha'iẏāchē. Ēbāra khānasāmā kē baliẏā bhōjanēra āẏōjana karā hōka” ēhēna kathā śuniẏā mējōba'u aśrupāta kariẏā nīrabē caliẏā gēlēna.)	মহেন্দ্র বললেন “অনেক সময় নষ্ট হয়েছে। এবার পরিচারক কে বলে খাবারের আয়োজন করা হোক” এই কথা শুনে মেজোবউ কান্না করে নীরবে চলে গেল। (Mahēndra balalēna “anēka samaẏa naṣṭa haẏēchē. Ēbāra paricāraka kē balē khābārēra āẏōjana karā hōka” ē'i kathā śunē mējōba'u kānnā karē nīrabē calē gēla.)
বকুলের কুঞ্জে কুঞ্জে অঞ্চল ভরিয়া ফুল তুলিত এই গাঁয়ের দুইটি বালক বালিকা। শুকতারা আকাশে ডুবিতে না ডুবিতে, উষার জলদমালা লোহিত না হইতে হইতেই সরসীর বক্ষে তরঙ্গ তুলিয়া ছিন্ন কমলদুটির ন্যায় পাশাপাশি সাঁতার দিয়া বেড়াইত। (Bakulēra kuñjē kuñjē añcala bhariẏā phula tulita ē'i gāṁẏēra du'iṭi bālaka bālikā. Śukatārā ākāśē ḍubitē nā ḍubitē, uṣāra jaladamālā lōhita nā ha'itē ha'itē'i sarasīra bakṣē taraṅga tuliẏā chinna kamaladuṭira n'yāẏa pāśāpāśi sāṁtāra diẏā bēṛā'ita.)	বকুলের লতায় লতায় আঁচল ভরে ফুল তুলতো এই গ্রামের দুটি ছেলে মেয়ে। শুকতারা আকাশে ডুবতে না ডুবতে, ভোরের মেঘ লাল না হতেই দীঘির বুকে ঢেউ তুলে ছিন্ন পদ্মদুটির মত পাশাপাশি সাঁতার দিয়ে বেড়াত।(Bakulēra latāẏa latāẏa āṁcala bharē phula tulatō ē'i grāmēra duṭi chēlē mēẏē. Śukatārā ākāśē ḍubatē nā ḍubatē, bhōrēra mēgha lāla nā hatē'i dighīra bukē ḍhē'u tulē chinna padmaduṭira mata pāśāpāśi sāṁtāra diẏē bēṛāta.)

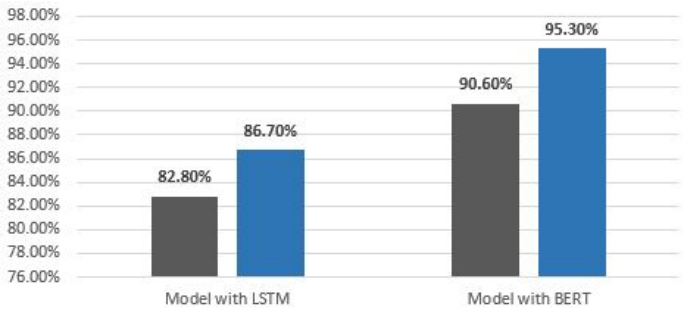

Figure 7: Performances of both models.

V. CONCLUSION AND FUTURE WORK

This paper demonstrates a comprehensive Bangla text simplification approach. The lack of a text simplification approach motivates us to develop a comprehensive text simplification approach in Bangla with the intention to increase the number of readers of Bangla ancient and complicated scripts. We have implemented two different models; however, the proposed model based on BERT gives us the best result with the accuracy of 90.6% during pre-training and 95.3% upon fine-tuning.

Although we have reached our primary goal, the work has some limitations. According to our study, the main limitation of our work is the size of the corpus and lexicon. Although the sizes are decent at this stage of the work, the model would have given more robust results with large datasets. The authors are already working to increase the size of the corpus and lexicon and fine-tune the model more precisely to achieve more precise results for the future version. Moreover, we would upload all the necessary codes and data in our GitHub repository for future researchers.

References

[1] J. Hays, “Bengalis,” http://factsanddetails.com, Jul 2017.

[2] “Summary by language size,” https://www.ethnologue.com, Feb 2021.

[3] R. Choudhury, *History of Bengali literature*. Utso Prokashan, 2010.

[4] J. Suter, S. Ebling, and M. Volk, “Rule-based automatic text simplification for german,” 2016.

[5] T. Wang, P. Chen, K. Amaral, and J. Qiang, “An experimental study of lstm encoder-decoder model for text simplification,” *arXiv preprint arXiv:1609.03663*, 2016.

[6] Y. Zhang, Z. Ye, Y. Feng, D. Zhao, and R. Yan, “A constrained sequence-to-sequence neural model for sentence simplification,” *arXiv preprint arXiv:1704.02312*, 2017.

[7] S. Ma and X. Sun, “A semantic relevance based neural network for text summarization and text simplification,” *arXiv preprint arXiv:1710.02318*, 2017.

[8] L. van den Bercken, R.-J. Sips, and C. Lofi, “Evaluating neural text simplification in the medical domain,” in *The World Wide Web Conference*, ser. WWW ’19. New York, NY, USA: Association for Computing Machinery, 2019, p. 3286–3292. [Online]. Available: https://doi.org/10.1145/3308558.3313630

[9] D. Nishihara, T. Kajiwara, and Y. Arase, “Controllable text simplification with lexical constraint loss,” in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, 2019, pp. 260–266.

[10] T. Maruyama and K. Yamamoto, “Extremely low-resource text simplification with pre-trained transformer language model,” *International Journal of Asian Language Processing*, vol. 30, no. 01, p. 2050001, 2020.

[11] R.-M. Botarleanu, M. Dascalu, S. A. Crossley, and D. S. McNamara, “Sequence-to-sequence models for automated text simplification,” in *Artificial Intelligence in Education*, I. I. Bittencourt, M. Cukurova, K. Muldner, R. Luckin, and E. Millán, Eds. Cham: Springer International Publishing, 2020, pp. 31–36.

[12] T. Mikolov, G. Corrado, K. Chen, and J. Dean, “Efficient estimation of word representations in vector space,” 01 2013, pp. 1–12.

[13] A. Sherstinsky, “Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network,” *Physica D: Nonlinear Phenomena*, vol. 404, p. 132306, 03 2020.

[14] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, pp. 1735–1780, 1997.

[15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” 10 2018.

[16] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber, “A novel connectionist system for unconstrained handwriting recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, pp. 855–868, 2009.

[17] “Notebooks documentation,” https://www.kaggle.com/docs/notebooks.