Automatic Grammatical Error Correction for Arabic

Enas Albasiri, Tawa Suleman and Tyler Lanni
The Graduate Center, CUNY
ealbasiri@gradcenter.cuny.edu
tSuleman@gradcenter.cuny.edu
tlanni@gradcenter.cuny.edu

Abstract

Automatic Grammatical Error Correction (GEC) aims to correct multiple types of errors such as orthography, grammar, lexical and morphological errors. Many GEC tasks have been completed in European languages like English, a language with low morphological diversity, but low sourced languages such as Arabic have received less attention. Prior work on Arabic GEC with a few exceptions used machine learning techniques and rule based approaches, methods that are largely outclassed by the advent of modern neural networks. In this study, we expanded beyond strictly statistical models into more sophisticated transformer models, an evolution of neural network architecture. We evaluated the performance of the model using M2 metrics, comparing our efficiency to that of past Arabic GEC experiments. We found that our model performed with a low accuracy compared to other applications of Arabic GEC based on multi-head attention models. However, the proposed method has great potential and could outperform other models if it is fine-tuned to accommodate the complexity of Arabic language morphology.

1 Introduction

A growing body of research in natural language processing has been done on Automatic Grammar Error Correction (GEC) due to the increasing number of bilingual speakers and second language learners around the world. GEC aims to correct multiple types of error including orthography, grammar, lexical, and morphological errors. Automatic GEC has proven to be a useful tool for natural language processing applications. GEC tasks were applied on multiple languages including English (Rozovskaya and Roth, 2011; Susanto et al., 2014; Yuan and Briscoe, 2016; Hoang et al., 2016; Chollampatt et al., 2016; Junczys-Dowmunt and Grundkiewicz, 2016; Mizumoto and Matsumoto, 2016; Rozovskaya and Roth, 2016; Jianshu et al., 2017; Chollampatt and Ng, 2018; Kaneko et al., 2020), Russian(Rozovskaya and Roth, 2019; Trinh and Rozovskaya, 2021), Chinese(Lee et al., 2016; Rao et al., 2017, 2018), and Arabic(Solyman et al., 2022; Zaghouanietal.,2014).

In this paper,we proposed an automatic GEC model for Arabic, a low-resource language. We will evaluate the performance of a transformer model to perform automatic text correction of Arabic script. By applying more recent developments in neural machine translation, the results study could make significant contributions to the field of natural language processing in general, and to Arabic NLP developments in particular, as we (1) make the resource available for research purposes; (2) provide benchmark results on this new corpus, using state-of-the-art models that are trained on synthetic data and learner data; (3) provide a linguistic error analysis of the models' performance to give an insight of what areas are the current system are highly achieving, and what areas cans

The remainder of this paper is organized as follows. Section 2 presents related work and highlights some language processing challenges that are specific to Arabic. Section 3 describes our experiments, data, and evaluation method. In Section 4 we detail our methods. Section 5 presents results and error analysis. We conclude in Section 6.

2 Background

2.1 Related Work

There has been increased work in automatic grammatical error correction, particularly of Arabic, after the successful Arabic text correction shared tasks QALB 2014 (Mohit et al., 2014) and QALB 2015 (Rozovskaya et al., 2015). (Rozovskaya et al., 2014) proposed a hybrid model of machine learning and rule-based approach. More recent work took the advantage of modern neural networks to perform automatic correction task. For example, Ahmadi (2018) applied sequence-to-sequence and attention-based model which was the first Arabic grammatical error correction model based on end-to-end deep neural networks. Solyman et al. (2022) also presented a sequence-to-sequence Transformer model which achieved the best F1 score compared to other grammatical error correction when applied on the two benchmarks QALB-2014 and QALB-2015. Belkebir and Habash (2021) presented an automatic error type annotation system for Modern Standard Arabic called ARETA. ARETA achieved a high performance of 85.8% on a manually annotated test data.

2.2 Arabic Language Processing Challenges

Arabic posits a challenging case for language processing in general and automatic GEC in particular for multiple reasons. First, Arabic has a rich morphology. Secondly, Arabic is a low-resource language.

3 Experiments

3.1 Data

For this task, we are using the Qatar Arabic Language Bank (QALB). The data are taken from online commentaries written to Aljazeera articles and was annotated and corrected by native speakers of Arabic. There are 2 million words in the training data, and roughly 50,000 words in the development and test data. The data consists of the original comments, the corrected sentences, and the corrected sentences with their proposed edits.

3.2 Data Processing

After standard tokenization and cleaning procedures, the data was processed using subword-nmt. This python package allows the text to be broken into subwords. The method, described by Sennrich et al. (2016), employs a technique known as Byte Pair Encoding (BPE) to break up sequences of characters into the most frequently used patterns.

Subword Neural Machine Translation (NMT) shows improvement in performance of dealing with out of vocabulary tokens (OOV). This method can segment potentially novel words into known subwords for easier translation. For instance, creating subwords out of the word <flying> would create two segments <fly> and <@@ing>. Suppose that <flying> was not in the initial training data but <fly> and <@@ing> was, the NMT would have the appropriate information to deal with the OOV word <flying>.

Given that the QALB data is Arabic, a language which is heavily diverse in its morphology, subword NMT will allow the model to a wide range of words without replacing them as unknown (<unk>) in the training, validation, and testing phases.

The BPE tokenizer was trained on L1 and L2 target language representations from the training data. A vocabulary of 40,000 subwords was created. The tokenizer was then applied to the source and target representations of the train, validation, and testing data.

After the BPE tokenization, the data was binarized and used to create a dictionary using the fairseq-preprocessing command. The preprocess data successfully lowered the number of words replaced by <unk>. Where cor is the target representation and incor is the source representation, the preprocessing produced the following output:

```
| [incor] Dictionary: 37472 types
| [incor] data/train/train.tok.incor: 17665 sents, 1143963 tokens, 0.161% replaced (by <unk>)
| [incor] Dictionary: 37472 types
| [incor] data/dev/dev.tok.incor: 927 sents, 63433 tokens, 0.377% replaced (by <unk>)
| [incor] Dictionary: 37472 types
| [incor] data/test/test.tok.incor: 924 sents, 62616 tokens, 0.434% replaced (by <unk>)
| [cor] Dictionary: 38992 types
| [cor] data/train/train.tok.cor: 17665 sents, 1100636 tokens, 0.0627% replaced (by <unk>)
| [cor] Dictionary: 38992 types
| [cor] data/dev/dev.tok.cor: 927 sents, 61734 tokens, 0.164% replaced (by <unk>)
| [cor] Dictionary: 38992 types
| [cor] data/test/test.tok.cor: 924 sents, 61451 tokens, 0.169% replaced (by <unk>)
```

The output shows that our preprocessing step was able to keep the amount of <unk> tokens used below 1% for each category in our vocabulary.

3.3 Evaluation

The model will be evaluated using the guidelines in the first and the second shared tasks for Arabic GEC. The test sets in both QALB-2014 and QALB-2015 will be used to report the precision, recall, and F1 scores. An example sentence and its corresponding corrections is given in (1):

EXAMPLE

اعداد القتلى في صفوف الارهابيين بالمئات و من

الصعب حصر اعداد القتلى بشكل دقيق بسبب
الحرب الشاملة التي يشنها الجيش العربي السوري
في اماكن تواجدهم.

Target sentence:

أعداد القتلى في صفوف الإرهابيين بالمئات، ومن
الصعب حصر أعداد القتلى بشكل دقيق بسبب
الحرب الشاملة التي يشنها الجيش العربي السوري
في أماكن تواجدهم.

In this example, the first line (S) is a document where a document can be a single sentence or a paragraph of multiple sentences written on a single line. The following corrections where made on the target example. (1) The first correction replaces the first token with the word اعداد. (2) The second correction (add_token) specifies an insertion of a comma in front of token 6. (3) The third correction merges tokens 6 and 7.

We evaluate model performance using the M2 evaluation script, M2Scorer (Dahlmeier and Ng, 2012). The M2Scorer compares the corrections generated by our model to the proposed gold corrections. The scorer calculates our model's precision, recall, and F1 score.

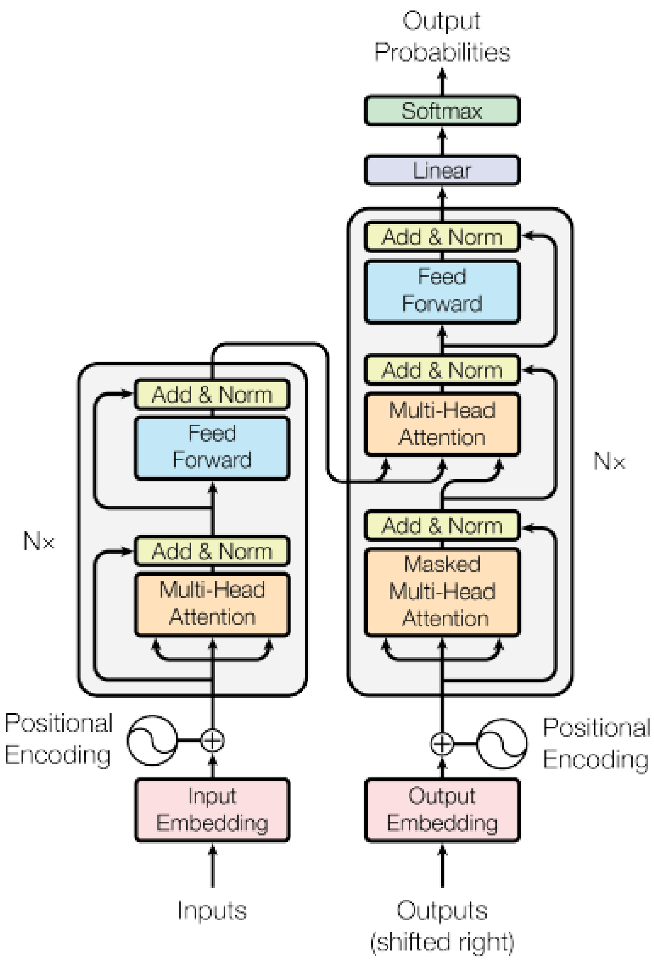

Figure 1: vaswani_wmt_en_de_big architecture

4 Methods

This experiment utilized Fairseq's pretrained transformer architectures. Following is the environment and transformer parameters used to preprocess, train, and test the GEC.

4.1 Environment

The experiment was carried out within a Google Colab notebook. The Colab employed a GPU during training and testing phases. Due to Colab's limited runtime availability, training had to be paused and then resumed from a checkpoint as to not go over the 12-hour limit per session.

4.2 Transformer

The pretrained transformer option used for this experiment was transformer_vaswani_wmt_en_de_big. This model is taken from the paper written by Vaswani et al. (2017). It was trained on the standard WMT 2014 English-German dataset which consists of 4.5 million sentence pairs (Luong and Manning, 2015).

The training hyperparameters can be seen in Table 1. Training took approximately 14 hours and a total of 97 epochs until valid-loss no longer decreased.

5 Results and Discussion

5.1 Results

The model was evaluated using the QALB-2014 test set and QALB-2015 L2 test data. We selected ten sentences from our generated corrections. The precision, recall, and F1 scores generated by the M2Scorer are reported in Table 2.

5.2 Linguistic Analysis of Performance Errors

In this section, we analyze some of the errors that were generated by the proposed model.

Hyperparameters	Values
adam-betas	(0.9,0.98)
dropout	0.3
clip-norm	1.0
criterion	label-smoothed cross entropy
label-smoothing	0.1
log-format	simple
log-interval	100
max-tokens	5000
warmup updates	8000
warmup-init-lr	1E-07
learning-rate	0.0005
learning-rate-scheduler	inverse square root
stop-min-lr	1E-09
update-frequency	16
keep-last-epochs	15
ddp-backend	no_10cd
patience	3

Table 1: Hyperparameters and values for Transformer model

CORRECT EDITS	35
PROPOSED EDITS	136
GOLD EDITS	1429
Precision	25.74%
Recall	2.45%
F1.0	4.47%

Table 2: : M2Scorer output of precision, recall, and F1 score. The table also includes the number of correct and proposed edits.

5.2.1 Multiple correction (Deletion)

The model generated multiple corrections for the sentence that involves deletions. Both suggestions were grammatical. For example:
Original: اين ذهبت تلك الأموال ؟
"Where did the money go?"
Hypothesis1: اين تلك الأموال ؟
"Where is that money?"
Hypothesis2: اين الأموال ؟
"Where is the money?"

5.2.2 Insertion of high frequent phrases

The model did not just correct error, it extended to generating new phrases that we speculate to be frequent in the corpus. For example:
Original: لكن في النهاية من كل قصة...
"but at the end of the story. . . "
Hypothesis: لكن كلى قيد الحياة ، إن شاء الله
"but for every story that is live, God willing"
In the above example, the phrase "God willing" was inserted even though it was not there in the original sentence.

5.2.3 Wrong application of nominalization

In Arabic, participle is an active non-past participial form derived from verbs. It could be used as the main verb in non-embedded clauses. Active participle can be used to describe a state of being (understanding, knowing). On the other hand, passive participles, like active participles, can act as adjectives. A passive participle may express a current state of being; a couple of examples would be (known, understood). The two forms are very close in morphology. Our model made the wrong prediction about the use of active vs. passive participle. For example:
Original: و الله قادر على أن ينصر عباده المظلومين
"Allah is able to able to help his oppressed servants "
Hypothesis: و الله قادر على أن ينصر عباده الظالمين
"Allah is able to able to help his unjust servants "
In the above example, the two words are derived from the same root but their derivations yield completely different meaning.

6 Conclusion

In this task, we expanded on previous Arabic GEC experiments. We attempted to use a state-of-the-art architecture to increase the performance and accuracy of grammar edits in Arabic, a low-resource language. Data was taken from the QALB database and divided into train, valid, and test sections. This data also underwent subword tokenization using the BPE algorithm during preprocessing. Preprocessing effectively allowed the model to lower all words replaced by <unk> to below 1% for each category.

Our model performed with a lower accuracy compared to other Arabic GEC models. However, the proposed method has great potential. Our model displayed creative edits and interesting paraphrases. Fine tuning another transformer architecture pretrained on Arabic might provide a different outcome and is something worth exploring.

References

Sina Ahmadi. 2018. Attention-based encoder-decoder networks for spelling and grammatical error correction.

Riadh Belkebir and Nizar Habash. 2021. Automatic error type annotation for Arabic. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 596–606, Online. Association for Computational Linguistics.

Daniel Dahlmeier and Hwee Tou Ng. 2012. 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*, pages 568–572, Montréal, Canada. Association for Computational Linguistics.

Minh-Thang Luong and Christopher Manning. 2015. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the 12th International Workshop on Spoken Language Translation: Evaluation Campaign*, pages 76–79, Da Nang, Vietnam.

Behrang Mohit, Alla Rozovskaya, Nizar Habash, Wajdi Zaghouani, and Ossama Obeid. 2014. The first QALB shared task on automatic text correction for Arabic. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 39–47, Doha, Qatar. Association for Computational Linguistics.

Alla Rozovskaya, Houda Bouamor, Nizar Habash, Wajdi Zaghouani, Ossama Obeid, and Behrang Mohit. 2015. The second QALB shared task on automatic text correction for Arabic. In *Proceedings of the Second Workshop on Arabic Natural Language Processing*, pages 26–35, Beijing, China. Association for Computational Linguistics.

Alla Rozovskaya, Nizar Habash, Ramy Eskander, Noura Farra, and Wael Salloum. 2014. 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)*, pages 160–164, Doha, Qatar. Association for Computational Linguistics.

Alla Rozovskaya and Dan Roth. 2019. Grammar error correction in morphologically rich languages: The case of Russian. *Transactions of the Association for Computational Linguistics*, 7:1–17.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. 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)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Aiman Solyman, Zhenyu Wang, Qian Tao, Arafat Abdulgader Mohammed Elhag, Rui Zhang, and Zeinab Mahmoud. 2022. Automatic arabic grammatical error correction based on expectation-maximization routing and target-bidirectional agreement. *Knowledge-Based Systems*, 241:108180.

Viet Anh Trinh and Alla Rozovskaya. 2021. New dataset and strong baselines for the grammatical error correction of Russian. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4103–4111, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.