Optimizing Grammar Error Correction Performance with Attention-Based Neural Networks

Medha Joshi*

Department of Artificial Intelligence & Data Sciences, IGDTUW, Delhi 110006, India

* Corresponding author

doi: https://doi.org/10.21467/proceedings.178.27

ABSTRACT

The task of correcting grammatical errors, known as Grammatical Error Correction (GEC) is addressed in this research. Various errors in text such as punctuation, spelling, grammatical, and choice of words errors are detected and corrected. Communication is crucial in everyone's daily lives, and language is key to share information in verbal and written communication. Among all languages, English has an important position due to its global use in business, education, and entertainment. Yet, mastering its grammar can be tough for learners. With rapid commercial use of Grammarly, ChatGPT and QuillBot there is a new focus on GEC domain. In this research, the seq2seq neural networks-based on attention are applied in Grammar error correction, using a special type of LSTM model with FastText embedding at the vectorization. The attention mechanism which has been so far popular in Language translation, Computer vision and sentiment analysis has now been deployed in English language grammatical error correction. Validation of the implemented model is done on Lang-8 Corpus. Results show that attention-based model optimizes Grammatical error correction with good GLUE score and less loss.

Keywords: Grammatical Error Correction, Natural Language Processing, Grammar Checking

1 Introduction

Today, lots of people are learning English as a Second Language (ESL). Learning a new language can be hard, so when one writes, they might make grammatical mistakes. In fact, studies have looked at writings from ESL learners, where the learners write, and English language experts or teacher correct them. These studies found that there are many different types of mistakes in what the learners write [1]. While learning and enhancing knowledge of English Language, learning it through regular writing exercises is one of the most common methods which is adopted by the masses, and therefore writing grammatical error free content indicates one's proficiency in English. Grammar errors could be committed by native writer as well as ESL (English as Second Language) as learning a language is a long process and adopting grammatical rule takes time. Apart from punctuation, verb form errors, there could be spelling mistakes as well, as enhancing vocabulary also takes its own course. Taking human help in correction of mistake committed in writing English Language content from an English teacher or experts is a difficult and cumbersome task. Therefore, utilizing artificial intelligence for supplementary error rectification is getting popular.

The method of GEC has also improved with the development of methods and technologies based on natural language processing (NLP). First among all method to detect and correct grammar error was the rule-based error correction method. In this approach language specialists or grammar authority sets the rules based on which errors are identified and corrected. The outcomes achieved through this rule-based approach are moderately precise. However, enlarging the rule database demands considerable efforts from the experts and presents challenges in addressing all categories of grammatical errors. Another

Proceedings DOI: 10.21467/proceedings.178; Series: AIJR Proceedings; ISSN: 2582-3922; ISBN: 978-81-984081-8-1

popular approach in GEC was statistical machine translation (SMT). This method does not require expert knowledge like rule-based methods. In place of rule from language expert, it develops and learns a rectification and correction model from collections of texts from learners that have been aligned at the sentence level and corrected for errors. The limitation of SMT based methods is that it is not easy to acquire collection of texts from learners (corpora). The advancement of deep neural networks (DNN) has found application in grammar and error correction field as well. However, most of these deep learning-based methods focus primarily on the internal composition of individual sentences, they neglect the broader contextual framework, in which each sentence is situated in whole text. In reality, sentences within a text are interconnected semantically to completed text rather than its being an isolated entity. Underscoring the importance of considering contextual cues for accurate grammatical error correction, this research paper proposes a deep architecture that captures sentence features across a wider span. By employing an encoder-decoder structure with attention mechanisms, implemented model incorporates contextual information from neighboring sentences during the error correction process, hence enhancing the grammatical correctness of original sentence. The paper can be outlined as follows: It enhances the conventional input encoder structure by incorporating the Luong dot attention mechanisms. Through the attention mechanism, distinct weights are allocated to each segment and processed within the decoder of the structure. As a result, the process of deciphering the source sentence can effectively leverage its context. This research paper refines the traditional word and character-based embedding algorithm by integrating the FastText embedding method.

2 Literature Survey

Initial work in the domain of GEC concentrated on identifiable categories of errors, including articles and prepositions. Researchers tackled this issue through a combination of manually crafted rules and statistical classification methods [2]. The parser implemented the written rules during error correction. Software company Microsoft developed all rule-based error correction open-source tool Language Tool in early days. However, the field of natural language has uncertainty, flexibility, and complexity beyond the static rule. To achieve more refined and evolving results, it is essential to expand the quantity of implemented rules, which can lead to a higher likelihood of rule clashes. Techniques which are inherently based on classifiers approach specific error types as classification problems [3] (Reference citation in text should be in sequential order), training classifiers with contextual relational features. Makarenko et al. [4] applied Long Short-Term Memory (LSTM) to train and selection of words as per the located context of sentence, offering clear gain in the learning of features related to context. Deep neural network-based models are also effective for correcting grammar errors. A study conducted by Hu et al. [5] demonstrated this effectiveness. An Encoder-Decoder architecture based on recurrent neural network (RNN) was designed by Xie and colleagues [6] employing a model which works at character level. They equipped an attention mechanism for handling out-of-vocabulary (OOV) words. They incorporated a beam search algorithm into the model and utilized an n-gram language on the encoder side. Lang et al. [7] examined relevant features in grammar error correction using a logistic regression model. They then condensed these features using a clustering algorithm. Through testing focusing on 10 prepositions and 11 types of grammatical errors, they illustrated the success of their approach. Shi Y [8] et al. constructed applied Bi-LSTM and sequence annotation model, which gave a new vision for Language and GEC. Chollampatt et al. [9] applied CNN neural feature within an encoder-decoder framework. They solved grammar errors while taking whole content in system which was based on phrase. Xie et and colleague [10] proposed an alternative approach, in which they applied character-level sequence to sequence neural model. This unique model was successful in removing the OOV mistake, but it cannot take advantage of word-level information for removing grammar errors. Zhou et al. [11] mechanism that has been applied for finding the appropriate items in history to improve suggestions, and they used attention mechanism with encoder-decoder. They employed a classification model approach to design the GEC model. Additionally, they continuously refined the model by considering the grammatical relationships and hierarchical structure between words. Cheng et al. [12] used a character-level model that works like a sequentially placed neural model. While this model solves the issues where predication is done correctly for the words which were not part of training vocabulary, it doesn't make good use of word-level details for Grammar Error Correction (GEC), even when combined with a separate word-based language model. Although it handles unknown words better, it still struggles with understanding the context of words within sentences for grammar correction. Some Grammer correction model from different language such as Indonesian [13] and Chinese [14] [21][22] are also studied and their approach to apply attention-based model in followed. GECToR [15] is utilizes transformer based seq2seq architecture in Grammer error correction but it is computationally expensive and need large data set and training time just to solve grammar error. Some advance method such as Grammarly and GPT3 are not limited to grammar correction task but also works on tone ,vocabulary enablement and plagiarism check etc. feature as well .There comparison with suggested model cannot fit in due to more parameter coming in picture [17]. All the mentioned methods do satisfactory work in grammar error correction, but they lack in applying the context derived from sentence is correction. This research paper examines the architecture and comparison of grammar correction model having encoder decoder architecture and an attention layer. The attention layer captures the context and the inherent details within the sentences.

Figure 1 below depicts typical encoder decoder model without attention mechanism added in it.

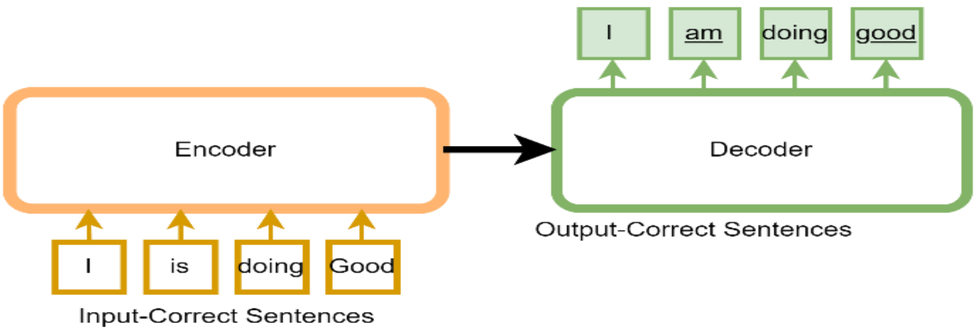

Figure 1: Grammar error correction using seq2seq Encoder-Decoder models

3 Research Methodology

To establish a stable and accurate model for Grammatical Error Correction (GEC), multiple model architectures were explored before converging on the attention-based model as the most effective approach. Each model was evaluated for its ability to handle diverse grammatical errors and produce high-quality corrections. Despite variations in the architecture of the models, certain preprocessing steps remained consistent across all models. These tasks included data cleaning to remove irrelevant part of data set, eliminating unnecessary prefixes from input sequences to streamline processing, and creating dictionaries or lookup tables to map grammatical errors to potential corrections efficiently. Such preprocessing ensured the training data was optimized for learning and improved the overall performance of the models. In following points data sets and data preprocessing steps are explained.

Proceedings DOI: 10.21467/proceedings.178
ISBN: 978-81-984081-8-1

Series: AIJR Proceedings
ISSN: 2582-3922

3.1 Dataset: lang8.train.auto.bea19.m2 data set

Among many available datasets on GEC such as NUCLE [19], ICNALE, CoNLL-2014 [20], FCE etc. Lang-8 Corpus of Learner English dataset has been picked from its original source [16]. The reason to pick this data set is that it encompasses a broader variety of grammatical mistakes, widely used in GEC research, making it easier to compare model performance with state-of-the-art systems. The sentences in this data set are in a specific format called M2. In this format a text line starting with English alphabet S signifies a sentence in its original form, while next line starting with English alphabet A represents an annotation done to edit the original sentence. Multiple annotations exist for an incorrect sentence.

Figure 2 shows an example of M2 format. The annotation describes that there is an error in noun which is referring to numerical is concept at number 10 position, in the sentence. The other annotation tells that there is an error in a Determiner, and it should be ‘a new’ album instead of ‘his new’. Therefore, the correct sentence for above example from data sentence is “This morning I found out that one of my favourite bands released a new album”.

The lang8.train.auto.bea19.m2 is of size 144 MB and this file contains 1037562 sentences ,some of them are grammatically correct sentence not needing any correction ,while some sentences have grammatical errors annotated. This is sufficient data as per dissertation report in GEC field by [17] to compare performance of based model on low scale training and testing computing resources.

```
S This morning I found out that one of my favourite band released his
new album
A 10 11|||R:NOUN:NUM|||bands|||REQUIRED|||-NONE-|||0
A 12 13|||R:DET|||a|||REQUIRED|||-NONE-|||0
```

Figure 2: An example of data set in M2 format

3.2 Data cleaning, Preprocessing, Exploratory Data Analysis and Tokenization

The original dataset was in specific format called M2 format. Transformation of it was into pairs of accurate and inaccurate sentences, which were saved in a NLP supporting format called comma-separated values (CSV). Duplicate and null values from this CSV were removed as first part of Data cleaning. There were a lot of characters in the dataset which are not part of English punctuation, were removed t using Regex. Unlike general NLP tasks data cleaning and preprocessing like stemming, stop word removal, converting each character to lower case, etc. are not relevant here as all these are essential part of English language punctuation.

Several exploratory data analysis (EDA) tasks were conducted to spot any pattern in the dataset. During the analysis of sentence lengths, observed that the majority of sentences consisted of fewer than 50 words, with very few exceeding this threshold, was observed. Through analysis done to measure percentile, it was determined that approximately 89% of accurate sentences with no grammar errors had lengths below 22, and 99% were shorter than 38 words. As a result, sentences longer than 25 words were excluded from whole dataset to reduce dictionary size and computational costs for grammar correction.

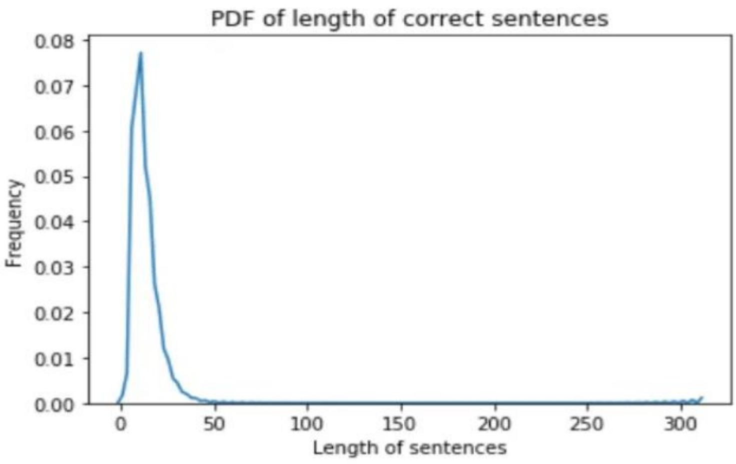

Figure 3: PDF of sentence length in EDA

As word-by-word data is fed to encoder, not as a whole sentence and then this data is input to decoder, therefore special character “#” is added to denote “start of the sentence” (this will be act as an input to decoder) and add “$” to denote “end of the sentence” (this will be decoder output).

For example, after doing above operation on data stream sentence “I am doing well.” decoder input will be “# I am doing well.” and decoder output will be “I am doing well. $”. The full stop punctuation should not be removed or replaced. Further sentences are tokenized and padded to make their length optimal for training.

Tokenized integer is converted to vectors. This conversion can be performed using GloVe, word-to-vec and fast-text embedding techniques. Out of which fast-text embedding was yielding more accuracy in later stage. Word-to-vec and Fast-Text vectorization were finalized for comparison in various model.

All the preprocessing steps along with model validations are summarized in figure 4.

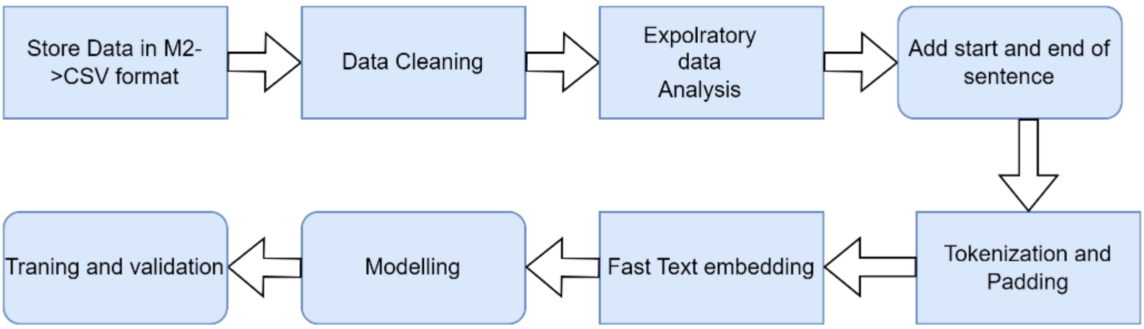

Figure 4: Workflow for Grammar Error correction

3.3 Grammer Error correction models

As part of this research, four models were implemented, developed and tested, serving as foundational approaches to evaluate the effectiveness of the attention-based architecture. The performance of these models was compared in terms of improvements in loss reduction and GLUE score, which are standard metrics for assessing grammatical correction accuracy and overall model performance.

3.3.1 Encoder-Decoder with character embedding model

In this model, the encoder takes the input sequence of characters and produces a fixed-length vector representation (embedding) of the entire sequence. The encoder processes each character in the input sequence sequentially and updates its hidden state at each time stamp. The decoder takes the fixed-length vector representation produced by the encoder and generates the output sequence of characters. Similar to the encoder, the decoder also utilizes RNN cells to process the output end one character at a time. At each time stamp, the decoder predicts the next character in the output sequence based on the current h state and the already generated characters.

The representative architecture of this model [9] is illustrated in Figure 5. Here an input word with spelling mistake "PRECISE" is corrected character by character. Each individual character such as P, R, E, C etc. is processed character by character. Whole word PRECISE is not processed.

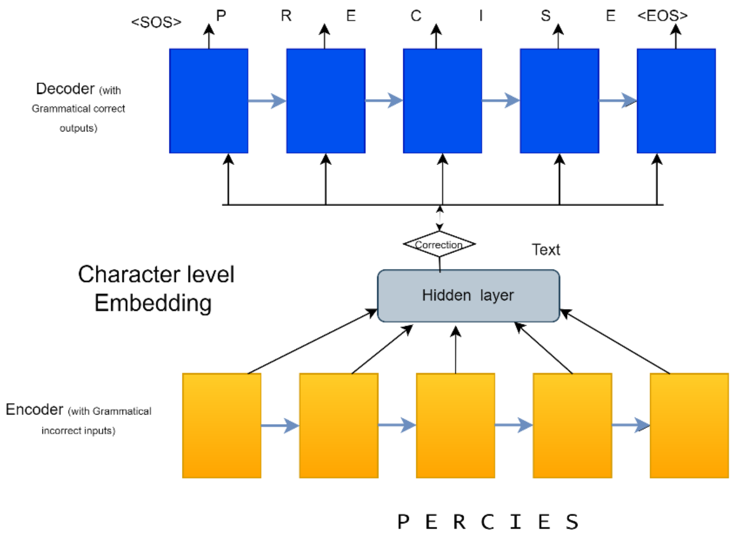

Figure 5: Encoder-Decoder model with character level embedding model

3.3.2 Encoder-Decoder with word embedding model

Unlike character-level model where individual character are input to encoder, in this model, input and output sequences are represented as sequences of words. This means that each word in the input sequence is individually encoded and processed by the model. The representative architecture of this model is illustrated in Figure 6. Here the input sentence stream "I have no money" is corrected word by word. The words such as, have, no and money are processed as a word rather than an individual character.

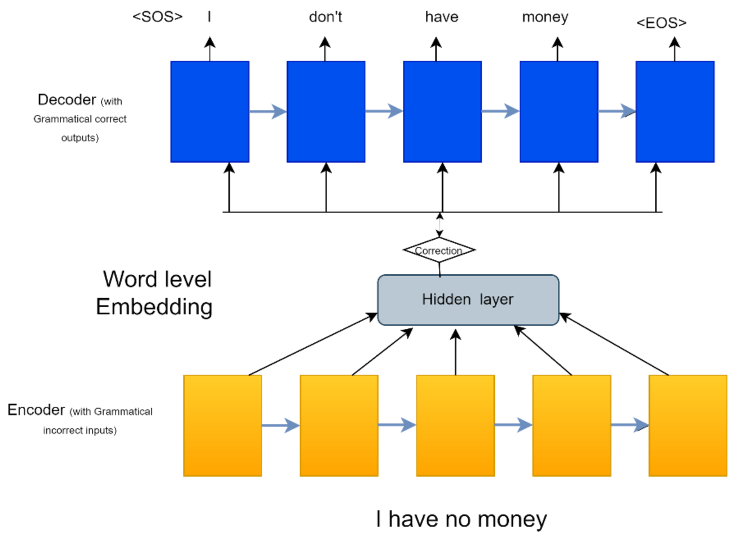

Figure 6: Encoder-Decoder model with word level model

3.3.3 Encoder-Decoder FastText embedding model

This encoder-decoder utilizes FastText embeddings to represent the input sequence. Each word in the input sequence is mapped to its corresponding FastText embedding vector. These embedding vectors capture the semantic and morphological information of the words, allowing the model to better understand the input text as per Bojanowski et al., 2017 [26].

3.3.4 Luong attention model with FastText embedding

Following the methodology of Luong et al. [27], their model was adapted and implemented for Grammatical Error Correction, as illustrated in Figure 7. The dot scoring function from the Luong attention mechanism was employed to enhance the model's performance. In earlier approaches, the end hidden state (h) alone was used to initialize the output decoder. However, for longer sentences or those with complex structures, models without an attention layer struggled to retain contextual information, leading to diminished accuracy. The attention mechanism addresses this limitation by utilizing the encoder's hidden states (h) at each time step during the prediction of individual words. The integration of attention and context through the hidden attention layer [29] is depicted in Figure 7.

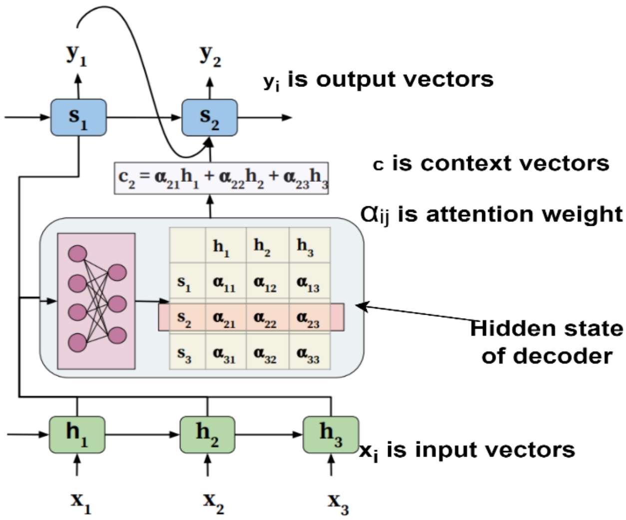

Figure 7: Encoder-Decoder model with Luong attention model referred from [29]

Proceedings DOI: 10.21467/proceedings.178
ISBN: 978-81-984081-8-1

Series: AIJR Proceedings
ISSN: 2582-3922

4 Mathematical Equations

The encoder-decoder model is a type of neural network architecture. It comprises: the encoder and the decoder figure [1]. Building blocks from Encoder Decoder are-

Encoder: The encoder takes an input sequence, which is input grammatically correct or incorrect sentences here, and represents it in the form of hidden states and cell states. Kersa library-based LSTM cell at the Encoder end is implemented.

Mathematically, the encoder computes the hidden states and cell states as follows:

$$h_t^{enc}, c_t^{enc} = \mathrm{EncRNN}(x_t, h_{t-1}^{enc}, c_{t-1}^{enc}) \qquad (1)$$

where x_t : input sample at time stamp t,

h_t^{enc} : hidden state at time stamp t of the encoder,

c_t^{enc} : cell state at time stamp t of the input encoder.

Decoder: Based on the representation produced by the input encoder, Decoder at the output end generates the output sequence. During initialization, the encoder representation is passed to the decoder along with a special token indicating the start of the sentence ("<start>"). The decoder then iteratively predicts the next word in the sequence based on the current hidden state and the ground truth tokens.

Mathematically, the decoder computes the hidden states and cell states as follows:

$$h_t^{dec}, c_t^{dec} = \mathrm{DecRNN}(y_t, h_{t-1}^{dec}, c_{t-1}^{dec}) \qquad (2)$$

where y_t: input at time step t,

h_t^{dec}: hidden state at time stamp t of the decoder

c_t^{dec}: cell state at time stamp t of the decoder.

The decoder predicts the next word in the sequence using the hidden state htdec and the softmax function:

$$y(t) = \mathrm{softmax}(W_{out} h_t^{dec} + b_{out}) \qquad (3)$$

where y(t): at time stamp t output probability distribution over the entire vocabulary,

W_{out} and b_{out}: weight matrix and bias vector of the output layer, respectively.

5 Results

Loss values (MSE) and GLUE of each implemented model are presented in following Table 1.

Table 1: *Summary of MSE and GLUE score of each GEC model implemented in this paper*

Model Type	Loss value	GLUE Score
Encoder-Decoder Character Embedding	.18	.23
Encoder-Decoder Word Embedding	.72	.19
Encoder-Decoder word level with FastText	.55	.28
Luong Attention model with FastText embedding	.44	.63

Character level embeddings demonstrate slightly better performance in GLUE scores compared to word level embeddings but show a more significant improvement in Mean Squared Error (MSE) values. The introduction of FastText embeddings further enhances GLUE scores while moderately reducing. Among the evaluated models, the sequence-to-sequence model with FastText embeddings and the Luong attention model with FastText embeddings, the attention-based model outperforms in both MSE and GLUE score metrics. This performance trend aligns with anticipated outcomes, reaffirming the expected effectiveness of the attention mechanism.

6 Conclusion

This paper implements basic LSTM model with encoder-decoder seq2seq features and adds features of attention layer to them to show improvement in GLUE score and MSE loss as well. The introduction of attention layer enables the decoder to dynamically concentrate on various sections of the input sequence while generating each output sequence word. Rather than depending only on the final hidden state of the encoder, the mechanism of attention takes into account each hidden states of the encoder throughout each time step. This enables the implemented model to weigh the importance of each hidden state based on its relevance to the current decoding step. The results in the table provide a comprehensive comparison of the performance of various Grammar Error Correction (GEC) models, highlighting the effectiveness of different architectures and embedding techniques. The Encoder-Decoder model with character embeddings exhibits the lowest loss value .18, indicating better optimization during training. While, its GLUE score .23 remains relatively low, suggesting that it struggles to accurately detect and correct grammatical errors despite its efficiency in minimizing error in training. The Encoder-Decoder model with word level embeddings, on the other hand, demonstrates the highest loss value .72 and the lowest GLUE score .19, pointing to its limitations in both error correction accuracy and optimization. Introducing FastText embeddings in the Encoder-Decoder word-level model leads to a marked improvement. The loss value reduces to .55 and the GLUE score increases to 0.28 This indicates that FastText embeddings enhance the model's ability to generalize and capture linguistic nuances, making it more effective than basic word embeddings. The Luong Attention model with FastText embeddings looks as the most robust and reliable approach. With a loss value of .44 and a significantly higher GLUE score .63, this model demonstrates the benefits of integrating attention mechanisms with advanced embedding techniques. The attention mechanism effectively captures context and long-range dependencies, resulting in improved performance in both error detection and correction tasks. Overall, this analysis highlights the critical role of attention layers and advanced embeddings in achieving superior performance in GEC. These findings validate the hypothesis that leveraging contextual embeddings and attention mechanisms is essential for advancing grammar error correction systems.

7 Future Work

In upcoming research, the aim is to expand the Grammar Error Correction (GEC) system by leveraging advanced techniques, with a focus on transitioning from the attention mechanism to transformer-based models. There is a plan to integrate transformer frameworks, including the Transformer model proposed by Vaswani and colleagues. [23, into this GEC system. These models have demonstrated impressive effectiveness across a range of natural language processing tasks by effectively capturing dependencies over long distances and context dependent correlations. By adopting transformers, improvements in the accuracy and efficiency are anticipated in implemented GEC system.

Proceedings DOI: 10.21467/proceedings.178
ISBN: 978-81-984081-8-1

Series: AIJR Proceedings
ISSN: 2582-3922

Plan to explore techniques for data augmentation and domain adaptation to improve the robustness of the GEC system is also explored. This includes generating synthetic training data using techniques such as back-translation and paraphrasing, as well as adapting the model to specific domains or writing styles through domain-specific fine-tuning or transfer learning. There is also plan to add Explanation AI [24] [25] where in this model which would explain every error correction. By integrating this explanatory capability, GEC system will not only correct errors but also empower users to comprehend the reasons behind the corrections, fostering language learning and improvement. There is also plan to incorporate a process to implement methods which would tackle noisy corrections and incorrect annotation [28] in lang8 data set.

8. Declarations

8.1 Competing Interests

The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

8.2 Publisher's Note

AIJR remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

How to Cite

Medha Joshi (2025). Optimizing Grammar Error Correction Performance with Attention-Based Neural Networks. *AIJR Proceedings*, 252-262. https://doi.org/10.21467/proceedings.178.27

References

[1] J. Bentley, "Report from TESOL 2014: 1.5 Billion English learners worldwide," *International TEFL Academy*, vol. 19, no. 12, pp. 2017-2017, Dec. 2014

[2] R. De Felice and S. Pulman, "Automatic detection of preposition errors in learner writing," CALICO Journal, vol. 26, no. 3, pp. 512–528, 2008.

[3] Z. Yuan and T. Briscoe, "Grammatical error correction using neural machine translation," *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, San Diego, CA, USA, pp. 380–386, Jun. 2016 http://dx.doi.org/10.18653/v1/N16-1042

[4] V. Makarenkov, L. Rokach, and B. Shapira, "Choosing the right word: Using bidirectional LSTM tagger for writing support systems," Engineering Applications of Artificial Intelligence, vol. 84, pp. 1–10, 2019 https://doi.org/10.1016/j.engappai.2019.05.003.

[5] L. Hu, Y. Tang, and X. Wu, "Considering optimization of English grammar error correction based on neural network," Neural Computing and Applications, vol. 34, no. 5, pp. 3323–3335, 2022 https://doi.org/10.1007/s00521-020-05591-2

[6] Z. Xie, A. Avati, and N. Arivazhagan, "Neural language correction with character-based attention," arXiv preprint arXiv:1603.09727, 2016. https://doi.org/10.48550/arXiv.1603.09727

[7] L. Hu, Y. Tang, and X. Wu, "Considering optimization of English grammar error correction based on neural network," Neural Computing and Applications, vol. 34, no. 5, pp. 3323–3335, 2021. https://doi.org/10.1007/s00521-020-05591-2

[8] Y. Shi, "Research on English grammar error correction technology based on BLSTM sequence annotation," Asian Conference on Artificial Intelligence Technology, 2021 10.4108/eetsis.v9i5.2011

[9] S. Chollampatt and H. T. Ng, "A multilayer convolutional encoder-decoder neural network for grammatical error correction," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, no. 1, 2018. https://doi.org/10.1016/j.engappai.2019.05.003

[10] Z. Xie, G. Genthial, and S. Xie, "Noising and denoising natural language: Diverse backtranslation for grammar correction," Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1, pp. 619–628, 2018. 10.18653/v1/N18-1057

[11] S. Zhou and W. Liu, "English grammar error correction algorithm based on classification model," Complexity, 2021. http://dx.doi.org/10.1155/2021/6687337

[12] L. Cheng, B. Peng, and Y. Qiao, "Research on automatic error correction method in English writing based on deep neural network," *Computational Intelligence and Neuroscience*, 2022. http://dx.doi.org/10.1109/ACAIT56212.2022.10137929

[13] N. Lin, B. Chen, X. Lin, K. Wattanachote, and S. Jiang, "A framework for Indonesian grammar error correction," *Transactions on Asian and Low-Resource Language Information Processing*, vol. 20, no. 1, pp. 1–12, 2021. http://dx.doi.org/10.1145/3440993

[14] X. Wu, P. Huang, J. Wang, Q. Guo, Y. Xu, and C. Chen, "Chinese grammatical error diagnosis system based on hybrid model," *Proceedings of the 2nd Workshop on Natural Language Processing Techniques for Educational Applications*, Beijing, China, pp. 117–125, Jul. 2015. https://doi.org/10.18653/v1/W15-4418

[15] M. Kaneko, S. Takase, and M. Komachi, "GECToR: Grammatical error correction with transformer," *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 6163–6172, 2019.

[16] B. Raad, "Exploring the profound impact of artificial intelligence applications (Quillbot, Grammarly, and ChatGPT) on English academic writing: A systematic review," *International Journal of Integrative Research (IJIR)*, 2023. https://doi.org/10.59890/ijir.v1i10.366

[17] NAIST, "Lang-8 Corpus of Learner English," *14th BEA Shared Task*, 2021.

[18] K. Sakaguchi, "Robust text correction for grammar and fluency," Ph.D. dissertation, Johns Hopkins University, 2018.

[19] J. Wieting, K. Gimpel, and G. Neubig, "NUCLE: A nuclear test set for evaluating open-domain grammatical error correction," *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, pp. 2971–2976, 2019.

[20] M. Junczys-Dowmunt and R. Grundkiewicz, "The AMU system in the CoNLL-2014 shared task: Grammatical error correction by data-intensive and feature-rich statistical machine translation," *Proceedings of the 18th Conference on Computational Natural Language Learning (CoNLL)*, Baltimore, MD, USA, pp. 25–33, Jun. 2014. https://doi.org/10.3115/v1/W14-1703

[21] X. Wu, P. Huang, J. Wang, Q. Guo, Y. Xu, and C. Chen, "Chinese grammatical error diagnosis system based on hybrid model," *Proceedings of the 2nd Workshop on Natural Language Processing Techniques for Educational Applications*, Beijing, China, pp. 117–125, Jul. 2015. https://doi.org/10.18653/v1/W15-4418

[22] L. Yang, X. Liu, Y. Zhang, H. Wang, and S. Li, "A deep learning approach for Chinese grammatical error correction," Proceedings of the 6th International Conference on Natural Language Processing and Information Retrieval (NLPIR), Hangzhou, China, pp. 123–134, Oct. 2021. https://doi.org/10.1007/978-981-18-6359-4_12

[23] A. Vaswani, N. Shazeer, N. Parmar, et al., "Attention is all you need," *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

[24] Y. Wang, S. Huang, C. Chou, and M. Chen, "An explainable AI approach for grammar error correction," *Proceedings of the 12th Language Resources and Evaluation Conference (LREC)*, pp. 2032–2037, 2020.

[25] J. Yin, C. Xiong, and A. Shukla, "Towards explainable grammatical error correction with neural attention," *Proceedings of the 28th International Conference on Computational Linguistics (COLING)*, pp. 2775–2786, 2020.

[26] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017.

[27] Luong, Minh-Thang & Pham, Hieu & Manning, Christopher, "Effective Approaches to Attention-based Neural Machine Translation" , *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (2015) http://dx.doi.org/10.18653/v1/D15-1166

[28] C. Bryant, M. Felice, and T. Briscoe, "Automatically generating reusable resources for multi-word expression identification in untagged text," *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, vol. 2, pp. 314–320, 2017.

[29] S. Chaudhari, et al., "Exploring intelligent grammar correction," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 5, issue 12,pp. 1–32, 2021 https://doi.org/10.1145/3465055

Proceedings DOI: 10.21467/proceedings.178
ISBN: 978-81-984081-8-1

Series: AIJR Proceedings
ISSN: 2582-3922