Data Augmentation for Spoken Language Assessment

Santiago Dubov, Corpus Christi College

Supervisor: Dr K M Knill

1 Introduction

English is the world's most spoken language with over 1.2 billion speakers and this Figure is predicted to increase. Combined with the fact that native speakers are vastly outnumbered by non-native speakers, this has led to significant growth in the field of computer assisted language learning, giving rise to apps such as Duolingo. These tools allow learners to receive reliable and meaningful feedback on their ability e.g their grammar/pronunciation, in an instantaneous and low cost manner. This feedback can then be used by the learner to improve their level. One of the ways in which feedback can be provided to learners on their speech is through automated grammatical error correction (GEC). GEC in speech is a challenging problem for a variety of reasons. Firstly, we must use an Automatic Speech Recognition (ASR) system to create a transcription from audio, which can introduce transcription errors. In addition, unlike written language, speech includes disfluencies such as hesitations, repetitions and full sentences are not always used. As annotated speech data is limited, current GEC and Grammatical Error Detection systems are trained on written data for which large corpora exist [1]. Although this provides a good model baseline, these models perform significantly more poorly than models which have then been fine-tuned/trained entirely on speech data.

Therefore, this project will focus on the augmentation of existing data sources to better improve models aimed at correcting grammatical errors in non-native spoken English. Thus, several augmentation techniques are considered, providing pseudo speech data to improve performance of GEC systems.

2 Spoken GEC

2.1 Task

Grammatical error correction is the task of correcting a sentence to ensure that it obeys the rules of English grammar. This becomes more difficult when dealing with speech due to disfluencies as mentioned earlier. In spoken GEC a learner's speech is transformed using an ASR system into a written transcript. This transcript is then fed through a deep learning model to produce a grammatically correct sentence which can then be shown to the learner as feedback (Figure 1). It is important that the model removes or ignores the speech disfluencies such as 's-'.

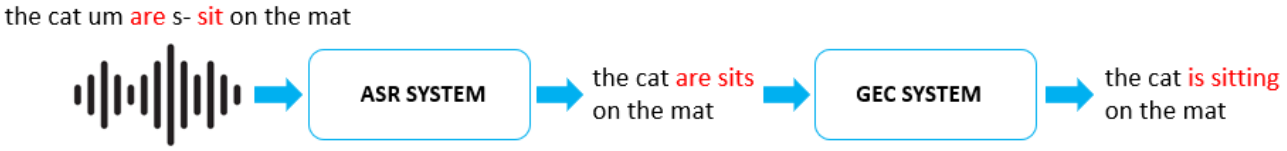

Figure 1: End-to-end spoken GEC system

3 Baseline Spoken GEC system

3.1 Model Structure

The proposed system is composed of a deep neural network which in this case is a transformer based on [2]. It is a sequence-to-sequence deep learning model that uses multihead attention to keep track of long range dependencies and an encoder-decoder architecture. Input text is first transformed into vectors known as word embeddings which are then passed into the model. Neural networks require a large amount of labelled data for training. However, labelled speech data is difficult to obtain as each sentence must be transcribed and annotated manually by someone trained in grammatical error correction. Even though a large amount of work has been done to produce corpora for written GEC, the number of spoken corpora is very limited. Consequently, the baseline model is trained on large written corpora and then evaluated on smaller spoken corpora. It is found that the performance achieved is noticeably lower than a model which has been fine-tuned on speech data. Thus, this motivates the use of augmentation techniques to try to create data that more closely matches speech data using the data resources available.

4 Data Augmentation

To investigate the possibilities for data augmentation, we first examine the corpora that we have available. The largest and most abundant corpora are written texts with grammatical error annotations [3] [4]. Additionally, several corpora composed of unlabelled native speech are also available. The former performs poorly due to the differences between speech and written text such as disfluencies. The latter is unusable in its current form as labelled data is required to train the neural networks. The two main methods for data augmentation which will be explored in this project are:

- Generation of grammatical errors in transcribed native speech corpora effectively transforming native speech into learner speech.
- Propagation of disfluencies in labelled learner written corpora to make more speech like.

Both of these methods aim to produce data equivalent to an annotated learner speech corpus but will still produce samples that would not occur naturally. A key part of this work will reside in the over generation of this augmented data followed by filtering to retain the most in-domain data.

4.1 Grammatical Error Generation

The generation of grammatical errors has been used before for data augmentation, such as in the work done by Manakul [5]. N-gram error statistics are used to create errors in speech corpora. Reverse neural machine translation is another method, which can be used to create grammatical errors by running the models described previously in reverse. However, as initial tests have shown this method to be unreliable in nature, the method in [6] is implemented here and henceforth referred to as the Kakao method. In this approach we create an error dictionary containing the most common errors found in a reference corpus and re-propagate them into a native speech corpus. This is similar to the n-gram error statistics approach, however, we also introduce an additional probability of a never before seen error occurring in nouns, verbs and prepositions. Some examples of augmented sentences using the data detailed in section 5.2 are shown below.

Original: it is impossible to reach any place early in the morning because of the traffic jams

Augmented: it is impossible to reach any place early in the morning because of with the traffic jams

Original: I hope you can help me

Augmented: I wish you can help me

4.2 Speech Disfluency Propagation

To create speech disfluencies in written corpora we focus on two types of disfluency, repetitions and false starts (the learner starts to respond but then stops and starts a new sentence). We define the maximum number of disfluencies in a sentence and the maximum length of a disfluency. We choose locations at random in the sentence and create a disfluency of a random length. This can occur either by repetition of the preceding words or by using a masked language model (MLM). A MLM is a large pre-trained model designed to predict the words hidden by a 'mask' in a sentence. Thus, by placing a mask token at this position the model gives us something akin to a false start which can be inserted into the original sentence. In this work, the Roberta MLM [7] was used.

4.3 Language Model Filtering

In some cases, it is possible that an error sequence or disfluency could be generated which is highly unlikely to be made by a learner. Therefore, a method of filtering generated sentences to yield a better-matching data set is needed. The approach adopted so far is to train a language model on learner speech which can then measure the similarity between the augmented sentences and learner speech. Filtering is then used to remove highly unlikely phrases. The model that is used here is an n-gram language model which assumes the probability of observing the sentence $w_1, \ldots, w_m$ as given by Equation 1. Thus, the model yields a measure of the probability of a sequence given the training data. These models have the advantage that they do not require data with corrections to be trained thus we can use un-annotated learner speech transcriptions.

$$P\left(w_1, \ldots, w_m\right) = \prod_{i=1}^{m} P\left(w_i \mid w_1, \ldots, w_{i-1}\right) \approx \prod_{i=1}^{m} P\left(w_i \mid w_{i-(n-1)}, \ldots, w_{i-1}\right) \tag{1}$$

$$PP(W) = \frac{1}{p(w_1, \ldots, w_m)}^{\frac{1}{M}} \tag{2}$$

To carry out filtering we calculate the perplexity of each sentence using Equation 2. Low perplexity indicates the probability distribution is good at predicting the sample. We then have two choices for filtering our sentences, to use an absolute threshold of perplexity and only accept sentences below this threshold, or to look at the perplexity difference between the augmented source and target sentence. In both cases a high value of perplexity indicates a highly unlikely construction.

5 Experiments

5.1 Data

The corpora used in this work are all manually annotated for errors and the spoken corpora are all manually transcribed. The written corpora are taken from examination scripts and language learning websites: Cambridge Learner Corpus [3], BEA Challenge: Lang-8, W&I,

LOCNESS [4]. For our experiments punctuation and capitalisation are removed, and spelling errors corrected to match the ASR output format. Thus, as previously mentioned, we use the written corpora to train the baseline model and the labelled spoken data sets (NICT, BULATS) to evaluate the model's performance. Switchboard is composed of transcriptions of native speaker telephone conversations.

Corpus	Written or Spoken	Data size # tokens	GEC labels	non-native
CLC	Written	25.0M	✓	✓
Lang-8	Written	11.0M	✓	✓
W & I	Written	800K	✓	✓
LOCNESS	Written	1.00M	✓	x
NICT	Spoken	157K	✓	✓
BULATS	Spoken	64.0K	✓	✓
Switchboard (SWBD)	Spoken	940k	x	x

Table 1: Data sets used in experiments

- **NICT-JLE**: Transcriptions of 167 oral proficiency tests for Japanese learners.
- **BULATS-EVAL3**: Transcriptions of the long free speaking section of the LinguaSkill Business English test.

5.2 Set up

The default parameters of 6 layers, 8 heads and 20% dropout as described in [2] were used for the transformer. Input words were mapped to randomly initialised word embeddings of dimension 512. The model used was borrowed from Yiting Lu and was coded in PyTorch. All data processing, filtering and grammatical error generation were implemented by the author. The Kakao method was implemented following the algorithms shown in [6]. We use an error dictionary created using the CLC corpus to corrupt Switchboard. The GEC performance metric used in this work is GLEU [8] which agrees strongly with the performance of GEC systems evaluated by humans. GLEU considers the overlap of n-grams in the correction and the reference. The 5-gram language model that was trained using the KenLM [9] toolkit was used for filtering. The data used for training was un-annotated learner speech taken from BULATS group C-E data that was not part of the evaluation set described previously. The generation of speech disfluencies was achieved using code provided by Yiting Lu.

5.3 Results

5.3.1 Baseline

A model was trained on the CLC and another on all the written data in Table 1 to measure the effect of the quantity of data on performance. The results are presented in Table 2. We see that the model's performance improves only marginally even when 50% more data is used.

Model Training Corpora	Data Size # tokens	NICT	BULATS
CLC	25.0M	0.475	0.493
CLC+BEA	37.8M	0.477	0.498

Table 2: GLEU scores for models trained on written data evaluated on spoken test sets. 'BEA' represents the combination of the Lang8, W&I and LOCNESS corpora.

To provide a valid comparison for the performance that could be obtained if labelled speech data was available, we applied K-fold cross validation fine-tuning our previous model on the NICT corpus. We split the corpus into 5 folds and concatenated our predictions for each fold before calculating a GLEU score shown in table 3. We observed that fine-tuning on speech data yields a drastic increase in performance on the spoken evaluation sets. This provides us with a good estimate for results that would be obtained for a model trained solely on speech data and motivates data augmentation.

Fine tuning	NICT
no	0.477
yes	0.598

Table 3: Effect of fine-tuning on speech data (K fold cross validation) for a model trained on written data (CLC + BEA).

5.3.2 Grammatical Error Generation

Preliminary experiments showed that filtering by absolute perplexity provided data with the greatest similarity to authentic speech corpora, so was used in the experiments reported here. The baseline CLC+BEA model was subsequently fine-tuned for an additional 3 epochs on 3 different versions of the augmented Switchboard. GLEU scores are shown for the best epoch in Table 4. Firstly we notice that despite the lower perplexity evaluated by our proposed language model, which showed that Switchboard, when augmented resembles the speech test sets more than the written data, fine-tuning results exhibited a decrease or, in the case of heavy filtering, a very small increase on BULATS. However, the results clearly demonstrate that filtering positively affects our results and that the more we filter the closer the domain of our augmented data to our evaluation data.

Fine tuning	Filtering	Perplexity threshold	Data Size # tokens	NICT	BULATS
no	no	N/A	37.8M	0.477	0.498
yes	no	N/A	940k	0.453	0.490
yes	light	250	480k	0.458	0.491
yes	medium	150	194k	0.468	0.497
yes	heavy	80	64.3k	0.474	0.504

Table 4: GLEU scores for the best epoch of the baseline CLC+BEA model fine-tuned on the augmented Switchboard with filtering applied.

GEG appears, at best, provide us with results similar to those obtained by training on a written corpus with no fine-tuning. Two contributing factors may be the cause of this. Firstly, although the native speech corpora contain disfluencies, they are present in both the source and the target and are never corrected. Thus, the model does not learn to filter these out and additionally learns incorrect relationships involving 'normal speech' such as repetition of words being normal. Secondly, while it is clear that filtering helps to remove unlikely error sequences and obtain an augmented corpus more similar to a learner corpus, it is impossible to fully replicate the distribution and type of errors found in a spoken corpus. Furthermore, the error distribution is specific to each corpus ;therefore we cannot say that the errors should be similar for corpora with different vocabularies dealing with different subjects. To conclude, grammatical error generation can be used to generate errors and make use of native speech corpora for fine-tuning of GEC models with some efficacy but will only ever be an approximation to a natural error process.

5.3.3 Speech disfluency generation

To investigate the effect of disfluencies, they were generated in the CLC corpus and then the corpus was filtered using the language model described previously. A maximum of 2 disfluencies each with a maximum length of 5 tokens were allowed to occur per sentence. E.g, "i need to use my computer to **do company** do company projects". Results are shown in Table 5. We see that due to the high percentage of disfluencies in NICT, the model improves substantially on the baseline. This initial experiment shows us that disfluency generation can be an effective method of increasing spoken GEC performance. However, as the results on BULATS depict, more work is needed to understand and apply this to all data sets.

Fine Tuning	Data Size # tokens	NICT	BULATS
No	37.8M	0.477	0.498
Yes	1.6M	0.566	0.485

Table 5: The baseline CLC+BEA model was fine-tuned for an additional 3 epochs on a disfluent version of CLC that had been filtered.

6 Conclusion and Next Steps

Grammatical error generation enables the use of native speech corpora to provide augmented learner data. However, results perform similarly to models trained on written data as they are unable to handle speech disfluencies. Generating speech disfluencies in written corpora and fine-tuning on this data can lead to large performance increases if the spoken evaluation set is highly disfluent. However, this method has yet to be investigated fully. Future work will include a detailed investigation into the propagation of speech disfluencies by considering the percentage of the corpus which contains disfluencies as well as disfluency length and number. The effect of filtering on a disfluent corpus will also be considered. Finally, GEG will be carried out on other native speech corpora to verify results found here using Switchboard.

References

[1] K. M. Knill et al. Automatic Grammatical Error Detection of Non-native Spoken Learner English. In *ICASSP 2019*, pages 8127–8131, 2019.

[2] Ashish Vaswani et al. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[3] Diane Nicholls. The Cambridge Learner Corpus: Error coding and analysis for lexicography and ELT. In *Corpus Linguistics 2003 conference (Volume 16)*, pages 572–581, 2003.

[4] Christopher Bryant et al. The BEA-2019 shared task on grammatical error correction. In *Proc. 14th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–75, 2019.

[5] P. P. Manakul. Automatic Assessment of English as a Second Language. 2019.

[6] Yo Joon Choee et al. A neural grammatical error correction system built on better pre-training and sequential transfer learning. In *Proc. 14th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 213–227, 2019.

[7] Yinhan Liu et al. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.

[8] Courtney Napoles et al. Ground truth for grammatical error correction metrics. In *Proc. 53rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 588–593, 2015.

[9] Kenneth Heafield. KenLM: Faster and smaller language model queries. In *Proc. 6th Workshop on Statistical Machine Translation*, pages 187–197, 2011.