IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 26 June 2023, accepted 10 July 2023, date of publication 13 July 2023, date of current version 19 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3294979

RESEARCH ARTICLE

Supervised Copy Mechanism for Grammatical Error Correction

KAMAL AL-SABAHI[1] **AND KANG YANG**[2]
[1]College of Computing and Information Sciences, University of Technology and Applied Sciences-Ibra, Ibra 400, Oman
[2]Key Laboratory of Software Engineering for Complex Systems, College of Computer, National University of Defense Technology, Changsha 410073, China

Corresponding author: Kang Yang (yangkang@nudt.edu.cn)

ABSTRACT AI has introduced a new reform direction for traditional education, such as automating Grammatical Error Correction (GEC) to reduce teachers' workload and improve efficiency. However, current GEC models still have flaws because human language is very variable, and the available labeled datasets are often too small to learn everything automatically. One of the key principles of GEC is to preserve correct parts of the input text while correcting grammatical errors. However, previous sequence-to-sequence (Seq2Seq) models may be prone to over-correction as they generate corrections from scratch. Over-correction is a phenomenon where a grammatically correct sentence is incorrectly flagged as containing errors that require correction, leading to incorrect corrections that can change the meaning or structure of the original sentence. This can significantly reduce the accuracy and usefulness of GEC systems, highlighting the need for improved approaches that can reduce over-correction and ensure more accurate and natural corrections. Recently, sequence tagging-based models have been used to mitigate this issue by only predicting edit operations that convert the source sentence to a corrected one. Despite their good performance on datasets with minimal edits, they struggle to restore texts with drastic changes. This issue artificially restricts the type of changes that can be made to a sentence and does not reflect those required for native speakers to find sentences fluent or natural sounding. Moreover, sequence tagging-based models are usually conditioned on human-designed language-specific tagging labels, hindering generalization and the real error distribution generated by diverse learners from different nationalities. In this work, we introduce a novel Seq2Seq-based approach that can handle a wide variety of grammatical errors on a low-fluency dataset. Our approach enhances the Seq2Seq architecture with a novel copy mechanism based on a supervised attention approach. Instead of merely predicting the next token in context, the model predicts additional correctness-related information for each token. This auxiliary objective propagates into the weights of the model during training without requiring extra labels at testing time. Experimental results on benchmark datasets show that our model achieves competitive performance compared to state-of-the-art(SOTA) models.

INDEX TERMS Supervised attention, supervised copy mechanism, grammatical error correction, sequence-to-sequence.

I. INTRODUCTION

Grammatical error correction refers to the process of identifying and correcting errors in written texts that violate the rules of grammar. These errors can range from simple mistakes in spelling, punctuation, and capitalization to more complex errors involving the use of verb tense, subject-verb agreement, sentence structure, and word choice [1]. In all GEC tasks, there are two objectives: first, to identify the errors and correct them with high accuracy. Second, to keep the already correct tokens as they are. Several approaches have been proposed ranging from statistical to neural network-based ones and from Seq2Seq to sequence tagging models. Despite GEC being intensively studied for years, the best models are still far from perfect. The Seq2Seq models have been proven to be effective in machine translation(MT) [2]. Due to the similarity between MT and GEC, an encoder-decoder model can be used for the latter as well. In which

The associate editor coordinating the review of this manuscript and approving it for publication was Anandakumar Haldorai.

the encoder is used to encode the erroneous sentence and the decoder generates the correct output sentence. However, there are still some issues in GEC Seq2Seq-based models. As studied by previous works [3], the frequent repetition and omission of tokens occur in the Seq2Seq generation process as a result of generating the sequence from scratch. More importantly, there is no guarantee that the generated sentences can keep all the original correct words while maintaining the semantic structures [4].

Recently, sequence tagging methods [5], [6], [7] consider GEC as a text editing task in which a set of edits would be predicted and applied to convert a source sentence to a corrected one, therefore bypassing some of the above-mentioned problems of Seq2Seq models. This might justify the superior performance of sequence tagging-based models over the Seq2Seq-based ones. After investigating the case, we found that sequence tagging models work better when the errors are at a minimum, whereas their performance drops drastically when the original sentences are too long or contain low-frequency tokens. Moreover, when it comes to corrections that need longer insertions, most of these sequence tagging methods rely on iterative corrections, which can reduce fluency. In practice, especially with low-level learners, human correctors may rewrite some parts of the sentence to make it fluent and more natural, which may not be possible in sequence tagging-based models.

We think that a generative model is in need to handle such an issue. However, as mentioned in several related work, [3], [4], [8], and [9] Seq2Seq models can suffer from the over-correction problem that reduces their precision. To this end, we propose a new model that can leverage the powerfulness of Seq2Seq generative ability when needed and just copy the correct tokens from the source to the target otherwise. At a high level, the copy mechanism is a way to allow the model to choose between copying parts of the input sequence into the output sequence or generating from scratch. This is particularly useful when dealing with tasks such as GEC, where the input and output sequences share common subsequences [10]. During decoding, the attention mechanism can choose to attend to either the encoder's hidden states or the input sequence based on the decoder state and the output tokens generated so far [11]. The challenge with regard to copying in Seq2Seq is that new machinery is needed to decide when to generate and when to copy.

Technically, the balance between copying and generating is controlled by a balancing factor learned implicitly from the attention model during the training in an end-to-end manner [12]. However, the attention methods adopted in the existing copy mechanism models are non-parametric or trained inside the model without explicit supervision, and the attention results are poorly explained. We think that we could supervise this balancing factor to guide the model in deciding whether to generate or copy. Therefore, we propose a token-level labeling task for the source sentence and assign each token in the source sentence a label indicating whether this token is correct/incorrect. This can be used to calculate the balancing factor as a learnable switch between copying and generating that can be learned in a supervised way. The training of the balancing factor is unified with the training of the GEC model in a multi-task learning settings, both in a supervised manner.

In this way, we conjecture that the proposed model will have the following advantages: (1) Supervised copy mechanism will encourage the model to copy the already correct tokens with no change, which alleviates the over-correction problem of Seq2Seq models. (2) The proposed model will be able to handle more complicated errors compared to the ones handled by sequence tagging models since the sequence tagging models generate edits based on human-crafted rules or vocabularies, especially when it comes to long insertions and low-quality texts.

The main contributions of this work are two folds:

1) We enhance the current neural Seq2Seq architecture by adding a supervised copy mechanism that enables the model to copy the unchanged words directly from the source sentence, just as humans do when they correct sentences.
2) The new model is evaluated on three benchmark datasets, CoNLL-2014 test set, BEA 2019, and JFLEG. The first two benchmarks are minimal edits test sets, and the third one is used to evaluate fluency. The model constantly achieves competitive scores compared to the recent SOTA.

II. PRELIMINARIES

A. SEQUENCE-TO-SEQUENCE MODELS

Sequence-to-sequence (seq2seq) models are a class of deep learning models that have gained significant traction in recent years [13], particularly in natural language processing (NLP) tasks, such as machine translation, text generation, Summarization, and GEC tasks. The most prominent seq2seq model is the Transformer [14]. It leverages self-attention mechanisms to enable efficient parallelization and better handling of long-range dependencies. It consists of an encoder and a decoder, each of which is composed of multiple layers of self-attention and feed-forward neural networks. The encoder processes the input sequence $x = (x_1, \ldots, x_n)$ and produces a set of hidden representations $h = (h_1, \ldots, h_n)$, while the decoder generates the output sequence $y = (y_1, \ldots, y_m)$ based on the encoder representations and the previously generated output tokens. This powerful architecture has inspired state-of-the-art models such as GPT-3 [15] and T5 [16]. In the context of this work, we use the Transformer as a core component in our proposed model due to its proven success and adaptability across various tasks [14], including machine translation, summarization, and dialogue generation.

B. ATTENTION MECHANISM

The attention mechanism is a key component in many deep learning models, particularly in the Transformer architecture [14]. It is a mechanism that allows the model to selectively focus on different parts of the input sequence when making predictions. It was first introduced in the context of neural machine translation by Bahdanau et al. in 2014 [17]. In a Transformer model, the attention mechanism is used to compute a weighted sum of the input representations, where the weights are determined by the similarity between the current hidden state of the decoder and the representations in the encoder [18], [19]. This allows the model to attend to different parts of the input sequence at each time step rather than processing all of the inputs in a fixed order.

Classically, an attention model can be learned implicitly, i.e., in an end-to-end manner based on the final objective of the model. However, due to sparsity issues caused by a large number of free parameters in large models trained on small datasets [20], overfitting might occur. Supervised Attention Mechanism is a type of attention mechanism used in deep learning models, particularly in NLP tasks such as machine translation [21], sentiment analysis [22] and text classification [23]. In a supervised attention mechanism, the model is trained with labeled data to learn where to focus its attention on the input sequence. It has been shown to improve performance on NLP tasks compared to models that use unsupervised attention mechanisms or no attention mechanism at all [21]. By incorporating explicit supervision, supervised attention mechanisms can learn to attend to the most relevant parts of the input more effectively, leading to improved predictions [24].

C. COPY MECHANISM

The copying mechanism is important to human language communication. It basically refers to locating a certain segment of the source sentence and copying it as it is to the target sequence [25]. In Seq2Seq models, the copy mechanism refers to a mechanism that allows the decoder to copy words directly from the input sequence to the output sequence [25]. This is especially useful in cases where the output sequence is a rephrased or translated version of the input sequence, and there are words or phrases in the input sequence that cannot be accurately captured by the language model's vocabulary. During decoding, the attention mechanism can choose to either attend to the encoder's hidden states or directly copy a token from the input sequence. The copy mechanism is implemented by adding a binary indicator for each token in the encoder's input sequence, indicating whether the token should be copied directly from the input sequence or generated by the language model [26]. This results in a hybrid generation approach that combines the strengths of both the language model and the copy mechanism. This approach can significantly improve the performance of Seq2Seq models in specific use cases, such as machine translation [21], [25], text summarization [12], GEC, and data-to-text generation [27]. In the context of GEC, the copy mechanism allows the model to directly copy already correct tokens from the source sentence to the target one.

D. MULTI-TASK LEARNING APPROACHES (MTL)

Multi-task learning is a type of machine learning approach that involves training a single model to perform multiple related tasks simultaneously. Instead of training separate models for each task, a multi-task learning model learns to share information and leverage the correlations between the tasks to improve overall performance [28]. In MTL, a shared representation is jointly learned from multiple tasks [29]. Theoretical results have shown that the joint training scheme with MTL is more sample efficient than single-task learning, at least under certain assumptions of task relatedness, linear features, and model classes [9]. A work proposed by [29] tried to study the influence of auxiliary tasks on multi-task learning for sequence tagging problems. They concluded that applying similarity measures to choose the auxiliary dataset for MTL has increased the main task performance. In the context of GEC, the work of [9] is the only work that mentions multi-task learning for GEC. They added token-level and sentence-level multi-task learning for the GEC task. However, their model is a Seq2Seq model that inherited all the Seq2Seq issues mentioned earlier in this section.

III. THE SUPERVISED ATTENTION-BASED MODEL

A. BASE ARCHITECTURE

The work of [9] is the most similar to ours. However, it may not be an appropriate baseline for our proposed approach. This is because the previous work [9] used different pretraining data and a different transformer version than our proposed approach. These differences in pretraining data and transformer version can significantly affect the performance of the model, making it difficult to compare the performance of the two approaches in a fair and meaningful way. Therefore, we implemented our own baseline that uses the same copy mechanism architecture as [9]. This baseline was trained and tested under our experimental settings, using the same pretraining data and transformer version as our proposed approach. This approach ensures that any performance differences between our proposed approach and the previous work are due to differences in the model architecture and training methodology rather than differences in pretraining data or the transformer version. Therefore, our baseline uses the most commonly used Transformer Architecture for GEC, Transformer (big) model [14], with a 6-layer encoder (*enc*) and a 6-layer decoder (*dec*) with 1,024 hidden units and copy mechanism. Regarding the copy mechanism, most of the current works, such as [9], follow the following pattern to calculate the final probability distribution, which will be used as a base model for this work, as shown in Fig 1. For each output token y_t at output position t, given source token sequence x=(x_1,...,x_T), the generation probability distribution over token vocabulary Υ is defined as:

$$p^{gen}(y_t \mid y_{1:t-1}; x) = softmax(W^{gen}[h_t^{dec}, o_t]), \ y \in \Upsilon \quad (1)$$

where W^{gen} is a learned parameter and:

$$h_t^{dec} = dec(y_{1:t-1}; H^{enc}) \quad (2)$$

$$H^{enc} = enc(x) \quad (3)$$

$$s_t = softmax(\frac{(h_t^{dec})^T H^{enc}}{\sqrt{d}}) \quad (4)$$

$$o_t = H^{enc} s_t \quad (5)$$

The copy probability can be calculated from the attention distribution as follows:

$$p^{copy}(y_t \mid y_{1:t-1}; x) = s_t \quad (6)$$

and the combined final probability is calculated using Eq.7:

$$p(y_t) = (1 - \alpha^{copy}).p^{copy}(y_t) + \alpha^{copy}.p^{gen}(y_t) \quad (7)$$

where α^{copy} is the balancing factor between p^{copy} and p^{gen} calculated using Eq. 8.

$$\alpha^{copy} = \sigma((W^{\alpha})^T o_t) \quad (8)$$

where W^{α} is a learned parameter. The balancing factor α^{copy} is learned implicitly by an attention mechanism during the training. However, the currently used attention methods are non-parametric or trained inside the model without explicit supervision. We think that supervising this balancing factor could guide the model to decide whether to generate or copy in a more efficient way. Therefore, we enhance the copy mechanism with a token-level labeling task that can be learned jointly with the Seq2Seq objective in multi-task learning settings. This leads to a refined calculation of the balancing factor that serves as a switch between copying and generating, learned through a supervised approach. We will delve into the details of this process in the subsequent sections.

B. THE PROPOSED MODEL

As shown in Figure 2, the following are the main components of the proposed model:

1) TOKEN-LEVEL LABELING TASK

We extract the labels for this task from the annotated GEC used in this work. To generate the labels, we assign a binary label for each token, indicating whether it is correct or incorrect. Under the assumption that each source token x_i can be aligned with a target token y_j, a token is correct if $x_i = y_j$, and incorrect otherwise. The alignment has been done using fast-align.[1]

2) MULTI-TASK LEARNING

For each token in the source sentence, a label is assigned indicating whether this token is correct or incorrect. Each token's label is predicted by passing the final state of the encoder through a *softmax* after an affine transformation, as shown in Fig 2. Two tasks will be jointly learned in our model, and each task has its corresponding loss function. The cross-entropy between the ground truth labels and the predicted labels can be considered as an auxiliary objective to be learned jointly with the GEC objective function. The total loss will be as follows:

$$L_{total} = l_{corr} + \lambda l_{det} \quad (9)$$

where $\lambda \in [0, 1]$ and l_{corr} and l_{det} are the individual losses of the correction and the detection task, respectively.

3) SUPERVISED COPY MECHANISM

In the new method, we followed the same calculations except for the way of calculating the alpha α^{copy}. Thus, Equations Eq.1, Eq.2, Eq.3, Eq.4, Eq.5, and Eq.6 remain the same as described in Section III-A. To learn the balancing factor, a token-level labeling task is used for the original sequence in which a binary label is assigned for each token in the source sentence indicating whether this token is correct or incorrect, as mentioned in Section III-B1. Each token's label is predicted by a binary classifier with a *softmax* and an affine transformation at the top of the encoder with the final state H_i^{enc} of the encoder as input, as shown in Eq. 10.

$$z_i^l = p(label_i \mid x_{1...N}) = softmax(W^T h_i^{enc}), \quad (10)$$

The balancing factor, α^{copy}, is then calculated as:

$$\alpha^{copy} = \sigma((W^{\alpha})^T c) \quad (11)$$

where:

$$c = H^{enc} z^l \quad (12)$$

The final probability distribution is calculated according to Eq. 13:

$$p(y_t) = (1 - \alpha^{copy}).p^{copy}(y_t) + \alpha^{copy}.p^{gen}(y_t) \quad (13)$$

In this way, we directly optimize the copy mechanism in a supervised way. It is worth mentioning that the supervised attention mechanism used in this work may play as a regularizer in the multi-task learning objective since it can mitigate the vanishing gradient problem during the back-propagation by adding supervision into the intermediate layers in the network [30].

a: AN ILLUSTRATION OF SUPERVISED COPY MECHANISM

From the calculation of the final probability in Eq. 13, we can see that the value of the α^{copy} plays as a balancing factor between the magnitude of the copy and generate probabilities, so:

- If α^{copy} is near one, the model leans towards generation.
- If α^{copy} approaches 0, the model favors copying.

[1] https:///github.com/clab/fast_align

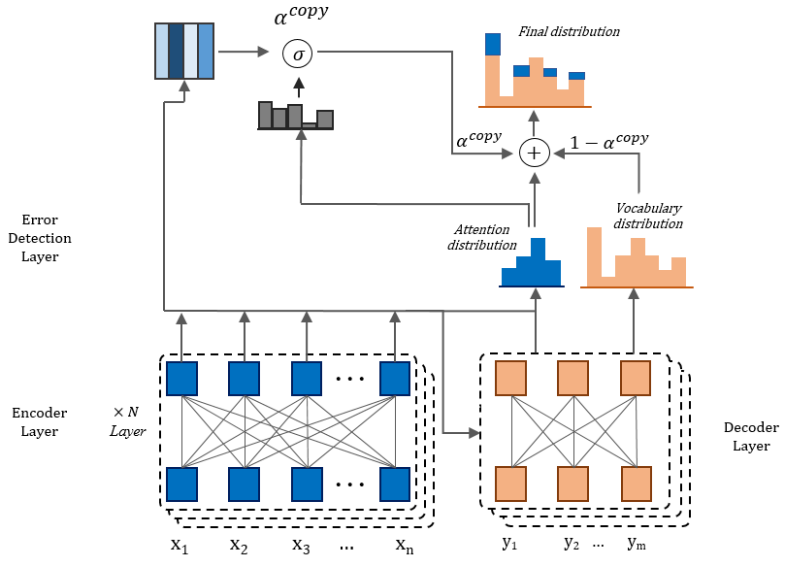


FIGURE 1. The traditional way of calculating the final probability distribution.

This balance is visualized as:

$$p(y_t) = \begin{cases} Copy & \alpha^{copy} \approx 0 \\ Generate & \alpha^{copy} \approx 1 \end{cases}$$

Following the newly proposed method for calculating α^{copy}, as in Eq.10, Eq.11, and Eq.12, the token label probability for each correct tokens y_t, will gravitate towards zero, favoring the copy score in the probability of y_t. Conversely, if y_t is incorrect, the token label probability leans towards one, giving the generation score a significant contribution to the probability of y_t. This dynamic nudges the model to copy when a token is correct.

Figure 3 provides a practical example of how α^{copy} values are calculated following the proposed approach for each token in the sentence, "A ten-year-old boy go to school." As per Eq. 13, the α^{copy} values aim to deter the model from altering correct tokens by maximizing their copy probability, while encouraging the model to rectify incorrect words by maximizing the generation probability. As a result, the α^{copy} is close to zero for all the correct tokens so that the copy probability will be dominant, and the model will copy those tokens to the output.

For instance, for the correct token "to", α^{copy} is near zero, hence copy probability dominates and the model copies the token to the output. The total probability calculation is as follows:

$$p(y_t) = (1-\alpha^{copy}).p^{copy}(y_t) + \alpha^{copy}.p^{gen}(y_t)$$
$$p(y_t) = (1-0.0005).p^{copy}(y_t) + 0.0005.p^{gen}(y_t)$$
$$p(y_t) = (0.99956).p^{copy}(y_t) + 0.0005.p^{gen}(y_t)$$

However, for the incorrect token "*go*", α^{copy} is close to one, pushing the model to favor generation:

$$p(y_t) = (1-0.86754).p^{copy}(y_t) + 0.86754.p^{gen}(y_t)$$
$$p(y_t) = (0.13246).p^{copy}(y_t) + 0.86754.p^{gen}(y_t)$$

In this case, the p^{gen} will be dominant so that the model will generate the correction.

IV. EXPERIMENTS

A. DATASETS

1) PRETRAINING DATA

Due to GEC public data scarcity [31], we followed the work of [5] and [32] to generate the pretraining dataset in which they make use of a publicly available English clean non-parallel dataset, One-Billion-Word dataset [33], as shown in Table 1. They applied some noising scenarios to inject several errors into clean sentences, thus generating an additional artificial dataset of noisy and clean sentence pairs.

2) TRAINING AND FINE-TUNING DATASETS

Following recent works in English GEC, we conduct experiments in the same setting with the restricted track of the BEA-2019 GEC shared task [34]. The public version of the Lang-8 corpus [35], NUCLE [36], the FCE corpus [37], and the Cambridge English Write & Improve training split described in the BEA-2019 shared task (BEA-19 train) [34] are combined and used for the first fine-tuning stage. For the second fine-tuning stage, we only used W&I+LOCNESS [34] (shown in Table 1).

B. EXPERIMENTAL SETTINGS

Transformer-big model is used with six layers for each encoder and decoder and a vocabulary size of 32k Byte Pair

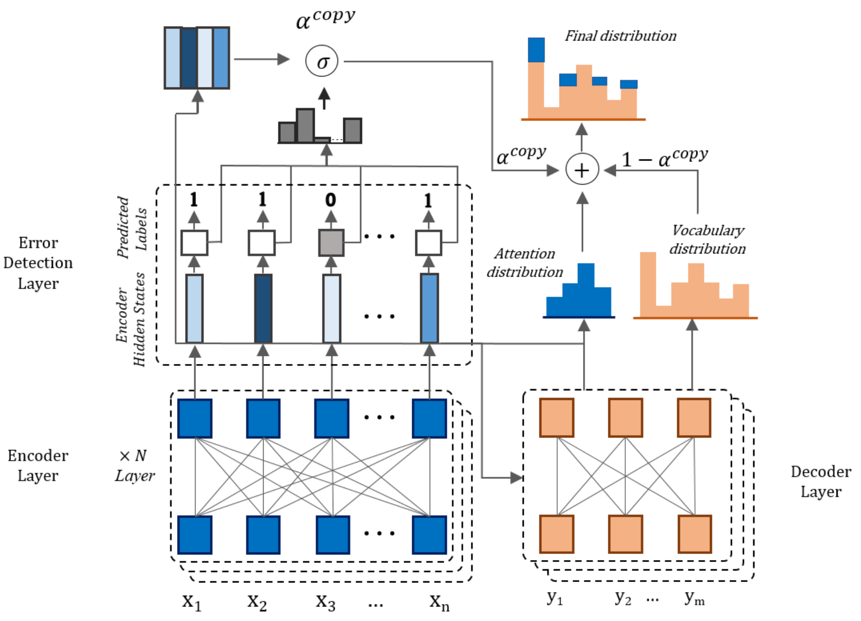
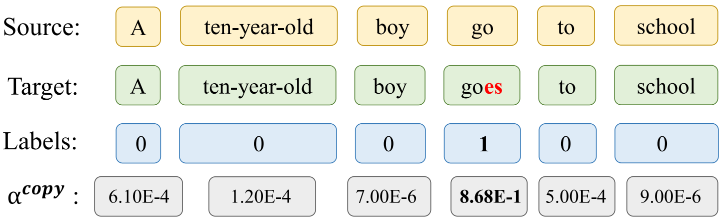


FIGURE 2. The proposed model flow chart. The token-level labeling task is added to the main task, Seq2Seq. The final layer calculates label predictions based on the encoder output. The *softmax* activation function is used to output a normalized probability distribution over all the possible labels for each token. α^{copy}, the balancing factor is calculated from the supervised attention after applying an affine transformation as in Eq. 11.



Source:	A	ten-year-old	boy	go	to	school
Target:	A	ten-year-old	boy	goes	to	school
Labels:	0	0	0	**1**	0	0
α^{copy} :	6.10E-4	1.20E-4	7.00E-6	**8.68E-1**	5.00E-4	9.00E-6

FIGURE 3. Example for α^{copy} Calculation.

Encoding [38]. A beam size of (beam=5) is used at inference time in all experiments. For a fair comparison with state-of-the-art models, we follow the works of [5], [39], and [32], which used CoNLL-2014 [40] and BEA 2019 [34] test sets for evaluation. BEA dev set [34] is used for validation. Moreover, the JFLEG test set [41] with GLEU metric is used to provide another aspect of evaluation with respect to fluency. Unlike CoNLL-2014 [40], and BEA 2019 [34], this dataset represents a wide range of language proficiency levels by providing more native-sounding corrections for the original text. To compare with the state-of-the-art approaches in English GEC that pre-train with synthetic data, we used the same pretraining dataset used by [5] and [32], as described in Section IV-A. Moreover, F0.5 and M2 scores are reported for BEA 2019 [34] and CoNLL-2014 [40], respectively, and GLEU is reported for the JFLEG test set.

C. EXPERIMENTAL RESULTS

In our work, we evaluate the performance of our proposed GEC model by comparing its predicted corrections with gold standard edits, focusing on a comparative evaluation against recently proposed GEC models. Table 2 demonstrates that our model achieved superior results on the JFLEG dataset, reflecting its ability to handle a broad range of fluency levels. However, it did not surpass sequence tagging-based models on minimal edit datasets like BEA 2019 and CoNLL-2014. We attribute this to the inherent strengths of sequence tagging models in correcting errors with small spans, whereas Seq2Seq models tend to generate output sequences from scratch. Despite this, sequence tagging models struggle to maintain comparable GLEU scores on the JFLEG dataset, which comprises holistic sentence rewrites without limiting corrections to minimal error spans or providing error coding. Our model's GLEU scores on the JFLEG dataset were 64.9 and 65.5 for single and ensemble models, respectively, highlighting its better performance compared to sequence tagging-based models in this context. Regarding the over-correction issue, it is clear from the result shown in Table 2 that our model's precision is higher than the previous models.

TABLE 1. Descriptions and statistics of the datasets.

Dataset	Description	Data size(#sentences)	Quality
PIE-Synthetic	Perturbed version of One-billion-word	9,000,000	-
Lang8	Online English learning site	947,344	Poor
NUCLE	College student essays	56,958	Good
FCE	ESL exam questions	34,490	Good
WI+Locness	English essays	34,304	Good

D. ABLATION STUDY ON THE MODEL ARCHITECTURE

1) THE EFFECT OF SUPERVISED ATTENTION

Table 3 highlights the superior performance of our model across all test datasets, including JFLEG, thereby emphasizing its ability to manage varying fluency levels effectively. This superior performance can be attributed to the implementation of supervised attention mechanisms, which have been proven to enhance performance in NLP tasks compared to unsupervised or no attention mechanisms. With explicit supervision, these mechanisms effectively focus on the most relevant input sections, leading to improved predictions, particularly in identifying incorrect tokens.

The left side of Table 3 shows that our model with the supervised copy mechanism exhibits a substantial reduction in over-correction, demonstrated by the decreased number of False Positives (FP=1654), compared to the two baselines. This is in line with our objective to minimize over-correction by preserving grammatically correct sentence portions. Furthermore, our model yields a higher precision score (63.40), outpacing both baselines. This improved precision signifies that our model is more adept at avoiding unnecessary or incorrect corrections, further reinforcing its ability to limit over-correction. With an F0.5 score surpassing both baselines, our model effectively balances precision and recall, thus consolidating its overall superior performance.

2) PRETRAINED ENCODER TYPE

Three main works [5], [32], [42], used as baselines in this work, are pre-trained based on powerful BERT-like language models, namely, BERT [43], RoBERTa [44], and XLNet [45]. Following the same, we use BART to initialize a 12+2 model. As shown in Table 3, after the fine-tuning stage, the proposed model with BART achieved a better result on all three datasets.

V. RELATED WORK

A. MACHINE TRANSLATION-BASED MODELS

Although GEC MT-based systems have become state-of-the-art approaches, GEC differs from translation since it only changes several words of the source sentence. Several GEC Seq2Seq-based models have been proposed, such as [6], [9], [42], [46], [47], and [48]. The main architecture of these models is the encoder-decoder with attention. Some of these use the copy mechanism to enhance the performance of the basic Seq2Seq models. In a work proposed by [9], they used a form of copy mechanism to encourage the GEC model to copy the correct tokens from the source to the target unchanged. Despite that, these models have the ability to mitigate the issue of over-correction as well as hold better generality and diversity in the generation results compared to the sequence tagging models [3]; the encoder-decoder attention serves as the copy distribution. In contrast, guaranteeing that important words in the source are copied remains a challenge [12].

B. SEQUENCE TAGGING-BASED MODELS

Sequence tagging models are a common approach for grammatical error correction (GEC) tasks [3], [5], [6], [7], [32]; in which each sentence is annotated with edits as labels. The target sentence can be recovered by applying those edits to the source sentence. Awasthi et al. [5] proposed a GEC model based on the idea of iteratively applying edits to the original text to produce a sequence of intermediate texts that approach the corrected text. A similar approach was proposed by Omelianchuk et al. [32]. It is based on a sequence tagger that uses a pre-trained BERT-like transformer as an encoder and two linear layers in place of the decoder. One main issue in these models is that their performance drops on data with more edit span. Moreover, the edits, such as the verb form transformations (e.g., VBD/VBZ) and prepositions (e.g., on/in), are usually constrained by human-designed or automatically generated lexical rules [5], [32] and vocabularies [5], [49], which limits the generality and transferability of these methods.

C. COPY MECHANISM-BASED MODELS

Copy mechanism is not a new topic; it has been there for a while. One of the early issues in the copy mechanism is how the model can decide when to copy and when to generate. It has been deployed to improve Seq2Seq models on several tasks, such as summarization [12], [50] and GEC [9], [34]. In most of the previous work, the balance between copying and generating is controlled by a balancing factor learned implicitly during the training [9]. Classically, the copy balancing factor can be calculated from the attention weights, which are learned implicitly, i.e., in an end-to-end manner based on the final objective of the model. Unfortunately, this might lead to extremely high weights for some parts of the sentence, leaving essentially negligible weights for the other important context in the sentences. Moreover, due to sparsity issues caused by a large number of free parameters, the attention model learned by implicit training generates low-quality attention maps. Furthermore, in practice, the variability in natural language is very large. The available annotated datasets are often too small to learn everything

TABLE 2. Performance in English GEC benchmarks (i.e., CoNLL-14 (M2 Score), BEA-19 test (ERRANT), and JFLEG). The single model scores are at the upper part, while the lower part shows the scores for the ensemble models.

	CoNLL-14			BEA-19 test			JFLEG
Method	P	R	F0.5	P	R	F0.5	GLEU+
gT5 xxl [6]	-	-	65.7	-	-	69.8	
gT5 base [6]	-	-	54.1	-	-	60.2	
PIE [5]	66.1	43.0	59.7	-	-	-	60.3
Stahlberg 2021 [7]	72.8	49.5	66.6	72.1	64.4	70.4	64.7
GECTor [34]	77.5	40.1	65.3	79.2	53.9	72.4	-
Ours with BART init	74.7	44.3	65.7	80.4	55.4	73.7	64.9
PIE [5]	68.3	43.2	61.2	-	-	-	61.0
Stahlberg 2021 [7]	75.6	49.3	68.3	77.7	65.4	74.9	64.7
GECTor [34]	78.2	41.5	66.5	79.4	57.2	73.7	-
Copy-augmented(4 ens) [9]	71.6	38.7	61.1	-	-	-	61.0
Ours with BART init(3 ens)	77.8	41.4	66.1	84.6	51.7	75.1	65.5

TABLE 3. Performance of the model variants related to BART initialization and supervised attention.

	BEA-19 dev						CoNLL-13	JFLEG-dev
Method	TP	FP	FN	Precision	Recall	F0.5	F0.5	GLEU
Transformer_big & copy mechanism (Base Model)	3002	2213	5120	57.63	36.92	51.81	48.7	57.1
Transformer_big & copy mechanism & BART	2989	2124	4796	58.57	38.2	52.93	50.9	57.6
Transformer_big & copy mechanism & BART & supervised attention	2865	1654	5110	63.39	35.86	54.95	52.2	59.6

automatically. The patterns discovered in the data might not always correspond to the behavior that we expect or desire our models to exhibit.

D. SUPERVISED ATTENTION-BASED MODELS

Supervised Attention with References refers to a type of machine learning model architecture that combines both supervised learning and attention mechanisms with references or additional inputs. In these models, the attention mechanism helps the model to focus on the most relevant information from the input. In contrast, the references or other inputs provide additional context that guides the attention mechanism toward a specific task [21]. For example, in natural language processing tasks, a supervised attention model might be trained to predict the next word in a sentence given the previous words while using reference inputs such as a pre-defined summary of the topic being discussed. This reference information helps the attention mechanism focus on the most relevant information in the input and make a more informed prediction. Supervised attention with references has been applied to a wide range of tasks, including machine translation [21], sentiment analysis [51], text classification [23], text summarization [12], and question answering [52]. These models often outperform traditional attention models or supervised learning models alone, as they are able to leverage both the strengths of attention mechanisms and additional contextual information. To the best of our knowledge, our work is the first to utilize supervised attention in the context of GEC.

VI. CONCLUSION

The copying mechanism basically refers to locating a certain segment of the source sentence and copying it as it is to the target sequence. Technically, the attention mechanism is used to determine when the model should copy rather than generate. In this work, the model is provided with explicit supervision in the form of attention labels, which indicate the important parts of the input that should be corrected. This supervision can be in the form of attention maps or binary attention masks. During training, the model optimizes its attention mechanism to match the provided labels, allowing it to learn to attend to the incorrect parts of the input and increase its generation probability while copying over the rest to the output unchanged. To the best of our knowledge, we are the first to use supervised attention to improve the copy mechanism in the context of GEC. It is arguably more similar to how humans write or edits text. The supervised copy mechanism can be used mainly in the GEC problem, but it can also be used on any problem that we have, or we can have alignment data between the source and the target. As a future work, we would like to expand this approach to new languages other than English. However, one possible limitation of this work is that it was evaluated on a specific set of datasets and may not generalize well to other datasets or contexts. While we demonstrated the effectiveness of our approach in handling low-fluency datasets and reducing over-correction, the model's performance may vary in other settings or with different types of errors. Further evaluation of a broader range of datasets and error types may be necessary to assess the generalizability of our approach fully.

REFERENCES

[1] R. N. Neupane, "Error analysis of written English composition: A case of basic level students," *Tribhuvan J.*, vol. 1, no. 1, pp. 101–109, Mar. 2023. [Online]. Available: https://www.nepjol.info/index.php/tribj/article/view/53517

[2] J. Lichtarge, C. Alberti, S. Kumar, N. Shazeer, N. Parmar, and S. Tong, "Corpora generation for grammatical error correction," in *Proc. Conf. North*, 2019, pp. 3291–3301. [Online]. Available: https://aclanthology.org/N19-1333

[3] J. Li, J. Guo, Y. Zhu, X. Sheng, D. Jiang, B. Ren, and L. Xu, "Sequence-to-action: Grammatical error correction with action guided sequence generation," in *Proc. AAAI Conf. Artif. Intell.*, Jun. 2022, vol. 36, no. 10, pp. 10974–10982. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/21345

[4] C. Park, Y. Yang, C. Lee, and H. Lim, "Comparison of the evaluation metrics for neural grammatical error correction with overcorrection," *IEEE Access*, vol. 8, pp. 106264–106272, 2020.

[5] A. Awasthi, S. Sarawagi, R. Goyal, S. Ghosh, and V. Piratla, "Parallel iterative edit models for local sequence transduction," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 4260–4270. [Online]. Available: https://aclanthology.org/D19-1435

[6] S. Rothe, J. Mallinson, E. Malmi, S. Krause, and A. Severyn, "A simple recipe for multilingual grammatical error correction," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics 11th Int. Joint Conf. Natural Lang. Process.*, 2021, pp. 702–707. [Online]. Available: https://aclanthology.org/2021.acl-short.89

[7] F. Stahlberg and S. Kumar, "Synthetic data generation for grammatical error correction with tagged corruption models," in *Proc. 16th Workshop Innov. Use NLP Building Educ. Appl.*, Apr. 2021, pp. 37–47. [Online]. Available: https://aclanthology.org/2021.bea-1.4

[8] Z. Tu, Z. Lu, Y. Liu, X. Liu, and H. Li, "Modeling coverage for neural machine translation," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, Berlin, Germany, 2016, pp. 76–85. [Online]. Available: https://aclanthology.org/P16-1008

[9] W. Zhao, L. Wang, K. Shen, R. Jia, and J. Liu, "Improving grammatical error correction via pre-training a copy-augmented architecture with unlabeled data," in *Proc. Conf. North*, 2019, pp. 156–165. [Online]. Available: https://aclanthology.org/N19-1014

[10] J. Gu, Q. Liu, and K. Cho, "Insertion-based decoding with automatically inferred generation order," *Trans. Assoc. Comput. Linguistics*, vol. 7, pp. 661–676, Nov. 2019.

[11] N. Jaitly, Q. V. Le, O. Vinyals, I. Sutskever, D. Sussillo, and S. Bengio, "An online sequence-to-sequence model using partial conditioning," in *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29. Red Hook, NY, USA: Curran Associates, Inc., 2016. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2016/file/312351bff07989769097660a56395065-Paper.pdf

[12] S. Xu, H. Li, P. Yuan, Y. Wu, X. He, and B. Zhou, "Self-attention guided copy mechanism for abstractive summarization," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 1355–1362. [Online]. Available: https://aclanthology.org/2020.acl-main.125

[13] K. Al-Sabahi, Z. Zuping, and Y. Kang, "Bidirectional attentional encoder–decoder model and bidirectional beam search for abstractive summarization," 2018, *arXiv:1809.06662*.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–25.

[15] T. B. Brown et al., "Language models are few-shot learners," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Red Hook, NY, USA: Curran Associates, Inc., 2020, pp. 1877–1901. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

[16] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.*, vol. 21, no. 140, pp. 1–67, 2020. [Online]. Available: http://jmlr.org/papers/v21/20-074.html

[17] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, *arXiv:1409.0473*.

[18] K. Al-Sabahi, Z. Zuping, and M. Nadher, "A hierarchical structured self-attentive model for extractive document summarization (HSSAS)," *IEEE Access*, vol. 6, pp. 24205–24212, 2018.

[19] F. Mohsen, J. Wang, and K. Al-Sabahi, "A hierarchical self-attentive neural extractive summarizer via reinforcement learning (HSASRL)," *Int. J. Speech Technol.*, vol. 50, no. 9, pp. 2633–2646, Sep. 2020.

[20] A. Javari, Z. He, Z. Huang, R. Jeetu, and K. C.-C. Chang, "Weakly supervised attention for hashtag recommendation using graph data," in *Proc. Web Conf.*, Apr. 2020, pp. 1038–1048, doi: 10.1145/3366423.3380182.

[21] L. Liu, M. Utiyama, A. Finch, and E. Sumita, "Neural machine translation with supervised attention," in *Proc. 26th Int. Conf. Comput. Linguistics, Tech. Papers*. Osaka, Japan, Dec. 2016, pp. 3093–3102. [Online]. Available: https://aclanthology.org/C16-1291

[22] Y. Zou, T. Gui, Q. Zhang, and X. Huang, "A lexicon-based supervised attention model for neural sentiment analysis," in *Proc. 27th Int. Conf. Comput. Linguistics*. Santa Fe, NM, USA, Aug. 2018, pp. 868–877. [Online]. Available: https://aclanthology.org/C18-1074

[23] S. Choi, H. Park, J. Yeo, and S.-W. Hwang, "Less is more: Attention supervision with counterfactuals for text classification," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2020, pp. 6695–6704. [Online]. Available: https://aclanthology.org/2020.emnlp-main.543

[24] G. Yang and H. Tang, "Supervised attention in sequence-to-sequence models for speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2022, pp. 7222–7226.

[25] J. Gu, Z. Lu, H. Li, and V. O. K. Li, "Incorporating copying mechanism in sequence-to-sequence learning," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, Berlin, Germany, 2016, pp. 1631–1640. [Online]. Available: https://aclanthology.org/P16-1154

[26] B. Liu, M. Zhao, D. Niu, K. Lai, Y. He, H. Wei, and Y. Xu, "Learning to generate questions by LearningWhat not to generate," in *Proc. World Wide Web Conf.*, New York, NY, USA, May 2019, pp. 1106–1118, doi: 10.1145/3308558.3313737.

[27] A. Shimorina and C. Gardent, "Handling rare items in data-to-text generation," in *Proc. 11th Int. Conf. Natural Lang. Gener.*, 2018, pp. 360–370. [Online]. Available: https://aclanthology.org/W18-6543

[28] S. Changpinyo, H. Hu, and F. Sha, "Multi-task learning for sequence tagging: An empirical study," in *Proc. 27th Int. Conf. Comput. Linguistics*. Santa Fe, NM, USA, Aug. 2018, pp. 2965–2977. [Online]. Available: https://aclanthology.org/C18-1251

[29] F. Schröder and C. Biemann, "Estimating the influence of auxiliary tasks for multi-task learning of sequence tagging tasks," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 2971–2985. [Online]. Available: https://aclanthology.org/2020.acl-main.268

[30] C. Li, M. Z. Zia, Q. Tran, X. Yu, G. D. Hager, and M. Chandraker, "Deep supervision with intermediate concepts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 1828–1843, Aug. 2019.

[31] J. Lichtarge, C. Alberti, and S. Kumar, "Data weighted training strategies for grammatical error correction," *Trans. Assoc. Comput. Linguistics*, vol. 8, pp. 634–646, Dec. 2020.

[32] K. Omelianchuk, V. Atrasevych, A. Chernodub, and O. Skurzhanskyi, "GECToR—Grammatical error correction: Tag, not rewrite," in *Proc. 15th Workshop Innov. Use NLP Building Educ. Appl.*, 2020, pp. 163–170. [Online]. Available: https://aclanthology.org/2020.bea-1.16

[33] C. Chelba, T. Mikolov, M. Schuster, Q. Ge, T. Brants, and P. Koehn, "One billion word benchmark for measuring progress in statistical language modeling," 2013, *arXiv:1312.3005*.

[34] C. Bryant, M. Felice, Ø. E. Andersen, and T. Briscoe, "The BEA-2019 shared task on grammatical error correction," in *Proc. 14th Workshop Innov. Use NLP Building Educ. Appl.*, Florence, Italy, 2019, pp. 52–75. [Online]. Available: https://aclanthology.org/W19-4406

[35] T. Mizumoto, M. Komachi, M. Nagata, and Y. Matsumoto, "Mining revision log of language learning SNS for automated Japanese error correction of second language learners," in *Proc. 5th Int. Joint Conf. Natural Lang. Process.*, Nov. 2011, pp. 147–155. [Online]. Available: https://aclanthology.org/I11-1017

[36] D. Dahlmeier, H. T. Ng, and S. M. Wu, "Building a large annotated corpus of learner English: The NUS corpus of learner English," in *Proc. 8th Workshop Innov. Use NLP Building Educ. Appl.*, Atlanta, GA, USA, Jun. 2013, pp. 22–31. [Online]. Available: https://aclanthology.org/W13-1703

[37] H. Yannakoudakis, T. Briscoe, and B. Medlock, "A new dataset and method for automatically grading ESOL texts," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics, Hum. Lang. Technol.*, Portland, OR, USA, Jun. 2011, pp. 180–189. [Online]. Available: https://aclanthology.org/P11-1019

[38] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, 2016, pp. 1715–1725. [Online]. Available: https://aclanthology.org/P16-1162

[39] M. Chen, T. Ge, X. Zhang, F. Wei, and M. Zhou, "Improving the efficiency of grammatical error correction with erroneous span detection and correction," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2020, pp. 7162–7169. [Online]. Available: https://aclanthology.org/2020.emnlp-main.581

[40] H. T. Ng, S. M. Wu, T. Briscoe, C. Hadiwinoto, R. H. Susanto, and C. Bryant, "The CoNLL-2014 shared task on grammatical error correction," in *Proc. 18th Conf. Comput. Natural Lang. Learn., Shared Task*, Baltimore, MD, USA, 2014, pp. 1–14. [Online]. Available: https://aclanthology.org/W14-1701

[41] C. Napoles, K. Sakaguchi, and J. Tetreault, "JFLEG: A fluency corpus and benchmark for grammatical error correction," in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics*, 2017, pp. 229–234. [Online]. Available: https://aclanthology.org/E17-2037

[42] X. Sun, T. Ge, F. Wei, and H. Wang, "Instantaneous grammatical error correction with shallow aggressive decoding," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics 11th Int. Joint Conf. Natural Lang. Process.*, 2021, pp. 5937–5947. [Online]. Available: https://aclanthology.org/2021.acl-long.462

[43] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pretraining of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1, Jun. 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423

[44] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*.

[45] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, *XLNet: Generalized Autoregressive Pretraining for Language Understanding*. Red Hook, NY, USA: Curran Associates Inc., 2019.

[46] Z. Yuan and C. Bryant, "Document-level grammatical error correction," in *Proc. 16th Workshop Innov. Use NLP Building Educ. Appl.*, Apr. 2021, pp. 75–84. [Online]. Available: https://aclanthology.org/2021.bea-1.8

[47] Y. Zhang and Q. Yang, "A survey on multi-task learning," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 12, pp. 5586–5609, Dec. 2022.

[48] M. Kaneko, M. Mita, S. Kiyono, J. Suzuki, and K. Inui, "Encoder–decoder models can benefit from pre-trained masked language models in grammatical error correction," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 4248–4254. [Online]. Available: https://aclanthology.org/2020.acl-main.391

[49] J. Mallinson, A. Severyn, E. Malmi, and G. Garrido, "FELIX: Flexible text editing through tagging and insertion," in *Proc. Findings Assoc. Comput. Linguistics, EMNLP*, 2020, pp. 1244–1255. [Online]. Available: https://aclanthology.org/2020.findings-emnlp.111

[50] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," 2017, *arXiv:1704.04368*.

[51] J. Tang, Z. Lu, J. Su, Y. Ge, L. Song, L. Sun, and J. Luo, "Progressive self-supervised attention learning for aspect-level sentiment analysis," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 557–566. [Online]. Available: https://aclanthology.org/P19-1053

[52] C. Gan, Y. Li, H. Li, C. Sun, and B. Gong, "VQS: Linking segmentations to questions and answers for supervised attention in VQA and question-focused semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Los Alamitos, CA, USA, Oct. 2017, pp. 1829–1838, doi: 10.1109/ICCV.2017.201.

KAMAL AL-SABAHI received the B.S. degree in computer science from Sana'a University, Sana'a, Yemen, in 2008, the M.S. degree in information technology from OUM University, Kuala Lumpur, Malaysia, in 2015, and the Ph.D. degree in computer science from Central South University, Changsha, China, in 2019. He is currently an Assistant Professor with the University of Technology and Applied Science-Ibra, Oman. His research interests include deep learning, natural language processing, knowledge engineering, and data mining.

KANG YANG received the B.S. degree in computer science from Wuhan University, China, in 2015, and the M.S. degree in information technology from Central South University, Changsha, China, in 2019. He is currently pursuing the Ph.D. degree in computer science with the National University of Defense Technology, China. His research interests include software engineering, deep learning, natural language processing, knowledge engineering, and data mining.