Journal of King Saud University – Computer and Information Sciences 35 (2023) 101572

HOSTED BY

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

Optimizing the impact of data augmentation for low-resource grammatical error correction

Aiman Solyman [a,*], Marco Zappatore [b], Wang Zhenyu [a], Zeinab Mahmoud [c], Ali Alfatemi [d], Ashraf Osman Ibrahim [e], Lubna Abdelkareim Gabralla [f]

[a] School of Software Engineering, South China University of Technology, Guangzhou, China
[b] Department of Engineering for Innovation, University of Salento, Lecce, Italy
[c] School of Computer Science, Wuhan University of Technology, Wuhan, China
[d] Computer Science, Graduate School of Arts and Sciences (GSAS), Fordham University, New York, United States
[e] Creative Advanced Machine Intelligence Research Centre, Faculty of Computing and Informatics, Universiti Malaysia Sabah, Jalan UMS, 88400 Kota Kinabalu, Sabah, Malaysia
[f] Department of Computer Science and Information Technology, College of Applied, Princess Nourah bint Abdulrahman University, P.O. BOX 84428, Riyadh 11671, Saudi Arabia

ARTICLE INFO

Article history:
Received 3 February 2023
Revised 29 April 2023
Accepted 29 April 2023
Available online 9 May 2023

Keywords:
Grammatical error correction
Data augmentation
Synthetic data
Deep learning

ABSTRACT

Grammatical Error Correction (GEC) refers to the automatic identification and amendment of grammatical, spelling, punctuation, and word-positioning errors in monolingual texts. Neural Machine Translation (NMT) is nowadays one of the most valuable techniques used for GEC but it may suffer from scarcity of training data and domain shift, depending on the addressed language. However, current techniques (e.g., tuning pre-trained language models or developing spell-confusion methods without focusing on language diversity) tackling the data sparsity problem associated with NMT create mismatched data distributions. This paper proposes new aggressive transformation approaches to augment data during training that extend the distribution of authentic data. In particular, it uses augmented data as auxiliary tasks to provide new contexts when the target prefix is not helpful for the next word prediction. This enhances the encoder and steadily increases its contribution by forcing the GEC model to pay more attention to the text representations of the encoder during decoding. The impact of these approaches was investigated using the Transformer-based for low-resource GEC task, and Arabic GEC was used as a case study. GEC models trained with our data tend more to source information, are more domain shift robustness, and have less hallucinations with tiny training datasets and domain shift. Experimental results showed that the proposed approaches outperformed the baseline, the most common data augmentation methods, and classical synthetic data approaches. In addition, a combination of the three best approaches *Misspelling*, *Swap*, and *Reverse* achieved the best F_1 score in two benchmarks and outperformed previous Arabic GEC approaches.

1. Introduction

In computational linguistics, the automatic task of spotting and then correcting grammatical, spelling, punctuation, and word-choice errors in a text so that a new, error-free, text is achieved is known as Grammatical Error Correction (GEC). Therefore, GEC is a monolingual text-rewriting task. Different approaches have been introduced in the last two decades to enable GEC automation and to improve its fluency. Rule-based systems (Moukrim et al., 2021), n-gram language models (Rozovskaya et al., 2014), and Neural Machine Translation (NMT) (Solyman et al., 2022) are the three main categories of GEC. NMT-based systems such as convolutional neural networks (CNN) and self-attention networks (SAN) have proven to outperform other GEC approaches (Solyman et al., 2021; Solyman et al., 2022). The main challenge of neural-based GECs is that it requires large amount of parallel data, which is not available for low-resource languages such as Czech, Latvian, and Arabic. As automatic GEC becomes more important, English

* Corresponding author.
E-mail addresses: seaiman@mail.scut.edu.cn, wangzy@scut.edu.cn (A. Solyman).

Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.jksuci.2023.101572
1319-1578/

and Chinese received much research attention (Lai et al., 2022; Liang and Li, 2023), which makes up 45.3% of the internet content [1], while few studies have been conducted on low resources languages due to the lack of training data. Arabic GEC (AGEC) introduces additional challenges including orthographic ambiguity and the complexity of grammar, richness of morphological features, and dialectal variations (Habash et al., 2018). Nevertheless, AGEC is becoming more and more important, because of the widespread use of the Arabic language. Arabic is one of the official languages of the United Nations and the official language of 22 Arab countries with 420 million native speakers. In addition, a large community of 1.9 billion non-Arab Muslims uses Arabic in their daily worship, which further increases the need for effective AGEC systems, as well as the growing number of foreign language students studying and practicing Arabic.

GEC approaches, especially those based on NMT for low-resource languages, are hampered by data scarcity and data sparsity issues, resulting from an insufficient number of training data. In order to address these challenges, numerous solutions have been proposed such as transfer learning (Mosin et al., 2023), training multilingual systems (Pajak and Pajak, 2022), and data augmentation (DA) (Pellicer et al., 2023). The need to solve the data sparsity problem leads to the use of more data, even if it does not belong to the same distribution of the original data, which causes data distribution mismatch problem (Wang et al., 2018a).

Let consider X, Y as the original sentence pairs of a parallel GEC data, and $\hat{p}(X, Y)$ as the distribution of the training data that is different from the distribution of the true data $p(X, Y)$. The canonical maximum likelihood estimation (MLE) in the seq2seq GEC frameworks is to maximize the probability $p(x \mid y)$ of the given training data (Lai et al., 2022). The possible drawback of MLE is that it cannot address the mismatch between $p(X, Y)$ and $\hat{p}(X, Y)$, since $\hat{p}(X, Y)$ is usually defined on the training data and $p(X, Y)$ has much support from the valid data. Moreover, there has been recent increase in the use of supervised learning techniques to construct synthetic training data. However, these methods may have limitations that result in the generation of training data with limited patterns, which may only include spelling errors (Solyman et al., 2021; Solyman et al., 2022).

The motivation behind this paper is proposes completely different data augmentation approaches to address two main problems associated with neural-based GEC systems: data sparsity and data distribution mismatch. This is inspired by recent work in NMT (Voita et al., 2021) which finds that the contribution of source tokens on the output reduces as the decoding advances. The proposed approaches expose the neural network to new scenarios during training where the target context cannot achieve low loss. Consequently, the system passes the burden to the encoder to increase its contribution when generating the correction output. However, these approaches are used to avoid the problem of out-of-distribution data by constructing an additional training set $q(X, Y)$ with expanded support from $\hat{p}(X, Y)$.

The proposed framework is named Grammar Error Correction Data Augmentation (GECDA). GECDA improves the performance of GECs, which extends the distribution of the authentic data to the augmented data, resulting in a standalone version that is up to four times larger than the original data. Furthermore, it is a straightforward framework that does not require data preprocessing or additional data besides the original training corpora, or training additional models. The experiment results discussed in this paper on two benchmarks QALB-2014 and QALB-2015 show that GECDA outperforms two powerful data augmentation methods: RMAL (Norouzi et al., 2016) and SwitchOut (Wang et al., 2018b), and two common synthetic data techniques: spell-confusion (Grundkiewicz et al., 2019) and back-translation (Kiyono et al., 2020), as same as previous AGEC systems. On the other hand, to the best of our knowledge, this is the first work in GEC that addresses the problem of data distribution mismatch. The research contributions proposed in this paper can be summarized as follows:

- We propose a GEC framework to correct grammar automatically called GECDA, specifically designed for low-resource languages. GECDA addresses the challenges of data sparsity and data distribution mismatch that often limit the performance of existing GEC systems in these languages.
- We propose seven data augmentation approaches that enhance the contribution of source information in GEC systems, resulting in significant improvements in GEC performance without requiring additional training data.
- We conduct a comprehensive performance analysis of GECDA in the context of Arabic GEC. The experiments demonstrate that GECDA outperforms state-of-the-art data augmentation methods, classical synthetic data approaches, and existing Arabic GEC systems.

The remainder of this paper is structured as follows. Next section overviews the state of the art about GEC approaches, Section 3 details the proposed GECDA framework. Section 4 describes the experimental settings, whereas Section 5 presents the experimental results and discussion in Section 6. Finally, Section 7 gives the conclusion. The code, trained models, and data files are available on (https://github.com/aimanmutasem/GECDA).

2. Related work

Automatic grammar correction has received research attention since the early 1970s. Rule-based systems were the most common technique used to correct grammar (Moukrim et al., 2021). Moreover, these systems demand considerable customization efforts, as language-specific rules are needed (i.e., different linguists are needed to create rules coping with specific linguistic phenomena of different languages). Later on, n-gram language models have been used to measure the probability of each word from large text to detect grammatical errors (Rozovskaya et al., 2014). More recently, NMT has been used notably to translate incorrect input sentences into the correct grammatical form, which required massive training data. Furthermore, the recent and state-of-the-art GEC systems are based on NMT techniques. English has received more research attention thanks to extensive resources compared to other languages such as Slovene, Russian, and Arabic. These resources include pre-trained language models, parallel training data, and open-access GEC systems. For example, OpenAI introduced GPT-3, a mega language model that has a capacity of 175 billion parameters that does not require fine-tuning to correct grammar (Brown et al., 2020). Another example is Pathways, a mega language model introduced by Google AI that has been trained with 540 billion parameters and can produce human-like text (Chowdhery et al., 2022). In this section, we focus on GEC for low resource languages and for the English low resource research track, since they are more challenging (Grundkiewicz et al., 2019).

2.1. English GEC

In response to the lack of training data in GEC, numerous approaches have been proposed in recent years. Ge et al. (2018)

[1] https://www.statista.com/statistics/262946/most-common-languages-on-the-internet/.

introduced the iterative routing process called "fluency boost learning," which employs CNN to achieve remarkable improvements in the accuracy and fluency of GEC systems. Acheampong and Tian (2021) proposed a notable cascading learning strategy-based GEC system that reduces the need for massive training data in neural GEC systems. Xie et al. (2018) used beam-search noising techniques to construct parallel GEC training data from monolingual data with comparable performance to the original data. Wan et al. (2020) proposed data augmentation for GEC by editing the latent representations of grammatical sentences, which increases the diversity of training examples. Grundkiewicz et al. (2019) employed a spell-checker to synthesize parallel training data from an out-of-domain monolingual corpus for training a multi-head attention network. For Indonesian GEC systems, Musyafa et al. (2022) proposed a copy-augmented method applied to Transformer-based to improve accuracy by copying correct or unmodified words from the source to the target text. Sun et al. (2022) proposed a generic and language-independent strategy for multilingual GEC systems, utilizing available resources such as parallel translation data and pre-trained cross-lingual language models. Hagiwara and Mita (2020) introduced the GitHub Typo Corpus, a large-scale multilingual GEC training dataset for 15 languages. Náplava and Straka (2019) introduced synthetic multilingual GEC training data for training Transformers, which achieved significant improvements in Czech, German, and Russian. Qorib et al. (2022) proposed a GEC system that is a combination of machine-learning and binary classification approaches, which uses logistic regression for binary classification. The method yields substantial improvements over the state-of-the-art on both the CoNLL-2014[2] and BEA-2019[3] test sets. Lai et al. (2022) proposed an approach to improving the performance of GEC models, called Type-Driven Multi-Turn Corrections, which addresses the exposure bias problem in existing models. The approach involves generating multiple training instances for each original instance, with each instance targeting a specific type of error correction. This enables models to be explicitly aware of the process of gradual corrections and the interdependence between different types of corrections. Tarnavskyi et al. (2022) proposed an approach for improving the GEC sequence tagging architecture by assembling Transformer-based encoders in large configurations, achieving a new state-of-the-art result without pre-training on synthetic datasets. Their approach involves majority voting on span-level edits, and they also used knowledge distillation to generate synthetic training datasets.

2.2. Arabic GEC

Arabic GEC has started to receive more attention after successful shared tasks in 2014 (Mohit et al., 2014) and 2015 (Rozovskaya et al., 2015). Despite the early attention, Arabic GEC still suffers from a lack of training data, since the only annotated Arabic training data consist of 20,430 examples only. Rozovskaya et al. (2014), introduced a hybrid Arabic GEC system made of rule-based and machine-leaning approaches. Nawar (2015) proposed another solution that used word patterns and rule-based statistics to detect and correct grammatical errors. Role-based systems may not be able to handle all types of grammatical errors, especially those that are more complex or involve semantic errors. Sina (2017) employed seq2seq RNN and the attention mechanism in AGEC. Abandah et al. (2022) adopted bidirectional long short-term memory (BidLSTM) to correct soft spelling errors in modern and classical Arabic texts based on character level. Madi and Al-Khalifa (2020) employed LSTM, BiLSTM, and SimpleRNN baselines to detect errors, which outperformed the commercial Arabic Grammar Checker (Microsoft Word 2007), and also introduced their own training data. Watson et al. (2018) discussed FasTest pretrained word embedding and seq2seq BidLSTM to obtain more linguistic information in GEC. Solyman et al. (2019) proposed an AGEC model based on CNN, which was extended in Solyman et al. (2021), a GEC framework made of a classical confusion method and CNN seq2seq model consisting of nine convolutional layers and attention mechanism. GEC RNN-based systems are widely recognized for their reliability in detecting errors, it is crucial to acknowledge their fallibility and susceptibility to occasional errors. Furthermore, it is worth noting that GEC systems based on RNNs and CNNs may not be optimal for correcting more complex sentence structures and longer-range dependencies. More recently, an AGEC model based on the self-attention network equipped with a combination of capsule networks and a bidirectional regularization term strategy was proposed (Solyman et al., 2022). Pajak and Pajak (2022) tuned a set of pre-trained multilingual models such as mBART, mT5, or xProphetNet for GEC in seven different languages, including Arabic and reported encouraging results.

To summarize, automatic grammar correction has evolved significantly over the past few decades, starting from rule-based systems and n-gram models to more advanced techniques such as NMT-based methods. The recent development of mega language models like GPT-3 and Pathways has led to a surge in interest and progress in this field. However, GEC for low-resource languages and English low-resource research tracks pose a significant challenge due to the lack of training data. Different approaches have been proposed to overcome this challenge, including fluency boost learning, cascading learning strategy, noising techniques, data augmentation, and synthetic GEC training data. Although the majority of GEC systems for low-resource languages adopt two strategies to overcome the data sparsity problem, there is a need to investigate the impact of data augmentation to address this issue more effectively. The aim of this paper is to propose data augmentation approaches that are able to increase the contribution of the source during decoding. This aspect has not been previously investigated, and our proposed work aims to fill this research gap.

3. Methodology

This section introduces the proposed approaches and describes how they have been applied with a modified version of the baseline Transformer-based architecture proposed in Vaswani et al. (2017), as a neural translation task. GEC using NMT aims to build a model that can automatically detect and correct grammatical errors in text by leveraging the Transformer architecture. This can be achieved by training the model to minimize the negative log-likelihood of generating the correct output given the input. This loss function is known as the Maximum Likelihood Estimation (MLE) loss, which can be expressed as:

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

In this expression, $\hat{p}(X,Y)$ denotes the empirical distribution of the input–output pairs in the training data, where x and y denote the input and output sequences, respectively. The central aim of GEC using NMT is to optimize the likelihood of generating the correct output sequence y for a given input sequence x. This objective is achieved by computing the expectation of the log probability of the output sequence given the input sequence over all feasible input–output pairs in the training data, as represented by $\hat{p}(X,Y)$.

In data augmentation, let $\widehat{X}$ and $\widehat{Y}$ be the corresponding augmented version. The training objective of the neural-based seq2seq GEC systems is to maximize the likelihood estimation of $P(y|x)$ as

[2] https://www.comp.nus.edu.sg/nlp/conll14st.html.
[3] https://codalab.lisn.upsaclay.fr/competitions/4057.

in Eq. (1), where $\hat{p}(X, Y)$ is the empirical distribution of all training pairs (x, y) usually defined for training data only, while $p(X, Y)$ is the distribution of the true data which receives large support from the valid data. The difference in distribution between $p(X, Y)$ and $\hat{p}(X, Y)$ leads to data distribution mismatch problem, that cannot be covered by MLE. This problem will increase when $\hat{p}(X, Y)$ is not sufficient to cover the entire dataset. In order to overcome this problem, seven data augmentation strategies (namely, *Misspellings*, *Swap*, *Token*, *Source*, *Reverse*, *Mono*, and *Replace*) are proposed to augment training pairs $(\hat{x}, \hat{y})$ with a new distribution of $q\left(\widehat{X}, \widehat{Y}\right)$, that provides more extensive support compared to $\hat{p}(X, Y)$. Accordingly, the new objective of the MLE using data augmentation will be as follows:

$$MLE = \mathbb{E}_{x,y\sim q\left(\widehat{X},\widehat{Y}\right)}[logP(\hat{y}|\hat{x})]. \tag{2}$$

In this work, instead of synthesizing source sentences, all these strategies address the target sentences in almost all experiments. Each augmented example is appended to the corresponding true example in the whole data for the training task. As a result, the network is exposed to new scenarios during training where the target context cannot reduce the loss. This strengthens the encoder and steadily increases its contribution, thus forcing the GEC model to pay more attention to the encoder representations during decoding.

In the following subsections, we briefly explain the proposed strategies and the expected impact of each strategy on the training dynamics of the system. As opposed to the classical noise-based approaches, our *Misspelling* approach (Section 3.1) addresses the target side and uses the augmented data as an auxiliary task (i.e., a task that must be accomplished to improve the performance of the primary task) during training, while the other six proposed approaches have never been investigated in GEC. Initially, two hyper-parameters α and t were identified, where α refers to the target words to be affected and t is the total number of input words.

3.1. Misspelling

It is one of the most popular approaches of synthetic data in GEC, which uses a simple confusion function to generate spelling errors, good examples were presented in Solyman et al. (2021), Grundkiewicz et al. (2019). The proposed approach is different since it is used to address the target side and the synthesized data is applied as an auxiliary task during training to strengthen the encoder and increase its influence during decoding. Initially, the PyArabic [4] library was utilized to tokenize the given sequence, and the value of α was set to 0.1, which is multiplied by the total number of input words t to get the actual number of words to change. Next, one of these two sub-approaches is applied, either (1) deleting a character c_i within the word w_i in the input sentence, or (2) inserting a random character c_i at position c_{i+1} within the randomly selected word w_i. It is worth mentioning that the value of α was tuned during training using 0.1, 0.15, and 0.2. The best performance was obtained with 0.1, which is a delicate process that strives for balance to choose the most appropriate value [5]. Details are illustrated in Fig. 1 and all the steps of the *Misspelling* data augmentation approach are shown in Algorithm 1.

Algorithm 1 Misspelling approach

```
Require: $(X, Y), \alpha$. ▷ Original training pair, and $\alpha$ in [1, 2, 3, 4, 5]
Ensure: $\left(X, \widehat{Y}\right)$. ▷ An augmented example that synthesized the target-side ($\widehat{Y}$)
  function ADDCHAR$T_i$
    $T_i = [k_1, k_2, \ldots, k_n], k_i \in [k_1, k_2, \ldots, k_n]$
    $\hat{k}_i \in [k_1, k_2, \ldots, k_n]$ ▷ Add $\hat{k}_i$ into index $i + 1$
    $\widehat{T}_i = \left[k_1, k_2, k_i, \hat{k}_i, \ldots, k_n\right]$ ▷ Detokenize the array of characters to $\widehat{T}_i$
    Return $\widehat{T}_i$
  end function
  function DELETECHAR$T_i$
    $T_i = [k_1, k_2, \ldots, k_n], k_i \in [k_1, k_2, \ldots, k_n]$
    Delete $k_i$
    $\widehat{T}_i = [k_1, k_2, \ldots, k_n]$ ▷ Detokenize the array of characters to $\widehat{T}_i$
    Return $\hat{T}_i$
  end function
  procedure MISSPELLING$\alpha, X$
    $\widehat{X} \longleftarrow X$
    $N \longleftarrow (\alpha * len(X))$
    Chs = $[AddChar(T_i), DeleteChar(T_i)]$
    for $N$ do
      $T_i \in \widehat{X}$
      $\widehat{T}$ = choice(Chs) ▷ Apply either “Add character” or “Delete character” function
      Update $\widehat{X}$
    end for
    Return $\widehat{X}$
  end procedure
```

3.2. Swap

Recently, NMT performance has improved thanks to deep neural approaches and extensive parallel training data. Numerous works have been introduced to extend this success to low-resource languages. Motivated by previous work in NMT such as Artetxe et al. (2018), this work seeks to leverage attention to monolingual data to improve the quality of GECs. Especially, this approach proposes to swap the words on the target sentence from their original position until the number of words $(1 - \alpha) \times t$ remains, as shown in the following equation:

$$\widehat{Y} = swap(Y, (\alpha \times t)) \tag{3}$$

where *swap* is the transformation function, Y is the input sentence, and $(\alpha \times t)$ is the actual number of words that should be swapped. *Swap* follows the same setup as *Misspelling* data augmentation, which works in the target sentences instead of the source sentences. Given w_i and w_n as random words in the input sentence Y, each is swapped at the other word position. This forces the encoder to truly learn the compositionality of its own input words independently and led the decoder to trust less the target prefix when predicting the next token.

3.3. Token

This approach proposes to replace a number of words with the value of $(\alpha \times t)$ in the input target sentence with the unknown

[4] https://pypi.org/project/PyArabic/.

[5] A low value may lead to less error patterns and a high value would lose the context.

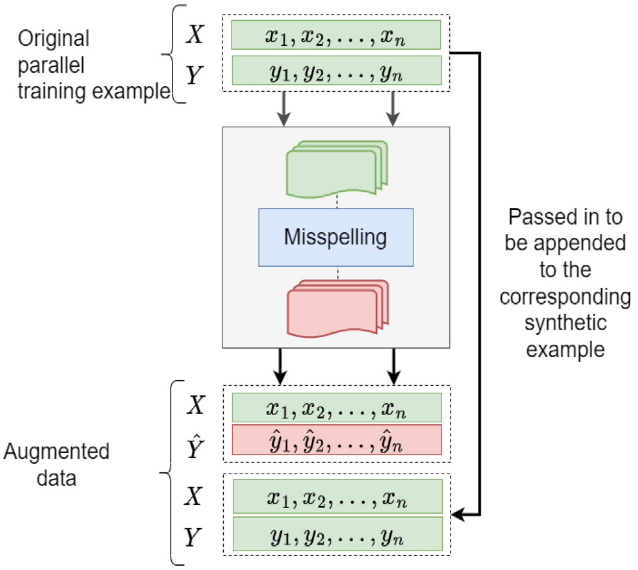


Fig. 1. Schema of the proposed misspelling data augmentation process, green refers to original data and red to synthesized data.

token <UNK>. It aims to weaken the decoder during training by replacing some or all of the conditioning information in the given sequences with a special token <UNK>. As a result, it makes the decoder prefixes less informative thus leading the system to uncertainty in the decoding when predicting a new word. In this scenario, the decoder predicts each word conditioned on the previous word and pays more attention to the encoder's representations. This forces the model to generate good predictions by relying on the latent variable. However, the default dropout was evaluated with the decoder trying to learn the model how to use the latent variable, but performance did not improve: therefore, we decided not to apply default dropout and we adopted the token-based approach. In addition, this approach is different from the word dropout, which is applied to the decoder rather than the feature extractor.

3.4. Source

The intuitive negative impact of copying training patterns to augmented data in GEC has never been investigated which leads the system to learn copying behavior when applies exceedingly after training. In the context of weakening the decoder and strengthening the encoder, a copy data augmentation strategy was proposed with different settings. Our approach seeks to leverage such a noise feature to the augmented data as an auxiliary task by transcribing the source sentence into the target and appending the augmented example with the corresponding original example to support the distribution of the synthetic data. Unlike what might be expected, the impact of the copied corpus will lead to improve accuracy of named entities and other correct words in the sentence as well as to increase the encoder's participation in predicting the final output. Another important feature of the auxiliary task in this approach is that the copied data has the same error rate and distributions as the authentic data, which is richer in training patterns compared to previous approaches.

3.5. Reverse

Recently, a notable study by Voita et al. (2021) analyzed the contribution of the encoder and the decoder in the final MT output. It was found that more training data lead the model to have more sharp tokens contributions and relying more on source information. In addition, the training process has several stages of different nature and it is not monotonic, and also the impact of the encoder decreases as the decoding progresses. To increase the influence of source-side in GEC, we propose to reverse the order of input words on the target-side from right-to-left to left-to-right. For instance, given $y = w_i, w_{i+1}, \ldots, w_l$ as the input target sentence, the reversed version $\hat{y}$ can be represented as $w_l, \ldots, w_{i+1}, w_i$. In our work, each augmented example was appended with the corresponding true example to support the distribution of the synthetic data. The impact of reversing the target sentences helps the system to increase the encoder's contribution when generating the corrected output sentences.

3.6. Mono

In NMT, stored compressed large bilingual parallel texts are known as *Bitext* and used as an intermediate representation in the compression process, which consist of sequence of *Biwords* pairs with a high probability of co-occurrence. Inspired by the *Biwords* approaches, this work proposes to leverage such linguistic knowledge from *Bitext* in GEC that can be represented as a detailed annotation of the source in the target. Accordingly, one-to-many word alignment was utilized for the source and target, and the value of α was set to 0.1. Next, the target-side words were reordered until the number of words $(1 - \alpha) \times t$ remained in their original positions. This causes monotonous alignment in target sentences and makes them less fluent; hence, it leads the system to increase the attention of the encoder's representations when generating a new word.

3.7. Replace

This approach proposes to use one-to-one word alignment for source and target sentences and then replaces the number of target words $(\alpha \times t)$ with random entries from the vocabulary of the training data. Let w_i and w_n be random words at random positions in the input sentence y, both of which are replaced by random words w_j and w_m from the training vocabulary placed in the same position. However, such a data augmentation approach supports the global objective of the current study which is to strengthen the encoder and force the system to pay more attention to the decoder and increase its contribution when predicting the correction output.

The implementation of each approach (described in Sections 3.1–3.7) follows the same guidelines and data flow listed in Algorithm 1. Each approach receives a pair (source and target) of parallel sentences (x, y) and generates a new synthetic pair $(x, \hat{y})$, which is then appended to the original example to build the augmented data. Table 1 shows the output of each approach, while Fig. 2 illustrates the components of the GECDA. GECDA uses the Transformer architecture, the encoder tokenizes and embeds the input text, followed by encoder layers with self-attention mechanisms. The decoder generates the output text and is trained to predict the correct word for each position in the output sequence. The predicted

Table 1
Sample sentences synthesized starting from a monolingual English sentence ("Input Sentence"), according to each proposed approach. The red color identifies synthesized words. The word order (with respect to the input sentence) is indicated by superscript numbers.

Approach	Explanation	Synthetic sentence	Input sentence
Misspelling	Generate spelling errors	Automatic^{1} grammattical^{2} error^{3} correction^{4} for^{5} low^{6} resource^{7} langages^{8}	Automatic^{1} grammatical^{2} error^{3} correction^{4} for^{5} low^{6} resource^{7} languages^{8}
Swap	Swap each pair of words randomly	Automatic^{1} languages^{8} low^{6} correction^{4} for^{5} error^{3} resource^{7} grammatical^{2}	
Token	Replacing random words in target sentences with special tokens *UNK*	Automatic^{1} grammatical^{2} UNK^{3} correction^{4} for^{5} low^{6} resource^{7} UNK^{8}	
Source	Copy the source sentence into the target sentence	Automatic^{1} grammatical^{2} error^{3} correction^{4} for^{5} low^{6} resourcelanguages^{7}	
Reverse	Reverse the order of target words from right-to-left to left-to-right	languages^{8} resource^{7} low^{6} for^{5} correction^{4} error^{3} grammatical^{2} Automatic^{1}	
Mono	Reordered target words to create a monotonous alignment between source and target	Automatic^{1} error^{3} grammatical^{2}correction^{4} for^{5} low^{6} languages^{8} resource^{7}	
Replace	Replaces the aligned source and target words with words from training vocabulary	Automatic^{1} grammatical^{2} North^{3} correction^{4} for^{5} low^{6} resource^{7} School^{8}	

output is post-processed to correct any errors before being obtained by passing the final vector through a linear layer and softmax function.

4. Experiments

4.1. Data

The seed data was a very small parallel corpus named QALB-2014 (Mohit et al., 2014). The source of QALB-2014 have been collected from English articles translated into Arabic, and Arabic Learners Written Corpus (CERCLL) (Alfaifi et al., 2014), as well as users' comments posted on the Aljazeera news channel. The collected data was corrected and annotated by a team of ten native speakers and linguistic experts. The whole data contains about 20,500 parallel examples subdivided into train and development sets. The augmented data used for the training task is two to four times larger than the original data. The synthetic data of back-translation and spell-confusion that were used are from Solyman et al. (2022) consists of 1,500,173 sentence pairs for training and development sets. To analyze the hallucinations, a very small corpus with high error rates has been used, partitioned into two subsets: L2-train-2015 with 311 training examples (43 k words) and L2-dev-2015, with 155 development examples (25 k words) (Rozovskaya et al., 2015). Regarding data preprocessing and also to address the problem of rare and unknown words, Byte Pair Encoding (BPEmb) was applied to split unknown tokens into subtokens (Sennrich et al., 2016).

4.2. Model setting

The complexity of the proposed model can be understood through a number of modifications made to a baseline Transformer-based architecture (Vaswani et al., 2017). First, the batch size was reduced from 2048 tokens to 128, and the model size was decreased from 512 to 256. Additionally, the number of layers was reduced to 4, while 8 attention heads were maintained per layer. To improve performance, the learned positional encoding approach was used instead of the static encoding method in the original paper by Vaswani et al. (2017). Similarly, label smoothing was not applied, following the setting of BERT (Devlin et al., 2019). A static Adam optimizer (Kingma and Ba, 2015) with a learning rate of 0.001 was used instead of warm-up and cool-down steps.

During training, each model was trained for 23 epochs using augmented data for each approach, followed by two epochs using QALB-2014 for fine-tuning. To prevent exploding gradients, gradient clipping was applied with a value of 1.0. Dropout with proba-

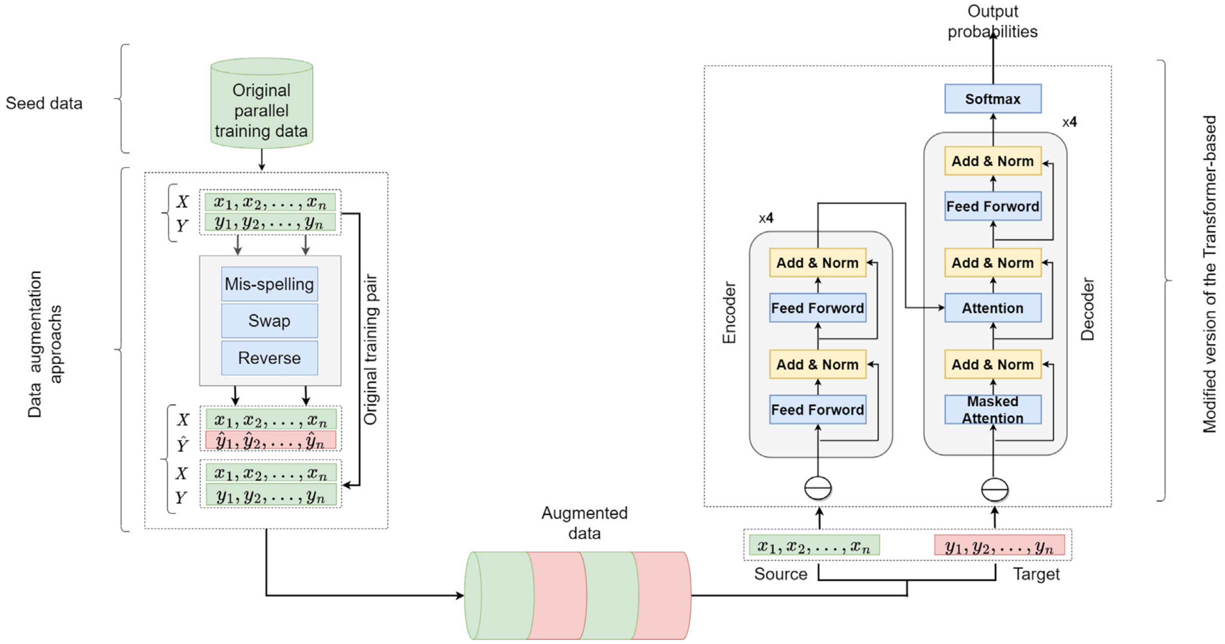


Fig. 2. An illustration of the main components of GECDA framework. The green color refers to the original training example while the red refers to the synthetic example which is used as an auxiliary task.

bilities of 0.15 and 0.10 was also used to avoid overfitting. The maximum number of input sequences during training and testing was set to 150 words. Tokenization was performed using the BPEmb algorithm with a vocabulary size of 1 k and 10 k and a dimension of 300. During inference, beam search was applied with a beam size of 5. The hyperparameters for the proposed model were selected manually based on prior knowledge and empirical evaluation. While an automatic parameter tuning process such as grid search or Bayesian optimization can lead to improved performance, we chose to select the hyperparameters manually in this study due to our experience with similar models and depending on the available computational resources. Nonetheless, we believe that the selected hyperparameters represent a reasonable starting point for future studies and can serve as a benchmark for evaluating the performance of automatic parameter tuning methods on this task. To ensure accurate decoding, data was cleaned after inference in case the decoder failed to correct repetitions of words or characters larger than 5, such as “cdcdcdcdcdcd” to “cdcd”. All experiments were conducted on two NVIDIA TITAN RTX GPUs with 25 GB RAM each, mounted in Scalable Link Interface configuration, an Intel Core i7-9700KF @ 3.6 GHz 12 cores, and NVIDIA CUDA Toolkit 10.2. The model was implemented in PyTorch using Python 3.6.

4.3. Evaluation

The proposed framework was evaluated on two benchmarks, and following the same guidelines in the first and second automatic grammar correction shared tasks (Mohit et al., 2014; Rozovskaya et al., 2015). MaxMatch (Dahlmeier and Ng, 2012) was applied to evaluate the performance using the same tool in the shared task to measure the word-level edits of each output compared to the golden target sentences, and reported precision, recall, and F_1 score using different scenarios during training. These metrics evaluate different aspects of a GEC system’s performance and provide useful insights into its strengths and weaknesses.

Precision is a measure of how accurate the GEC system is in correcting errors. It is defined as the ratio of true positives to the sum of true positives and false positives:

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{4}$$

In the context of GEC, true positives are the errors that the system correctly corrects, while false positives are the errors that the system incorrectly corrects. A high precision score indicates that the system is good at identifying and correcting errors without introducing new errors. Recall is a measure of how well the GEC system detects errors. It is defined as the ratio of true positives to the sum of true positives and false negatives:

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{5}$$

In the context of GEC, false negatives are the errors that the system fails to detect. A high recall score indicates that the system is good at identifying errors, even if it may not always be able to correct them. F1-score is the harmonic mean of precision and recall. It provides a single score that balances the trade-off between precision and recall, as shown in Eq. (6). A high F1-score indicates that the system is good at both identifying and correcting errors.

$$F1\text{-}score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{6}$$

In addition, *BLEU*-4 score was applied to evaluate the quality of the machine correction, and a case study is provided for human evaluation. *BLEU*-4 score is a measure of the similarity between the corrected output and the reference output. It is calculated using the n-gram precision of the corrected output with respect to the reference output, up to a maximum n-gram size of 4:

$$BLEU\text{-}4 = \mathrm{BP} \times \exp\left(\sum_{n=1}^{4} w_n \log p_n\right) \tag{7}$$

where w_n is the weight assigned to n-grams (typically set to $\frac{1}{4}$), and p_n is the precision of n-grams. The *BLEU*-4 score takes into account not only the accuracy of the corrections but also the fluency and naturalness of the corrected output. To calculate the *BLEU*-4 score, a brevity penalty is applied to penalize overly short corrected outputs. The brevity penalty is 1 if the length of the corrected output is greater than the length of the reference output, and it is calcu-

Table 2
Comparisons of precision, recall, F_1, and *BLEU*-4 score of the proposed approaches, and the baseline, RAML, SwitchcOut, spell-confusion, and back-translation using two benchmarks.

Model	QALB-2014				QALB-2015			
	Prec.	Recall	F_1	*BLEU*-4	Prec.	Recall	F_1	*BLEU*-4
Baseline	70.21	53.46	60.70	73.20	68.37	57.49	62.46	75.78
Misspelling	73.87	**56.03**	**63.72**	**75.22**	72.13	**60.37**	65.73	78.27
Swap	73.22	54.27	62.34	74.84	72.11	58.99	64.89	77.97
Token	71.53	54.28	61.72	73.90	70.60	58.81	64.17	77.20
Source	**74.94**	50.93	60.65	73.78	**74.04**	56.73	64.24	77.80
Reverse	73.85	53.77	62.23	74.78	73.28	59.72	**65.81**	**78.73**
Mono	71.57	53.81	61.43	73.94	70.08	58.41	63.72	77.06
Replace	72.36	54.30	62.04	74.29	71.33	59.11	64.65	77.73
RAML	71.66	54.16	61.69	73.62	71.02	57.86	63.76	77.31
SwitchOut	72.70	53.91	61.91	73.78	72.56	57.28	64.02	77.47
Spell-confusion	72.68	54.20	62.09	73.81	72.63	57.63	64.26	77.76
Back-translation	73.16	54.13	62.22	73.92	72.54	58.36	64.68	78.01

lated using an exponential function of the ratio of the reference output length to the corrected output length otherwise:

$$BP = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{otherwise} \end{cases} \tag{8}$$

where c is the length of the corrected sentence, and r is the length of the reference sentence.

5. Experimental results

5.1. Impact of data augmentation

Table 2 shows the performance of the proposed approaches measured in precision, recall, F_1, and *BLEU*-4 score. Across all results in Table 2, *Misspelling* (generating spelling errors), *Swap* (swapping each pair of words randomly), *Reverse* (reversing the order of target words from right-to-left to left-to-right), and *Replace* (replaces the aligned source and target words), each achieved the best performance, respectively. Furthermore, *Mono* (reordered target words to create a monotonous alignment between source and target), *Token* (replacing random words in target sentences with special tokens *UNK*), and *Source* (copying the source sentence into the target sentence), have achieved lower performance. The decrease in performance is due to a completely different vocabulary as shown in *Token* or unsystematic word order; however, it still outperforms the baseline. Fig. 3 illustrates the performance of all approaches. Furthermore, statistical results show that *Misspelling*, *Swap*, and *Reverse* auxiliary tasks perform better than the baseline (i.e. Transformer-based model has become the standard baseline model in the NLP that achieved state-of-the-art performance on various benchmark datasets) (Vaswani et al., 2017; Solyman et al., 2022), two data augmentation methods RMAL (Norouzi et al., 2016) and SwitchOut (Wang et al., 2018b), the classic spell-confusion method (Grundkiewicz et al., 2019), and back-translation (Kiyono et al., 2020) in the two benchmarks.

As a second step, taking into account the achievements of the best approaches over the baseline system in Table 2, we then investigated the performance of three sub-approaches as a combination of (*Misspelling* + *Swap*), (*Misspelling* + *Swap* + *Reverse*), and (*Misspelling* + *Swap* + *Reverse* + *Replace*). The second approach (spelling + swap + reverse) performed better compared to the other two, as shown in Table 3. Accordingly, we named the combined approach with the best performance as GECDA. On the other hand, we analyzed the impact of the combination of the most common data augmentation and synthetic data approaches including spell-confusion, back-translation, SwitchOut, and RAML. The combination of (spell-confusion + back-translation + SwitchOut) perform better as compared with (spell-confusion + back-translation) and (spell-confusion + back-translation + SwitchOut + RAML) in $F1$ and BLUE scores as shown in Table 3; however, GECDA still outperforms these approaches.

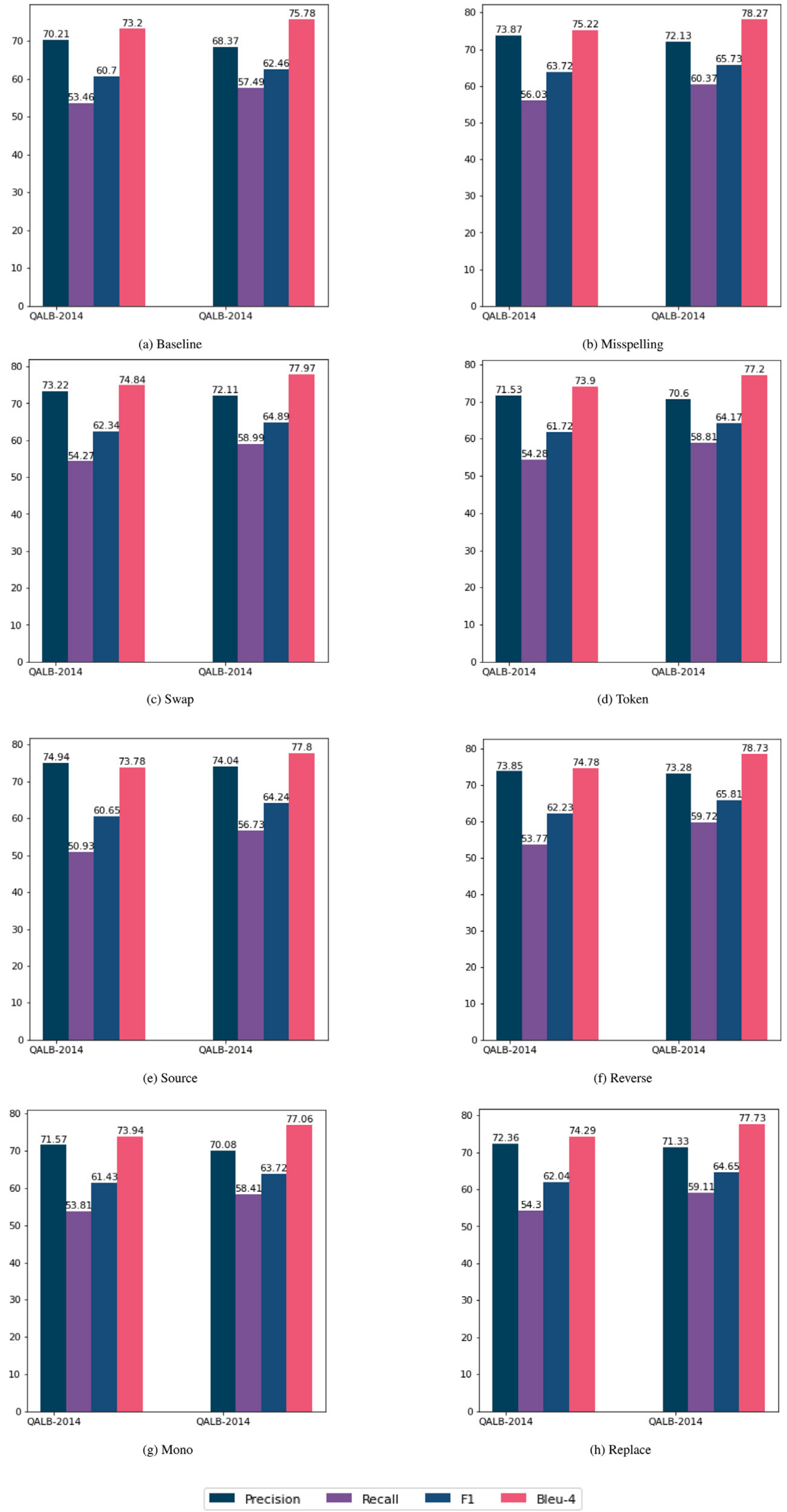
5.2. Contributions of source and target

This subsection analyses the relative contributions of source and target to GECDA output decision to predict the corrections of grammatical errors. Similarly, we aim at confirming that the improvement in output prediction quality and systematic robustness is associated with the encoder being exposed to more scenarios during training when a good source representation is required. Voita et al. (2021) has contributed that more training data lead NMT systems to tend more toward source representations; however, this motivated us to investigate such a feature using GECDA in GEC. To analyze the relative contribution of each source and target words on GECDA output decision, a layer-wise relevance propagation (LRP) has been applied with the transformer as in Voita et al. (2021). LRP was used to calculate the relative contributions $R_t(x_i)$ of the source word x_i and $R_t(y_j)$ of the target word y_i to the predictions that the network had made at the time t. The value of relevance at time step t it can be represented as $R_t(x) + R_t(y) = 1$, whereas for all time steps can be represented as the following equation.

$$\sum_i R_t(x_i) + \sum_j R_t(y_j) = R_t(x) + R_t(y) = 1, \tag{9}$$

This paper uses the same technique as Voita et al. (2021) to achieve reliable comparisons of the relative contributions for each equal-length subset of the source and target using held-out data. We teacher-force the reference predictions while computing LRP in order to obtain predictions with the exact same length, which allows us to evaluate various approaches to some extent. The held-out data was a combination of the development sets from QALB-2014 and QALB-2015, and the chosen subset are parallel examples with the same lengths and at least 16 words on both sides. To this end, we retrained the baseline, the best performing auxiliary tasks *Misspelling*,*Swap*, a combination of (*Misspelling* + *Swap*), and finally GECDA to compute the relative contributions using the same toolkit of Voita [6]. Fig. 4 illustrates the contribution of source representations at each time step t for different baselines using QALB-2014 and QALB-2015. The first time step has been skipped when the target prefix is not available, and the contribution of the source token EOS is also shown. As depicted in Fig. 4, the

[6] https://github.com/lena-voita/the-story-of-heads.



(a) Baseline

(b) Misspelling

(c) Swap

(d) Token

(e) Source

(f) Reverse

(g) Mono

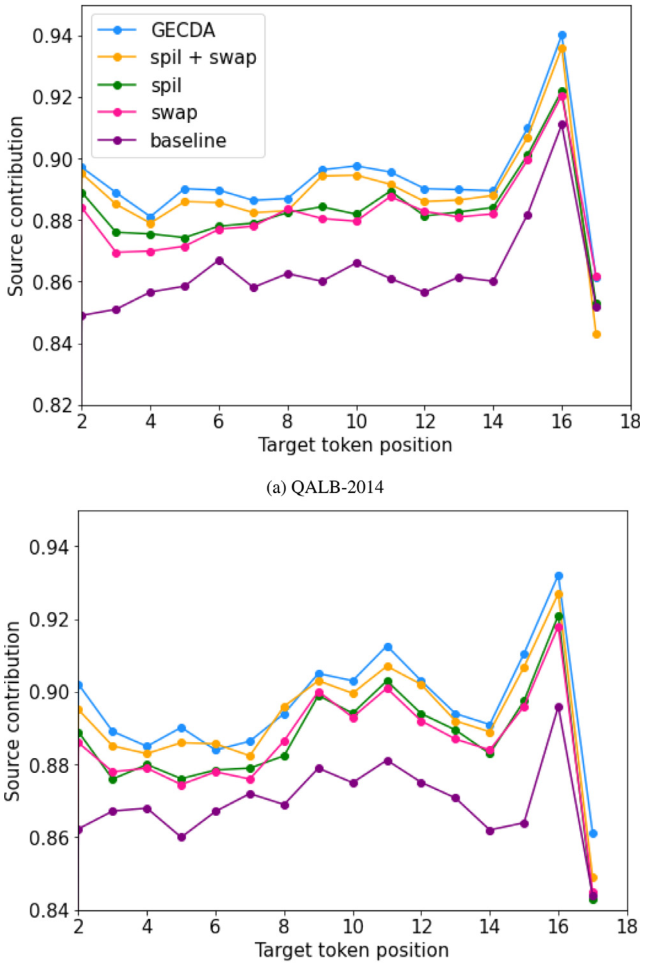
(h) Replace

Fig. 3. Performance achieved by the baseline and the proposed data augmentation approaches.

source influence decrease as decoding progresses, and the penultimate token shows a clear peak. This is because the decoder is checking whether the entire source sentence has been predicted before issuing a full stop at the end of the sequence. These results demonstrate the efficacy of GECDA since the auxiliary tasks increase the source influence and the baseline reported the smallest source influence in the two benchmarks. The differences in contributions are big when the decoding starts and remains throughout the sentence,

Table 3
Comparisons of a combination of three different sets of the proposed approaches as compared to a combination of existing data augmentation and synthetic data approaches in precision, recall, F_1, and *BLEU*-4 score.

Model	QALB-2014				QALB-2015			
	Prec.	Recall	F_1	*BLEU*-4	Prec.	Recall	F_1	*BLEU*-4
Misspelling + Swap	75.15	57.10	64.89	76.15	74.24	62.03	67.58	79.36
Misspelling + Swap + Reverse	**75.99**	**58.29**	**65.98**	**77.03**	**74.83**	**63.31**	**68.59**	**80.26**
Misspelling + Swap + Reverse + Replace	75.43	57.84	65.47	76.51	74.56	62.67	68.09	79.77
Back-translation + Spell-confusion	75.06	56.19	64.26	75.92	73.92	61.80	67.31	78.87
Back-translation + Spell-confusion + SwitchOut	75.68	57.10	65.09	76.31	74.20	62.13	67.63	79.27
Back-translation + Spell-confusion + SwitchOut + RAML	75.52	56.94	64.92	76.23	74.03	61.98	67.47	79.11



(a) QALB-2014

(b) QALB-2015

Fig. 4. The source contribution of GEC prediction for the baseline, *Misspelling*, *Swap* auxiliary tasks, a combination of (*Misspelling* + *Swap*), and GECDA.

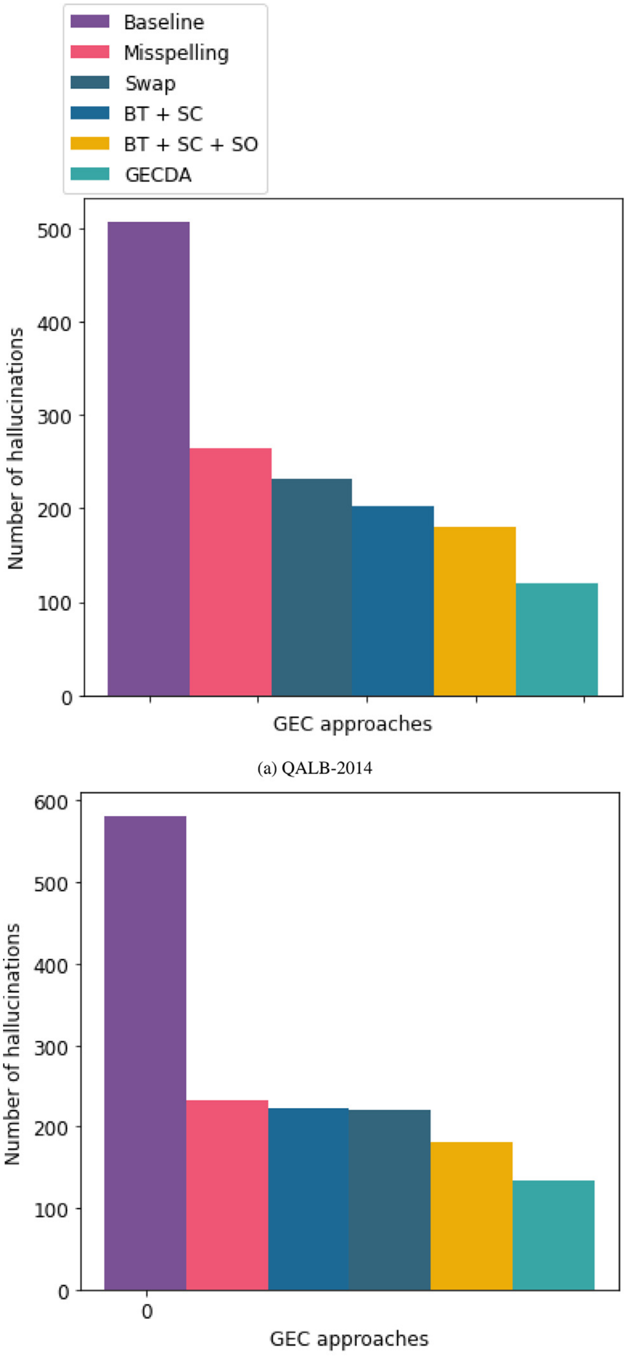
while there is no distinct difference in the source influence between *Misspelling*, and *Swap* auxiliary tasks. The highest source influence is reported by GECDA, which is a combination of multiple auxiliary tasks, this confirms the complementarity of the proposed model.

5.3. Hallucinations

In this section, we investigate the impact of GECDA in the system's encoder to reduce hallucinations when using very small training data. Raunak et al. (2021) define the hallucinations in NMT as completely inadequate translations that appear because the system depends too much on the target context. To the best of our knowledge, hallucinations have never been investigated in GEC, where they assume the form of repetitions or totally inconsistent syntax. We hypothesize that systems trained with GECDA produce less hallucination. To substantiate this hypothesis, we performed a hallucination analysis on the learners of Arabic as a foreign language (L2), which are very small training data and have high error rates. In this context, we applied the proposed method by Raunak et al. (2021) to measure the hallucinations which were used to detect training examples that indicate generating hallucination when injecting spurious tokens with input sentences. This method has been adapted in GEC to measure the number of system outputs that appear to be hallucinations. Hence, a modified version of BLEU score was applied as same as in Raunak et al. (2021), which only considers the precision of unigrams and bigrams with weights of 0.8 and 0.2, respectively. In our experiments, we classified tokens emitted correction as a hallucination if the adjusted BLEU score of the sentence level was less than 25 since the source and target are in the same language. The impact of hallucination has been evaluated on the baseline as compared to *misspelling* and *swap* auxiliary tasks, a combination of (*misspelling* + *swap*), and finally GECDA. The current method considers only the sentences relevant to hallucination; hence, we count the sentences that induce a hallucination in each system separately. The data that have been used to evaluate hallucinations are reported in Section 4.1. Fig. 5 shows the number of hallucinations in each system of the two benchmarks QALB-2014 and QALB-2015, shorter bars are better. As illustrated in Fig. 5(a) and 5(b), GECDA achieved the lowest score, this demonstrates the utility of GECDA to reduce hallucinations in the GEC systems.

5.4. Data preprocessing and decoding improvement

In this subsection, we investigate the impact of data preprocessing and decoding techniques that include multi-pass decoding and re-ranking L2R. In all the previous experiments, BPEmb was applied with a vocabulary size of 10 k. Furthermore, we decided



(a) QALB-2014

(b) QALB-2015

Fig. 5. Impact of hallucination of the baseline, the auxiliary tasks *Misspelling* and *Swap*, as same as GECDA, a combination of (back-translation (BT) + spell-confusion (SC)), and (BT + SC + SwitchOut (SO)), in two benchmarks (short bars are the best).

to investigate performance with different settings, which use BPEmb with a vocabulary size of 1000 (Heinzerling and Strube, 2018). Numerous studies in English GEC such as Wan et al. (2020) have used BPEmb with a vocabulary of 30 k. Table 4 shows the performance of GECDA with different settings of BPEmb. GECDA with 1 k vocabulary size performed better than 10 k and large gap when 30 k is used, which reduced the F_1 score by 06.07 and 08.08 for QALB-2014 and QALB-2015, respectively, less than GECDA with 1 k BPEmb. It is worth mentioning that GECDA with 1 k BPEmb requires more computational resources and more time to fill the gap in the training data, which performed better to correct spelling errors in Arabic GEC. Furthermore, BPEmb with 30 k is more powerful for capturing context representation compared to 1 k vocabulary size when a large dataset is available. These models were trained from left to right, to investigate performance using different settings, a new version was introduced that trained from right to left. Table 4 shows that the performance of GECDA R2L is better than that of GECDA L2R. This is because GECDA R2L captures the prefixes more efficiently due to the fact that the Arabic writing system is from right to left.

As a result of human language complexity, it is inefficient to correct multiple grammatical errors in a single round for low-resource scenarios. Ge et al. (2018) introduced a fluency boost learning to overcome this challenge and correct sentences in multiple rounds, which improves the performance of GEC systems. In this work, we investigated the impact of Fluency Boost Learning with GECDA using a simple modified version. In inference, the output of GECDA L2R was fed as input to the GECDA R2L model. Table 4 shows that F_1 and BLEU scores were improved, where the precision decreasing by 0.87 and 0.30 for QALB-2014 and QALB-2015, respectively. The decline in precision is a result of the low-rate unbalancing problem between L2R and R2L models (Liu et al., 2016). In the context of improving the performance of GECDA, we investigated the impact of re-ranking L2R. Initially, four different versions of GECDA were trained on both sides R2L and L2R. Then, an n-best list and the corresponding probabilities scores were generated for each model on both sides. Next, the R2L candidate list was passed to GECDA L2R to compute the scores. Finally, the scores on both sides were summed and used to re-rank the n-best list. GECDA R2L with re-ranking achieved the best F_1 and BLEU scores and also solved the unbalancing problem as shown in Table 4.

5.5. Statistical analysis of existing GEC models

Table 5 presents a comparison of the performance of the proposed GECDA framework with existing Arabic GEC models, using two benchmark datasets: QALB-2014 and QALB-2015. The table includes the best-performing systems in each benchmark, along with other neural-based models that used either synthetic data or fine-tuned pre-trained models. Overall, the results demonstrate that GECDA, with the use of BPEmb and re-ranking L2R, outperformed all existing GEC models that based on the pre-trained models and achieved the highest F_1 score in both benchmarks. GECDA achieved an F_1 score of 71.03 on the QALB-2014 dataset, surpassing the best-performing system's score of 70.39. Similarly, on the QALB-2015 dataset, GECDA achieved an F_1 score of 73.52, outperforming the best-performing system, which achieved an F_1 score of 73.19.

It is worth noting that GECDA leverages the synthetic data generated during training and strengthens the encoder during decoding, enabling it to achieve superior performance even when training data is limited. In contrast, some existing GEC models, such as those that rely solely on synthetic data, have limitations in terms of limited training patterns. In summary, the results in Table 5 and Fig. 6 confirm the effectiveness of the proposed GECDA framework in improving the performance of low-resource GEC tasks.

5.6. Case study

In the final analysis, GECDA has been investigated using a real-world example from the QALB-2015 test set. Table 6 shows the source, target, translation, and output of Arabic GEC models, including the baseline (Transformer) + GECDA + BPEmb 1 k vocab-

Table 4
Comparisons of precision, recall, F_1, and *BLEU*-4 score of the proposed approaches, and the impact of different vocabulary size 1 k and 30 k, as well as the performance of GECDA with multi-pass decoding and re-ranking L2R.

Model	QALB-2014				QALB-2015			
	Prec.	Recall	F_1	*BLEU*-4	Prec.	Recall	F_1	*BLEU*-4
GECDA + BPEmb 30 k	64.58	55.87	59.91	62.03	61.47	58.88	60.51	62.59
GECDA + BPEmb 1 k	77.96	60.40	68.07	84.66	76.71	65.78	70.83	87.37
GECDA + BPEmb 1 k + R2L	76.93	62.64	69.05	89.35	75.22	67.44	71.12	91.01
GECDA + BPEmb 1 k + R2L + multi-pass decoding	76.06	**65.62**	70.46	90.46	74.92	**71.05**	72.93	92.14
GECDA + BPEmb 1 k + R2L + re-ranking L2R	**78.66**	65.04	**71.03**	**90.79**	**77.68**	69.78	**73.52**	**92.98**

Table 5
Comparisons of F_1 of the proposed framework and existing AGEC models using two benchmarks.

System	2014	2015
Rozovskaya et al. (2014)	67.91	N/A
Solyman et al. (2021)	N/A	70.91
Nawar (2015)	N/A	72.87
Sina (2017)	50.34	N/A
Watson et al. (2018)	70.39	73.19
Pajak and Pajak (2022)	N/A	69.81
GECDA	**71.03**	**73.52**

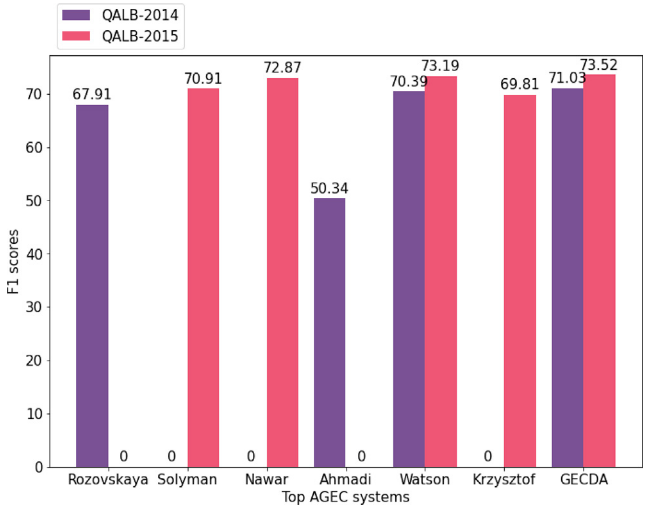


Fig. 6. F_1 score of the top systems in AGEC using QALB-2014 and QALB-2015 benchmarks.

ulary, and finally GECDA R2L with L2R re-ranking model. The given example contains 12 errors categorized as follows: Errors 1(s) and 2(s) are spelling errors, while errors 3(p), 5(p), 9(p), 11(p), and 12 (p) are punctuation errors. Furthermore, errors 4(m) and 8(m) are missing space errors, where 7(g) and 10(g) are grammatical errors, and error 6(g) is a grammatical error in the target sentence, which demonstrates that the training data is not clean, all errors are in red color. Initially, the baseline corrected most of the errors except the punctuation errors in 3(p) and 9(p) and the grammatical errors in 6(g), 7(g), and 10(g). Moreover, the second version, Transformer + GECDA, corrected all the errors except the punctuation errors in 3(p) and 9(p) and also caused a new error, which was labeled as new. The same model with re-ranking L2R corrected all errors except the punctuation error number 9(p) that should be “Shawla” an Arabic comma.

GECDA has been shown to be effective in correcting grammatical errors automatically without additional data. In other words, a summarizing sentence to say that the best-performing GECDA configuration was the one with R2L, BPEmb 1 k, etc. It achieved state-of-the-art results in F_1 score that was calculated from the edited words compared to its given golden words. However, it is still far from being perfect as it could not correct some punctuation errors, such as error number 9(p), since there are no punctuation rules in Arabic. In addition, it shows some weaknesses in challenging examples where standard Arabic has been mixed with dialectal words. Accordingly, extra efforts are needed to improve performance to correct complex errors such as punctuation and dialectal words.

6. Discussion

The results presented in Section 5 demonstrate the effectiveness of the proposed data augmentation approaches, which leads to significantly improving the performance of the baseline GEC system. This section aims to discuss the implications of these results and potential future research directions. The experiments indicate that GECDA outperforms two state-of-the-art data augmentation approaches, SwitchOut and RAML, as well as two synthetic data methods, back-translation and spell-confusion, in terms of F_1 and *BLEU*-4 scores on two benchmark. Furthermore, we investigated the relative contributions of source and target to GECDA output decision using LRP, revealing that source representations played a crucial role in the improvement of GECDA. GECDA combines three simple but effective data augmentation techniques, namely *Misspelling*, *Swap*, and *Reverse*. The combination of these techniques provides synthetic data that have almost the same distribution as the authentic corpus, and allows GECDA to better capture the diversity of grammatical errors in the target language. We found that source representations played a crucial role in the improvement of GECDA, increased the system domain robustness, and makes GECDA suffer less from hallucinations with very small training data. This proves that source representations can improve the model’s ability to handle various types of errors.

Despite the promising results, there are some limitations to our work. First, we only evaluated the proposed approach on two benchmark datasets, which may not fully represent the diversity of grammatical errors in other languages. Therefore, further experiments on other datasets and languages are needed to validate the generalization of our model. Second, while our experiments show that the proposed model outperforms several state-of-the-art data augmentation and synthetic data methods, there may be other methods that perform better. Thirdly, it is important to note that the proposed model did not include automatic parameters tuning. This may have limited the effectiveness of the model, as parameter tuning can help optimize the model’s performance. Therefore, incorporating an automatic parameter-tuning approach could potentially improve the proposed GEC model.

As part of our future work on improving our GEC model, we plan to investigate the use of two deep learning methods: Multi-layer Extreme Learning Machines (M-ELM) (Zhang et al., 2020) and Physics-informed deep learning (PIDL) (Zhang et al., 2022).

Table 6
Example of corrections in different versions of our GEC models. Error words are red-highlighted, numbered, and marked in terms of error type (i.e., s: spelling; g: grammar; p: position; m: missing space).

Type	Example
Source	اللغة العربية تجسد هوية العرب $^{1(s)}$والمسلبمين ، وهي $^{2(s)}$لغه الدين والشريعة $^{3(p)}$، رغم ذلك نجد العرب لا يهتمون $^{4(m)}$بهافقد شاعت الأخطاء فيها $^{5(p)}$_ تجد $^{6(g)}$مسؤلين $^{7(g)}$كبارا يرتكبون $^{8(m)}$أخطاءنحوية جسيمة وهم عرب أقحاح فهاذا مما يتأسف له $^{9(p)}$_ نرجو من مسؤولي العرب أن $^{10(g)}$يتحملو مسؤولياتهم تجاه لغتهم $^{11(p)}$_ وشكرا $^{12(p)}$_
Target	اللغة العربية تجسد هوية العرب والمسلمين ، وهي لغة الدين والشريعة ، رغم ذلك نجد العرب لا يهتمون بها ، فقد شاعت الأخطاء فيها ، تجد مسؤلين كبار يرتكبون أخطاء نحوية جسيمة وهم عرب أقحاح ، فهذا مما يتأسف له ، نرجو من مسؤولي العرب أن يتحملوا مسؤولياتهم تجاه لغتهم ، وشكرا .
English translation	Arabic language embodies the identity of Arabs and Muslims and is the language of religion and Sharia. Nevertheless, we find Arabs do not care for it, they become common mistakes, that you find high officials committing serious grammatical errors while they are native Arabs, which we regret. We hope that Arab officials will take their responsibility towards their language, thank you.
Baseline (Transformer)	اللغة العربية تجسد هوية العرب والمسلمين ، وهي لغة الدين والشريعة $^{3(p)}$_ ذلك نجد العرب لا يهتمون بها فقد شاعت الأخطاء فيها ، تجد $^{6(g)}$مسؤلين $^{7(g)}$كبارا أخطاء نحوية جسيمة وهم عرب أقحاح فهاذا مما يتأسف له $^{9(p)}$_ نرجو من مسؤولي العرب أن $^{10(g)}$يتحملو مسؤولياتهم تجاه لغتهم ، وشكرا .
GECDA R2L	اللغة العربية تجسد هوية العرب والمسلمين ، وهي لغة الدين والشريعة $^{3(p)}$_ ذلك نجد العرب لا يهتمون بها فقد شاعت الأخطاء ^{new}في ، تجد مسؤولين كبار يرتكبون أخطاء نحوية جسيمة وهم عرب أقحاح فهاذا مما يتأسف له $^{9(p)}$. نرجو من مسؤولي مسؤولي العرب أن يتحملوا مسؤولياتهم تجاه لغتهم ، وشكرا .
GECDA R2L + BPEmb 1 K + Re-ranking L2R	اللغة العربية تجسد هوية العرب والمسلمين ، وهي لغة الدين والشريعة ، ذلك نجد العرب لا يهتمون بها فقد شاعت الأخطاء فيها ، تجد مسؤولين كبار يرتكبون أخطاء نحوية جسيمة وهم عرب أقحاح فهاذا مما يتأسف له $^{9(p)}$. نرجو من مسؤولي العرب أن يتحملوا مسؤولياتهم تجاه لغتهم ، وشكرا .

These methods have shown promising outcomes in reducing the training time of deep neural networks and improving accuracy. We expect these methods will significantly reduce the time required to train GEC models and improve their overall performance. Furthermore, we plan to use a logic mining approach that incorporates supervised learning through association analysis (Kasihmuddin et al., 2022). This will enhance the effectiveness of the GEC techniques. In addition, we are interested in integrating the use of logic mining techniques such as Discrete Hopfield Neural Network (DHNN) (Jamaludin et al., 2022) with neural-based GEC models, and investigating alternative methods for feature selection and logical rule formulation.

7. Conclusion

In this paper, a GEC framework (named GECDA) has been presented for data augmentation in low-resource languages to correct Arabic grammar as a case study. In this context, seven aggressive transformation approaches were designed (namely, *Misspelling*, *Swap*, *Reverse*, *Replace*, *Mono*, *Token*, and *Source*). The proposed solution deviates from classical approaches, which strengthen the encoder and tend more to the source representations during decoding. Moreover, GECDA aims to generate synthetic data that have almost the same distribution as the authentic corpus. The augmented data introduces new contexts when the target prefix is not helpful for the next word prediction; hence, the system passes the burden to the encoder. Experimental results on two benchmarks showed that the proposed approach achieved remarkable improvement over the baseline system as well as over two data augmentation methods and the classical synthetic data approaches. Similarly, it also outperformed the existing Arabic GEC systems that used synthetic data and pre-trained models. GECDA performed well to overcome the challenges of data scarcity and mismatch data distribution, which increased the contribution of the source tokens, system domain robustness, and suffered less from hallucinations with very small training data. In summary, using the encoder representation to minimize training losses and increasing its contribution to generate output corrections will improve the performance of GEC systems without the need to use additional data or train additional models.

However, experimental results on two benchmarks showed remarkable improvement, and we also acknowledged some limitations. The evaluation was limited to only two benchmark datasets, and other methods may perform better. In addition, the proposed GEC model lacks automatic parameter tuning which could have limited its effectiveness. In the future, we aim to control synthetic error types, data augmentation using error type tags. Furthermore, we are interested in exploring neural-based approaches that can reduce training time and increase accuracy. In addition, we aim at investigating the impact of GECDA on other GEC tasks such as text-to-speech or speech-to-speech.

Funding

The APC was funded by Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2023R178), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2023R178), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

References

Abandah, G., Suyyagh, A., Khedher, M.Z., 2022. Correcting arabic soft spelling mistakes using bilstm-based machine learning. Int. J. Adv. Comput. Sci. Appl. 13. https://doi.org/10.14569/IJACSA.2022.0130594.

Acheampong, K.N., Tian, W., 2021. Toward perfect neural cascading architecture for grammatical error correction. Appl. Intell. 51, 3775–3788.

Alfaifi, A., Atwell, E., Hedaya, I., 2014. Arabic learner corpus (alc) v2: a new written and spoken corpus of arabic learners. Proceedings of Learner Corpus Studies in Asia and the World 2014, vol. 2. Kobe International Communication Center, pp. 77–89.

Artetxe, M., Labaka, G., Agirre, E., 2018. Unsupervised statistical machine translation. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Brussels, Belgium, pp. 3632–3642. https://doi.org/10.18653/v1/D18-1399. URL: https://aclanthology.org/D18-1399.

Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., 2020. Language models are few-shot learners.

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H.W., Sutton, C., Gehrmann, S., et al., 2022. Palm: Scaling language modeling with pathways.

Dahlmeier, D., Ng, H.T., 2012. Better evaluation for grammatical error correction. In: Proceedings of the 2012 Conference of the North American: Human Language Technologies.

Devlin, J., Chang, M.-W., Lee, K., Toutanova, K., 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2019.

Ge, T., Wei, F., Zhou, M., 2018. Fluency boost learning and inference for neural grammatical error correction. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics.

Grundkiewicz, R., Junczys-Dowmunt, M., Heafield, K., 2019. Neural grammatical error correction systems with unsupervised pre-training on synthetic data. In: Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications.

Habash, N., Eryani, F., Khalifa, S., Rambow, O., Abdulrahim, D., Erdmann, A., Faraj, R., Zaghouani, W., Bouamor, H., Zalmout, N., et al., 2018. Unified guidelines and resources for arabic dialect orthography. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018).

Hagiwara, M., Mita, M., 2020. Github typo corpus: A large-scale multilingual dataset of misspellings and grammatical errors. In: Proceedings of the 12th Language Resources and Evaluation Conference, pp. 6761–6768.

Heinzerling, B., Strube, M., 2018. Bpemb: Tokenization-free pre-trained subword embeddings in 275 languages. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018).

Jamaludin, S.Z.M., Romli, N.A., Kasihmuddin, M.S.M., Baharum, A., Mansor, M.A., Marsani, M.F., 2022. Novel logic mining incorporating log linear approach. J. King Saud Univ.-Comput. Informat. Sci. 34, 9011–9027.

Kasihmuddin, M.S.M., Jamaludin, S.Z.M., Mansor, M.A., Wahab, H.A., Ghadzi, S.M.S., 2022. Supervised learning perspective in logic mining. Mathematics 10, 915.

Kingma, D.P., Ba, J.L., 2015. Adam: A method for stochastic optimization. In: ICLR 2015: International Conference on Learning Representations 2015.

Kiyono, S., Suzuki, J., Mizumoto, T., Inui, K., 2020. Massive exploration of pseudo data for grammatical error correction. IEEE/ACM Trans. Audio, Speech, Language Process. 28, 2134–2145.

Lai, S., Zhou, Q., Zeng, J., Li, Z., Li, C., Cao, Y., Su, J., 2022. Type-driven multi-turn corrections for grammatical error correction. In: Findings of the Association for Computational Linguistics: ACL 2022. Association for Computational Linguistics, Dublin, Ireland, pp. 3225–3236.

Liang, Y., Li, L., 2023. Heterogeneous models ensemble for chinese grammatical error correction. International Conference on Artificial Intelligence, Virtual Reality, and Visualization (AIVRV 2022), vol. 12588. SPIE, pp. 111–120.

Liu, L., Utiyama, M., Finch, A., Sumita, E., 2016. Agreement on target-bidirectional neural machine translation. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.

Madi, N., Al-Khalifa, H., 2020. Error detection for arabic text using neural sequence labeling. Appl. Sci. 10, 5279.

Mohit, B., Rozovskaya, A., Habash, N., Zaghouani, W., Obeid, O., 2014. The first qalb shared task on automatic text correction for arabic. In: Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP), pp. 39–47.

Mosin, V., Samenko, I., Kozlovskii, B., Tikhonov, A., Yamshchikov, I.P., 2023. Fine-tuning transformers: Vocabulary transfer. Artif. Intell. 103860.

Moukrim, C., Abderrahim, T., Tarik, A., et al., 2021. An innovative approach to autocorrecting grammatical errors in arabic texts. J. King Saud Univ.-Comput. Informat. Sci. 33, 476–488.

Musyafa, A., Gao, Y., Solyman, A., Wu, C., Khan, S., 2022. Automatic correction of indonesian grammatical errors based on transformer. Appl. Sci. 12, 10380.

Náplava, J., Straka, M., 2019. Grammatical error correction in low-resource scenarios. In: Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019), pp. 346–356.

Nawar, M., 2015. CUFE@QALB-2015 shared task: Arabic error correction system. In: Proceedings of the Second Workshop on Arabic Natural Language Processing.

Norouzi, M., Bengio, S., Jaitly, N., Schuster, M., Wu, Y., Schuurmans, D., et al., 2016. Reward augmented maximum likelihood for neural structured prediction. Adv. Neural Informat. Process. Syst. 29.

Pajak, K., Pajak, D., 2022. Multilingual fine-tuning for grammatical error correction. Expert Syst. Appl. 116948.

Pellicer, L.F.A.O., Ferreira, T.M., Costa, A.H.R., 2023. Data augmentation techniques in natural language processing. Appl. Soft Comput. 132, 109803.

Qorib, M., Na, S.-H., Ng, H.T., 2022. Frustratingly easy system combination for grammatical error correction. In: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 1964–1974.

Raunak, V., Menezes, A., Junczys-Dowmunt, M., 2021. The curious case of hallucinations in neural machine translation. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 1172–1183.

Rozovskaya, A., Habash, N., Eskander, R., Farra, N., Salloum, W., 2014. The Columbia system in the QALB-2014 shared task on Arabic error correction., In: Proceedings of the EMNLP 2014 Workshop on Arabic.

Rozovskaya, A., Bouamor, H., Habash, N., Zaghouani, W., Obeid, O., Mohit, B., 2015. The second qalb shared task on automatic text correction for arabic. In: Proceedings of the Second workshop on Arabic Natural Language Processing, pp. 26–35.

Sennrich, R., Haddow, B., Birch, A., 2016. Neural machine translation of rare words with subword units. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics.

Sina, A., 2017. Attention-based encoder-decoder networks for spelling and grammatical error correction.

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).

Solyman, A., Zhenyu, W., Qian, T., Elhag, A.A.M., Toseef, M., Aleibeid, Z., 2021. Synthetic data with neural machine translation for automatic correction in arabic grammar. Egypt. Informat. J.

Solyman, A., Zhenyu, W., Qian, T., Elhag, A.A.M., Rui, Z., Mahmoud, Z., 2022. Automatic arabic grammatical error correction based on expectation maximization routing and target-bidirectional agreement. Knowl.-Based Syst. 108180

Sun, X., Ge, T., Ma, S., Li, J., Wei, F., Wang, H., 2022. A unified strategy for multilingual grammatical error correction with pre-trained cross-lingual language model, 4367–4374. URL: https://www.ijcai.org/proceedings/2022/606.

Tarnavskyi, M., Chernodub, A., Omelianchuk, K., 2022. Ensembling and knowledge distilling of large sequence taggers for grammatical error correction. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 3842–3852.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I., 2017. Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008.

Voita, E., Sennrich, R., Titov, I., 2021. Analyzing the source and target contributions to predictions in neural machine translation. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing.

Wan, Z., Wan, X., Wang, W., 2020. Improving grammatical error correction with data augmentation by editing latent representation. In: Proceedings of the 28th International Conference on Computational Linguistics, 2020.

Wang, X., Pham, H., Dai, Z., Neubig, G., 2018a. SwitchOut: an efficient data augmentation algorithm for neural machine translation. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing.

Wang, X., Pham, H., Dai, Z., Neubig, G., 2018b. Switchout: an efficient data augmentation algorithm for neural machine translation. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 856–861.

Watson, D., Zalmout, N., Habash, N., 2018. Utilizing character and word embeddings for text normalization with sequence-to-sequence models. In: Proceedings of the 2018 Conference on Empirical Methods.

Xie, Z., Genthial, G., Xie, S., Ng, A.Y., Jurafsky, D., 2018. Noising and denoising natural language: Diverse backtranslation for grammar correction. In: Proceedings of the 2018 Conference: Human Language Technologies.

Zhang, J., Li, Y., Xiao, W., Zhang, Z., 2020. Non-iterative and fast deep learning: Multilayer extreme learning machines. J. Franklin Inst. 357, 8925–8955.

Zhang, J., Zhao, Y., Shone, F., Li, Z., Frangi, A.F., Xie, S.Q., Zhang, Z.-Q., 2022. Physics-informed deep learning for musculoskeletal modelling: Predicting muscle forces and joint kinematics from surface emg. IEEE Trans. Neural Syst. Rehabil. Eng.