Toward Utilizing Bidirectional Multi-head Attention Technique for Automatic Correction of Grammatical Errors

Zeinab Mahmoud, Natalia Kryvinska, Mohammed Abdalsalm, Aiman Solyman, Ali Alfatemi, and Ahmad Musyafa

***Abstract*—Automatic Grammar Error Correction (GEC) models identify and correct a wide range of grammatical errors. Various strategies have been proposed for GEC, with the Neural Machine Translation (NMT) approach being the most effective. However, NMT-based GEC models with encoder-decoder layers rely heavily on the highest layer, leading to potential inaccuracies. Additionally, during inference, exposure bias can cause the model to substitute previously targeted words with incorrect alternatives. Another challenge is data scarcity. This paper introduces a GEC model leveraging the seq-to-seq Transformer framework, specifically designed for low-resource languages like Arabic. We propose a method to generate noise in the text to create synthetic parallel data, addressing the data constraints. Inspired by Capsule Networks (CapsNet), we incorporate CapsNet in GEC to dynamically aggregate information from multiple layers. In order to mitigate exposure bias, we incorporated a bidirectional training approach and a regularization term using Kullback-Leibler divergence to align left-to-right and right-to-left models. Experiments on two benchmarks demonstrate that our model outperforms current Arabic GEC models, achieving the highest scores. The code is available on GitHub (https://github.com/Zainabobied/ArabicGEC).**

***Index Terms*—Neural Machine Translation, Grammar Error Correction, Data Scarcity, Arabic Language Processing.**

I. Introduction

THE demand for automated Grammatical Error Correction (GEC) tools has increased with the rise in the global population learning a second language. Neural Machine Translation (NMT) and neural-based techniques such as multi-head attentions, have become essential in offering GEC solutions for correcting text. The sequence-to-sequence (seq2seq) architecture has demonstrated remarkable improvements in performance. This effectiveness relies on deep learning models such as RNN and CNN [1, 2]. However, GEC seq2seq neural-based approaches require massive parallel training data, which is not available especially in low-resource languages such as Vietnamese, Hindi, and Arabic.

In neural network-based GEC models that utilize neural networks, it is typical to use a multi-layered architecture, in which the uppermost layer in this structure serves a pivotal role in subsequent operations. However, research in the field of NMT has shown that utilizing representations from all layers outperforms methods that only use the top layer [3]. Traditional GEC techniques use static aggregation, overlooking valuable contextual information within the hidden representations. In this scenario, the application of dynamic aggregation within GEC can enhance correction accuracy by leveraging the utilization of contexts that have not been explored before.

Furthermore, the autoregressive structures in GEC systems often lead to exposure bias problems [4]. This issue arises when the model replaces previous target words during inference with words it has formulated, potentially introducing early-generated errors that can negatively impact correction accuracy and result in incorrect prefixes and suffixes. To mitigate this challenge, the literature proposes an additional model that generates corrections from right-to-left (R2L). This model can distinguish the correct corrections in the original list of candidates of the left-to-right model (L2R) [5]. The drawback of this technique is that the R2L list is also susceptible to the issue of exposure bias.

To improve the performance of our model and overcome the limitations of the previous GEC systems, we propose a noise method called Generate and Augment Synthetic Data (GASD). GASD comprises two baselines: Swap Data Augmentation (SDA) and Synthetic Error Generation (SEG). In addition, a GEC neural network structure is proposed that employs the Transformer-based architecture [6], which has proven to be effective in various NLP tasks [7, 8]. This architecture utilizes a combination of Capsule Network (CapsNet), which is inspired by the field of computer vision [9], and a bidirectional agreement technique. CapsNet has the advantage of being able to dynamically aggregate information and optimize language capture features throughout the network, leading to improved model performance. To overcome the problem of exposure bias in GEC, our model enforces an enhanced agreement process between the R2L and L2R models, which is different from traditional re-ranking strategies. To achieve this, we aligned the training objectives of these two models using a Kullback-Leibler regularization term; this led to reduced divergence and enhanced the correction quality. The main contributions of this paper include:

Manuscript received March 30, 2024; revised September 21, 2024.

Zeinab Mahmoud is a PhD candidate in the School of Computer Science and Technology, Wuhan University of Technology, China (e-mail: zainabobied2019@whut.edu.cn).

Natalia Kryvinska is a Professor in the Department of Information Management and Business Systems, Comenius University, Slovakia (e-mail: natalia.kryvinska@uniba.sk).

Mohammed Abdalsalm is a PhD candidate in the School of Computer Science and Technology, Wuhan University of Technology, China (e-mail: 79834@whut.edu.cn).

Aiman Solyman is a Postdoctoral Researcher at the Department of Computer Science, University of Milan, Italy (e-mail: aiman.solyman@unimi.it).

Ali Alfatemi is a PhD candidate in the CIS Department, Fordham University, Bronx, NY 10458, USA (e-mail: aalfatemi@fordham.edu).

Ahmad Musyafa is the head of the Department of Informatics Engineering at Pamulang University, South Tangerang, 15310, Indonesia. (E-mail: dosen00668@unpam.ac.id).

- Propose GASD, a noise method used to generate synthetic parallel dataset for GEC that contains over 14.21 million examples.
- Propose the use of CapsNet with a seq2seq Transformer-based model for more efficient aggregation of linguistic information in GEC.
- Propose a regularization approach that applies KL-divergence-based as a term between R2L and L2R models to mitigate exposure bias in GEC.

The paper is organized into seven sections. Section II introduces the research context. Section III gives an overview of the related work. Section IV provides details of our research methodology. Section V outlines the experimental details. Section VI is devoted to results and discussion. Finally, Section VII provides the conclusion of the paper.

II. Research context

Deep layer representations play a pivotal role in the domain of GEC [10, 11, 12]. These non-linear transformation layers convert incorrect sentences into grammatically correct outputs. The architecture utilizes multi-layer encoders and decoders, with the first layer used for word embedding. In mathematical terms, the structure of each encoder layer can be represented as follows:

$$H_e{}^{l} = \textsc{Layer}_e(H_e{}^{l-1}) + H_e{}^{l-1}. \quad (1)$$

The $\textsc{Layer}(\cdot)$ incorporates various neural network techniques, such as RNNs, CNNs, or Transformers. These layers are connected through residual connections, an essential architectural component that enables information flow between layers [6]. In the same context, the decoder comprises a sequence of L consecutive layers. These layers are designed in a specific structure, which is represented as follows:

$$H_d{}^{l} = \textsc{Layer}_d(H_d{}^{l-1}, H_e{}^{L}) + H_d{}^{l-1}. \quad (2)$$

Each decoder layer depends on its previous one ($H_d{}^{l-1}$) and the information from the top encoder layer ($H_e{}^{L}$), as in Equation 2. The output of the GEC model is primarily derived from the top decoder layer ($H_d{}^{L}$). Therefore, both the encoder and decoder operations mainly involve stacking layers sequentially, with a preference for using only the top layer. In the realm of deep learning, it is observed that deeper layers are adept at extracting both contextual and semantic information. Thus, the encoder and decoder operations mainly involve stacking layers sequentially, utilizing the top layer for output. In deep learning, deeper layers extract contextual and semantic information more efficiently. However, the performance of the top layer in consistently yielding good results is not guaranteed. To mitigate the inefficiencies in deep learning methodologies, the concept of residual connections is employed for layer aggregation. Nevertheless, it is important to highlight that a single-step operation within a residual connection might not be enough to achieve adequate aggregation, as established in [13].

Neural-based GEC models generally use a seq2seq encoder-decoder structure. The encoder takes in the erroneous input sequence, denoted as x, and transforms it into a sequence of hidden vectors, represented as $h = (h_1, h_2, ..., h_T)$, through the use of a neural network. Following this, the decoder uses the hidden states, referred to as h, to predict the corrected output sequence y. During inference, GEC faces the exposure bias problem, which can result in errors due to the substitution of correct words with incorrect alternative words [4].

To address these limitations, we introduce a GEC model that leverages the strengths of deep layer representations and also effectively mitigates the impact of exposure bias in text correction.

III. Related work

This section highlights advancements in GEC, particularly the use of NMT for low-resource languages. The dominant approach in GEC was rule-based, relying on grammar rules created by linguists. The introduction of the N-gram language model marked a significant shift in GEC. This method detects grammatical errors by evaluating the probability of each n-gram occurring within a substantial corpus of text. However, the landscape of GEC has been transformed with the introduction of SMT and NMT techniques. These techniques allowed the conversion of inaccurate sentences into their accurate forms. Pre-trained language models, such as BERT, have greatly benefited English GEC models. BERT was trained on a monolingual corpus to understand the complexities of text coherence for NLP applications [14]. Subsequently, it was fine-tuned for grammatical corrections. Similarly, OpenAI introduced GPT-3, a language model based on the Transformer architecture that utilizes a text corpus of 570 GB, including 175 billion parameters [15]. OpenAI further advanced this technology with the introduction of GPT-4, showcasing the capabilities of these models in replicating human-like text generation for various NLP tasks, including GEC.

The main objective of research in this domain is to overcome the shortage of training data. The authors in [16] used beam search noising techniques to generate additional training data from a monolingual corpus. Additionally, [17] evaluated the effectiveness of cascading learning methodologies. The scarcity of GEC resources is a widespread issue in Asian languages. To address this problem, [2] introduced an Indonesian GEC system that utilizes LSTM for multi-classification tasks, specifically designed to fix common POS errors in Indonesian text. Nahid et al. [18] present a transformer-based model called Panini for correcting grammatical errors in Bangla text. The model leverages a large-scale parallel corpus and transfer learning, enhancing the precision and eloquence of the Bangla language.

In the realm of GEC, Arabic is often considered to have limited resources. The sole annotated training dataset available is the QALB-2014 dataset, which comprises 20,430 sentences. To access more linguistic data, [19] utilized a pre-trained embedding model combined with a character-based and Bidirectional Recurrent Neural Network (BRNN) GEC model. Ahmadi et al. [20] introduced a seq2seq model leveraging BRNN and an attention mechanism. Another study [21] proposed a GEC model utilizing multiple layers of CNN as its fundamental structure. This approach was further advanced by [22], which used synthetic data for pre-training a GEC model consisting of 278,770 instances. Solyman et al. [1] introduced an AGEC model based on data augmentation techniques to augment data during training, demonstrating

the potential of data augmentation in enhancing GEC models' performance, particularly for low-resource languages like Arabic.

In summary, previous studies have focused on generating synthetic training sets. Arabic GEC approaches employed simple confusion techniques, resulting in relatively small data sets. Moreover, most Arabic GEC systems rely on rule-based methods and traditional multi-layer architectures that primarily utilize the top layer. The objective and motivation of this paper are to address and overcome these limitations.

IV. Research Methodology

A. Model Outline

We proposed a noise method to augment the training dataset with synthetic errors, enhancing the model's generalization capacity by mimicking real-life GEC data. Inspired by the work of [9] in computer vision, our model incorporates CapsNet and EM routing to aggregate information across multiple layers efficiently. To address exposure bias in seq-to-seq models, we introduced a method to align two models: R2L and L2R. Moreover, we incorporated the Kullback–Leibler divergence into the training process, which encouraged these models to learn from each other's strengths. The architecture of this approach is illustrated in Figure 1.

B. Noise Method

The scarcity of available training data limits the task of GEC. In Arabic GEC, the QALB corpus, containing 19,411 examples, is the only annotated data available. Compared to training datasets for GEC in other languages, this dataset is relatively small. To address this challenge and increase the training dataset, we proposed a noise approach leveraging a monolingual corpus. We utilized the OSIAN dataset as a seed corpus for our technique. OSIAN is a public monolingual Arabic dataset comprising 477,556 articles, 15 million sentences, and 367.5 million words. The data covers various subjects, including education, economy, health, sports, and stories. However, before adding the OSIAN corpus to the training data, the corpus was combined into a single dataset, and some preprocessing steps were applied. These steps included eliminating links, non-UTF8 encoding, mentions, diacritics, and extraneous white spaces. Subsequently, the PyArabic library [23] was applied to segment the text into sentences, each containing at least ten words. This process yielded a substantial training resource in 14.21 million sentences.

The objective of GASD is to produce reliable synthetic data by generating grammatical errors that reflect common mistakes found in human writing. This method involves using SDA and SEG. SDA swaps words within the sentence, while SEG introduces spelling mistakes or standardizes Arabic characters with similar shapes. After organizing the data such that each sentence was on a distinct line, the sentences were input into GASD. SDA augments the data by swapping words equivalent to 10% of the total word count in each sentence. Experiments determined that exceeding 10% resulted in a loss of sentence context. This phase generated syntax and grammatical errors. Following this, SEG normalized Arabic characters and generated spelling errors at a proportion of 10%, similar to SDA.

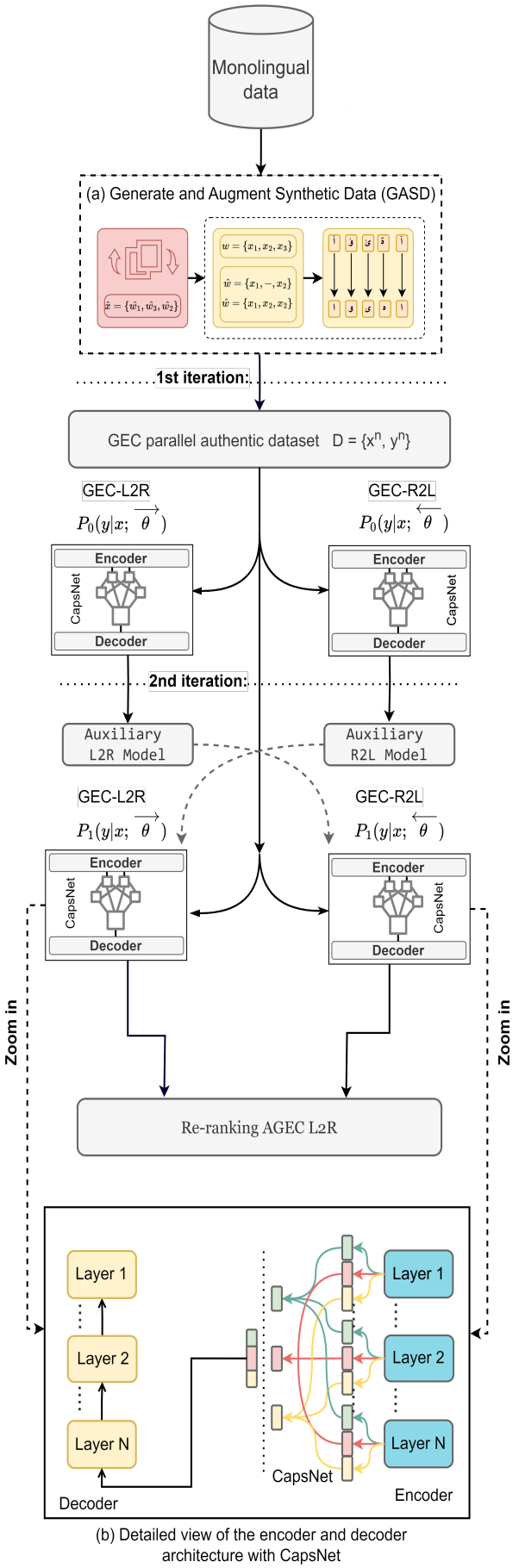

Fig. 1. Illustration of the Model Architecture Integrating GASD. (a) Highlights the Bidirectional GEC-R2L and GEC-L2R Models Over Two Iterations Governed by a Regularization Term. (b) Provides a Detailed View of the Encoder and Decoder Architecture with CapsNet.

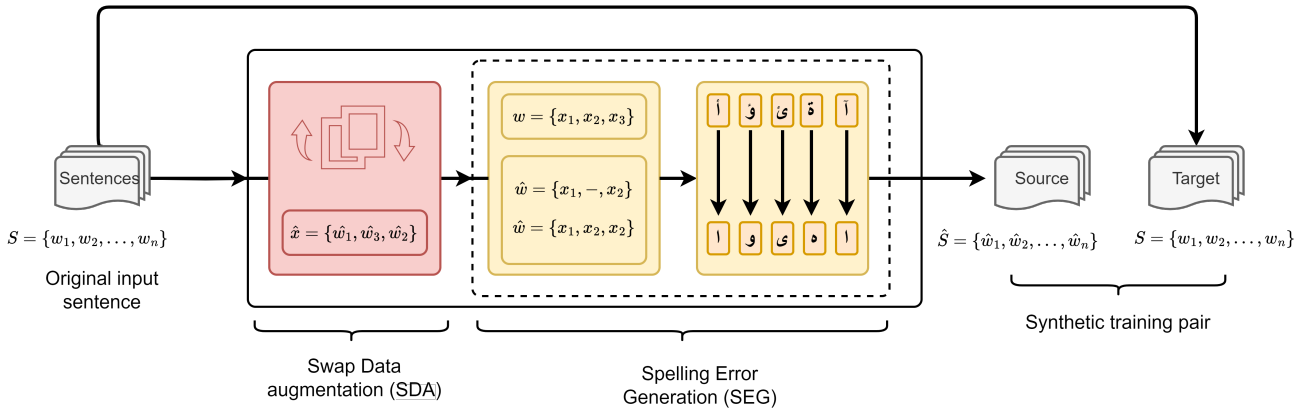

Fig. 2. Architecture Overview of the Proposed Noise Method. The Red Box Indicates SDA and the Yellow Represents SEG.

Exploring the complexities of the noise method, Algorithm 1 outlines all stages and is illustrated in Figure 2. It is worth noting that this dataset will be available for a broad spectrum of uses, including academic investigations, commercial ventures, and initiatives spearheaded by governmental agencies.

C. GEC model architecture

We utilize CapsNet, a collection of artificial neurons that effectively extracts features from input data in various contexts [24]. Recently, [9] introduced an EM routing algorithm for general-purpose applications. This algorithm activates each output capsule based on two factors: its "cost to ignore" and its "benefit to use" from earlier network layers. The success of CapsNet inspired this work to integrate it into our GEC model, which dynamically aggregates information across network layers.

In our approach, we consider the layers denoted as $H^1, ..., H^L$ within the GEC model as input. Each input capsule comprises a set of hidden states, with each state representing distinct linguistic features. The input capsules in our model propose vectors V that determine the amount of information to be transferred to the output capsules. Subsequently, the output capsules are merged to align with the dimensions of the initial hidden layer, thereby producing an enhanced representation, symbolized as $O = \Omega^1, ..., \Omega^N$. This enhancement improves CapsNet's ability to extract rich contextual representations from multi-layered input data, making it a valuable technique for various NLP tasks.

1) Dynamic Combination: A dynamic linear combination technique, inspired by [25], was incorporated. This approach allows the GEC model to generate dynamic weights, departing from the traditional practice of using fixed weights. The dynamic weights are computed using the following equation:

$$W_l = \text{FFN}_l(H^1, ..., H^L) \qquad \in \mathbb{R}^{J \times d}. \tag{3}$$

In this context, the l-th layer represents the input hidden states utilized as input for a distinct feed-forward network denoted by $\text{FFN}_l(\cdot)$, and J denotes the length of the hidden layer H^l. The output weight, which has the same dimension as H^l, is derived using the representations of the entire layer. This dynamic combination mechanism in the GEC system can accurately weigh various contextual information layers during inference, addressing a limitation of classical seq2seq GEC systems.

In the GEC model, EM routing is utilized. This technique accepts two tensors as inputs: the scores $a_i^{(inp)}$ and the capsules $\mu_{icd}^{(inp)}$. It then outputs three tensors: the scores $a_j^{(out)}$, the capsules $\mu_{jch}^{(out)}$, and the variances ${\sigma_{jch}^{(out)}}^2$. The computation of input scores $a_i^{(inp)}$ for the input capsules $\hat{H}^l$ is performed before the initiation of the routing loop as follows:

$$a_i^{(inp)} = W_l \hat{H}^l. \tag{4}$$

In this equation, W_l refers to a transformation matrix. Additionally, the weight V_{ijch} is calculated for each element (ch) in the output capsule. These weights connect the i-th input capsule to the j-th output capsule, as shown in the following equation:

$$V_{ijch} = \sum_d W_{jdh}\, \mu_{icd}^{(inp)} + B_{jch}. \tag{5}$$

However, the CapsNet loop consists of three primary stages: E-Step, D-Step, and M-Step.

E-Step: This is a critical stage in the dynamic routing-by-agreement procedure. In this stage, we determine the routing probabilities C_{ij}, which have dimensions $n^{(inp)} \times n^{(out)}$ and determine which output capsule receives the i-th input capsule. For the first iteration, we assign C_{ij} to $\frac{1}{n^{(out)}}$, where $n^{(out)}$ denotes the average count of output capsules. In the iterations that follow, we calculate C_{ij} using the following equation:

$$C_{ij} = \frac{f(a_j^{(out)}) P_{ij}}{\sum_j f(a_j^{(out)}) P_{ij}}. \tag{6}$$

Here, $a_j^{(out)}$ is the output score for the j-th output capsule, f stands for the logistic function, and P_{ij} indicates the product probability of the votes.

D-Step: This step decides the distribution of data from each i-th input capsule to each j-th output capsule, established by the following equation:

$$D_{ij}^{(use)} = f(a_i^{(inp)}) \cdot C_{ij}. \tag{7}$$

Here, f represents the logistic function, and C_{ij} are the routing probabilities. The variable $D_{ij}^{(use)}$ can range from 0 to 1. A value of 0 means that the i-th input capsule is

Algorithm 1: Noise Method (GASD)

Input: $S = \{w_1, w_2, ..., w_n\}$.
Output: $\hat{S} = \{\hat{w}_1, \hat{w}_2, ..., \hat{w}_n\}$
initialization $\hat{S} = S$;
for N *iterations* **do**
  ▷ Swap Data Augmentation (SDA)
  if $len(\hat{S}) \geq 10$ **then**
    Choose two random words in the sequence to sweep.
    $X = \{w_x, w_{x+1}, w_{x+2}, w_{x+3}\}$
    Feed X to SDA to get
    $\hat{X} = \{\hat{w}_x, \hat{w}_{x+1}, \hat{w}_{x+2}, \hat{w}_{x+3}\}$
    Update $\hat{S}$ using $\hat{X}$ considering each word indexes.
  end
  ▷ Spelling Error Generation (SEG) - spelling errors
  Select a random word in $\hat{S}$ = w_{i1}
  Decomposing word w_{i1} to array of characters = $[c_1, c_2, ..., c_n]$
  for $w_{i1} = [c_1, c_2, ..., c_n]$ **do**
    Select a random character in $[c_1, c_2, ..., c_n]$ = c_i
    Add c_i in the index of c_{i+1}
  end
  Composing $[c_1, c_2, ..., c_n]$ to $\hat{w}_{i1}$
  Select a random word in $\hat{S}$ = w_{i2}
  Decomposing word w_{i2} to array of characters = $[c_1, c_2, ..., c_n]$
  for $w_{i2} = [c_1, c_2, ..., c_n]$ **do**
    Select a random character in $[c_1, c_2, ..., c_n]$ = c_i
    Delete c_i
  end
  Composing $[c_1, c_2, ..., c_n]$ to $\hat{w}_{i2}$
  Overwrite $\hat{w}_{i1}$ and $\hat{w}_{i2}$ to the corresponding indexes in $\hat{S}$.
  Update $\hat{S}$
  ▷ Spelling Error Generation (SEG) - character-based normalization
  $Ch = \{$ aa, ee, ah, eh, oo, ah, ea $\}$, $\hat{Ch} = \{$ ah, ah, ah, ee, oo, h, ee $\}$
  if $\hat{S}$ *include a word that has a character in* Ch **then**
    for $1 \rightarrow 5$ **do**
      Pick a word w_i in $\hat{S}$ that has a character c_i in Ch.
      Replaces c_i with the corresponding character in $\hat{Ch}$ to get $\hat{w}_i$.
      Overwrite $\hat{w}_i$ with in $\hat{S}$ in the index of w_i
      Update $\hat{S}$
    end
  end
end

completely ignored by the j-th output capsule, while a value of 1 indicates that the j-th output capsule fully employs the i-th input capsule. The function f maps real numbers from the range of [-∞, ∞] to the interval [0, 1]. Hence, the values of $D_{ij}^{(use)}$ are always limited by $f(a_i^{(inp)})$, as shown in the subsequent equation:

$$0 \leq D_{ij}^{(use)} \leq f(a_i^{(inp)}) \leq 1. \tag{8}$$

The proportion of disregarded data $D_{ij}^{(ign)}$ for each i-th input capsule is calculated using the following equation:

$$D_{ij}^{(ign)} = f(a_i^{(inp)}) - D_{ij}^{(use)}. \tag{9}$$

$D_{ij}^{(use)}$ and $D_{ij}^{(ign)}$ are calculated for all capsules, including both input and output. Additionally, $1 - f(a_i^{(inp)})$ is factored in, representing the data that is effectively "gated off" by the logistic function f.

$$D_{ij}^{(use)} + D_{ij}^{(ign)} + (1 - f(a_i^{(inp)})) = 1. \tag{10}$$

M-Step: The last phase in each iteration fulfills two functions, including calculating the output scores of $a_j^{(out)}$, which are means $\mu_{jch}^{(out)}$, and variances ${\sigma_{jch}^{(out)}}^2$, representing the optimized output capsule.

D. Training approach

A bidirectional training approach was introduced to overcome the challenge of exposure bias in the GEC task, which often results in unsatisfactory corrections and poor predictions for prefixes and suffixes. This approach aims to improve alignment between two GEC models by incorporating the Kullback-Leibler (KL) divergence into the training process. This term, denoted as D_{kl}, measures the difference in probabilities between the models that read R2L and those that read from L2R. The updated training objective is designed to optimize the conventional likelihood for both models individually. This optimization is achieved through the use of EM routing, and the bidirectional approach strives to minimize the divergence between the two models, ensuring consistent results.

In simple terms, the R2L and L2R GEC models process text sequences in different directions. Despite these distinct approaches, both aim to compute the probability of a corrected output. Each model analyzes the sequences individually, but the ultimate objective is to predict the same probability. Ideally, both models should yield identical probability outputs, as shown in the following equation:

$$\log P(y|x; \overleftarrow{\theta}) = \log P(y|x; \overrightarrow{\theta}). \tag{11}$$

Achieving Equation (11) is challenging when the models are independently trained using Maximum Likelihood Estimation (MLE). To address this issue, a divergence regularization term D_{kl} is introduced into the MLE training objective of GEC as follows:

$$L(\overleftarrow{\theta}) = \sum_{n=1}^{N} \log P(y^n|x^n; \overleftarrow{\theta}) + \lambda \sum_{n=1}^{N} D_{kl}(P(y|x^n; \overleftarrow{\theta})||P(y|x^n; \overrightarrow{\theta})). \tag{12}$$

In this equation, the first term $\sum_{n=1}^{N} \log P(y^n|x^n; \overleftarrow{\theta})$ refers to the conventional loss of the R2L model, and λ is the model hyper-parameter. A higher value of λ emphasizes reducing the divergence between the two models. If Equation (11) is fulfilled, the regularization term reduces to zero, indicating that both models are aligned and their predictions are consistent. If this is not the case, the training process continues to minimize the divergence between the two models, ensuring alignment and leading to more reliable predictions. However, any deficiencies in the L2R model could destabilize the balance between these models and affect the training process of the R2L model. This is because the L2R model influences the R2L model's training through the regularization term in the objective function. Conversely, the balance maintained between the two models enables the L2R model to detect and dismiss any erroneous output generated by the R2L model, a crucial aspect of the training process for ensuring prediction accuracy and reliability. Accordingly, the L2R model's training objective is defined as:

$$L(\overrightarrow{\theta}) = \sum_{n=1}^{N} \log P(y^n|x^n; \overrightarrow{\theta}) + \lambda \sum_{n=1}^{N} D_{kl}(P(y|x^n; \overleftarrow{\theta})||P(y|x^n; \overrightarrow{\theta})). \quad (13)$$

The two models operate as supplementary systems in a combined training procedure, providing mutual support through regularization. The D_{kl} term steers the training process to a point where no additional improvements are possible, minimizing discrepancies and improving both models. The training objective is defined as the sum of both objectives:

$$L(\theta) = L(\overleftarrow{\theta}) + L(\overrightarrow{\theta}). \quad (14)$$

The overall training process begins with pre-training both models using synthetic data. Following this initial phase, the models proceed with iterative and coordinated training. During the RTL model's training, the MLE and EM routing methods are utilized to optimize the L2R model. The process is then reversed to fine-tune the R2L model, as detailed in Equation 14.

V. Experimental details

A. Training data

The training dataset comprises both authentic and synthetic collections. The data source is the monolingual Arabic news corpus OSIAN [26], which includes a total of 15 million sentences. The synthetic dataset was created using the noise method introduced in Section IV-B. This method facilitated the production of synthetic data sufficient for training a neural-based GEC model. The training set consisted of 14 million pairs, while the development set included 210,000 pairs. These sentence pairs provided comprehensive training examples for both models. For fine-tuning, the QALB-2014 dataset was utilized. To address the challenge of rare and unknown words, the BPE method was employed, decomposing unknown words into sub-words [27]. This preprocessing step enhanced the model's ability to handle a wider vocabulary, further improving its performance.

B. Model settings

The experiments utilized a tailored GEC model built upon the Transformer architecture as referenced in [6]. This model was customized in various aspects to meet specific needs. Initially, the model size was reduced from 512 to 256, and the layer count decreased from 6 to 4, while maintaining the head attention at 8 heads as in the original model. Learned positional encoding, as applied in BERT [14], was chosen for positional encoding. This approach enhances the model's ability to understand sequential dependencies in the data, leading to improved performance. During the training stages, learning rates were set at 1e-3 and 1e-1. Similar to the BERT model, label smoothing was not incorporated, based on experimental results showing no significant improvement in performance. Training involved an initial iteration of 25 epochs using synthetic data, followed by 15 epochs with the QALB-2014 training set, saving checkpoints after each epoch to preserve the best-performing versions of the models. To enhance performance and prevent overfitting, gradient clipping with a threshold of 1.0 and dropout with a probability of 0.15 were applied, ensuring manageable gradients and minimized instability. Beam search with a beam size of five was utilized to create synthetic examples, enabling the model to explore various possibilities and choose optimal examples for estimating the Kullback-Leibler (D_{KL}) divergence. During training and testing, a maximum sentence length of 500 tokens was imposed. All models were trained on a system equipped with two TITAN RTX GPUs and Python 3.6, utilizing CUDA 10.2 Production for optimized computing efficiency.

C. Evaluation

To evaluate the models, the test sets from the QALB-2014 and QALB-2015 shared tasks were used, comprising 948 and 920 sentences, respectively. Each pair consists of a sentence with grammatical errors and its corrected version. The MaxMatch algorithm [28] was used to calculate word-level edits for every corrected sentence by comparing it with the respective golden target sentence. The assessment was conducted using precision, recall, and the $F1$ score, which are established standards for measuring effectiveness in the GEC domain. These metrics provided insights into the model's accuracy in correcting errors and its ability to produce outputs that closely align with the reference sentences.

VI. Results and Discussion

The evaluation initially focused on the QALB-2014 dataset, using the BPE technique to address the challenges of rare and unknown words. The results showed that the R2L model achieved an F1 score of 60.90, while the L2R model secured an F1 score of 57.77, which means that the R2L model performed better in correcting the Arabic language's right-to-left script. Table I presents the precision, recall, and F2 scores for both models, highlighting a 3.13-point difference in their F1 scores.

A. Synthetic Data

To assess the effectiveness of GASD in generating a reliable synthetic dataset for GEC, the models were pre-trained

TABLE I
PERFORMANCE OF THE GEC TRANSFORMER-BASED MODELS UTILIZING BPE.

GEC	Precision	Recall	F_1
GEC-L2R	72.53	48.01	57.77
GEC-R2L	73.06	52.22	60.90

using the constructed data. The synthetic data improved the performance of both models. As shown in Table II, the models achieved an increase of 8.13 and 6.35 points in the $F1$ score, respectively. These results validate the effectiveness of the proposed method in producing reliable synthetic data. However, additional efforts are required to further expand the training dataset for Arabic GEC.

TABLE II
RESULTS OF THE CONSTRUCTED DATA OVER THE TWO GEC MODELS.

GEC	Precision	Recall	F_1
GEC-L2R + Pre-training	77.53	57.31	65.90
GEC-R2L + Pre-training	78.51	58.83	67.25

B. Dynamic Combination

The influence of dynamic combination and CapsNet on GEC was investigated. CapsNet is known for its ability to capture hierarchical relationships within data. The critical parameter of CapsNet, the number of iterations, was initially set to its original value of 3.

The work of [9] was adapted to allow capsule networks to accept variable input lengths, which is particularly advantageous for NLP tasks. Incorporating dynamic combinations into the model improved the agreement estimation during routing, leading to a performance boost for the pre-trained models. The F_1 score increased by 1.03 for the L2R model and 1.15 for the R2L model, as shown in Table III. These results underscore the advantages of applying CapsNet to GEC systems. However, it is important to note that the introduction of CapsNet led to an increase in the number of model parameters during training, necessitating a longer training duration. The findings substantiate the potential of dynamic combination as enhancements for CapsNet GEC models, resulting in improved accuracy without significantly increasing model complexity. Further research can explore the fine-tuning of CapsNet parameters to optimize its performance in NLP tasks.

TABLE III
THE IMPACT OF THE DYNAMIC COMBINATION ON TWO DIFFERENT VERSIONS OF OUR ARABIC GEC MODEL.

GEC	Precision	Recall	F_1
GEC L2R + Pre-training + EM-Routing	77.47	58.92	66.93
GEC R2L + Pre-training + EM-Routing	79.13	60.24	68.40

C. Impact of the bidirectional training approach

During training, both models were compared based on their probabilities to guide the process. The results are presented in Table IV. After the second iteration, the agreement between the models improved, and the performance on the development set stabilized. This indicates that two iterations were sufficient to achieve consistency between the models in this case.

This finding highlights the effectiveness of the introduced regularization term in addressing exposure bias, thereby enhancing the performance of each model. Additionally, the difference in F1 scores between the two models decreased from 3.13 to 0.24 points. This indicates that the regularization term successfully promotes model agreement and coherence. To better understand the influence of the bidirectional training approach, Table IV provides detailed information. Integrating regularization terms in the bidirectional training approach is valuable in mitigating exposure bias issues and improving model performance. However, it is important to note that the iterative training approach doubles the training time compared to MLE. Despite this increase in computational cost, the resulting performance improvements justify the additional time required.

D. Multi-Pass Error Correction

Correcting sentences with multiple grammatical errors is challenging due to the inherent complexity of human language. Single-pass inference methods often fail to handle the intricate details of such sentences effectively. Recognizing this challenge, [5] introduced the concept of multi-round correction to enhance the accuracy of GEC systems. This approach entails iteratively correcting a sentence in multiple rounds, improving its fluency and grammatical correctness.

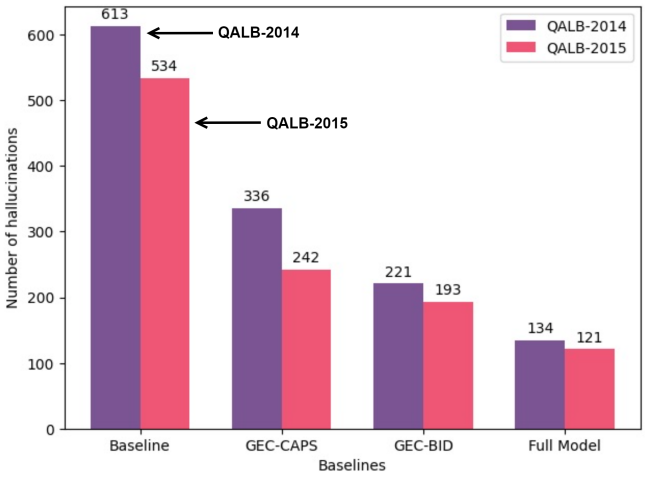

Fig. 5. Impact of Hallucinations Across Two Benchmarks: Comparison of GEC-CAPS (GEC R2L + CapsNet), GEC-BID (GEC R2L + Bidirectional), and GEC Full Model (GEC R2L + CapsNet + Bidirectional).

The multi-pass correction paradigm was applied and adapted to our GEC models. Specifically, GEC-R2L initiates the process by taking the input sentence S and producing an intermediate correction denoted as $\bar{S_1}$ through a single-round correction. However, $\bar{S_1}$ is not considered the final output; instead, an additional layer of correction is introduced. $\bar{S_1}$ is fed into GEC-L2R, which performs a subsequent round of correction, ultimately generating the system output correction $\bar{S_2}$.

To evaluate the efficiency of this multi-pass correction approach, two comparison baselines were set up. For the

TABLE IV
PERFORMANCE OF OUR GECS AFTER TWO ITERATIONS OF COMBINED TRAINING USING THE REGULARIZATION TERM TECHNIQUE. THE TERM "FULL MODEL" REFERS TO THE MODE THAT IS PRE-TRAINED WITH CAPSNET.

GEC model		Precision	Recall	F_1
1st iteration	GEC L2R + (full model)	79.13	59.11	67.67
	GEC R2L + (full model)	79.29	61.29	69.13
2nd iteration	GEC L2R + (full model)	79.22	61.27	69.09
	GEC R2L + (full model)	79.69	61.41	69.33

TABLE V
COMPARISON OF BLEU SCORES FOR THE PROPOSED GEC SYSTEM AND BASELINE TRANSFORMER-BASED ACROSS DIFFERENT SENTENCE LENGTHS.

Sentence Length	BLEU-unigram		BLEU-bigram		BLEU-trigram		BLEU-four-gram	
	GEC	Baseline	GEC	Baseline	GEC	Baseline	GEC	Baseline
05 - 19	54.3 ± 2.4	47.7 ± 2.2	56.7 ± 2.1	49.6 ± 2.3	67.9 ± 2.1	58.3 ± 2.8	77.6 ± 1.3	68.7 ± 1.9
20 - 39	66.4 ± 2.8	56.7 ± 2.9	69.2 ± 2.7	58.9 ± 2.9	76.2 ± 1.2	63.8 ± 2.5	79.7 ± 2.1	70.6 ± 2.8
40 - 59	77.1 ± 2.4	66.8 ± 2.8	79.4 ± 2.4	69.2 ± 2.8	82.3 ± 1.4	71.7 ± 2.1	85.2 ± 1.3	74.8 ± 2.4
60 - 79	78.2 ± 0.8	69.4 ± 1.4	81.0 ± 2.4	73.5 ± 1.5	86.7 ± 2.0	75.0 ± 1.9	88.9 ± 2.3	78.2 ± 1.5
80 - 99	82.3 ± 1.1	73.6 ± 2.1	86.2 ± 2.6	77.4 ± 1.9	89.3 ± 1.7	80.7 ± 2.3	94.2 ± 1.8	83.6 ± 2.1
$\geq$ 100	**85.1 ± 2.1**	76.7 ± 2.4	**88.3 ± 1.6**	78.3 ± 2.5	**94.3 ± 2.2**	81.5 ± 0.9	**96.3 ± 2.3**	83.9 ± 1.6

TABLE VI
PERFORMANCE OF EMPLOYING A MULTI-PASS CORRECTION APPROACH IN GECS.

GEC Model	Precision	Recall	F_1
L2R ⇒ R2L	78.28	65.94	71.58
R2L ⇒ L2R	78.64	66.13	71.84

TABLE VII
PERFORMANCE OF RERANKING TECHNIQUE FOR GEC MODEL USING TWO BENCHMARKS.

Benchmark	Precision	Recall	F_1
QALB 2014	80.14	67.20	73.10
QALB 2015	81.35	70.43	75.49

TABLE VIII
EVALUATING THE PERFORMANCE OF THE PROPOSED MODEL AGAINST THE EXISTING MODELS USING TWO BENCHMARKS, AND (+/-) REFERS TO A DECREASE AND INCREASE IN THE SCORE.

Systems	2014		2015	
	F_1	△	F_1	△
Rozovskaya [29]	67.91	+ 5.19	N/A	-
Nawar [30]	N/A	-	72.87	+ 2.62
Ahmadi [20]	50.34	+ 22.76	N/A	-
Watson [19]	70.39	+ 2.71	73.19	+ 2.30
Aiman [22]	70.91	+ 2.19	N/A	-
Solyman [1]	71.03	+ 2.07	73.52	+ 1.97
Mahmoud [31]	71.51	+ 1.59	74.03	+ 1.46
Our GEC Model	**73.10**		**75.49**	

first baseline, the output from GEC-R2L is used as input for GEC-L2R. Conversely, in the second baseline, the correction sequence is inverted by using the output from GEC-L2R as input for GEC-R2L. As summarized in Table VI, the results indicate an improvement in both recall and F_1 score across both baselines, with an average increase of 5 points. However, it is noteworthy that precision decreased by 0.94 and 1.05 points for GEC-L2R and GEC-R2L, respectively. This decrease in precision could potentially be attributed to a delicate balance issue between the two correction models, as highlighted in previous literature [32].

Incorporating multi-pass error correction into GEC models is a promising strategy for enhancing the correction of sentences in low-resource languages. It boosts recall and F_1 score, though at the expense of a decrease in precision. This highlights the need for further research into addressing model agreement and balance concerns in multi-pass correction scenarios.

E. Re-ranking approach

To mitigate the variance between both models and the reduction in precision, a reranking approach was applied to the GEC L2R outputs. This technique is based on the work of [33], where using the L2R model during the decoding stage improved NMT tasks.

Initially, three distinct GEC models were trained for both the L2R and R2L directions. Following this, the GEC models in the R2L direction were used to generate a list of top n-best candidate corrections, each with its corresponding conditional probability score. The candidates were subsequently assessed by the three GEC models operating in the L2R direction, with each model assigning a score to the candidates. Ultimately, the R2L n-best candidates were reordered based on the combined scores from both models.

Table VII presents the evaluation of the proposed GEC model with the reranking L2R approach, utilizing the QALB-2014 and QALB-2015 test sets. The results demonstrate improvements in precision and F1 score relative to previous experiments. This enhancement is attributed to the collaborative exploration between GEC models in R2L and L2R directions, which effectively mitigates exploration bias and promotes a more comprehensive search for correction candidates.

F. Impact of hallucinations

In this subsection, the influence of the proposed system on minimizing hallucinations when utilizing minimal training data is explored. In GEC, hallucinations manifest as either repeated segments or entirely disjointed syntax. It is posited that models trained with the proposed technique generate

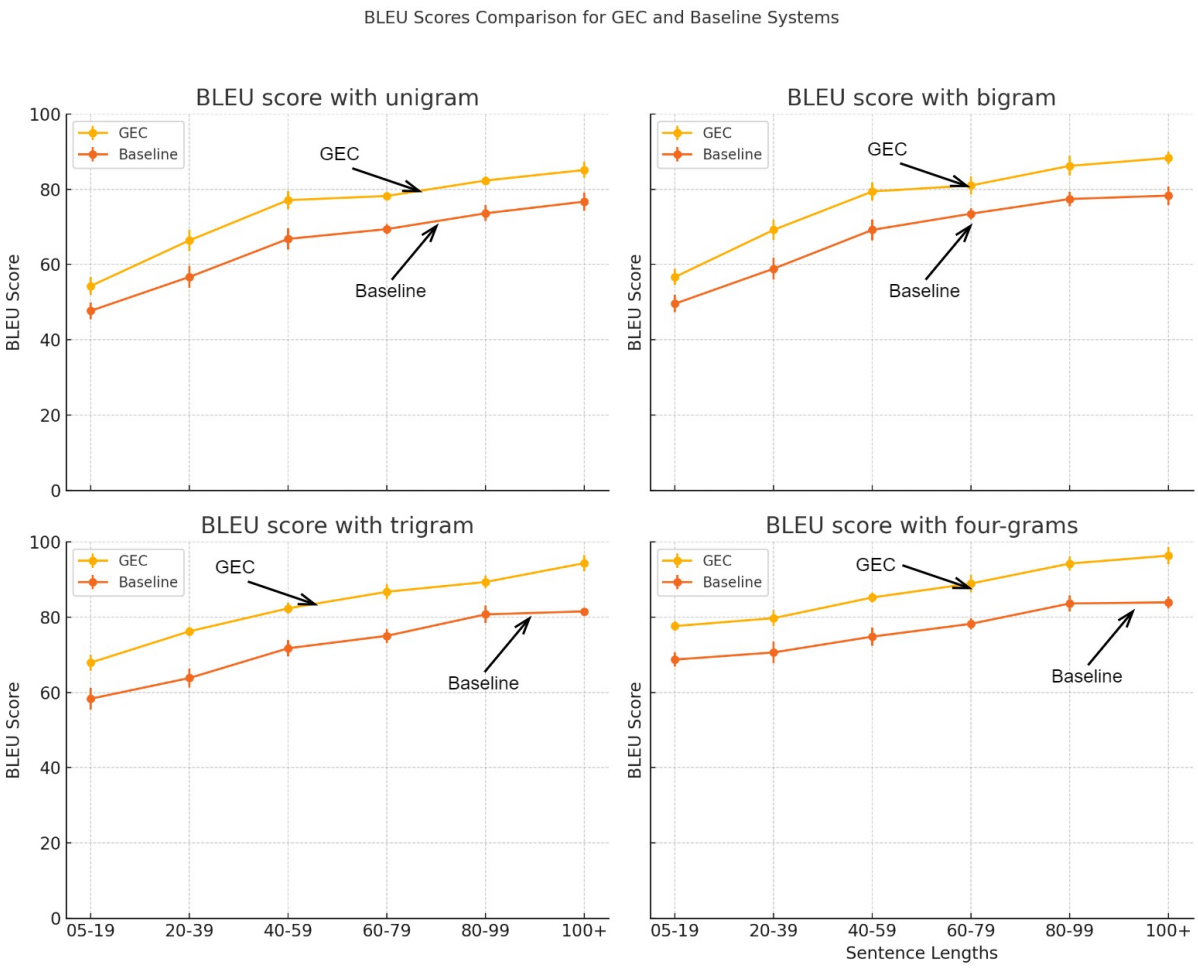

Fig. 3. Illustration of BLEU Scores for GEC and Baseline (Transformer-based) Across Different Sentence Lengths and N-gram Settings.

fewer hallucinations. To demonstrate this, hallucinations were analyzed using the L2-2015 dataset [34]. The dataset consists of 311 sentences for the training set and 155 for the validation set. It also exhibits a high rate of errors per sentence, as it was collected from students learning Arabic as a second language. This high error rate in the dataset renders it well-suited for training and testing the model on hallucinations. To quantify hallucinations, the method outlined in [35] was employed. This approach initially creates training examples with hallucinations by inserting irrelevant tokens into input sentences. Subsequently, it utilizes the BLEU score to assess the presence of hallucinations in the sentences. This method was adapted in GEC to determine the frequency of hallucinations in system outputs. Specifically, the BLEU score calculation was modified to focus on the precision of unigrams with a weight of 0.8 and bigrams with a weight of 0.2. An output was classified as a hallucination if its BLEU score fell below 25.

The influence of hallucinations was evaluated using several models: the baseline (Transformer-based), GEC R2L with CapsNet, GEC R2L with Bidirectional structure, and the full model of GEC R2L with both CapsNet and Bidirectional structure. This evaluation focused solely on sentences relevant to hallucinations. Therefore, the frequency of sentences containing hallucinations was calculated for each respective system, and Figure 5 shows the frequency of hallucinations over QALB-2014 and QALB-2015. A shorter bar signifies better performance. As depicted in Figure 5, the full model reported the lowest frequency of hallucinations. This result underscores the efficacy of the proposed model in minimizing hallucinations in GEC systems.

G. BLEU score with multiple grams

In this subsection, the performance of the proposed GEC system was evaluated against a Transformer-based baseline framework using the QALB-2014 benchmark dataset. Sentences from the dataset were grouped by length into six categories: 05-19, 20-39, 40-59, 60-79, 80-99, and 100+ words. For each length category, BLEU scores with n-gram settings ranging from 1 to 4 (Unigram to Four-gram) were computed. The BLEU score, a metric for evaluating the quality of text generated by a model, was used to quantify the performance of both systems. Scores were calculated for unigrams, bigrams, trigrams, and four-grams to assess accuracy at different levels of text granularity. As shown in Table V, each BLEU score is presented with its respective standard deviation to indicate variability in performance. The results demonstrate that the proposed GEC system outperforms the baseline across all sentence lengths and n-gram settings, highlighting its effectiveness in correcting grammatical errors in texts.

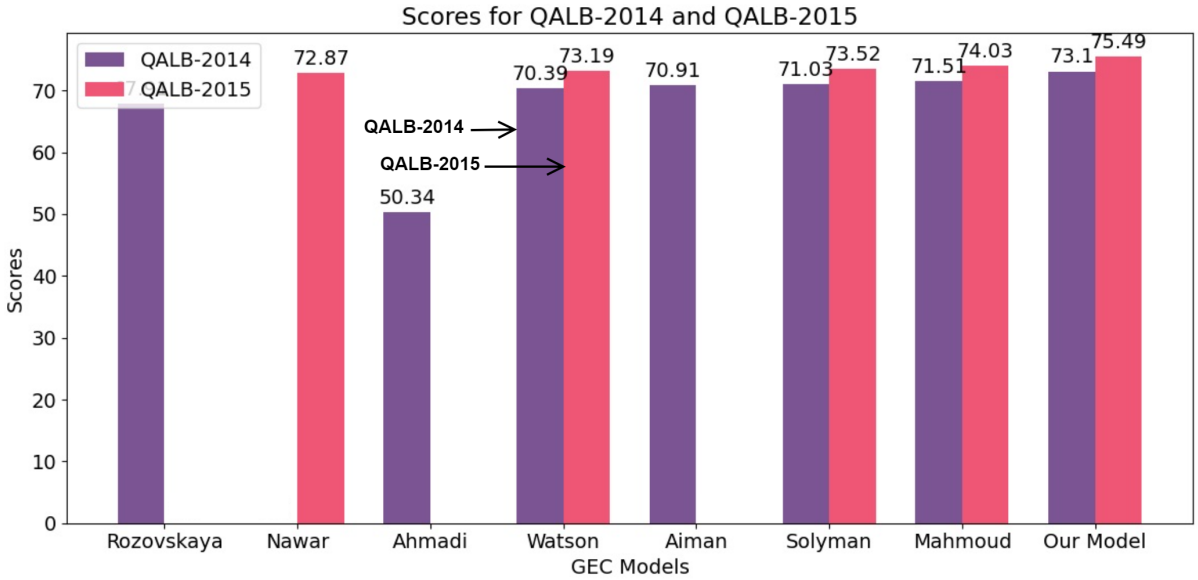

Fig. 4. Illustration of F1 Scores for Top and Recent Models in Arabic GEC Over Two Benchmarks.

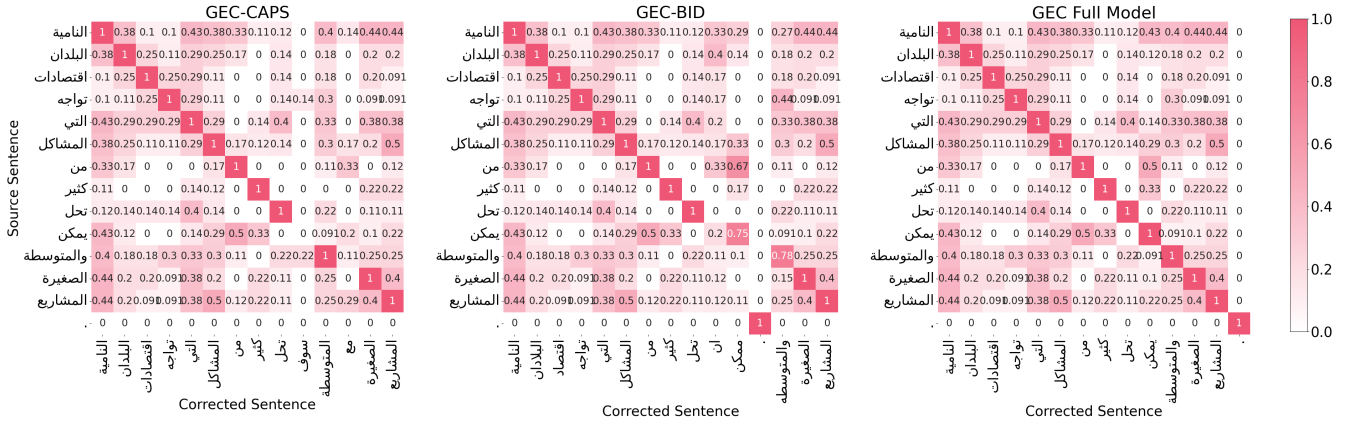

Fig. 6. Comparative analysis of correction outputs from GEC-CAPS, GEC-BID, and GEC Full Mode Models Against a Golden Target Sentence.

H. Case study

Figure 6 compares the outputs of three different versions of the GEC model: GEC-CAPS, GEC-BID, and the GEC Full Model, against a golden target sentence. Each model's output is presented in a separate column, and the corrected sentences are shown in separate rows. The differences between the model outputs and the golden target sentence are highlighted in pink, with varying shades indicating the degree of alignment with the target. Darker shades represent higher alignment scores, while lighter shades indicate lower ones. The GEC Full Model outperforms the other two models, followed by GEC-BID and then GEC-CAPS. This comparison visually represents how these different GEC models perform against a standard reference.

The most advanced model, which incorporates GEC R2L, CapsNet, Bid-agreement, and reordering of L2R, yielded promising results. It has outperformed the top systems in Arabic GEC. Table VIII provides a comprehensive statistical evaluation of the top and recent Arabic GEC models. The proposed model demonstrated strong performance by achieving the highest scores for both QALB-2014 and QALB-2015 test sets. This system represents a promising solution for automatic grammar correction in Arabic. It delivers consistent results and demonstrates its efficiency and effectiveness.

VII. Conclusion

This paper introduced an automatic grammar correction model designed for languages with limited resources, focusing on correcting grammatical and spelling errors in Arabic. The model was built on a seq2seq Transformer-based framework and addressed the challenge of limited training data by generating parallel data from an out-of-domain corpus, expanding the dataset to 14.21 million parallel examples. This size surpasses existing Arabic GEC datasets. CapsNet was integrated to identify complex linguistic patterns to improve the model's ability to correct errors. A regularization term was added to address the exposure bias problem and enhance consistency between the two versions of the GEC model (R2L and L2R). This term uses Kullback–Leibler divergence to determine the difference between two probability distributions. This training approach allows each model to iteratively evaluate and strengthen the other, leading to improved performance. The results confirm the effectiveness of CapsNet and the bidirectional training approach for GEC systems in limited resource scenarios. The proposed model

outperforms all current systems in Arabic GEC, achieving the highest scores in two benchmarks.

Future work aims to extend the model to other languages with limited resources in the GEC domain, utilizing both CapsNet and the bidirectional training approach. Additionally, CapsNet's capabilities in other NLP tasks will be investigated.

References

[1] A. Solyman, M. Zappatore, W. Zhenyu, Z. Mahmoud, A. Alfatemi, A. O. Ibrahim, and L. A. Gabralla, "Optimizing the impact of data augmentation for low-resource grammatical error correction," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 6, p. 101572, 2023.

[2] A. Musyafa, Y. Gao, A. Solyman, C. Wu, and S. Khan, "Automatic correction of indonesian grammatical errors based on transformer," *Applied Sciences*, vol. 12, no. 20, 2022.

[3] Z.-Y. Dou, Z. Tu, X. Wang, S. Shi, and T. Zhang, "Exploiting deep representations for neural machine translation," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 4253–4262.

[4] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *3rd International Conference on Learning Representations, ICLR 2015*, vol. 1409–0473, 2015.

[5] T. Ge, F. Wei, and M. Zhou, "Fluency boost learning and inference for neural grammatical error correction," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Jul. 2018, pp. 1055–1065.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, p. 6000–6010.

[7] Q. Tong, S. Yue, J. Zhang, Y. Liu, and Z. Han, "A seinier cyber public opinion propagation prediction model with extreme emotion mechanism." *IAENG International Journal of Computer Science*, vol. 51, no. 5, pp. 477–488, 2024.

[8] H. Yang, L. Wang, and Y. Yang, "Named entity recognition in electronic medical records incorporating pre-trained and multi-head attention," *IAENG International Journal of Computer Science*, vol. 51, no. 4, pp. 401–408, 2024.

[9] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 3859–3869.

[10] M. Junczys-Dowmunt and R. Grundkiewicz, "Phrase-based machine translation is state-of-the-art for automatic grammatical error correction," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Nov. 2016, pp. 1546–1556.

[11] S. Chollampatt and H. T. Ng, "A multilayer convolutional encoder-decoder neural network for grammatical error correction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018, pp. 7913–7920.

[12] R. Grundkiewicz, M. Junczys-Dowmunt, and K. Heafield, "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*. Association for Computational Linguistics, Aug. 2019, pp. 252–263.

[13] F. Yu, D. Wang, E. Shelhamer, and T. Darrell, "Deep layer aggregation," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2403–2412.

[14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "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, Volume 1*, Jun. 2019, pp. 4171–4186.

[15] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 1877–1901.

[16] Y. Belinkov and Y. Bisk, "Synthetic and natural noise both break neural machine translation," in *6th International Conference on Learning Representations, ICLR 2018*, vol. abs/1711.02173, 2018, pp. 27–40.

[17] K. N. Acheampong and W. Tian, "Toward perfect neural cascading architecture for grammatical error correction," *Applied Intelligence*, vol. 51, pp. 3775–3788, 2021.

[18] N. Hossain, M. H. Bijoy, S. Islam, and S. Shatabda, "Panini: a transformer-based grammatical error correction method for bangla," *Neural Computing and Applications*, vol. 36, no. 7, pp. 3463–3477, 2024.

[19] D. Watson, N. Zalmout, and N. Habash, "Utilizing character and word embeddings for text normalization with sequence-to-sequence models," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Oct.-Nov. 2018, pp. 837–843.

[20] S. Ahmadi, "Attention-based encoder-decoder networks for spelling and grammatical error correction," Ph.D. dissertation, PARIS DESCARTES UNIVERSITY, 2017.

[21] A. Solyman, Z. Wang, and Q. Tao, "Proposed model for arabic grammar error correction based on convolutional neural network," in *2019 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)*, 2019, pp. 1–6.

[22] A. Solyman, W. Zhenyu, T. Qian, A. A. M. Elhag, M. Toseef, and Z. Aleibeid, "Synthetic data with neural machine translation for automatic correction in arabic

grammar," *Egyptian Informatics Journal*, vol. 22, no. 3, pp. 303–315, 2021.

[23] M. H. kawakib, N. M. Hussien, and Y. M. Mohialden, "A pyarabic python library to create arabic applications," *Webology*, vol. 19, no. 5, 2022.

[24] G. E. Hinton, S. Sabour, and N. Frosst, "Matrix capsules with EM routing," in *International Conference on Learning Representations*, vol. 6, 2018, pp. 88–96.

[25] Z.-Y. Dou, Z. Tu, X. Wang, L. Wang, S. Shi, and T. Zhang, "Dynamic layer aggregation for neural machine translation with routing-by-agreement," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, Jul. 2019, pp. 86–93.

[26] I. Zeroual, D. Goldhahn, T. Eckart, and A. Lakhouaja, "OSIAN: Open source international Arabic news corpus - preparation and integration into the CLARIN-infrastructure," in *Proceedings of the Fourth Arabic Natural Language Processing Workshop*. Association for Computational Linguistics, Aug. 2019, pp. 175–182.

[27] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Aug. 2016, pp. 1715–1725.

[28] D. Dahlmeier and H. T. Ng, "Better evaluation for grammatical error correction," in *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Jun. 2012, pp. 568–572.

[29] A. Rozovskaya, N. Habash, R. Eskander, N. Farra, and W. Salloum, "The Columbia system in the QALB-2014 shared task on Arabic error correction," in *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*. Association for Computational Linguistics, Oct. 2014, pp. 160–164.

[30] M. Nawar, "CUFE@QALB-2015 shared task: Arabic error correction system," in *Proceedings of the Second Workshop on Arabic Natural Language Processing*. Association for Computational Linguistics, Jul. 2015, pp. 133–137.

[31] Z. Mahmoud, C. Li, M. Zappatore, A. Solyman, A. Alfatemi, A. O. Ibrahim, and A. Abdelmaboud, "Semi-supervised learning and bidirectional decoding for effective grammar correction in low-resource scenarios," *PeerJ Computer Science*, vol. 9, p. e1639, 2023.

[32] L. Liu, M. Utiyama, A. Finch, and E. Sumita, "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*. Association for Computational Linguistics, Jun. 2016, pp. 411–416.

[33] R. Sennrich, B. Haddow, and A. Birch, "Improving neural machine translation models with monolingual data," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Aug. 2016, pp. 86–96.

[34] A. Rozovskaya, H. Bouamor, N. Habash, W. Zaghouani, O. Obeid, and B. Mohit, "The second QALB shared task on automatic text correction for Arabic," in *Proceedings of the Second Workshop on Arabic Natural Language Processing*. Association for Computational Linguistics, Jul. 2015, pp. 26–35.

[35] V. Raunak, A. Menezes, and M. Junczys-Dowmunt, "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*. Association for Computational Linguistics, Jun. 2021, pp. 1172–1183.