Arabic Grammatical Error Detection Using Transformers-based Pretrained Language Models

Sarah AlOyaynaa [1*], *Yasser* Kotb[1,2]

[1]Information Systems Department, College of Computer and Information Sciences, Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh, Saudi Arabia.
[2] Computer Science Division, Department of Mathematics, Faculty of Science, Ain Shams University, Cairo, Egypt.

Abstract. This paper presents a new study to use pre-trained language models based on the transformers for Arabic grammatical error detection (GED). We proposed fine-tuned language models based on pre-trained language models called AraBERT and M-BERT to perform Arabic GED on two approaches, which are the token level and sentence level. Fine-tuning was done with different publicly available Arabic datasets. The proposed models outperform similar studies with F1 value of 0.87, recall of 0.90, precision of 0.83 at the token level, and F1 of 0.98, recall of 0.99, and precision of 0.97 at the sentence level. Whereas the other studies in the same field (i.e., GED) results less than the current study (e.g., F0.5 of 69.21). Moreover, the current study shows that the fine-tuned language models that were built on the monolingual pre-trained language models result in better performance than the multilingual pre-trained language models in Arabic.

1 Introduction

Language is often considered the hallmark of human intelligence. Thus, developing systems that can understand human language are one of the main obstacles in the quest for artificial general intelligence.

Machine learning (ML) and Deep Learning (DL) which are subdomains of Artificial Intelligence (AI) have succeeded extensively and outperformed other technologies in many AI aspects (Q. Zhang et al., 2018) such as speech recognition, image analysis, and natural language processing (NLP). NLP is "*a theory-motivated range of computational techniques for the automatic analysis and representation of human language*" (Young et al., 2017). It is a field of ML dealing with linguistics that builds and develops Language Models (Singh & Mahmood, 2021). There are a lot of NLP tasks such as text classification, name entity recognition, question and answering, and machine translation (Montejo-Ráez & Jiménez-Zafra, 2022).

Previously, there was a misunderstanding that machine learning tasks can only achieve good results if there is a huge amount of data that have similar distribution and feature space.

* Corresponding author: saloyiana@sm.imamu.edu.sa

Hence, that making implementing machine learning tasks and their collection of labeled training data are expensive and difficult. Transfer learning came to tackle these challenges by reducing the need for gathering labeled training data for new tasks as well as reemploying the knowledge gained from a different task (Sarhan & Spruit, 2020).

The transformers model is one of the most famous ML architectures that was introduced by Google (Vaswani, 2017) which is a sequence-to-sequence (S2S) architecture and mainly proposed to perform neural machine translation (NMT) (Karita et al., 2019). Its architecture consists of the encoder and decoder layers, and one is coupled to the other through layers of the feed-forward network and multi-head attention. The cosine and sine functions, which produce positional encoding, assist the model, and recall the order and position of words. Self-attention is used as a main method by the encoder and decoder layer's multi-head attention layer (Chouikhi & Alsuhaibani, 2022).

The recent launches of improved Transformer (Vaswani, 2017) based models like BERT (Devlin et al., 2019) turned out to be a climacteric year for the NLP world. These models were trained on large datasets to create pre-trained language models. Then, transfer learning was used to fine-tune these models on a small, labeled dataset for task-specific features resulting in significant performance enhancement on several NLP tasks (Elmadani et al., 2020)(Sun et al., 2020).

Grammatical Error Detection and Correction (GED&C) systems are commonly used to identify grammatical writing errors and then correct them automatically. Grammatical Error Detection (GED) is one of the key components of the grammatical error correction (GEC) community (Q. Wang & Tan, 2020). Even earlier, it was the first step of GEC, especially, the GEC approaches that are based on hand-crafted rules. Those rules are different for various grammatical error types, thus, detecting errors in the given text is the basis of the correction system (Q. Wang & Tan, 2020). Thereafter, such systems are handy for both natives and learners. In terms of language learners, an effective and automatic GED system is useful for them to find whether the texts written themselves have errors or not.

Moreover, the implementations of the GED&C systems are not new in the field and there are many efforts to master these tasks. Furthermore, some of them gained great success such as implementing such a system using statistical machine translation techniques (Brockett et al., 2006). However, in terms of the Arabic language, there are a few and still not those efficient efforts, and implementing such systems requires many resources which could be challenging.

Arabic is one of the world's five major languages with over 290 million native speakers and a total of 422 million world speakers (UNESCO, 2019) (Ethnologue, 2020). It is a Semitic, highly structured, and derivational language where morphology and syntax play a very important role (Shaalan, 2005). However, as it is a widely spoken language but in terms of NLP applications and studies especially GED&C is still new in the research field compared with other prominent languages (e.g., English). Hence, that could lead to hardening the contribution process.

The current study tackles the low-resource problem by taking the pre-trained language models that were built based on transformers architecture (Vaswani, 2017) to perform the GED task based on two approaches which are: the token level classification and sentence level classification, both in Arabic Language.

Table 1 represents the covered errors of the token level classification task in the current study (Shaalan, 2005) (Zaghouani et al., 2014):

Table 1. Covered errors in token level task.

Type of Error	Definition	Example
Spelling	Deleting at least one character of a word or inserting an extra character of a word.	سموه **ودعى**
Morphology	Incorrect derivation or inflection.	التنمية **الطموحة** ذهب علي إلى **حديقات** جميلة.
Syntactic	Wrong agreement in gender, number, or case, wrong case assignment, wrong order of words, wrong tense use, missing or redundant words.	أنا أدرس في الجامعة **الجديد**. أن لا **تقل** عدد المواقف

Whereas **Table 2** shows the covered errors of the sentence level classification as follows:

Table 2. Covered errors in sentence level.

Type of Error		Definition	Example
Spelling	Missing words	Deleting at least one word	لعب الولد القدم
	Duplicate words	Duplicate at least one word without any meaning	لا **تفرح** تفرح بكونك مسيطرا
	Missing character	Deleting at least one character of a word	هناك **قضيا** مشتركة بينهما
	Duplicate character	Duplicate at least one character without any meaning	هناك قضايا مشتركة **بينههههما**

Moreover, a comparison between the pre-trained language model that was originally trained on many different languages such as English, Chinese, etc., (i.e., multilingual language model) and the pre-trained language model which was originally trained on one language only (i.e., monolingual language model) (Pires et al., 2020) will be provided.

The paper structure is as follows: in the next section, we will provide a review of the literature. After that, the following methodology will be presented. The third section will cover the main results of the study as well as the discussion part. Finally, we will conclude and provide the future direction and some study limitations.

2 Literature Review

With advancements in industry and information technology, large volumes of electronic documents such as newspapers, emails, weblogs, and these are produced daily. Producing electronic documents has considerable benefits. Therefore, the existence of automatic systems such as spell and grammar-checker/correctors can help to improve their quality (Ehsan, 2014).

The GED&C task is not a new application in the field of NLP. In the literature, NLP researchers put the effort to achieve the GED&C task with high performance at different times, via different approaches and architectures, and diverse languages.

Parnow et al. (2020) introduced a transformer-based model with formulating the GEC task as a semi-supervised learning problem. The model was trained using three labeled GEC corpora and a couple as plain text corpora. It tested on the CoNLL-2014 test benchmark and achieved F0.5 of 60.2 without a fully pre-trained encoder. The proposed method showed a competitive result with the present leading models.

A study done by Grammarly (Alikaniotis et al., 2019), highlighted the effectiveness of the transformer architecture approach in the GEC task. It used three different pre-trained language models: BERT, GPT, and GPT-2 with no annotated data to be trained. To evaluate the proposed model, it employed two standard publicly available datasets: CoNLL-2014 and FCE. The result showed that this approach leverages the state-of-the-art against the previous state-of-the-art trained on large corpora with transformer architecture. However, the study discouraged following the same employed methodology and invited new improvements.

Qiu & Qu (2019) adopted a two stage-model to incrementally correct Chinese grammatical errors. In the first stage, it checked for spelling mistakes and correct them based on the language model. It used a language model to select the best probable word from candidates. The second stage focused on correcting the grammatical errors and remaining spelling mistakes based on the sequence-to-sequence transformer model. It treated the correction task as a translation task. The model training was done on NLPCC 2018 shared task 2. The proposed system outperformed the compared state-of-the-art systems with F0.5 of 32.01.

Language grammars are complex in nature, and when we talk about Arabic it became more complex. Even Arabic-speaking people nowadays are not fully familiar with the grammar of their language. Thus, Arabic grammatical checking is considered a difficult task. The difficulty comes from several sources (K. Shaalan et al., 1993) such as the length of the sentence and the complex Arabic syntax; the omission of diacritics (vowels) in written Arabic ‘at-ta.sk¯il’; the free word order nature of Arabic sentence; and the presence of an elliptic personal pronoun ‘ad.-dam¯iral-mustatir’(K. F. Shaalan, 2005).

Arabic language in nature has many forms. The modern form of Arabic and most used by humans is called Modern Standard Arabic (MSA). MSA is a simplified form of Classical Arabic and follows the same grammar of it (K. Shaalan, 2010). The main differences between Classical and MSA are that MSA has a larger and more modern vocabulary and does not use some of the more complicated forms of grammar found in Classical Arabic.

The authors in (Solyman et al., 2019) proposed a hybrid model for Arabic GEC based on deep encoder-decoder architecture. It used Multi convolutional layers (i.e., nine convolutional encoder and nine decoder layers), with an attention mechanism. Also, it was tested using the testing set of a large Arabic corpus. The model gains precision of 70.23, recall of 72.10, and F1 of 71.14.

Therefore, the findings showed high statical results out of the proposed model.(Madi & Al-Khalifa, 2018) proposed a model for the Arabic GED tool based on DL. It employed a simple recurrent neural network (RNN) architecture. Nonetheless, this study used a handcrafted and very small corpus. The results showed that the tool was good to identify the error to the user, but, with no suggested corrections. As an improvement of the previous work, (Madi & Al-Khalifa, 2020) developed a set of systems based on RNN, Long short-term memory (LSTM), and A Bidirectional LSTM (BiLSTM) architectures, all the work done on a large Arabic corpus, unlike the previous work. The result revealed that the BiLSTM performed the best of all systems in terms of F0.5 of 81.55% in the training set and LSTM was the worst performance. However, on the test set, the results show the opposite overall measures.

Al-qaraghuli (2021) developed a vanilla transformer to correct soft spelling mistakes in Arabic. The model handled four types of soft spelling mistakes, namely, confusion among shapes, both shapes at the end of the word, hamza alef (همزة الألف) insertion and omission of

after aljamaea (الجماعة), and alef waw confusion among teh,marbuta (تاء مربوطة), and at the end of the teh heh (تاء) word. Moreover, to train the model, they used Tashkeela set to train one model and the Wiki-40B to train another model. Moreover, they developed a dataset containing artificial errors generated following a stochastic error injection approach that was proposed (Khedher, 2021) to use in the model testing phase. In terms of the Tashkeela model performance, the best result achieved is accuracy of 99.62% whereas, the wiki model revealed the best accuracy of 99.77%.

SCUT AGEC is a model that is introduced by (Solyman et al., 2021), it is a sequence-to-sequence model that was built on Convolutional Neural Network (CNN) architecture which consists of nine encoder-decoder layers with an attention mechanism. It was pre-trained and finetuned on a dataset that was developed during the study called SCUT which is synthetic data consisting of 18,061,610 words divided into training and development sets. The model achieved state-of-the-art results against the neural AGEC models with the strong competitiveness of the hybrid AGEC systems on the Qatar Arabic Language Bank (QALB) test set with F1 of 70.64%. **Table 3** provides a summary of the reviewed studies:

Table 3. Summary of the literature review.

Study	Year	Language	Scope	Result
(Solyman et al., 2021)	2021	Arabic	Correction	Precision of 70.23 Recall of 72.10, F1 of 71.14
(Al-qaraghuli, 2021)	2021	Arabic	Correction	Best accuracy of: 99.77
(Parnow et al., 2020)	2020	English	Correction	F05 of 60.2
(Madi & Al-Khalifa, 2020)	2020	Arabic	Detection	F0.5 of 69.21 Recall of 41.61 Precision of 82.97
(Qiu & Qu, 2019)	2019	Chinese	Detection and Correction	F05 of 32.01

After all, the efforts in applying GED&C models to Arabic are low compared with other languages (i.e., English, Indian, and Chines). Therefore, this study aims to provide a new contribution to the Arabic language field. Also, help native and non-native Arabic speakers improve their writing skills and become more confident regarding their documentation.

More recently, the method of pre-training language models on an amount of unlabeled data and fine-tuning in downstream tasks has made a breakthrough in several natural language understanding tasks (Antoun et al., 2020)(Devlin et al., 2019). To the best of our knowledge, this method has not been utilized to serve Arabic GED&C yet despite its noticeable results in the literature. Therefore, the current study will use the popular pre-trained language models that were built based on the transformers (Vaswani, 2017) architecture to help improve the Arabic GED task. Moreover, there will be an evaluation of the existing pre-trained language models in terms of performing the GED task to highlight the possibility of using such a method in that regard.

3 Methodology

The proposed work followed two different tasks which are the token-level GED task and the sentence-level GED task. The upcoming section explains both in detail. **Fig 1** provides a high-level overview of the study's methodology:

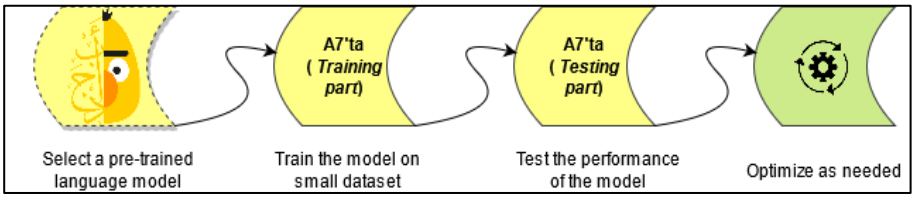

Fig. 1. Methodology Overview.

3.1 Token-level grammatical error detection

In this task, the grammatical correctness of the word level (i.e., token) will be checked.

3.1.1 Datasets

First, we used A7'ta (أخطاء) dataset which is a dataset that was published to rich the GED domain. A7'ta (Madi & Al-Khalifa, 2019) is a collection of modern Arabic sentences. It contained 470 erroneous sentences and their 470 error-free counterparts, and they are categorized into eight main categories: syntax errors, morphological errors, semantic errors, linguistic errors, stylistic errors, spelling errors, punctuation errors, and the use of informal as well as borrowed words. This dataset was developed mainly to serve the grammar-checking model training. However, there is a data imbalance issue declared by a dataset consumer (Madi & Al-Khalifa, 2020).

Due to the lack of prepared datasets to be consumed in the targeted field, and despite the imbalance issue, A7'ta is a highly prepared dataset for the current task. Therefore, we tried to tackle the dataset issue and improve it by taking a portion of other publicly available datasets in the field (i.e., QALB (Mohit et al., 2015)) and SCUT (Solyman et al., 2021)) and reformat them to match A7'ta format. Then, merging all of them into one dataset. Finally, the final dataset was split into two datasets which are the training dataset to be used in the fine-tuning step and the test dataset for testing the model performance. The splitting was applied randomly with a percentage of 70%-20% for train-test datasets with a random state of 200.

3.1.2 Data Preprocessing

To be able to start our experiments, first, we started with the tokenization process which is a step where the sentences decompose into tokens (i.e., splitting the sentences into words, each word called a "token". As we aim to use BERT based model, we had to add the special [CLS] symbol at the start of every text and the token [SEP] at the end of each sentence as that is the supported format by the model. We utilized a tokenizer class in the hugging face library named "*BertTokenizerFast*" with its default values except the keys in **Table 4**:

Table 4. Tokenizer settings in token level.

Key	Value
Return offsets mapping	true
Padding	128
Truncation	true
Max length	128

After we had individual tokens in the required format, we needed to convert them to numbers (i.e., IDs), and each number represents a token of the data. Lastly, we converted those numbers into tensors to be understood by the model. For that, we used a class in PyTorch called: "as_tensor" with its default values.

3.2 Sentence-level grammatical error detection

In this task, the grammatical correctness of the sentence level will be checked.

3.2.1 Datasets:

First, we used SCUT which is a publicly available dataset that was published to rich the GEC domain, so it has two columns with the names: source (i.e., incorrect erroneous sentences) and targets (i.e., correct sentences). It is a dataset developed by (Solyman et al., 2021) to overcome the lack of resources in the Arabic language in terms of the GEC task. It was developed in an unsupervised manner to generate a large-scale parallel synthetic corpus based on the confusion function. The source of the data was the Al-Watan corpus which is an open-access consisting of 10,000,000 (ten million) words written in Modern Standard Arabic (MSA) by professional journalists. A portion of this data was taken (i.e., 18,061,610 million words) and the confusion function has been used to randomly generated errors. The dataset focuses mainly on the errors related to spelling errors and is categorized as follows: missing words, duplicate words, a missing character, and duplicate character.

Since there is not that much data related to the targeted task, we reformat the dataset to be suitable for such a task. The reformatting was in two steps, the first one is adding a new column named "label", then, classifying the target as correct (i.e., 1), and the source as incorrect (i.e., 0). And that was done on a portion of the main dataset. Then, it was saved in a new file and used while the finetuning process. Finally, the final dataset was split into two datasets which are the training dataset to be used in the fine-tuning step and the test dataset for testing the model performance. The splitting was applied randomly with a percentage of 90%-10% for train-validation datasets. Also, to generate the test dataset, following the same reformat approach done on another portion of the same main dataset.

3.2.2 Data Preprocessing

The data preprocessing was the same as the token-level experiment as we used the same pre-trained model starting with the tokenization until converting the token into tensors. However, the used tokenizer class here was " *BertTokenizer*". Again, we used the default values except the keys in **Table 5**:

Table 5. Tokenizer settings in sentence level.

Key	Value
return_attention_mask	true
pad_to_max_length	true
add_special_tokens	true
Max length	64

4 Experiments

The current study's experiments mainly scoped on the BERT-based pre-trained models and used the hugging face library to pull the models from it. Since we target the Arabic language, we had to filter the models to only the Arabic-supported ones which are not that many. We conducted multiple experiments to evaluate the performance of a different set of available models. After the observation, we concluded with two main models to be focused on during this study which are the M-BERT (Pires et al., 2020): multilingual language model, and AraBERT (Antoun et al., 2020): monolingual language model with the same settings.

Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019) had been introduced by Google to tackle the unidirectional problem of the standard language model by using a "masked language model" (MLM) objective which randomly masks tokens from the input to predict the original vocabulary id of the masked word based on its context only. Moreover, it aims to master the next sentence prediction task. It was built based on the transformer encoder architecture provided by (Vaswani, 2017). It showed its effectiveness in different tasks of NLP such as question answering (Devlin et al., 2019), text classification (Garrido-Merchán & González-Carvajal, 2020), sentiment classification (Gao et al., 2019), and text summarization (Elmadani et al., 2020).

M-BERT (Pires et al., 2020) is like the original English BERT model but instead of being trained only on monolingual English data (Pires et al., 2020), it is trained on the Wikipedia pages of 104 languages with a shared word piece vocabulary thus multilingual language model.

Whereas AraBERT (Antoun et al., 2020) is an extension of BERT but it is pre-trained in the Arabic language only. It follows BERT architecture, and it aims to improve the state-of-the-art Arabic NLP tasks. It was pre-trained on more than a billion Arabic words and had been applied to perform many NLP tasks in Arabic such as Sentiment Analysis, and Question Answering, and showed new results that were not achieved before (Antoun et al., 2020).

Both models were fine-tuned with an initial learning rate of 1e-5 with a batch size of 8 and went over 4 training epochs. Whereas all the rest hyperparameters were kept at their default values. The fine-tuning process was done on the Google Colab with a GPU runtime using Python programming language and PyTorch framework.

5 Results

The main finding of this study illustrates in **Table 6**:

Table 6. Study Results.

Approach	Model	Accuracy	F1	F0.5	Precision	Recall
Token Level	(A)	0.86	0.87	0.85	0.83	0.90
	(B)	0.85	0.85	0.83	0.81	0.91
Sentence Level	(A)	0.98	0.98	0.97	0.97	0.99
	(B)	0.96	0.96	0.95	0.94	0.98

In the GED on the token level classification, the model that was built on the monolingual language model (i.e., model A), achieved a higher result than the multilingual language model (i.e., model B). As you see in Table 6, model A has F1 of 87% and accuracy of 86%. Whereas model B results in F1 of 85% and accuracy of 85% as well. It can be concluded that the monolingual models can perform better in such tasks.

Moreover, the conducted experiments on GED were also done on the sentences level classification with the same approach performed on the token-level model that was built on the monolingual language model (i.e., model A), achieved a higher result than the multilingual language model (i.e., model B). As you see in the Table 6, model A has F1 of 98% and accuracy of 98% whereas model B result in F1 of 96% and accuracy of 96% as well.

Despite the relatively low-resource fine-tuning, the results suggest that the pre-trained models might tremendously be beneficial, especially in GED tasks. Thus, the results can summarize that the popular approach (i.e., fine-tuning pre-trained language) is applicable in the Arabic language generally, and grammar checking in particular. The charts in Fig 2 and Fig 3 represent the accuracy results of both models:

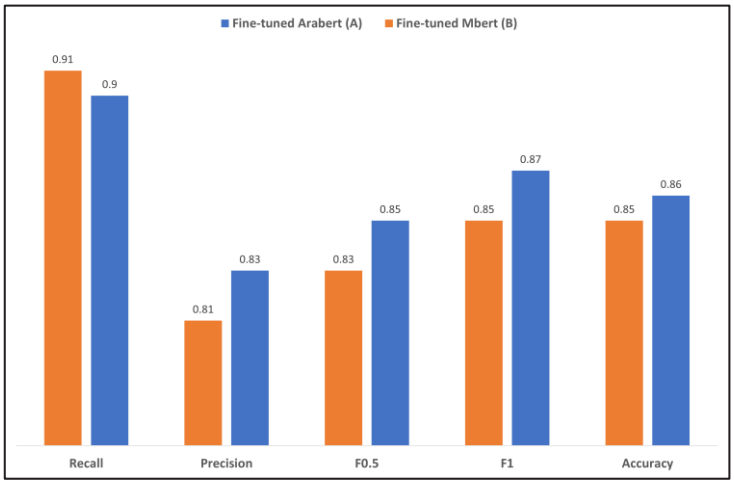

Fig. 2. Token Level Results.

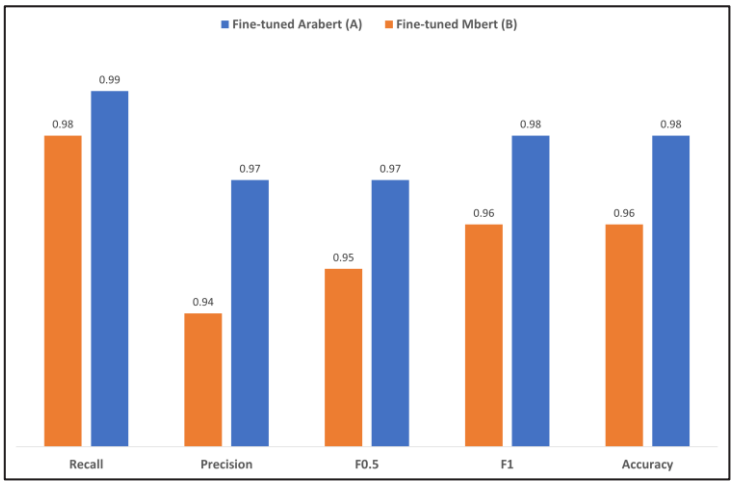

Fig. 3. Sentence Level Results.

The following examples show the performance of our fine-tuned models whether it is multilingual or monolingual and both on token-level and sentence level comparing them with the original pre-trained model without any fine-tuning. Note that the golden column is the real value in the datasets:

5.1 Model (A) Example

Table 7 shows examples of the Token-Level and Sentence level outputs of the model (A).

Table 7. Model (A) Output.

Full Sentence		هناك قضيا مشتركة بينههههما		
Input		**Golden (correct)**	**Output**	
			Arabert Without Finetuning	**Fine-tuned Arabert**
Token-level	هناك	Correct	Incorrect	Correct
	قضيا	Incorrect	Correct	Incorrect
	مشتركة	Correct	Incorrect	Correct
	بينههههما	Incorrect	Correct	Incorrect
Sentence-level	هناك قضيا مشتركة بينههههما	Incorrect	Incorrect	Incorrect

5.2 Model (B) Example

Table 8 shows examples of the Token-Level and Sentence level outputs of the model (B).

Table 8. Model (B) Output.

Full Sentence		هناك قضيا مشتركة بينههههما		
Input		**Golden (correct)**	**Output**	
			Mbert Without Finetuning	**Fine-tuned Mbert**
Token-level	هناك	Correct	Incorrect	Correct
	قضيا	Incorrect	Incorrect	Incorrect
	مشتركة	Correct	Correct	Correct
	بينههههما	Incorrect	Incorrect	Incorrect
Sentence-level	هناك قضيا مشتركة بينههههما	Incorrect	Incorrect	Incorrect

6 Discussion

As **Table 6** shows, the fine-tuned models produced very well results in terms of GED despite the fact that the fine-tuning was done on relatively small data and super few training rounds. Generally, this proves the effectiveness of the pre-trained models in performing NLP tasks which GED is one of them and agrees with the literature.

The models that were built on the monolingual pre-trained models resulted in better performance than the ones trained in multiple languages. Thus, it can be concluded that the more pre-trained models are dedicated to one language training, the better the results in that language generally, and in Arabic in particular. However, the multilingual pre-trained models

are still effective, especially in low-resource languages, and do not have a monolingual pre-trained language model yet.

Again, as you see in the above section, the experiments were conducted on two different methods to detect grammar errors in Arabic which are on the token level and the sentence level. The results on classifying the sentences as correct or incorrect were higher in terms of statics, it may be due to the fact that the model fine-tuned on a relatively large amount of data in comparison with the amount of the token level model. And the work to optimize these differences is left to the future.

Even though (Al-qaraghuli, 2021) focused on Arabic grammatical correction, not detection only, the current study's result agrees with it. It agrees with the promising performance of the models that are built on the transformers in terms of NLP tasks, especially in Arabic. The (Al-qaraghuli, 2021) results in Arabic grammatical correction with accuracy of 99.77% in correcting the artificial grammar errors. However, the following approach in the study was super time-consuming and required a lot of effort as well as costs since they started to build a transformer from scratch (i.e., vanilla transformers) then train it, and fine-tune it. Compared with our result which is relatively close to the (Al-qaraghuli, 2021), we can say that the possibility of utilizing the popular pre-trained models and fine-tuning it directly may result in the same result with much less time, effort, and even resources.

Madi & Al-Khalifa, 2020, conducted an error detection task in Arabic following a neural network approach but with different architecture than the one used in the current study. It used SimpleRNN, LSTM, and BiLSTM-based models and trained them on the same dataset used in the current study. The best result was the BiLSTM with precision of 82.97%, recall of 41.61%, and F0.5 of 69.21 %. In comparison with our result, our approach outperforms all the (Madi & Al-Khalifa, 2020) results with F0.5 of 0.85 in the token level and F0.5 of 0.97 in the sentence level. This might be an indication that the GED is more suitable to be executed on pre-trained models that were built on the transformer's architecture-based models. It might be due to that fact the self-attention mechanism.

Furthermore, agree with the current approach, but in other language settings like English, German, Czech, and Russian, (Katsumata & Komachi, 2020). Katsumata & Komachi, 2020 conducted a study following a similar approach to the current study. However, it employed another pre-trained model both called BART and the mBART (i.e., the multilingual version of Bart), and the focus of the study was on the GEC at the end. The performance was remarkable with a minimum training cycle (i.e., fine-tuning). The best results achieved after the fine-tuning on a publicly available dataset was F0.5 of 65.6% in terms of the English language.

The noticeable difference between the token level, and the sentence level statics results, is due to the remarkable difference in the used data mount during the finetuning process and that agrees with (Katsumata & Komachi, 2020) in terms of the Russian language. As the token level dataset is not available in terms of the Arabic language. Also, agreeing with (Madi & Al-Khalifa, 2020), the imbalance issue that was described in the methodology section might lead to this as well, even after trying to tackle it. These can be considered tiny limitations and call for improvements in future works as the token level detection can be more accurate in terms of grammar-checking tasks. Moreover, the imbalance issue led to an inconsistency in the accuracy, F0.5, and other metrics results, with that, however, it had never gone below 80% across all metrics which is still high and considerable results.

The results prove that the employment of transfer learning and the pre-trained models that were built based on the transformers can improve the performance in terms of GED on many levels such as the needed resources to fine-tune the models whether it is the data, or the hardware required to be used while conducting the experiments (Pajak, 2021). Thus, this approach can benefit the applications in low-resources languages and is a plus to the widely

used languages. Moreover, it can be considered a fast approach which leads to saving more time and increasing overall productivity.

With all that said, a good and balanced dataset, despite it is size, is required. As we mentioned earlier, the difference in the result between the sentence-level and token-level experiments is slightly high, even though the main difference in the application itself was in the dataset. The one used in the sentence level was balanced (i.e., (i) class items equal (c) class items) in both the training set and test set. On the other hand, at the token level, even though we tried to make it balanced as much as possible when splitting the dataset for training and testing, the imbalance issue appeared again since the sentence tokens class (i.e., i and, c) differ from one to another. And that can be a call to future work.

7 Conclusions

The conducted study investigated and concluded that fine-tuning the pre-trained language models (i.e., mBERT & AraBERT) can be applied successfully to perform GED tasks in Arabic. Our experiments show a remarkable result in performing both aimed NLP tasks. The proposed models related to the token level classification task achieved F1 of 87% and accuracy of 86% in the monolingual language model, and F1 of 85% and accuracy of 85% in the multilingual language model. Moreover, the proposed models related to the sentence-level classification task achieved F1 of 98% and accuracy of 98% in the monolingual language model, and F1 of 96% and accuracy of 96% in the multilingual language model. With taking the effectiveness of the following method, one of the main tasks to be tested in the recent future is the Arabic GEC.

References

1. Al-qaraghuli, M. (2021). Correcting Arabic Soft Spelling Mistakes Using Transformers. 146–151.
2. Antoun, W., Baly, F., & Hajj, H. (2020). AraBERT: Transformer-based Model for Arabic Language Understanding. ArXiv, May, 9–15.
3. Chouikhi, H., & Alsuhaibani, M. (2022). Deep Transformer Language Models for Arabic Text Summarization: A Comparison Study. Applied Sciences (Switzerland), 12(23). https://doi.org/10.3390/app122311944
4. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference, 1(Mlm), 4171–4186.
5. Ethnologue. Arabic language statistics, 2020.
6. Karita, S., Wang, X., Watanabe, S., Yoshimura, T., Zhang, W., Chen, N., Hayashi, T., Hori, T., Inaguma, H., Jiang, Z., Someki, M., Soplin, N. E. Y., & Yamamoto, R. (2019). A Comparative Study on Transformer vs RNN in Speech Applications. 2019 IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2019 - Proceedings, 9(4), 449–456. https://doi.org/10.1109/ASRU46091.2019.9003750
7. Madi, N., & Al-Khalifa, H. (2020). Error detection for Arabic text using neural sequence labeling. Applied Sciences (Switzerland), 10(15), 1–14. https://doi.org/10.3390/APP10155279

8. Madi, N., & Al-Khalifa, H. S. (2018). A Proposed Arabic Grammatical Error Detection Tool Based on Deep Learning. Procedia Computer Science, 142, 352–355. https://doi.org/10.1016/j.procs.2018.10.482
9. Montejo-Ráez, A., & Jiménez-Zafra, S. M. (2022). Current Approaches and Applications in Natural Language Processing. Applied Sciences (Switzerland), 12(10), 10–15. https://doi.org/10.3390/app12104859
10. Parnow, K., Li, Z., & Zhao, H. (2020). Grammatical Error Correction : More Data with More Context. 24–29. https://doi.org/10.1109/IALP51396.2020.9310498
11. Pires, T., Schlinger, E., & Garrette, D. (2020). How multilingual is multilingual BERT? ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, 4996–5001. https://doi.org/10.18653/v1/p19-1493
12. Qiu, Z., & Qu, Y. (2019). A Two-Stage Model for Chinese Grammatical Error Correction. IEEE Access, 7, 146772–146777. https://doi.org/10.1109/ACCESS.2019.2940607
13. Sarhan, I., & Spruit, M. (2020). Can we survive without labelled data in NLP? Transfer learning for open information extraction. Applied Sciences (Switzerland), 10(17). https://doi.org/10.3390/APP10175758
14. Shaalan, K. F. (2005). Arabic GramCheck : a grammar checker for Arabic. September 2004, 643–665. https://doi.org/10.1002/spe.653
15. Singh, S., & Mahmood, A. (2021). The NLP Cookbook : Modern Recipes for Transformer Based Deep Learning Architectures. 68675–68702. https://doi.org/10.1109/ACCESS.2021.3077350
16. Solyman, A., Wang, Z., & Tao, Q. (2019). Proposed model for arabic grammar error correction based on convolutional neural network. Proceedings of the International Conference on Computer, Control, Electrical, and Electronics Engineering 2019, ICCCEEE 2019. https://doi.org/10.1109/ICCCEEE46830.2019.9071310
17. Solyman, A., Zhenyu, W., Qian, T., Abdulgader, A., Elhag, M., & Toseef, M. (2021). Synthetic data with neural machine translation for automatic correction in arabic grammar. Egyptian Informatics Journal, 22(3), 303–315. https://doi.org/10.1016/j.eij.2020.12.001
18. Zaghouani, W., Mohit, B., Habash, N., Obeid, O., Tomeh, N., Rozovskaya, A., Farra, N., Alkuhlani, S., & Oflazer, K. (2014). Large scale Arabic error annotation: Guidelines and framework. Proceedings of the 9th International Conference on Language Resources and Evaluation, LREC 2014, 2362–2369. https://doi.org/10.1184/R1/6373136.v1
19. UNESCO. World Arabic language day, Dec 2019. https://www.unesco.org/ar/days/world-arabic-language
20. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.