ORIGINAL ARTICLE

Transformers to the rescue: alleviating data scarcity in arabic grammatical error correction with pre-trained models

Karim Ismail[1] · Sherif Abdou[2] · Mohamed Farouk[3] · Ahmed Salem[1]

Received: 13 March 2024 / Accepted: 20 February 2025 / Published online: 24 April 2025

Abstract

Grammatical error correction (GEC) in Arabic presents unique challenges arising from complex morphology and contextual intricacies. Current methodologies predominantly rely on neural machine translation (NMT) models, hindered by adequately annotated training data scarcity. This research introduces a novel approach utilizing pre-trained transformers, specifically sequence-to-sequence (seq2seq) models, such as AraT5 and AraBART, alongside their multilingual variants (mT5 and mBART), to address Arabic GEC. These transformers, initially designed for diverse natural language processing tasks, demonstrate promising results in GEC, particularly when parallel data are limited. Employing tokenization and preprocessing techniques on publicly accessible GEC datasets, we train the transformers using a supervised approach. The experimental results showcase superior performance, surpassing previous models with an F1 score of 92.1% on the QALB 2014 dataset, 89.4% on the QALB 2015 native test data, and 83.6% on non-native data. This highlights the effectiveness of the proposed methodology in rectifying various grammatical errors in Arabic text. In conclusion, this study contributes to advancing the field of Arabic GEC by leveraging transfer learning with pre-trained transformers. The findings underscore the potential of this approach to overcome challenges posed by limited data availability, with AraBART emerging as a practical choice. This research opens avenues for further exploration in low-resource languages. It suggests potential applications in high-resource languages, encouraging future comparative studies.

Keywords Grammatical error correction · Arabic NLP · neural machine translation · Low-resource languages · Pre-trained transformers · Transfer learning

1 Introduction

Natural language processing (NLP) is a fundamental discipline within artificial intelligence (AI) [1, 2]. NMT and text summarization serve distinct purposes [3–5]. The task of GEC involves identifying and rectifying faults present within a given document. In an alternative interpretation, the process involves taking a text that encompasses a multitude of diverse flaws and afterward reconstructing it as a text devoid of any errors. In contemporary discourse, the task of GEC has been categorized as an NMT problem [6–9]. This classification arises from the observation that GEC can be conceptualized as translating an erroneous text into a grammatically correct one while preserving the underlying semantic content [7, 8]. Another significant aspect of the GEC is its ability to assist writers in composing paragraphs in languages other than their native tongue, thereby minimizing linguistic

https://doi.org/10.1007/s00521-025-11145-1

errors. However, there is room for development in their writing style to enhance intelligibility [6, 10]. Arabic is recognized as one of the six official languages of the United Nations (UN). It serves as the primary language for about 300 million individuals residing in 22 countries within the Middle East region. Arabic serves as the primary language of the Islamic faith, with approximately 1.8 billion adherents worldwide using Arabic in their everyday religious practices [11, 12]. Arabic exhibits various linguistic variations, with two prominent forms being Modern Standard Arabic (MSA) and Classical Arabic (CA). MSA is a simplified and standardized version of Arabic commonly employed in mass media and online platforms. On the other hand, CA represents the linguistic form utilized in the Qur'an [11–13]. The work of Arabic GEC is a significant challenge because of the intricate grammatical structure and extensive morphological aspects inherent in the Arabic language. These morphemes contribute to the increased ambiguity of language. From these morphemes, the prefix, suffix, and affix as the word سيلعبون means they will play, the سيلعب means he will play, and ستلعب means she will play [6, 11–13]. Several methodologies have been suggested for addressing the Arabic GEC task. These include rule-based systems, n-gram models, and NMT-based systems, primarily relying on Seq2Seq models [6, 11–15]. The implementation of these models involves the utilization of neural network (NN) architectures such as recurrent neural network (RNN) and self-attention networks (SAN) [16–18]. One of the primary difficulties encountered in NMT bases systems is the insufficiency of annotated data, leading to a data sparsity issue [16, 19, 20]. The Transformer architecture has been widely employed in NLP tasks due to its high efficiency. The baseline model employed in prior studies has demonstrated superior efficiency to conventional deep learning models [21, 22]. Transfer learning has emerged as a highly effective technology in image processing and NLP. The primary idea revolves around utilizing a pre-trained model to adapt it to a novel task by fine-tuning the model weights. This approach can potentially address the challenge of data limitations [23]. Pre-trained transformer models such as bidirectional encoder representation transformer (BERT) and generative pre-trained transformer (GPT) have demonstrated significant efficacy in various NLP tasks, including text summarization and classification. However, their existing architecture could be more suitable for accommodating seq2seq tasks such as NMT and GEC [16, 24]. Models such as bidirectional and auto-regressive transformers (BART) and text-to-text transformers (T5) are examples of pre-trained transformers that can address seq2seq problems. These models primarily rely on an encoder–decoder architecture, making them suitable for various NLP tasks such as text generation and NMT [25, 26]. Arabic adaptations of BART and T5 exist in Arabic languages, known as AraBART and AraT5, respectively. Both models have undergone pre-training exclusively in the Arabic language [27, 28]. Additionally, alternative iterations of BART and T5, known as mBART and mT5, exist, respectively. These models have undergone pre-training in various languages, including Arabic [29, 30]. This study aims to leverage pre-trained models to enhance the GEC challenge. The primary challenge of this problem is the limited availability of data. The only datasets accessible for training are the Qatar Arabic Language Bank (QALB) from 2014 and 2015. Unfortunately, these datasets do not contain a sufficient number of documents to effectively train a deep neural network or a baseline transformer model [31–33].

1.1 Problem statement

Arabic GEC systems indicate a higher complexity level than systems intended for other languages, including English. Its complicated structures represent a challenge for conventional learners to comprehend, and the same holds true for deep learning models [12, 14, 33, 34]. One of the languages that is written and read from right to left is considered to be unique. The letter "ا" can be represented by other forms, including "إ" and "أ". Similarly, the letter "ى" has multiple variations and can be written as "34] "ي]. Every letter form in the Arabic language possesses a distinct semantic meaning. An additional challenge arises from the fact that the meaning of similar words may vary depending on their context within the same or separate sentences. Another challenge occurs when equivalent words are used in the same or particular sentences but with different contextual meanings [34, 35]. However, a notable difference between Arabic and other languages is the absence of capitalization to indicate the beginning of a sentence [12, 34]. Any GEC system should correct the Arabic error categories specified in Table 1 [33], which presents examples of different errors and their accompanying fixes.

1.2 Novelty of work

The primary contributions of this research to address the issue of grammatical correction are:

1. Employ the transfer learning approach to develop an Arabic grammatical error correction system as a novel downstream task for diverse pre-trained seq2seq transformers.
2. demonstrates that the lack of data challenge may be solved without synthetic data generation.
3. Train the QALB 2014 and QALB 2015 training data individually on the proposed model to demonstrate that there is no need to assemble them to obtain superior results, particularly for the non-native dataset.
4. Adapting the transformer encoder and decoder to process sentences longer than the default length, thus reducing truncation issues
5. Investigate alternate methods to the beam search for text-generation approaches.

The remainder of the paper is structured as follows: Sect. 2 provides a concise overview of transformers and pre-trained transformers. Section 2.3 presents the related work of this research, including a detailed explanation of the experiments and results from past research, along with a concise summary. Section 3 pertains to the research technique, while Sect. 4 explains the dataset and setting used in the studies. Section 5 encompasses the comprehensive findings and analysis of all the experiments conducted in this research. Section 7 serves as the final segment of this article, encompassing the conclusion and proposed future work.

Table 1 Grammatical errors in arabic and its correction

Error Type	**Description**	**Incorrect Sentence**	**Correct Sentence**
Spelling	Deleted or substituted characters for another, or insert an extra character in the word	كرأ الطالب الكتاب The student kara the book	قرأ الطالب الكتاب The student read the book
Punctuation	Using a comma in the wrong place	، قرأ الطالب الكتاب ,The student read the book	قرأ الطالب الكتاب The student read the book
Syntactic Error	Errors associated with a wrong agreement on gender, quantity, wrong tense use, wrong order or missing words	يقرأالطالب الكتاب يوم امس The student is reading the book yesterday	قرأ الطالب الكتاب يوم امس The student read the book yesterday.
Word Choice Error	Use incorrect words choice within the sentence	ذاكر الطلاب كرة القدم The students studied football	لعب الطلاب كرة القدم The students played football
Morphology Error	Errors related to a wrong decision or inflection or incorrect concatenative morphology	انقسم الطالب إلى شعبتين The student was divided into two groups	انقسم الطلاب إلي شعبتين The students were divided into two groups
Proper Name Error	Errors that occur in the spelling of names, particularly those of foreign which it incorrectly translation	لدن مدينة عريقة Ldon is an ancient city.	لندن مدينة عريقة London is an ancient city
Dialectal Usage correction	Use dialectal words within MSA sentence	بداية المباراة العاشرة بليل The game starts at 10 pm	بداية المباراة العاشرة ليلاً The game starts at 10 pm

2 Literature survey

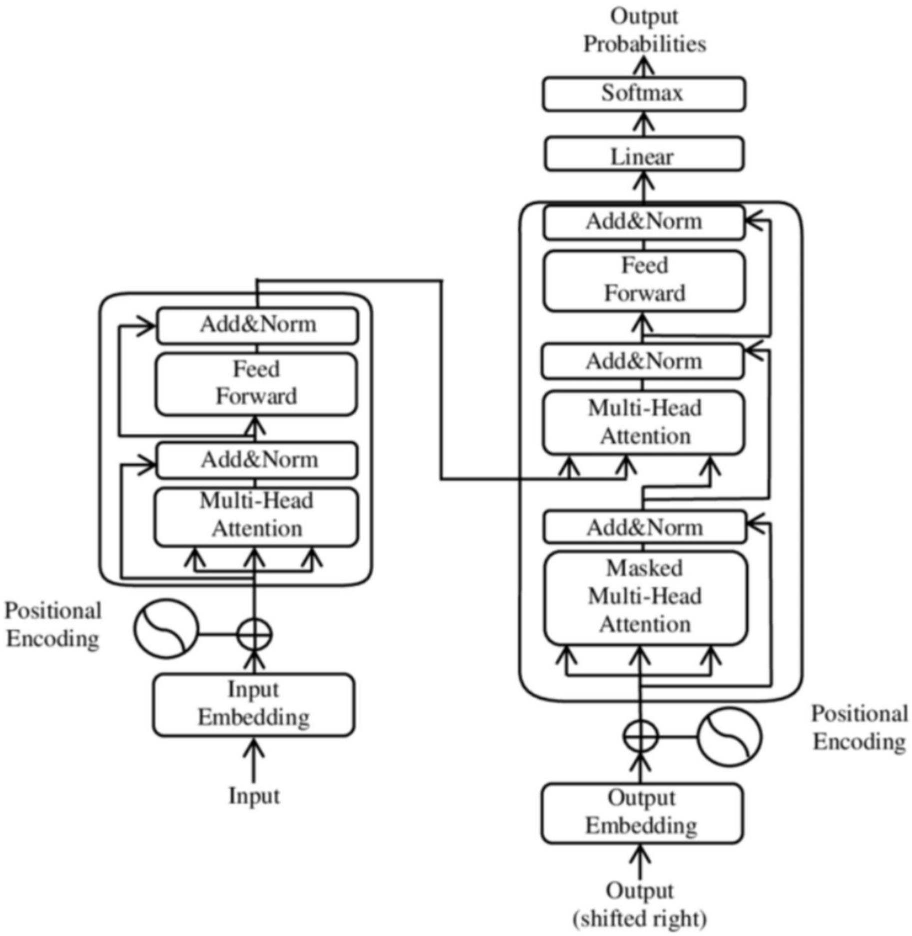
2.1 Transformers

The introduction of a network known as a transformer occurred in 2017, as documented by A. Vaswani et al. in their research publication titled “Attention is All You Need.” [36] The primary purpose of its invention, illustrated in Fig. 1, was to address the NMT problem in natural language NLP. Following its successful demonstration of efficacy in the designated job compared to deep learning models, this approach was subsequently employed in many other NLP tasks, including document summarization, text classification, and question answering [24, 36–38]. The system’s primary components consist of an encoder and decoder, incorporating self-attention and multi-head attention layers. Additionally, positional encoding is applied to the embedding layer in both the encoder and decoder blocks to address the limitation of self-attention in capturing sequence order information [36, 38, 39].

2.2 Pre-trained transformers

The limited amount of datasets poses challenges in the domain of machine learning. Various approaches address this issue, such as data augmentation and, more recently, transfer learning [23, 40]. In NLP, data augmentation encompasses various techniques, including using thesauri, text generation, and pre-trained word embedding models such as Word2Vec and Glove [4, 40, 41]. Transfer learning is a highly efficacious approach for addressing the issue of insufficient data. The concept revolves around the process of adjusting the acquired weights of a model that has been trained on a prior task. Pre-trained transformers refer to models that have undergone training on extensive datasets and can be fine-tuned on smaller datasets to do specific downstream tasks as needed [23, 42]. In addition

Fig. 1 Transformer architecture



to their many extensions, pre-trained transformers encompass a range of utilized models, including BERT, BART, GPT, and T5 [43–47].

2.2.1 T5

T5 is a pre-trained transformer model developed by Google, designed to perform language encoder and decoder tasks by converting language problems into a text-to-text format [46, 48]. The input for this process is raw textual data; the output is another text that does not involve converting the input data into numerical word embeddings. The T5 model's text-to-text framework enables it to acquire knowledge for various NLP tasks without modifying its maximum likelihood training objective function. Consequently, a single set of hyperparameters may fine-tune the model for each downstream task [29, 48]. T5 can be employed in several supervised tasks, like text summarization and question answering, and unsupervised tasks, like text production. The model was trained on the Colossal Clean Crawled Corpus (C4), encompassing textual data totaling 750 gigabytes. The T5 model's inputs consist of a prefix that signifies the task associated with the expected outcome [29, 49].

2.2.2 AraT5

AraT5 refers to a T5 transformer model that has been trained exclusively in Arabic. The model has been pre-trained using a dataset of 70GB of Modern Standard Arabic (MSA) text, which had around 7.1 billion tokens. The data sources consisted of various open-source data, such as AraNews and data obtained from Twitter. The Arabic Language GENeration (ARGEN) benchmark has been evaluated, encompassing seven tasks: Machine Translation (MT), Code-Switched Translation, Text Summarization, News Title Generation, Question Generation, Paraphrasing, and Transliteration. Compared to other models assessed on this benchmark, it has demonstrated commendable performance [27].

2.2.3 mT5

The mT5 model is a variant of the T5 transformer architecture that has undergone training on a diverse dataset comprising more than 100 languages, including Arabic. It aimed to produce a multilingual model that deviates as little from the recipe used to create T5. The model encompasses a range of models that vary in the size of their parameters, ranging from 60 million to 11 billion parameters. Additionally, the performance of AraT5 was assessed on the ARGEN benchmark for the Arabic language, where it exhibited significantly superior outcomes compared to other models [29].

2.2.4 BART

BART is a denoising autoencoder with a seq2seq paradigm, rendering it suitable for various end jobs. The proposed model is a pre-trained transformer architecture integrating bidirectional and auto-regressive transformer components [44]. The process of pre-training consists of two distinct steps. Initially, the act of distorting text involves the application of an arbitrary noising function. Furthermore, a seq2seq model is trained to reproduce the original text effectively. In an alternative interpretation, it facilitates associating a corrupted document with its uncorrupted counterpart. The primary goal of the training is to maximize the log probability of the original document [26, 44]. The BART model has been made available for fine-tuning. It has demonstrated its efficacy in several NLP applications, such as text generation, MT, and text summarization [44].

2.2.5 AraBART

AraBART, similar to AraT5, is a pre-trained transformer model based on BART designed explicitly for Arabic NLP tasks. The primary purpose of its design was to facilitate the abstractive summary task. The AraBART model, which has a parameter count of 134 million, was trained using a dataset of 20 gigabytes specifically composed of Arabic language data. The model's performance in the text summarization task surpassed that of other pre-trained transformers such as AraBERT and AraT5 [28, 50].

2.2.6 mBART

mBART is a pre-trained transformer model derived from BART, which has been trained on diverse corpora sourced from multiple languages. The model was trained to establish a universal set of trainable parameters that could be further refined for any given language pair. There exist multiple versions of mBART, such as mBART25, which is trained on a corpus of 25 languages, and mBART50, which encompasses training data from 50 languages. The performance of mBART25 was assessed in the context of text summarization, in addition to text production and machine translation, and demonstrated favorable results. However, when it comes to the Arabic assignment, AraBART demonstrates superior performance [30].

2.3 Arabic GEC previous researches

2019 saw the proposal of an Arabic GEC model-based MT by Aiman Solyman et al. [11], who employed an encoder–decoder model with nine CNN layers and an attention mechanism. They incrementally trained their model based on words with rare segmentation to overcome the encoder–decoder strategy's drawback. To overcome the problem of the small dataset, they included subword information for each word vector and used the FastText pre-trained embedding for word representation. The authors used the QALB challenge to test their suggested paradigm. They pooled the training sets from 2014 and 2015 to have a large corpus. They employed the QALB native data for the testing set and received an F1 score of 71.14 %.

In 2019, Chouaib Moukrim et al. [13] introduced a system for correcting syntactic errors in Arabic. This innovative method operates by automatically generating correct sentences, a unique approach in the field. Their main objective was to develop a novel method that could automatically handle the syntactic errors of the Arabic language by utilizing its ontology. Their process involved initially adapting a dictionary through translation. They were expressing the grammatical information by representing it as a quadruplet structure. Furthermore, using domain ontology involves applying the Ontology of Arabic Syntax (OAS) to represent Arabic grammar as mathematical entities. The employed techniques included text segmentation into sentences and sentence segmentation into words. A set of syntactic features was assigned to each sentence for sentence generation. Finally, the syntactically accurate sentences generated in the previous phase were meticulously compared to the original. The approach was rigorously tested on a manually collected dataset from the Arab spellchecker called Arramooz Alwaseet. The dataset contains 50,000 words, with 10,000 verbs and 40,000 nouns. They achieved an F1 score of 88.27%.

In 2020, Manar Al-Khatib et al. [14] proposed a method that explores the use of deep neural network technology to detect errors in Arabic text. Their objective was to develop a model that can detect and correct the many spelling and grammatical mistakes that frequently arise in Arabic text, utilizing recent developments in artificial neural networks. Their approach involved utilizing one hot encoding to represent the word, followed by employing AraVec pre-trained word embedding with a dimension of 300 and utilizing the Skip-gram method for language representation. Afterward, the language encoding occurs by using the BiLSTM with an inter-attention mechanism. They were employing a polynomial classifier for error detection. The researchers applied their methodology to a dataset from various sources, including Watan-2004 and Essex Arabic Summaries Corpus (EASC). This dataset consisted of 31 million words, which were manually annotated for use as training data. The researchers acquired data from the British Broadcast (BBC) and CNN Arabic corpus for the testing data. An expert manually annotated this data. After removing the attention mechanism, they achieved a f0.5 score of 95.15% and a f0.5 score of 90.94%.

Aiman Solyman et al. [33] extended their research into 2021 and suggested an unsupervised method for producing massive synthetic datasets to address the difficulty of the shortage of training data. For training an Arabic GEC model, they presented a confusion function to create 278,770 samples. The model was trained with synthetic data created by the authors, who then fine-tuned it with QALB training data. The authors apply the same model architecture as in their prior publication. The same native test set from QALB 2015 was employed, and the final F-score was 70.91%

In 2021, Gheith A. Abandah et al. [34] introduced a BiLSTM recurrent neural network for detecting and correcting spelling errors in classical and MSA. They presented two models. The first architecture has an input masking layer, two BiLSTM layers, and a fully connected layer. The training of this model involved using transformed input and stochastic error injection. The second model corresponds to the first model, adding a dropout layer to handle the overfitting issue. The models were trained on the training sets of Tashkeela and Arabic Treebank Part 3 (ATB3), which had 2617 K words. The test sets were then utilized to evaluate the models. The F1 score for Tashkeela was 98.7%, while the F1 score for ATB3 was 98.2%.

By putting up an Arabic GEC seq2seq model based on the transformer in 2022, Aiman Solyman et al. [51] expanded their research. To attain the same goal of producing more synthetic data, they first presented a noise method based on back translation and direct injection to produce various forms of grammatical errors. They produced 1,500,171 samples for development and 11,833,758 samples for training. All of the generated data have been applied to character normalization. Second, they developed right-to-left and left-to-right models, and to address the issue of exposure bias, they used expectation-maximizing routing from the CapsNet network in computer vision. Additionally, they included a bidirectional regularization term in the training objective to enhance the agreement between the two models. In addition to the native test set of QALB 2015, the experiment was carried out using the training and testing sets of the QALB 2014 challenge. On the QALB 2014 test set, the experiment's F-score was 71.82%, and on the QALB 2015 native test set, it was 74.18%.

Krzysztof Pajk and Dominik Pajk [52] suggested in 2022 that a pre-trained seq2seq language model may be adjusted for the GEC task. They sought to demonstrate the ability of a single multi-language model to fix various grammatical errors in many languages. They aimed to train a multi-language model that could effectively and extensively correct all grammatical errors in seven different languages: English, Arabic, German, Czech, Romanian, and Chinese. On a corpus that compiles all the datasets of the seven languages, their suggested model involved training multi-language T5 and BART transformer versions. They used the QALB 2014 challenge's training set for the Arabic dataset. And the QALB 2015 challenge's native test set. The mBART25 transformer's F-score of 69.81% represented the best performance for the Arabic test data.

Aiman Solyman et al. [32] 2023 developed aggressive transformation methods for data augmentation during training as part of their ongoing research. When the target prefix is insufficient to anticipate the next word, they aim to use augmented data to create additional contexts. To solve the problems of data sparsity and data distribution mismatch, they proposed various ways of data augmentation in place of the prior synthetic data. The augmentation techniques were missing, swap, source, reverse, mono, and replace. To strengthen the encoder, improve its contribution, and pay closer attention to the encoder representation during the decoding, the target sentences in the applicable experiments were generated rather than the source sentences. The acquired F-score result was 71.04 % for the QALB 2014 test and 73.52% for the QALB 2015 native test after conducting numerous trials, incorporating decoding improvement approaches as BPEmb, and reusing the re-ranking L2R from the prior research.

Bashar Alhafni et al. [31] first used pre-trained sequences to sequence the AraT5 and AraBART models for the GEC task in 2023. They first improved the multiclass Arabic grammatical error detection (GED) pre-trained transformers. In addition to the morphological preprocessing, they employed the GED data as an additional input to fine-tune the pre-trained transformers for the GEC. They used the ZAEBUC, QALB 2014, and 2015 datasets to train the model. They combined the training sets from the 2014 and 2015 challenges to experiment with the challenge's native and non-native test sets, and they combined the training sets from the three datasets to experiment with the ZAEBUC data to have as large a corpus as possible. The highest F1 score obtained for the QALB 2014 test set was 74.7%, followed by the QALB 2015 native test set at 77.1%, the non-native test set at 54.7%, and the ZAEBUC test set at 79.1%. After eliminating all datasets' punctuation errors, they reran the studies. This method produced F1 scores of 84.4%, 86.7%, 53.9%, and 83.2% for the same test sets, respectively.

In 2023, Sang Yun Kwon et al. [53] introduced techniques for generating synthetic data with ChatGPT by emphasizing the few-shot strategy. They contrasted this with AraT5v2, which is not pre-trained in MSA Arabic, and compared it with the Seq2edit method. The ChatGPT was utilized to create synthetic data consisting of 11 million sentences from the QALB2014 and 2015 datasets. The AraT5v2 pre-trained model achieved an F1 score of 67.7% on the QALB 2014 test data and 72.11% on the QALB 2015 native data after being trained on synthetic data.

In conclusion, previous research on Arabic GEC depends on generating synthetic text using various confusion functions or from external corpora, and subsequent research concentrated on the issue of insufficient training data by using various data augmentation techniques. Even though there are more training samples in each study, the outcomes only slightly improve when evaluated. Some methods use generated data to train the baseline transformer. In contrast, other methods use static word embedding, such as Fast Text, for word representation with conventional deep learning models, such as BiLSTM and CNN, and an attention mechanism. However, these methods could have produced better results. The training dataset in some models was normalized, which could harm the experiment's outcome because correcting all types of errors is one of the jobs that must be handled. In their most recent publication, Bashar Alhafni et al. conducted studies to eliminate punctuation errors that frequently need to be corrected. Because each language has its unique structure and this method may need more balanced data, fine-tuning a multi-language pre-trained transformer on a dataset of numerous distinct languages may not be an effective strategy. The study by Sang Yun Kwon et al. primarily concentrated on using ChatGPT to create synthetic data, noting limitations in generating high-quality Arabic language. The model AraT5v2 was utilized without pre-training on MSA Arabic, perhaps leading to training struggles. Table 2 represents all the previous studies including their methodology, used data set in training and testing, results, advantages, and limitations.

Algorithm 1 Applying transformers for Arabic GEC

```
Input: Raw Arabic sentences with different grammatical mistakes and their corresponding corrections $C$
Output: Fine-tuned GEC model that took grammatical error sentences and generated corrected versions for them $M$
  Data Preprocessing:
  Clean $C$ by removing sentence ids: $C_{clean} = c'_1, c'_2, ..., c'_N$
  Split each $c'_i$ into source $x_i$ and target $y_i$ sequences :$X = x_1, x_2, ..., x_N$ $Y = y_1, y_2, ..., y_N$

  Tokenization:
  Tokenize $X$ and $Y$ using tokenizer $t$: $\hat{X} = t(X)$ $\hat{Y} = t(Y)$

  Input Formatting:
  Add special tokens <START>, <EOS>, <PAD>:
  $\hat{X}, \hat{Y}$ = add_special _tokens $(\hat{X}, \hat{Y})$

  Pre-trained Model Loading:
  Load pre-trained transformer $f_\theta$ with parameters $\theta$
  Create attention masks $A_X$, $A_Y$

  Training:
  Randomly initialize $\theta$

  for epoch $e = 1$ to $E$ do
      for batch $b \in (\hat{X}, \hat{Y}, A_X, A_Y)$ do
          $\theta = \theta - \eta \bigtriangledown_\theta loss(f_\theta(b))$
      end for
  end for

  Decoding & Evaluation:
  Generate $\hat{Y}$ using $f_{\theta*}$ with beam search
  Post-process $\hat{Y}$ via detokenization $d$: $\tilde{Y} = d(\hat{Y})$
  Evaluate $f_{\theta*}$ using M$^2$ scorer
```

Table 2 Previous research summary

Authors	Methodology	Dataset	F1 score result (%)	Advantages	Limitations
Aiman Solyman et al. [11]	Apply fast embedding pre-trained word embedding - Applying encoder–decoder model using CNN and attention mechanism	Training Data: QALB 2014 and 2015 Training sets Testing Data: QALB 2015 Native	71.14	Using encoder–decoder architecture for GEC	Using the static fast embedding is not efficient method
Moukrim et al. [13]	Dictionary adopting - Using Domain Ontology	Arramooz Alwaseet Consists of 50,000 words divide into 10,000 verbs and 40,000 nouns	88.27	Using the language domain ontology for generating corrected sentences	The used data set is too small and its error types is unknown. - The methodology was not tested on a known benchmark to compare its result with other researches
Manar Al-Khatib et al. [14]	Language encoding - Model Learning	Training dataset: Collecting a corpus of 31 M word Testing dataset: BBC Arabic corpus CNN Arabic Corpus	93.89	Using a large corpus annotated by an expert for model training	Static embedding method which will not capture the complex errors in Arabic as morphological errors. - the types of errors corrected in the testing are unknown - The methodology was not tested on a known benchmark to compare its result with other researches
Aiman Solyman et al. [33]	Synthetic data generation - Fine-tuning - Model Learning	Training Data: Al-Watan It consists of 10 M words beside QALB training set Testing Data: QALB 2015 native	70.91	Generating synthetic data to overcome the issue of data sparsity	The quality of the generated synthetic is not good for the model training
Gheith A. Abandah et al. [34]	Using a BiLSTM for correcting spelling mistakes in classical and MSA	Training Data: -Tashkeela and ATB3 Training sets Testing Data: -Tashkeela and ATB3 Testing sets	Tashkeela test set: 98.7% ATB3 test set: 98.2%	Using a model for correcting mistakes in both classical and MSA	The model is just for correcting spelling mistakes only - There is no any other comparison with research on the same dataset
Aiman Solyman et al. [51]	Data Preprocessing - Synthetic data generation: - Model Learning - Training Optimizing	Training Data: Open Source International Arabic News (OSIAN) It consists of 367 M Word QALB 2014 and 2015 training sets Testing Data: QALB 2014 and 2015 test sets	QALB 2014 test set: 71.82% QALB 2015 native test set: 74.18%	Generating large number of synthetic data Adding training optimization method	The quality of the synthetic data is still an issue because as the generated synthetic is larger the result is not largely enhanced especially on the QALB 2014 test set
Krzysztof Pajk and Dominik Pajk [52]	Fine-tuning a single multi-language pre-trained seq2seq transformers mBART and mT5 to correct different errors in different languages including Arabic	Training data: Assembling datasets for different languages for Arabic they used QALB2014 and 2015 training sets Testing data: QALB 2014 and 2015 Testing sets	QALB 2014 test set: 65.98% QALB 2015 native test set: 70.04%	Using the transformer architecture for GEC	Fine-tuning a model for different languages includes Arabic is not efficient method because Arabic has a completely different structure than the other languages

Table 2 (continued)

Authors	Methodology	Dataset	F1 score result (%)	Advantages	Limitations
Aiman Solyman et al. [32]	Data augmentation - Model Learning - Decoding improvement	Training Data: QALB 2014 and 2015 training sets Testing Data: QALB 2014 and 2015 Testing sets	QALB 2014 test set: 71.04% QALB 2015 native test set: 73.52%	Applying data augmentation techniques rather than generating synthetic data	Using a base transformer might be not an efficient learning model for this problem. - The results of this research have been reduced compared to their prior research
Bashar Alhafni et al. [31]	morphological preprocessing - Grammar Error Detection - Grammar Error Correction	Training data: QALB 2014 and 2015 training sets ZAEBUC training set Testing Data: QALB 2014 and 2015 (native and non-native) testing set ZAEBUC testing set	QALB 2014 test set: 74.7% QALB 2015 native test set: 77.1% QALB 2015 non-native test set: 54.7% ZAEBUC test set: 79.1%	Using pre-trained seq2seq transformers for GEC - Apply the model on the QALB 2015 non-native dataset	Applying the morphological preprocessing will remove the hamza from all words from all sentences this will affect the training process - Using the traditional AraT5 is not the best choice for correcting MSA data. - The AraT5 and AraBART default settings are generating sentences of max length 512 and 1024 and the used testing data has sentences of larger length which will not be handled correctly
Sang Yun Kwon et al. [53]	Synthetic data generation using Chat GPT 4 - Grammar Error correction Using AraT5 V2	Training data: QALB 2014 and 2015 Training set Testing data: QALB 2014 and 2015 native test sets	QALB 2014 test set: 67.7% QALB 2015 native test set: 72.11%%	Applying the few-shot strategy for generating Synthetic data	Chat GPT still has some issues in generating correct Arabic text and this will affect the quality of the synthetic data - The AraT5 v2 has also a maximum length of 1024 and not pre-trained on MSA Arabic as the testing data which will not produce accurate result

3 Methodology

The following section presents the proposed methodology for the current study. The structure of the proposed Ara GEC transformer model is illustrated in Fig. 2 and described in Algorithm 1. The process comprises two primary stages. Phase 1 encompasses The initial phase of the model, which involves performing preprocessing on the training dataset. During this phase, the source sentences, which include various errors, and their matching target phrases, which are the corrected versions obtained from the provided.m2 file. Subsequently, all of these sentences go through a process of cleaning by removing the sentence ID. During tokenization, sentences are segmented into a sequential of tokens. The choice of tokenizers depends on the selection of pre-trained transformers. An illustration of this in seen in the T5 transformer, which utilizes the T5 tokenizer, and the BART transformer, which uses the BART tokenizer. The research employed the AraT5 and AraBART versions and the identical tokenizers utilized in the original study [27, 28]. Text normalization is not employed during the preprocessing stage, as the primary aim of this study is to rectify all types of mistakes that are found in any dataset. Moreover, it is crucial to recognize that the removal of punctuation is a mistake since it is another error that the proposed model has to correct. This study differentiates itself from previous research, including the investigations conducted by Bashar Alhafni [31], by utilizing identical datasets and transformers. Both tokenizers perform the same function of converting raw text into a series of tokens. However, the T5 tokenizer is specifically designed for the T5 transformer architecture, which operates within a text-to-text framework. In this model, the input and output texts are tokenized similarly, with a distinct delimiter separating them. In T5 models, the unique token "grammar:" was added to sentences to identify that it is a correction task. The BART tokenizer employs a typical approach to tokenize text, distinguishing between input and output by utilizing classic, unique tokens such as [CLS] and [SEP] to demarcate them. To formalize the BART tokenizer, three unique tokens were utilized in this study: <START> to signal the beginning of a sentence, <END> to indicate the end of a phrase, and <PAD> to indicate the padding of the sentence. Attention masks were constructed at the end of this tokenization phase to indicate which tokens are legitimate inputs and which are padding. Upon the completion of the tokenization phase, the transformers are prepared for the training process. This study utilized the AraT5 and mT5 variants derived from the T5 model and the AraBART and mBART variants derived from the BART model. The training process utilizes the training set, while the validation process uses the development set. The current pre-trained models can receive input data the encoder component will then convert the tokenized input into numerical embeddings. Subsequently, incorporate supplementary positional encodings with the token embeddings to provide the model with insights into the specific location of each token inside the input sequence. Subsequently, the data transits the stacked transformer layer, which exhibits a distinction between T5 and BART. In T5, this layer has a multi-head self-attention mechanism that enables the model to allocate weights to different input text segments. This facilitates the model in capturing contextual associations among words [49]. The model incorporates a position-wise feed-forward layer, propagating the attention information through a feed-forward neural network. This network operates independently in each location. The BART transformers employ stacked layers that include a bidirectional self-attention layer. This layer enables the model to effectively capture contextual information from every token's left and proper contexts. This technique is commonly observed in the training phase for the bidirectional objective. The auto-regressive decoder layer is a component that is included in the model architecture. During auto-regressive pre-training, the model generates text by producing one token at a time, employing a decoder configuration [44]. The model exclusively focuses on the preceding context on the left side to guarantee the resulting content's coherence. There is a distinction between the current model and T5, as the latter employs a bidirectional encoder for input and output. Following the stacked layer phase, the transformers go to the training phase. The objective function of the T5 training process training model the model to minimize the negative log-likelihood associated with creating the correct output, given a specific input. The loss function being referred to is commonly referred to as the maximum likelihood estimation (MLE) loss, which may be mathematically represented as [32]:

$$MLE = \mathbb{E}_{x,y \sim \hat{p}(X,Y)}[\log P(y \mid x)] \tag{1}$$

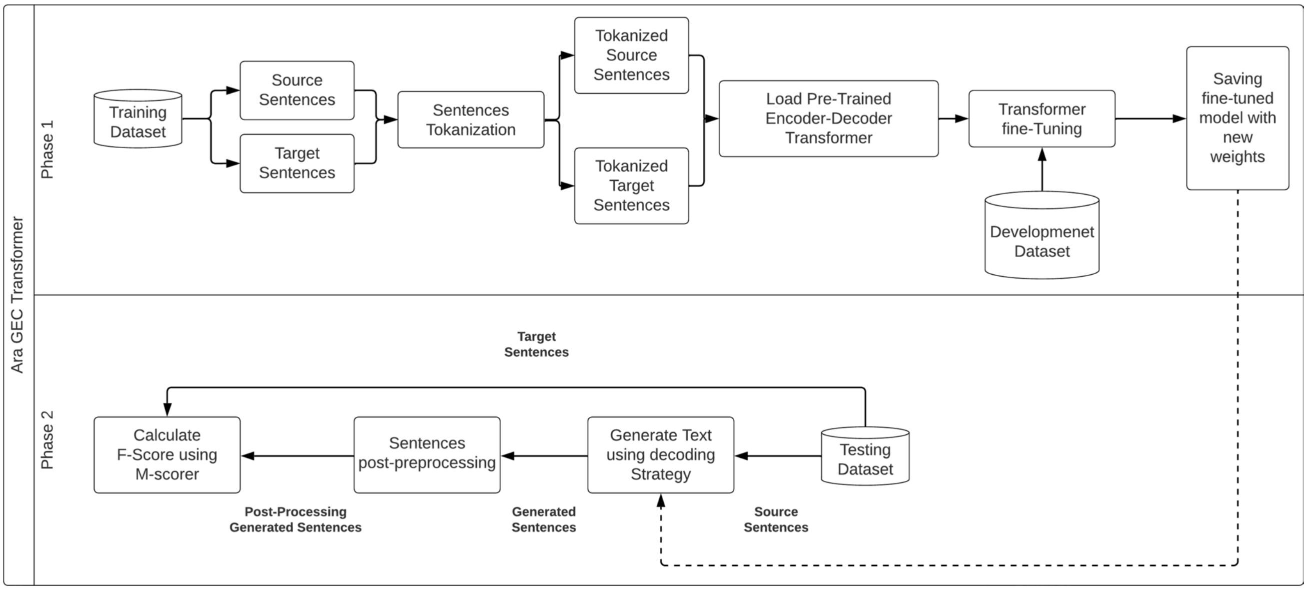

Ara GEC Transformer
Phase 1
Training Dataset
Source Sentences
Target Sentences
Sentences Tokanization
Tokanized Source Sentences
Tokanized Target Sentences
Load Pre-Trained Encoder-Decoder Transformer
Transformer fine-Tuning
Developmenet Dataset
Saving fine-tuned model with new weights
Phase 2
Target Sentences
Calculate F-Score using M-scorer
Sentences post-preprocessing
Generate Text using decoding Strategy
Testing Dataset
Post-Processing Generated Sentences
Generated Sentences
Source Sentences

Fig. 2 Proposed model—Ara GEC transformer

where p (X,Y) represents the empirical distribution of the input–output pairs found in the training data. In this context, x and y represent the input and output sequences.

The BART training objective is a training objective that combines bidirectional and auto-regressive training objectives. During bidirectional pre-training, BART employs a masking technique to obscure and predict random text spans. In the context of auto-regressive pre-training, the task involves making predictions about the subsequent token inside a given sequence. This dual pre-training methodology allows BART to comprehend language in both directions and produce coherent and contextually appropriate content [26, 44]. Upon the completion of this training phase, the trained models were preserved for subsequent application on various test texts. During phase 2, the trained model underwent experimentation by applying the test set. The model applies each source sentence to construct a corresponding target sentence. The sentence is generated from the model using a decoding approach. In this study, various decoding strategies, such as greedy decoding and nucleus sampling, were employed in addition to the commonly utilized algorithm known as beam search. Despite the superiority of beam search, the rationale behind employing greedy decoding is its lower computational complexity and the potential for achieving satisfactory decoding outcomes if the model is well-trained [54–56]. The rationale behind utilizing nucleus sampling is its capacity to dynamically select the quantity of created possibilities. The decoded sentences underwent post-processing through de-tokenization and the removal of unique tokens. This modification resulted in the sentences being presented in a way that individuals understand. The final stage of the model involves evaluating its performance in the context of correction. The sentences generated by the model were compared to the target sentences using the M^2 scorer, as done in earlier studies, to facilitate the comparison of the results.

4 Experiments

This section presents the experiments executed for this research, including the settings and datasets used. The details of the settings and datasets are described and analyzed. This section describes the performance measurement of the experiments and its significance to the GEC task.

4.1 Dataset

The research was conducted using the QALB dataset, which underwent two challenges: The first was introduced in 2014, followed by the second in 2015. The corpus under consideration was a joint endeavor undertaken by Columbia University and Carnegie Mellon University Qatar, with financial support provided by the Qatar National Research Fund. This dataset represents a subset of annotated data available for Arabic grammar error correction. The data was obtained from the written materials from those who commented on the Al Jazeera news program. The primary distinction between the datasets from 2014 and 2015 lies in the composition of the test data. Specifically, the 2014 dataset only comprises test data authored by native Arabic speakers. In contrast, the 2015 dataset encompasses two distinct test sets: L1, consisting of papers written by native Arabic speakers, and L2, comprising documents produced by non-native speakers. Notably, the L2 test set demonstrates a higher frequency of complex errors. [31, 33, 51]. Tables 3 and 4 provide comprehensive information about the QALB datasets for 2014 and 2015. In this study, an investigation was conducted on each dataset to collect data regarding the mean and maximum sentence lengths. This information is crucial for determining the appropriate decoding size of the applied transformer since it directly impacts the resultant outcomes. The analyses above can be visually represented by figures ranging from Figs. 3, 4, 5, 6, 7, 8, 9.

Tables 3 and 4 indicate that QALB 2015 exhibits greater complexity than 2014. This is mainly attributed to the limited training data and more intricate errors in the L2 testing data, as highlighted in Table 1. In prior studies, the researchers addressed the constraint of QALB data by merging the two datasets and employing synthetic data approaches. However, in the current experiments, each dataset was trained independently as distinct data. One additional concern with the QALB 2015 L2 test set is its lack of prior experimentation, which is essential given the complexity of its nature. Therefore, the applied experiments have a unique characteristic, including training the model exclusively on the training data, which comprises only 310 phrases. Another reason for training each dataset independently is to demonstrate the efficacy of transfer learning methodology and the utilization of pre-trained transformers trained on extensive datasets. This approach is expected to outperform traditional deep learning techniques or baseline transformers trained on artificially generated synthetic data. Additionally, it aims to establish that even a limited amount of data can yield superior outcomes.

4.2 Experiments setting

The primary aim of this study is to implement grammatical error correction in the Arabic language as a downstream task for pre-trained transformers. AraT5 and AraBART were utilized as transformer models that underwent complete training exclusively in the Arabic language, whereas mT5 and mBART were employed as Multilanguage transformers that encompass Arabic as one of their supported languages [29, 30]. In the QALB 2014 experiments, multiple versions of AraT5 were made accessible. The selected versions for the experiments were AraT5 MSA small and AraT5 MSA base. These versions were chosen due to their training in MSA Arabic, which aligns with the type of Arabic present in the used dataset. One notable distinction between the small and base versions is in the disparity of trainable parameters, with the base version exhibiting a more significant number of such parameters than the small version. The AraBART currently provides only one version, the base version [28]. The reason for not utilizing the mT5 and mBART models in the QALB 2014 task was more hardware capabilities. Specifically, the hardware could not train 19,411 words using a model with 580 M trainable parameters for mT5

Table 3 QALB 2014 Dataset Description

	Type of the dataset	Number of sentences	Number of Words	Level of speaker
QALB 2014	Training	19,411	1,021,165	Native
	Development	10,17	53,737	Native
	Testing	968	51,285	Native

Table 4 QALB 2015 dataset description

	Type of the dataset	Number of sentences	Number of Words	Level of speaker
QALB 2015	Training	310	43,353	Non-native
	Development	154	24,742	Non-native
	L1-Testing	920	48,547	Native
	L2-Testing	158	22,808	Non-native

and 611 M trainable parameters for mBART. For the 2015 QALB, all the models used in the 2014 dataset were used in addition to mT5 and mBART. All of the used models were trained on the specifications listed in Table 5 and the settings in Table 6.

During the testing phase, two additional decoding strategies, greedy decoding and nucleus sampling, were employed alongside the previously utilized beam search decoding methodology, the only method in the prior study. This study employed several hyperparameter configurations for all decoding methods, focusing on beam search and greedy algorithms. Specifically, beam search was tested using 2, 5, and 7 beams, while the greedy algorithm was evaluated with n-gram values of 2, 5, and 7. The rationale behind doing these trials with various parameter settings is to optimize the performance of the correction process.

Fig. 3 QALB 2014 training data analysis

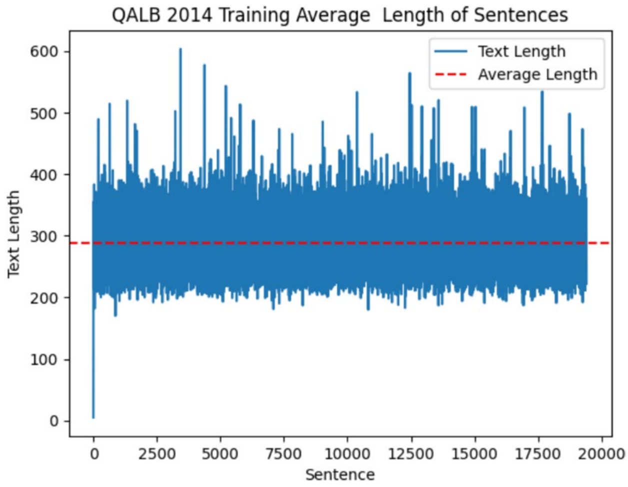


Fig. 4 QALB 2014 development data analysis

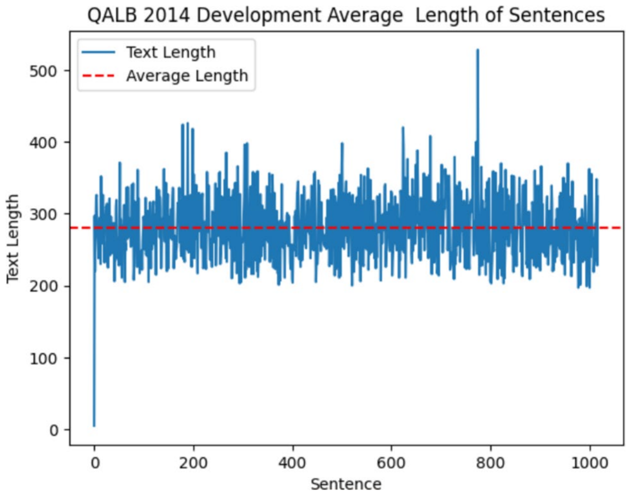


Fig. 5 QALB 2014 testing data analysis

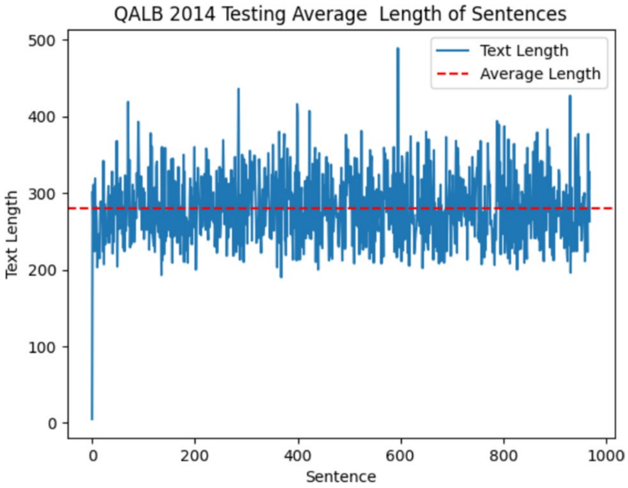


Fig. 6 QALB 2015 training data analysis

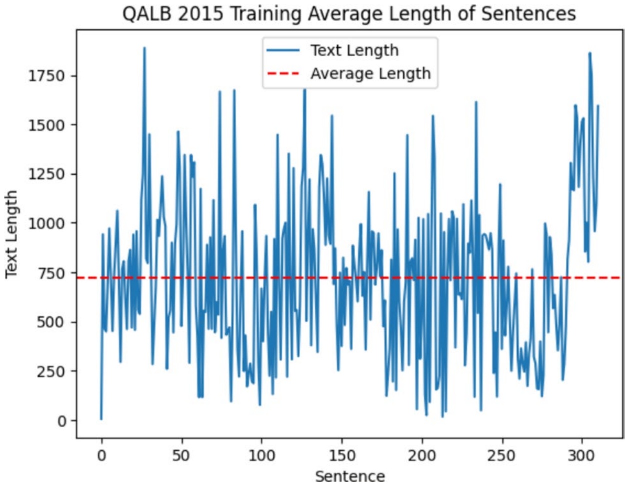
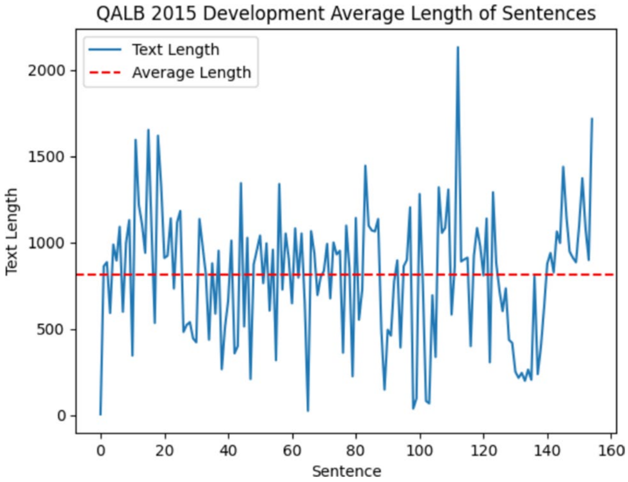


Fig. 7 QALB 2015 development data analysis



4.3 Performance measurement

The Max match tool supplied by the QALB dataset was utilized in this study to comprehensively compare the research findings with previous studies. The max match algorithm is utilized to forecast the sentences predicted

Fig. 8 QALB 2015 L1-testing data analysis

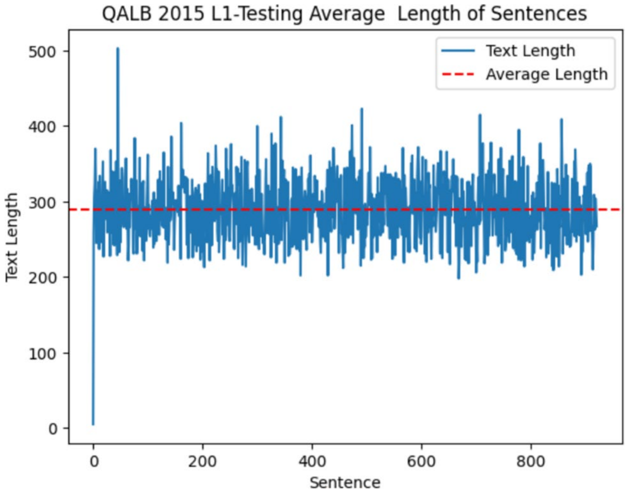
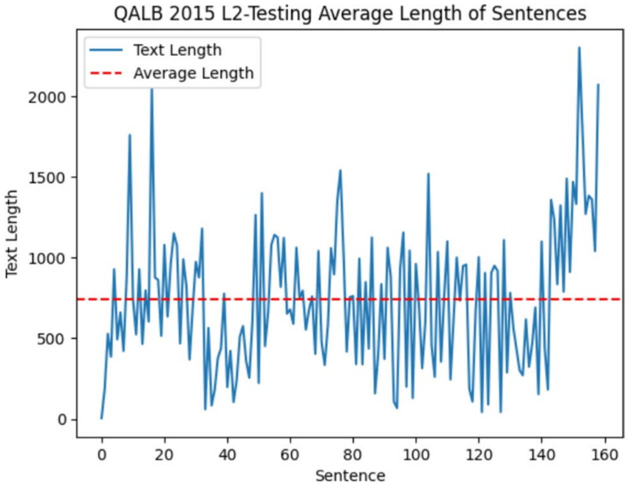


Fig. 9 QALB 2015 L2-testing data analysis



by the trained transformers [31, 51, 52]. These predictions are compared against the gold sentences, representing the test datasets' target sentences. The algorithm then calculates and outputs the precision, recall, and F-score. Precision refers to the accuracy of the GEC system in identifying the errors. Precision (*P*), which is known as the true positive rate, is calculated by dividing the number of calculated true positives by the total of true positives and false positives. This is represented mathematically using the following equation: [13, 25]

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{2}$$

where true positives (*TP*) refer to the errors that the system accurately identifies, and false positives (*FP*) denote the errors that the system wrongly corrects. In the field of GEC, recall (*R*) refers to the proficiency of GEC systems in accurately identifying errors. The recall refers to the extent to which the GEC system accurately corrects the errors. The term recall is also known as the *TP* rate; it is calculated by dividing the number of *TP* by the sum of *TP* and false negatives *FN*. The following equation can represent this mathematical relationship [25]:

$$\mathrm{R} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{3}$$

where *FN* refers to the instances in which the system fails to detect faults.

Table 5 Hardware and software specifications

Aspect	Specification
GPU	Nvidia RTX 3070 @1.73GHZ with 8 GB vRAM
Memory	32GB RAM
CPU	Intel I7 12700f @3.4GHZ
OS	Windows 11
Implementation platform	Spyder IDE in the Anaconda navigator
Implantation Language	Python 3.10
Libraries	-CUDA toolkit version 12.3 -PyTorch version 12.1 -hugging face library

Table 6 Training settings

Parameter	Value
learning_rate	5e-5
Batch size	1
Number of epochs	30
Training optimizer	AdamW

In GEC systems, a high *P* score indicates the system's proficiency in accurately correcting the errors. Conversely, a high *R* score indicates the system's effectiveness in correcting errors, while it may need to possess the capability to correct them consistently. The F-score is the harmonic mean of *P* and *R*. The metric is a comprehensive measure that reaches a balance between the trade-off of *P* and *R*. This study evaluated the F-score using two metrics: the F1 score and the F0.5 score, which are represented by the following mathematical equations [25, 31]:

$$\text{F1 score} = \frac{2 \times \text{P} \times \text{R}}{\text{P} + \text{R}} \tag{4}$$

The distinction lies in the fact that F0.5 assigns greater significance to *P* compared to *R*. The primary objective of the GEC is to rectify errors with *P* effectively.

$$F_\beta = \left(1 + \beta^2\right) \cdot \frac{\text{P} \cdot \text{R}}{\left(\beta^2 \cdot \text{P}\right) + \text{R}} \tag{5}$$

where β is the parameter that controls the balance between *P* and *R*.

5 Results and discussion

This section presents and discusses the results of all conducted tests. Each model's experimental findings are shown in separate tables, utilizing different decoding techniques and hyperparameters. The outcomes of each table are then explained. Furthermore, the conclusion of this part presents and analyzes the findings of this research compared to previous studies.

The comparative analysis of experimental outcomes is presented in Tables 7, 8, 9, 10, which demonstrates that the proposed experiments yielded superior findings compared to the previous research. The AraT5 base model, when utilized in conjunction with beam search decoding, yielded the most optimal outcomes. The AraT5 base is more significant than the smaller AraT5 variant, so it encompasses more parameters. Consequently, this expanded parameter space enables the AraT5 base to attain more precise attention scores during the learning process, ultimately leading to superior outcomes [27]. However, AraT5 small also achieved

Table 7 QALB test results using AraT5 small-MSA and beam search decoding

QALB Test Results Using AraT5 Small-MSA and Beam Search Decoding												
No of beams	2014				2015 L1-Test				2015 L2-Test			
	P	R	F1	F0.5	P	R	F1	F0.5	P	R	F1	F0.5
2	92.8	68.9	74.5	81.7	80.7	78.3	79.4	80.1	80.4	73.7	76.4	78.6
5	92.9	71.7	76.9	83.3	81.1	79	80	80.6	80	73	77	78.8
7	93.1	72.5	93.1	72.5	81.1	79	80	80.7	80	74	77	78.8

Table 8 QALB test results using AraT5 small-MSA and greedy search decoding

QALB Test Results Using AraT5 Small-MSA and Greedy Search Decoding												
No N-grams	2014				2015 L1-Test				2015 L2-Test			
	P	R	F1	F0.5	P	R	F1	F0.5	P	R	F1	F0.5
2	81.9	67.2	69.2	74.0	72.3	77.1	74.4	73.1	63.6	71.3	66.7	64.7
5	90.8	67.0	72.4	79.6	79.5	78	78.6	79.2	78.2	74.3	76	77.2
7	91.4	66.8	72.4	79.8	78.8	78	78.8	79.4	79.1	74.2	76.3	77.9

superior performance compared to conventional deep learning models. The performance of beam search decoding was superior to that of greedy search. This is because beam search creates a more significant number of probabilities for word prediction than the greedy approach, which relies solely on selecting the best word at each timestep. However, both methods' outcomes are similar, indicating that AraT5 models have effectively learned contextual word representations. The number of beams and n-gram hyperparameters exhibits a marginal outcome disparity. Based on previous research, it has been determined that the ideal value for the number of beams is 5. However, in this study, alternative possibilities exist, as the number of beams set to 7 yielded more favorable results in certain studies. In contrast, other beam numbers produced results near this optimal value. The concept above is also observed in greedy search, where the outcomes obtained from various n-gram values exhibit high proximity.

The correction findings obtained using the AraBART transformer are presented in Tables 11 and 12. The evidence demonstrates that AraBART achieved superior outcomes compared to AraT5. The difference in training objectives between AraBART and AraT5 lies in using a denoising autoencoder objective by AraBART [28]. The system is designed to effectively restore a document by utilizing a corrupted version of the same document. This enhances the transformer's capacity to generate higher attention scores during the training phase, improving the generated text's accuracy. The tests evaluated beam search and greedy search decoding techniques using various hyperparameters. The beam search algorithm yielded consistent results across various beams, suggesting a high confidence level in the model. The greedy search approach yielded lower accuracy scores for the number of grams equal to 2 than the accuracy scores obtained for n-grams equal to 5 and 7. This is because beam search decoding prioritizes the exploration of more predictable terms, increasing the range of high probabilities [54].

Tables 13 and 14 present the results obtained from the training of the mT5 base. The utilization of the base version of mT5 is justified due to the following rationale: Given that the AraT5 basic version achieved a higher score than the AraT5 small version, it was anticipated that the mT5 base version would also yield superior results compared to the mT5 small version. Despite the availability of larger models such as mT5 large, XL, and XXL, which are expected to yield improved results due to their more significant number of parameters [29], these models were not utilized in the experiment. This decision was primarily driven by hardware limitations, specifically the lack of the GPU memory to accommodate the substantial number of parameters associated with these models. Consequently, the experiment involving the application of these models to the QALB 2014 dataset was unsuccessful, as this dataset contains a more significant number of

Table 9 QALB test results using AraT5 base-MSA and beam search decoding

QALB Test Results Using AraT5 Base-MSA and Beam Search Decoding												
No of beams	2014				2015 L1-Test				2015 L2-Test			
	P	R	F1	F0.5	P	R	F1	F0.5	P	R	F1	F0.5
2	93.5	58.8	67.4	77.6	81	79.7	80.3	80.7	81.6	76.9	79	80.3
5	93.7	63.4	71.2	80.3	81	79.8	80.3	80.7	81.6	77.7	79.4	80.7
7	93.7	64.3	71.8	80.7	81	79.8	80.3	80.7	81.7	77.8	79.6	80.8

Table 10 QALB test results using AraT5 Base-MSA and greedy search decoding

QALB Test Results Using AraT5 Base-MSA and Greedy Search Decoding												
No N-grams	2014				2015 L1-Test				2015 L2-Test			
	P	R	F1	F0.5	P	R	F1	F0.5	P	R	F1	F0.5
2	84.0	53.8	60.2	69.4	74.2	78	76	74.8	69.4	73.3	70.9	69.8
5	91.4	53.2	62.2	73.4	80	79.8	80	80.2	80.39	78.2	79.1	79.8
7	92.1	53.3	62.4	73.8	80.4	79.8	80.	80.2	80.82	78	79.2	80.1

Table 11 QALB test results using AraBART and beam search decoding

QALB Test Results Using AraBART and Beam Search Decoding												
No of beams	2014				2015 L1-Test				2015 L2-Test			
	P	R	F1	F0.5	P	R	F1	F0.5	P	R	F1	F0.5
2	94.7	90.3	92.1	93.3	89.56	89.1	89.3	89.4	84.2	83	83.6	84
5	94.8	90.3	92.1	93.3	89.56	89.1	89.3	89.4	84.2	83	83.6	84
7	94.8	90.3	92.1	93.3	89.56	89.1	89.3	89.4	84.2	82.9	83.5	84

Table 12 QALB test results using AraBART and greedy search decoding

QALB Test Results Using AraBART and Greedy Search Decoding												
No N-grams	2014				2015 L1-Test				2015 L2-Test			
	P	R	F1	F0.5	P	R	F1	F0.5	P	R	F1	F0.5
2	91.7	90	90.7	91.4	87.8	88.1	87.8	87.7	79.9	81.8	80.6	79.2
5	94.	90.4	92	93.2	89.2	89	89.1	89.2	83.9	82.9	83.4	83.7
7	94.	90.4	92	93.3	89.3	89	89.1	89.2	84	82.9	83.4	83.8

records than QALB 2015, necessitating additional GPU memory for training. Furthermore, the findings indicate that mT5, which experiences pre-training on a multilingual corpus consisting of 101 languages, including Arabic, outperforms AraT5, which is exclusively trained in Arabic. This difference can be attributed to the reason that the mT5-base has a larger parameter count of 600 million, whereas the AraT5 small has 300 million parameters [27]. Consequently, the increased parameter count of mT5 contributes to its superior performance. The beam search algorithm demonstrated higher performance compared to the greedy search algorithm but with a slightly smaller margin of difference. In general, the outcomes of this research experiment were superior to those of the research that utilized mT5 for a compiled Arabic corpus. This discrepancy can be attributed to the fact that the languages within this corpus possess distinct structural characteristics from Arabic, negatively affecting both the training and decoding phases.

The results obtained using mBART, which has a fixed size of 610 M parameters, are presented in Tables 15 and 16. It has been observed that mBART achieved lower scores than mT5, although it has more parameters.

Table 13 QALB test results using mT5 base and beam search decoding

QALB Test Results Using mT5 Base and Beam Search Decoding

No of beams	2015 L1-Test				2015 L2-Test			
	P	R	F1	F0.5	P	R	F1	F0.5
2	86.2	85.4	85.8	86.0	82	79.4	80.6	81.4
5	86.3	85.4	85.8	86.1	82	79.6	80.7	81.4
7	86.3	85.4	85.8	86.1	82	79.6	80.7	81.4

Table 14 QALB test results using mT5 base and greedy search decoding

QALB Test Results Using mT5 Base and Greedy Search Decoding

No N-grams	2015 L1-Test				2015 L2-Test			
	P	R	F1	F0.5	P	R	F1	F0.5
2	75.3	83.1	78.6	76.5	60.89	69.8	64.1	61.8
5	85.4	84.9	85.	85.2	79.2	78.8	78.1	79.0
7	85.8	85.1	85.3	85.6	80.4	79.3	79.7	80.1

Additionally, AraBART outperformed AraT5 in terms of outcomes. This adjustment is because mT5, with its 24 layers, has more potential for learning complex patterns than mBART, which only has 12 layers [30]. Consequently, the estimated attention scores are expected to be superior in mT5. In these experiments, both beam search and greedy search yielded similar results. However, the number of beams and n-gram changes, indicating that the model will not further optimize if the number of epochs increases.

As previously stated, one of the research aims is to find an alternative for beam search decoding. Table17 displays the results of nucleus sampling, sometimes known as top-p sampling and represented mathematically as [54, 56]:

$$\sum_{x \in V^{(p)}} P\left(x \mid x_{1:i-1}\right) \geq p \tag{6}$$

where $V^(p)$ is the smallest possible of tokens. $P(x|\ldots)$ is the probability of generating token x given the previous generated tokens x from 1 to i-1.

Setting p to 0.8 for all of the models under evaluation allows for a harmonious equilibrium by selecting elements that are highly likely to occur, potentially leading to recurring or predictable patterns, while also allowing for some degree of randomness by analyzing only a subset of the total distribution. The rationale for selecting this decoding technique is its unique objective, which sets it apart from both beam and greedy decoding methods and determines the minimum gathering of words with a cumulative probability that exceeds the probability p. Thus, the size of the term set can flexibly expand based on the probability distribution of the subsequent word. The beam search aims to predict the next word based on the context. Instead of selecting only the most likely word, beam search keeps the gathering of the top candidates using the beam width. These candidates are augmented with supplementary terms, increasing their probability by twofold. Subsequently, the set is reduced to retain only the most promising candidates. It is seen that AraBART outperformed araT5 in the QALB 2014 and 2015 datasets, both for native and non-native languages. Additionally, mT5 achieved better results than mBART. However, when comparing these results to beam search, it is evident that it yields slightly superior outcomes compared to nucleus sampling. This suggests that the nature of the issue data and training method is more aligned with the decoding purpose of beam search.

The figures displayed show the results of each dataset after applying various transformers and carrying out experiments. Figure 10 depicts the results of QALB 2014, indicating that AraBART outperformed

Table 15 QALB test results using mBART large and beam search decoding

QALB Test Results Using mBART Large and Beam Search Decoding								
No of beams	2015 L1-Test				2015 L2-Test			
	P	R	F1	F0.5	P	R	F1	F0.5
2	83.79	78.5	80.4	82	63.6	71.3	62.3	70.4
5	84.1	79.7	81.5	82.9	69.2	72.8	69.8	70.4
7	84.1	79.8	81.6	82.9	69.3	73.2	70.6	70.4

Table 16 QALB test results using mBART large and greedy search decoding

QALB Test Results Using mBART Large and Greedy Search Decoding								
No N-grams	2015 L1-Test				2015 L2-Test			
	P	R	F1	F0.5	P	R	F1	F0.5
2	66.6	76.6	70.1	82.0	46.8	70.3	53.7	61.3
5	82.8	76.6	78.5	82.9	51.9	78.2	60	68.4
7	82.9	76.9	78.8	82.9	52.0	78.2	60.1	68.4

Table 17 QALB test results using nucleus sampling decoding on AraT5 and AraBART

QALB Test Results Using Nucleus sampling decoding on AraT5 and AraBART												
Model	2014				2015 L1-Test				2015 L2-Test			
	P	R	F1	F0.5	P	R	F1	F0.5	P	R	F1	F0.5
AraT5 Small- MSA	92.9	54.3	63.4	74.9	78	76	76.9	77.5	78.7	73.3	75.7	77.4
AraT5 Base- MSA	91.8	65.5	71.5	79.2	80	78.8	79.3	79.7	81	77.1	78.9	80.1
AraBART	94	90.3	92.1	93.3	89.3	88.9	89.3	89.4	84.1	82.9	83.6	83.9
mT5 Base	–	–	–	–	85.2	84.2	84.6	85	81.4	76.7	78.7	80.1
mBART Large	–	–	–	–	81.2	73.6	76	78.3	79.3	60	65.9	71.7

AraT5 small and Base in precision, recall, F1, and F0.5 scores. This is due to the bidirectional and denoising autoencoder training objective, which involves generating corrupted versions of each training sample and learning to predict the original document by randomly masking and permuting parts of it. This process helps in generating improved attention scores for the training samples. The decoding strategy of beam search, with beam sizes of 2, 5, and 7, yielded identical results. Therefore, selecting a beam size of 2 is optimal for this strategy, as it incurs lower computational complexity. Nevertheless, it is important to note that the nucleus sampling technique yielded identical outcomes to the beam search method. Consequently, nucleus sampling may be a preferable alternative to beam search, given its computational efficiency, as the time complexity of the beam search $O(|V| \log |V|)$, where K represents the beam width and |V| represents the vocabulary size, while the time complexity of nucleus sampling is $O(|V| \log |V|)$

Figures 11 and 12 display the outcomes of applying the same models on QALB 2015 L1 (the native data) and QALB 2015 (non-native data), together with mT5 and BART. Additionally, AraBART demonstrated superior precision, recall, F1, and F0.5 scores. Five score outcomes in both datasets other than AraT5 versions were the same as the QALB 2014 results. Furthermore, when beams were searched using different parameters, they yielded the same findings alongside the nucleus sampling. Therefore, opting for nucleus sampling would also be a superior decision. Surprisingly, mT5 outperformed mBART in these datasets, which can be attributed to the fact that mT5 possesses a larger number of layers, enabling it to capture more complex patterns.

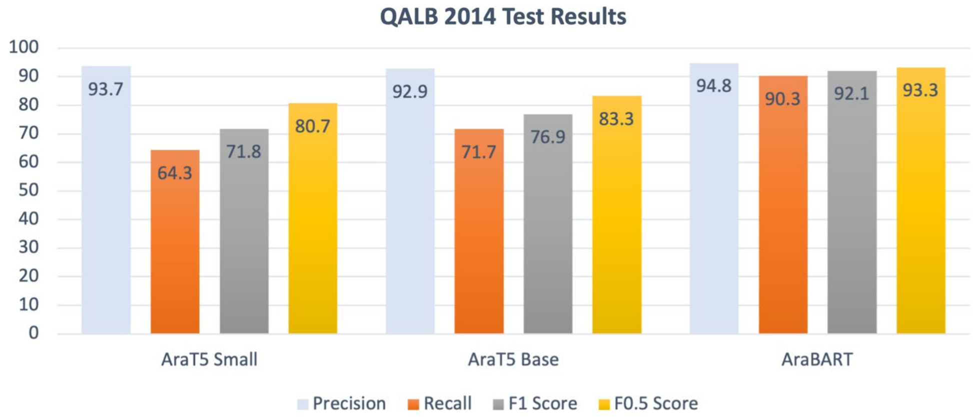


Fig. 10 QALB 2014 Testset Results Summary.

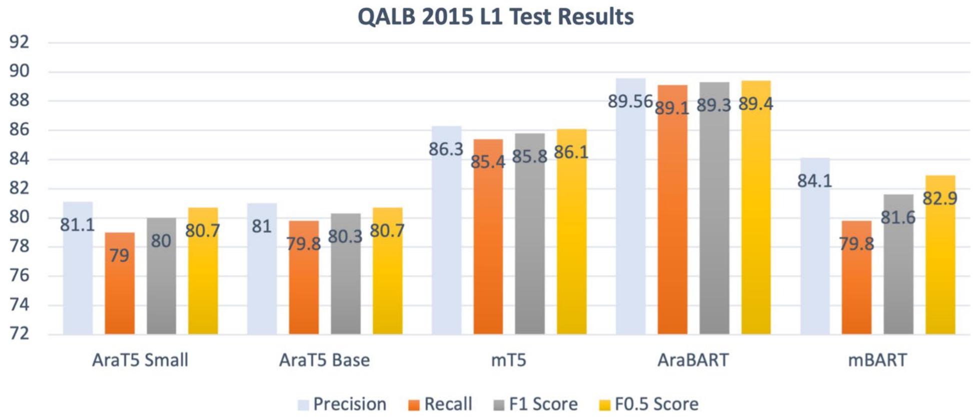


Fig. 11 QALB 2015 L1 testset results summary

Table 18 presents the results of this research in comparison with prior research conducted on the same datasets. This research has demonstrated superior results compared to all previous results. This is because earlier research utilized different synthetic data derived from other corpora, which were not part of the original dataset. The primary objective of these studies was to generate a significant number of words for training standard deep learning models or baseline transformers. In addition, they merged the training datasets from both QALB 2014 and 2015 (Table 19).

However, many of them only evaluate their models on the native data from QALB 2015, disregarding the non-native data. Krzysztof Pajk et al. [52] employ a methodology that integrates diverse datasets containing various languages and trains the mT5 and mBART transformers using these datasets. However, this strategy could have produced a satisfactory result. On the other hand, employing these transformers and exclusively training them on the Arabic dataset has yielded superior results. Bashar Alhafni et al. [31] utilized the AraT5 and AraBART models on QALB 2014 and 2015 datasets, both for native and non-native languages. However, they employed morphological preprocessing techniques to eliminate data errors before applying the models. Additionally, they initially used the models on a GED task to enhance the accuracy of the results. However, it corrected the various grammatical errors previously mentioned in Table 1. This is demonstrated in Table 20, where examples from the native dataset are provided, showcasing the source, target, and generated sentences. The same pattern is observed in Table 21, displaying examples of source, target, and generated sentences from the non-native data. Both tables show the efficiency and quality of the generated sentences compared to the target.

Table 19 displays the duration of training for each utilized pre-trained model on every training dataset. All of these time intervals are expected to be reduced. However, the training batch was restricted to one due to limited GPU memory, which can only load one sample at a time. QALB 2014 required a longer training period

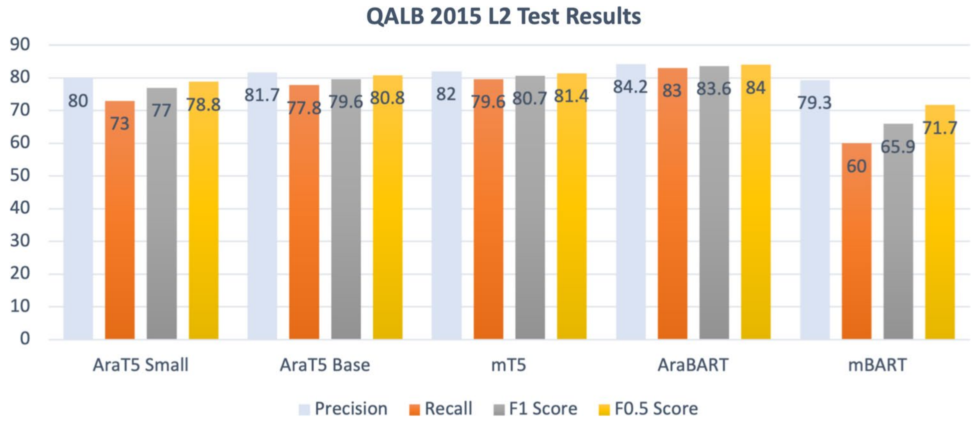


Fig. 12 QALB 2015 L2 test-set results summary

than QALB 2015 because of its larger number of samples. Additionally, training QALB 2014 on mT5 and mBART models was impossible due to their significant size, which exceeded the hardware limitations. The training time for the QALB 2015 dataset was long despite the tiny sample size. This was caused by changing the model's maximum length to 2500 to accommodate the maximum length of the data and achieve better correction results.

6 Limitations

Although the suggested research produced better results than previous studies, limitations remain. This model's sentence length is a constraint, as it has been trained on the maximum length of the given testing dataset. If there are different testing data with longer lengths, it may result in unexpected outcomes. This issue is open for future research. The data quality remains limited as the dataset lacks noise, such as mentions and hashtags. This issue could be addressed by implementing a data-cleaning procedure prior to sentence correction. Another data quality issue involves handling sentences with diacritics. If available, this can be addressed by training the model using supervised discretized sentences. The final concern pertains

Table 18 Comparison between the achieved results and the previous ones

Researchers	Year	F1 Score on QALB 2014	F1 score on QALB 2015. native data	F1 score on QALB 2015 non-native data
Aiman Solyman et al	2019	–	71.14	–
Aiman Solyman et al	2021	–	70.91	–
Aiman Solyman et al	2022	71.82	74.18	–
Krzysztof Pajk et al	2022	65.98	70.04	–
Aiman Solyman et al	2023	–	73.52	–
Bashar Alhafni et al	2023	74.7	77.1	62.4
Sang Yun Kwon et al	2023	67.7	72.11	–
Proposed Work Results	2024	92.1	89.4	83.6

Table 19 Training time for each model

Training set	AraT5 MSA Small	AraT5 MSA base	AraBART	mT5-Base	mBART large
QALB 2014	9 h	18.5 h	12 h	–	–
QALB 2015	6 h	14 h	8 h	16 h	11 h

https://doi.org/10.1007/s00521-025-11145-1

Table 20 Examples for the proposed model correction from the native data

Type of sentence	Samples
Source	الحمد لله الذي بعث محمدا نبيا و جعلنا مسلمين و بعد ارسل رسالتي لمن رسم الرسوم و أسأله هل رأى الرسول حتى يقوم برسم صورته ، هل تعلم من هو محمد لن اقول عندنا نحن المسلمين بل عند الغرب الذين كتبوا عنه ، هو أعظم العظماء المائة الذين كتب عنهم
Target	:الحمد لله الذي بعث محمدا نبيا وجعلنا مسلمين ، وبعد أرسل رسالتي لمن رسم الرسوم وأسأله هل رأى الرسول حتى يقوم برسم صورته ؟ هل تعلم من هو محمد ؟ لن أقول عندنا نحن المسلمون بل عند الغرب الذين كتبوا عنه ، هو أعظم العظماء المائة الذين كتب عنهم
Generated	:الحمد لله الذي بعث محمدا نبيا ، وجعلنا مسلمين . وبعد ، أرسل رسالتي إلى من رسم الرسوم ، أسأله هل رأى الرسول حتى يقوم برسم صورته ؟ هل تعلم من هو محمد ؟ لن أقول عندنا نحن المسلمون بل عند الغرب الذين كتبوا عنه ، هو أعظم العظماء المائة الذين كتب عنهم
Source	مع احترامي للتعليقات التى وان اختلفت كثيرا لكنها تعبر عن وجهات نظر محترمة بس المقال الجميل للأستاذ / عمر محمد ورسمة يجسد الواقع الحالي للصومال لان الصومال بلد واحد اللغة واحدة والديانة واحدة بل حتى المذهب أعتقد انه مذهب واحد فلم اذن الجري وراء الاقاليم والدويلات الصغيرة التى من المستحيل اكتفاءئها بنفسها دون باقي البلد
Target	مع احترامي للتعليقات التي وإن اختلفت كثيرا لكنها تعبر عن وجهات نظر محترمة ، ولكن المقال الجميل للأستاذ عمر محمد ورسمه يجسد الواقع الحالي للصومال ، لأن الصومال بلد واحد ، اللغة واحدة ، والديانة واحدة ، بل حتى المذهب أعتقد أنه مذهب واحد ، فلم إذن الجري وراء الأقاليم والدويلات الصغيرة التي من المستحيل اكتفاءها بنفسها دون باقي البلد
Generated	مع احترامي للتعليقات التي وإن اختلفت كثيرا لكنها تعبر عن وجهات نظر محترمة ، ولكن المقال الجميل للأستاذ عمر محمد ورسمة يجسد الواقع الحالي للصومال ؛ لأن الصومال بلد واحد واللغة واحدة والديانة واحدة ، بل حتى المذهب أعتقد أنه مذهب واحد ، فلم إذن الجري وراء الإقليمية والدويلات الصغيرة التي من المستحيل اكتفاءئها بنفسها دون باقي البلد

to the model's focus on MSA and its importance in evaluating it on other Arabic language variants, such as CA, which poses a greater difficulty, and dialectal Arabic, which presents a research challenge due to variations in word meanings across dialects. This issue can be addressed by acquiring a dataset with labels to train the transformers or by developing a pre-trained transformer specifically designed for translating dialectal sentences to MSA.

Table 21 Examples for the proposed model correction from the non-native data

Type of sentence	Samples
Source	رحلتي إلى بلدي خلال إجازة الحج لما وصلت إلى السعودية في هذا الفصل كنت عازما عاى تأجيل الفصل ورجوعي إلى بلدي لأن زوجتي كان على وشك الولادة ، ولم يكن وعها أحد ، لكني فوجئت بخبر محزن وهو أنه لا يمكن تأجيل الدبلون ، إما أن أمسحه كله فأرجع في السنة القادمة للكلية ، وإما أن أسافر وأرجع سريعا وأيضا أخبرت بأن زوجتي مقبولة بجامعة الأميرة نورة بداية من هذا الفصل ، ففرحت وحزنت في نفس الوقت . سبب الفرح كان أنه سوف يتحقق ما كنت أتمنى منذ عشر سنوات ، وذلك أن أذهب يوما ما إلى السعودية لدراسة العلوم الإسلامية وأن أتزوج وأحضر زوجتي معي فندرس معا ، وسبب حزني كان معرفتي بأنه من الصعب جدا أن أرجع الآن إلى بلدي وأن زوجتي ستواجه صعوبات الولادة وآلامها بنفسها ولن أكون معا حتى أساعدها في ما أستطيع مساعدتها ، وقد كنت وعدتها قبل مجيئي أني سأرجع بسرعة وأن لا تحزن . كنت أشعر كأني ختنها ، لكن ماذا أفعل ؟ قدر الله وما شاء فعل بعد أن تدبرت جيدا جميع الاختيارات المتاحة بين يدي ، كتبت الخطاب أطلب فيه من مجلس المعهد أن يسمحوا لي بالسفر إلى بلدي لإحضار زوجتي ، وبينت حالي بالتفصيل . ثم وافقوا على أن يسمحوا لي ذلك من ١١ / ١٠ / ٢٠١٢ الموافق ل ٢٥ / ١١ / ١٤٣٣ إلى ٤ / ١١ / ٢٠١٢ الموافق ل١٩ / ١٢ / ١٤٣٣ ، وكذلك فعلت
Target	رحلتي إلى بلدي خلال إجازة الحج . لما وصلت إلى السعودية في هذا الفصل ، كنت عازما على تأجيل الفصل ورجوعي إلى بلدي ؛ لأن زوجتي كانت على وشك الولادة ، ولم يكن معها أحد ، لكني فوجئت بخبر محزن ، وهو أنه لا يمكن تأجيل الدبلوم ، إما أن أمسحه كله فأرجع في السنة القادمة للكلية ، وإما أن أسافر وأرجع سريعا وأيضا أخبرت بأن زوجتي مقبولة بجامعة الأميرة نورة بداية من هذا الفصل ، ففرحت وحزنت في نفس الوقت ؛ سبب الفرح كان أنه سوف يتحقق ما كنت أتمنى منذ عشر سنوات ، وذلك أن أذهب يوما ما إلى السعودية لدراسة العلوم الإسلامية ، وأن أتزوج وأحضر زوجتي معي فندرس معا ؛ وسبب حزني كان معرفتي بأنه من الصعب جدا أن أرجع الآن إلى بلدي ، وأن زوجتي ستواجه صعوبات الولادة وآلامها بنفسها ولن أكون معها حتى أساعدها في ما أستطيع مساعدتها به ، وقد كنت وعدتها قبل مجيئي أني سأرجع بسرعة وأن لا تحزن . كنت أشعر كأني ختنها ، لكن ماذا أفعل ؟ قدر الله وما شاء فعل بعد أن تدبرت جيدا جميع الاختيارات المتاحة بين يدي ، كتبت الخطاب أطلب فيه من مجلس المعهد أن يسمحوا لي بالسفر إلى بلدي لإحضار زوجتي ، وبينت حالي بالتفصيل . ثم وافقوا على أن يسمحوا لي بذلك من ١١ / ١٠ / ٢٠١٢ الموافق لـ ٢٥ / ١١ / ١٤٣٣ ، إلى ٤ / ١١ / ٢٠١٢ الموافق لـ ١٩ / ١٢ / ١٤٣٣ ، وكذلك فعلت .
Generated	رحلتي إلى بلدي خلال إجازة الحج . لما وصلت إلى السعودية في هذا الفصل كنت عازما على تأجيل الفصل ورجوعي إلى بلدي ؛ لأن زوجتي كانت على وشك الولادة ، ولم يكن معها أحد ، لكني فوجئت بخبر محزن وهو أنه لا يمكن تأجيل الحفل كله ، إما أن أمسحه كله فأرجع في السنة القادمة للكلية ، وإما أن أسافر وأرجع سريعا وأيضا أخبرت بأن زوجتي مقبولة بجامعة الأميرة نورة بداية من هذا الفصل ، ففرحت وحزنت في نفس الوقت . سبب الفرح كان أنه سوف تتحقق ما كنت أتمنى منذ عشر سنوات ، وذلك أن أذهب يوما ما إلى السعودية لدراسة العلوم الإسلامية ، وأن أتزوج وأحضر زوجتي معي فندرس معا . وسبب حزني كان معرفتي بأنه من الصعب جدا أن أرجع الآن إلى بلدي ، وأن زوجتي ستواجه صعوبات الولادة وآلامها بنفسها ، ولن أكون معا حتى أساعدها في ما أستطيع مساعدتها ، وقد كنت وعدتها قبل مجيئي أني سأرجع بسرعة وأن لا تحزن . كنت أشعر كأني ختنها ، لكن ماذا أفعل ؟ قدر الله وما شاء فعل . بعد أن تدبرت جيدا جميع الاختيارات المتاحة بين يدي ، كتبت الخطاب أطلب فيه من مجلس المعهد أن يسمحوا لي بالسفر إلى بلدي لإحضار زوجتي ، وبينت حالي بالتفصيل . ثم وافقوا على أن يسمحوا لي بذلك من ١١ /١٠ /٢٠١٢ الموافق ل ٢٥ /١١ /١٤٣٣ إلى ٤ / ١١ /٢٠١٢ الموافق لـ ١٩ / ١٢ / ١٤٣٣ ، وكذلك فعلت

7 Conclusion and future work

The Arabic GEC is an enormous task in NLP due to its diverse range of intricate errors and limited data availability. This research utilized transfer learning with seq2seq pre-trained transformers to address the issue of limited data quantity. It was observed that this approach yielded superior results compared to previous methods that relied on generating synthetic data from various corpora. However, it should be noted that the models in those prior methods were trained with more data than what was used in this research. The significance lies in the fact that the quality of the training data has greater importance than its quantity. This research shows that AraBART is currently the more practical option, and nucleus sampling is better than beam search for text decoding. For future research endeavors, this methodology has the potential to be applied to additional low-resource languages such as Arabic. Additionally, it might be used in high-resource languages like English to compare the results with the current research findings. Another avenue for future research involves developing a methodology capable of generating a greater quantity of high-quality data samples to improve the outcomes of our proposed study.

Funding Open access funding provided by The Science, Technology & Innovation Funding Authority (STDF) in cooperation with The Egyptian Knowledge Bank (EKB).

Data availability The corresponding author can provide the data supporting the study's conclusions upon request, as it is not publicly available owing to sensitivity. Data are kept in controlled access data storage at AASTMT.

Declarations

Conflict of interest The authors disclose no conflict of interest relevant to the content of this article. Furthermore, the authors do not have any relevant financial or non-financial interests to state.

References

1. Pouyanfar S, Sadiq S, Yan Y, Tian H, Tao Y, Reyes MP, Shyu M-L, Chen S-C, Iyengar SS (2018) A survey on deep learning: algorithms, techniques, and applications. ACM Comput Surv 51(5):1–36. https://doi.org/10.1145/3234150
2. Dargan S, Kumar M, Ayyagari MR, Kumar G (2020) A survey of deep learning and its applications: a new paradigm to machine learning. Arch Comput Methods Eng 27:1071–1092. https://doi.org/10.1007/s11831-019-09344-w
3. Dong S, Wang P, Abbas K (2021) A survey on deep learning and its applications. Comput Sci Rev 40:100379. https://doi.org/10.1016/j.cosrev.2021.100379
4. Li Y, Yang T (2018) Word embedding for understanding natural language: a survey. In: Guide to big data applications, pp 83–104, https://doi.org/10.1007/978-3-319-53817-4_4
5. Yang H, Luo L, Chueng LP, Ling D, Chin F (2019) Deep learning and its applications to natural language processing. In: Deep learning: fundamentals, theory and applications, pp 89–109, https://doi.org/10.1007/978-3-030-06073-2_4
6. Wang Y, Wang Y, Liu J, Liu Z (2005) A comprehensive survey of grammar error correction. arxiv 2020, arXiv preprint arXiv:2005.06600, https://doi.org/10.48550/arXiv.2005.06600
7. Yanghui Zhong and Xiaorui Yue (2022) On the correction of errors in english grammar by deep learning. J Intell Syst 31(1):260–270. https://doi.org/10.1515/jisys-2022-0013
8. Katsumata S, Komachi M (2020) Stronger baselines for grammatical error correction using pretrained encoder-decoder model arXiv preprint arXiv:2005.11849, https://doi.org/10.48550/arXiv.2005.11849
9. Liang H, Tang Y, Xinli W, Zeng J (2022) Considering optimization of english grammar error correction based on neural network. Neural Comput Appl 5:1–13
10. Zhu J, Shi X, Zhang S (2021) Machine learning-based grammar error detection method in english composition. Sci Program 2021:1–10. https://doi.org/10.1155/2021/4213791
11. Solyman A, Wang Z, Tao Q (2019) Proposed model for arabic grammar error correction based on convolutional neural network. In: 2019 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE), pp 1–6, IEEE, https://doi.org/10.1109/ICCCEEE46830.2019.9071310
12. Nora Madi and Hend Al-Khalifa (2020) Error detection for arabic text using neural sequence labeling. Appl Sci 10(15):5279. https://doi.org/10.3390/app10155279

13. Moukrim C, Abderrahim T, Tarik A et al (2021) An innovative approach to autocorrecting grammatical errors in Arabic texts. J King Saud Univ-Comput Inf Sci 33(4):476–488. https://doi.org/10.1016/j.jksuci.2019.02.005
14. Alkhatib M, Monem AA, Shaalan K (2020) Deep learning for Arabic error detection and correction. ACM Trans Asian Low-Resource Lang Inf Process 19(5):1–13. https://doi.org/10.1145/3373266
15. Vinyals OI, Le Sutskever QV (2014) Sequence to sequence learning with neural networks. Adv Neural Inf Process Syst 5:963
16. Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A et al (2020) Language models are few-shot learners. Adv Neural Inf Process Syst 33:1877–1901
17. Zhou M, Duan N, Liu S, Shum H-Y (2020) Progress in neural nlp: modeling, learning, and reasoning. Engineering 6(3):275–290. https://doi.org/10.1016/j.eng.2019.12.014
18. Mohamed A, Elaziz DA, Abualigah L, Yu L, Alshinwan M, Khasawneh AM, Lu S (2021) Advanced metaheuristic optimization techniques in applications of deep neural networks: a review. Neural Comput Appl 7:1–21. https://doi.org/10.1007/s00521-021-05960-5
19. Sutskever I, Vinyals O, Le QV (2014) Sequence to sequence learning with neural networks. Adv Neural Inf Process Syst 27:632
20. Bahdanau D, Cho K, Bengio Y (2014) Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473, https://doi.org/10.48550/arXiv.1409.0473
21. Hu D (2019) An introductory survey on attention mechanisms in NLP problems. Adv Intell Syst Comput 8:432–448
22. Galassi A, Lippi M, Torroni P (2020) Attention in natural language processing. IEEE Trans Neural Netw Learn Syst 32(10):4291–4308. https://doi.org/10.1109/TNNLS.2020.3019893
23. Kenneweg P, Stallmann D, Hammer B (2023) Novel transfer learning schemes based on siamese networks and synthetic data. Neural Comput Appl 35(11):8423–8436. https://doi.org/10.1007/s00521-022-08115-2
24. Francisca A, Acheampong HN-M, Chen W (2021) Transformer models for text-based emotion detection: a review of bert-based approaches. Artifi Intell Rev 7:1–41
25. Kumar S, Solanki A (2023) An abstractive text summarization technique using transformer model with self-attention mechanism. Neural Comput Appl 8:1–20. https://doi.org/10.1007/s00521-023-08687-7
26. Asa CS, Li X, Ghazvininejad M (2020) Recipes for adapting pre-trained monolingual and multilingual models to machine translation. arXiv preprint arXiv:2004.14911, https://doi.org/10.48550/arXiv.2004.14911
27. Mosa DT, Nasef NA, Lotfy MA et al (2023) A real-time Arabic avatar for deaf–mute community using attention mechanism. Neural Comput Applic 35:21709–21723. https://doi.org/10.1007/s00521-023-08858-6
28. Eddine MK, Tomeh N, Habash N, Roux J Le, Vazirgiannis M (2022) Arabart: a pretrained arabic sequence-to-sequence model for abstractive summarization. arXiv preprint arXiv:2203.10945, https://doi.org/10.48550/arXiv.2203.10945
29. Xue L, Constant N, Roberts A, Kale M, Al-Rfou R, Siddhant A, Barua A, Raffel C (2020) mt5: a massively multilingual pre-trained text-to-text transformer. arXiv preprint arXiv:2010.11934, https://doi.org/10.48550/arXiv.2010.11934
30. Liu Y, Jiatao G, Goyal N, Li X, Edunov S, Ghazvininejad M, Lewis M, Zettlemoyer L (2020) Multilingual denoising pre-training for neural machine translation. Trans Assoc Comput Ling 8:726–742. https://doi.org/10.1162/tacl_a_00343
31. Alhafni B, Inoue G, Khairallah C, Habash N (2023) Advancements in arabic grammatical error detection and correction: an empirical investigation arXiv preprint arXiv:2305.14734, https://doi.org/10.48550/arXiv.2305.14734
32. Solyman A, Zappatore M, Zhenyu W, Mahmoud Z, Alfatemi A, Ibrahim AO, Gabralla LA (2023) Optimizing the impact of data augmentation for low-resource grammatical error correction. J King Saud Univ-Comput Inf Sci 35(6):101572. https://doi.org/10.1016/j.jksuci.2023.101572
33. Solyman A, Zhenyu W, Qian T, Elhag A, Toseef M, Aleibeid Z (2021) Synthetic data with neural machine translation for automatic correction in Arabic grammar. Egy Inf J 22(3):303–315. https://doi.org/10.1016/j.eij.2020.12.001
34. Abandah GA, Suyyagh A, Khedher MZ (2021) Correcting arabic soft spelling mistakes using bilstm-based machine learning. arXiv preprint arXiv:2108.01141, https://doi.org/10.48550/arXiv.2108.01141
35. Alayba AM, Palade V, England M, Iqbal R (2018) A combined cnn and lstm model for arabic sentiment analysis. In: Machine Learning and Knowledge Extraction: Second IFIP TC 5, TC 8/WG 8.4, 8.9, TC 12/WG 12.9 International Cross-Domain Conference, CD-MAKE 2018, Hamburg, Germany, August 27–30, 2018, Proceedings 2, pp 179–191, Springer, https://doi.org/10.1007/978-3-319-99740-7_12
36. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. Adv Neural Inf Process Syst 30:963
37. Ludan R, Qin J (2022) Survey: transformer based video-language pre-training. AI Open 3:1–13. https://doi.org/10.1016/j.aiopen.2022.01.001
38. Humayun MA, Yassin H, Shuja J, Alourani A, Abas PE (2023) A transformer fine-tuning strategy for text dialect identification. Neural Comput Appl 35(8):6115–6124. https://doi.org/10.1007/s00521-022-07944-5
39. Dharaniya R, Indumathi J, Uma GV (2022) Automatic scene generation using sentiment analysis and bidirectional recurrent neural network with multi-head attention. Neural Comput Appl 34(19):16945–16958. https://doi.org/10.1007/s00521-022-07346-7
40. Shorten C, Khoshgoftaar TM, Furht B (2021) Text data augmentation for deep learning. J Big Data 8:1–34. https://doi.org/10.1186/s40537-021-00492-0

41. Pennington J, Socher R, Manning CD (2014) Glove: global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 1532–1543
42. Casola S, Lauriola I, Lavelli A (2022) Pre-trained transformers: an empirical comparison. Mach Learn Appl 9:100334. https://doi.org/10.1016/j.mlwa.2022.100334
43. Devlin J, Chang M-W, Lee K, Toutanova K (2018) Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, https://doi.org/10.48550/arXiv.1810.04805
44. Lewis M, Liu Y, Goyal N, Ghazvininejad M, Mohamed A, Levy O, Stoyanov V, Zettlemoyer L (2019) Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461, https://doi.org/10.48550/arXiv.1910.13461
45. Luciano Floridi and Massimo Chiriatti (2020) Gpt-3: Its nature, scope, limits, and consequences. Mind Mach 30:681–694. https://doi.org/10.1007/s11023-020-09548-1
46. Ni J, Ábrego GH, Constant N, Ma J, Hall KB, Cer D, Yang Y (2021) Sentence-t5: scalable sentence encoders from pre-trained text-to-text models. arXiv preprint arXiv:2108.08877, https://doi.org/10.48550/arXiv.2108.08877
47. Sabharwal N, Agrawal A, Sabharwal N, Agrawal A (2021) Future of bert models. In: Hands-on question answering systems with BERT: applications in neural networks and natural language processing, pp 173–178, https://doi.org/10.1007/978-1-4842-6664-9_7
48. Grover K, Kaur K, Tiwari K, Rupali, Kumar P (2021) Deep learning based question generation using t5 transformer. In: Advanced Computing: 10th International Conference, IACC 2020, Panaji, Goa, India, December 5–6, 2020, Revised Selected Papers, Part I 10, pp 243–255, Springer, https://doi.org/10.1007/978-981-16-0401-0_18
49. Bird JJ, Ekárt A, Faria DR (2023) Chatbot interaction with artificial intelligence: human data augmentation with t5 and language transformer ensemble for text classification. J Ambient Intell Humaniz Comput 14(4):3129–3144. https://doi.org/10.1007/s12652-021-03439-8
50. Antoun W, Baly F, Hajj H (2020), Arabert: transformer-based model for arabic language understanding. arXiv preprint arXiv:2003.00104, https://doi.org/10.48550/arXiv.2003.00104
51. Solyman A, Wang Z, Tao Q, Elhag A, Zhang R, Mahmoud Z (2022) Automatic arabic grammatical error correction based on expectation-maximization routing and target-bidirectional agreement. Knowl-Based Syst 241:108180. https://doi.org/10.1016/j.knosys.2022.108180
52. Krzysztof Pająk and Dominik Pająk (2022) Multilingual fine-tuning for grammatical error correction. Expert Syst Appl 200:116948. https://doi.org/10.1016/j.eswa.2022.116948
53. Sang YK, Bhatia G, Nagoudi El MB, Abdul-Mageed M (2023) Beyond english: evaluating llms for arabic grammatical error correction. arXiv preprint arXiv:2312.08400, https://doi.org/10.48550/arXiv.2312.08400
54. Holtzman A, Buys J, Li D, Forbes M, Choi Y (2019) The curious case of neural text degeneration. arXiv preprint arXiv:1904.09751, https://doi.org/10.48550/arXiv.1904.09751
55. Welleck S, Kulikov I, Roller S, Dinan E, Cho K, Weston J (2019) Neural text generation with unlikelihood training arXiv preprint arXiv:1908.04319, https://doi.org/10.48550/arXiv.1908.04319
56. Wenhao Yu, Zhu C, Li Z, Zhiting H, Wang Q, Ji H, Jiang M (2022) A survey of knowledge-enhanced text generation. ACM Comput Surv 54(11s):1–38. https://doi.org/10.1145/3512467

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Karim Ismail[1] · Sherif Abdou[2] · Mohamed Farouk[3] · Ahmed Salem[1]

✉ Ahmed Salem
a.salem@aast.edu

[1] College of Computing and Information Technology, Arab Academy for Science Technology and Maritime Transport (AASTMT), Cairo, Egypt

[2] Faculty of Computers and Artificial Intelligence, Cairo University, Giza, Egypt

[3] College of Computing and Information Technology, Arab Academy for Science Technology and Maritime Transport (AASTMT), Alexandria, Egypt