Detection of Typical Sentence Errors in Speech Recognition Output

Bohan Wang[1,†], Ke Wang[1,†], Siran Li[1,†] and Mark Cieliebak[2,*]

[1]Section of Electrical and Electronic Engineering, École Polytechnique Fédérale (EPFL), Lausanne, Switzerland
[2]Centre for Artificial Intelligence, Zurich University of Applied Sciences (ZHAW), Winterthur, Switzerland

Abstract
This paper presents a deep learning based model to detect the completeness and correctness of a sentence. It's designed specifically for detecting errors in speech recognition systems and takes several typical recognition errors into account, including false sentence boundary, missing words, repeating words and false word recognition. The model can be applied to evaluate the quality of the recognized transcripts, and the optimal model reports over 90.5% accuracy on detecting whether the system completely and correctly recognizes a sentence.

1. Introduction

Automatic Speech Recognition (ASR) systems develop technologies to recognize and translate spoken language into text by machines [1]. Sentence error detection on ASR systems is important for the two reasons: a) This can help to set proper punctuation marks; b) For multiple speakers, speaker recognition often fails at the change between two speakers, which results in single words at beginning or end of an utterance being assigned to the wrong person. A practical application domain of our work is to detect complete and correct sentences in ASR systems to mitigate the aforementioned problems.

In prior works, research focused mainly on grammatical error detection [2, 3]. In this paper, we focus on dealing with the specific errors emerging in speech recognition, such as missing words or incorrect sentence boundaries (detailed in Sec. 3.3). In addition, previous works on enriching speech recognition emphasize on finding correct sentence boundaries in whole transcripts [4, 5]. However, in real-time speech recognition, we have access to only individual sentences instead of full transcripts, and they don't take other typical speech recognition errors (apart from incorrect sentence boundaries) into account [6].

Recently, transformer models have shown state-of-art performance in generating word embeddings and extracting intrinsic features of word sequences. In specific, Bidirectional Encoder Representations from Transformers (BERT) [7], Generative Pre-trained Transformer (GPT) [8] and BIG-BIRD [9] have achieved promising performance to learn high quality language representations from large amounts of raw text. The token representations produced by these transformers pre-trained on unsupervised tasks also help improve the performance of a supervised downstream task.

In this paper, we fine-tune the pre-trained transformers (BERT, GPT2 and BIG-BIRD) on the speech recognition error detection task, to build a binary classification model detecting speech recognition errors. The performance of sequentially linking BERT embedding and a down-stream text classification network is also studied. We compare and analyze the performances of several classification models. The models are ensembled through a Random Forest to further improve the performance. Finally, we analyse the performance of BERT-based classifier on a multi-label dataset.

The paper is structured as follows: In Sec. 2, we explain the models and experimental design. In Sec. 3, we describe how the dataset is generated. We discuss the experimental results in Sec. 4.

2. Methods

2.1. Models

In this section, we use three state-of-art transformer models BERT [7], GPT2 [8], BIG-BIRD [9] are considered.

Besides, we also test the performance of using BERT embedding plus a downstream text classification network. For the classification networks, we use either a bi-direction LSTM and a TextCNN. We use a one-layer TextCNN with kernels sizes to be 2, 3 and 4. For LSTM, we use a one-layer bi-directional LSTM network [10], followed by an attention layer and a fully connected layer. The number of hidden states is 256. Specifically, the attention layer is found to be essential.

SwissText 2022: Swiss Text Analytics Conference, June 08–10, 2022, Lugano, Switzerland
*Corresponding author.
†These authors contributed equally.
bohan.wang@epfl.ch (B. Wang); k.wang@epfl.ch (K. Wang); siran.li@epfl.ch (S. Li); ciel@zhaw.ch (M. Cieliebak)

2.2. Ensemble learning

We ensemble the five trained classifiers with random forest. Configuration and the final classification performance are shown in Sec. 4.2.

3. Data preparation

3.1. Dataset sources

For the model to have better generalizing capacity, a training set from diverse sources covering diverse topics and occasions is necessary. The following corpora are included in our proposed dataset:

News reports [11]: 143, 000 articles from 15 American publications

Ted 2020 Parallel Sentences Corpus [12]: around 4000 TED Talk transcripts from July 2020

Wikipedia corpus [13]: over 10 million topics

Topical-Chat [14]: nearly 10 thousand human dialog conversations spanning 8 broad topics

3.2. Dataset Creation

To make the selected datasets suit our speech recognition model, we remove some non-English tokens, sentence ending symbols ('.', '!', '?'), duplicated sentences and also short sentences (less or equal to 5 words) to avoid some recognition errors. After pre-processing on the data from the sources, we create the following two datasets:

Standard Dataset: contains 0.3 million sentences from News reports, 0.3 million sentences from Ted corpus, 0.3 million sentences from Wikipedia corpus, 0.2 million sentences from Topical-Chat, in total 1.1 million sentences. We split the Standard Dataset randomly over all data sources into train set, ablation set and test set, with a proportion of 8:1:1.

Large Dataset: contains 2.3 million sentences from News reports, 0.4 million sentences from Ted corpus, 2 million sentences from Wikipedia corpus, 0.2 million sentences from Topical-Chat; in total 5 million sentences. We split it into train and test set, with a proportion of 19:1.

We train and compare performances of various models on the Standard Dataset. As a comparison, we evaluate the performance of BERT trained on the large dataset to see how an enlarged training set affects generalization ability for this task.

3.3. Generate positive and negative samples

For creating positive samples, punctuation is removed (except abbreviations such as it's, Mr., I've, etc.) and words are converted to lower case.

For creating negative samples, we mimic typical errors of the speak recognition system, which are detailed in the following, and we propose corresponding methods to create negative samples with respect to typical errors.

False sentence boundary: When a speech recognition system fails to correctly separate two sentences, the first sentence would be cut off in the middle and part of the sentence would be assigned to the next sentence (illustrated in Fig. 1 (a)). For such negative samples, we group the sentences by three, and randomly separate the three sentences into 2-4 sentences (so that on average negative samples created in this way would have equal length with positive samples). While choosing random separating points, the genuine sentence separations points, punctuation and typical words for starting subsentences (e.g. that, which, because, etc.) are avoided, and thus reduce the probability that a generated sample is still a complete sentence by chance (e.g. 'I like you because you are beautiful' to 'I like you'.)

Missing words: A speech recognition system can fail to recognize one or several words from a sentence, and as a result some words may be missing in the produced transcripts (Fig. 1 (b)). For such negative samples, we randomly remove 1 word for sentences up to 3 words, and 2-4 words from longer sentences.

Repeating words: The system can record speakers' unintended repeated words (Fig. 1 (c)). For such negative samples, we randomly repeat 1 word for sentences within 3 words, and 1-3 words from longer sentences.

False word recognition: The system can mistakenly recognize one word as another word (Fig. 1 (d)). For such negative samples, we randomly replace 1 word for sentences within 3 words, and 1-3 words from a longer sentences, by random words from another sentence.

Finally, the punctuation is removed and words are converted to lower case.

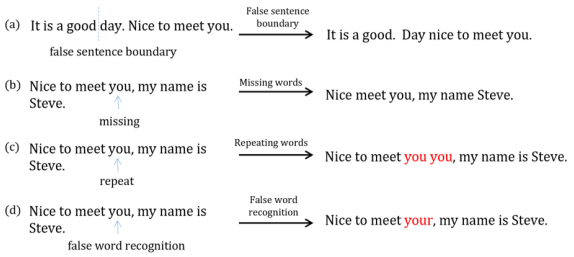

Figure 1: Typical errors in speech recognition system

After creating the positive and negative samples, the sentences longer than 100 words are removed, for they are too long to appear in speech recognition. We create the same number of negative samples as that of positive samples, so that we have a balanced dataset. The ratio between different types of negative samples is 2:1:1:1. The type *False Sentence Boundary* corresponds to two times

the number of other negative sample types since *False Sentence Boundary* contains two types of false sentences, those which are cut off and those which are assigned with extra words.

4. Experiments and Discussion

In this section, we report the results of our experiments. We describe below the setup, and then evaluate the different models in Sec. 4.1. In Sec. 4.2, based on the models, we train a Random Forest classifier to further aggregate the models and improve the performance. In Sec. 4.3, we compare the performance of BERT trained on Standard and Large Dataset. Finally, we show the result of BERT trained on a Multi-Labeled Dataset in Sec. 4.4.

Training details: We train each model for 5 epochs with batch size 64 using Adam optimizer. The initial learning rate is set as $3e-5$ for fine-tuning transformer models and $1e-3$ for downstream classification networks. To prevent overfitting, we only save the model with optimal performance on test set after each epoch.

4.1. Results on Standard Dataset

As explained in Sec. 2, we train five models on the Standard Dataset containing 1 million proper sentences and 1 million non-proper sentences to evaluate their performances.

The results of this experiment are presented in Table 1.

Model	Test Accuracy
BERT	89.27%
GPT-2	88.67%
BIG-BIRD	**90.26**%
BERT embedding + Bi-LSTM	86.33%
BERT embedding + TextCNN	81.40%

Table 1
Test accuracy of five models on Standard Dataset

From the results, we can see that the transformers provide much better results than the models sequentially linking BERT embedding and either a BiLSTM or TextCNN. Specifically, BIG-BIRD provides the optimal performance, with 90.26% test accuracy. BERT and GPT2 provide similar test accuracy, 89.27% and 88.67% respectively.

4.2. Ensemble learning with Random Forest

In this section, we combine the five trained models (in Table 1) with random forest in order to produce one optimal predictive model. The idea of the ensemble learning is to train a random forest classifier with the combination of the predicted classes from the models. The random forest classifier can generate a final classification through a majority vote mechanism.

To prevent random forest from overfitting the train set, we use a separate ablation set, instead of the train set which the models are trained on. The best parameters after 10-fold cross-validation are 100 decision trees, and a maximum depth of 3. The test accuracy of the random forest reaches 90.51%, higher than the optimal accuracy among the individual models (90.26%), but not to a large extent. This is probably since the transformers (along with their embedding) share similar structures and do not diverge much on decisions.

4.3. Results on Large Dataset

In this section, we train BERT on the large dataset (5 times the size of the Standard Dataset) with less epochs (1 epoch in contrast to 5 epochs). Overall, the model is trained with the same iterations as with Standard Dataset. With the same training details described before (but only for one epoch), results show that training with Large Dataset provides a higher test accuracy (90.36%), compared with the accuracy trained with Standard Dataset (89.27%).

The results suggest that, provided with enough computational capacity, we can further improve our model's generalization ability by training on a larger dataset.

4.4. Result on multi-label dataset

In this section, we further create a Multi-Label Dataset, which contains the same samples as the Standard Dataset, whereas the negative samples are distinctively labeled (including *false sentence boundary*, *false word recognition*, *missing words*, and *repeating words*) instead of uniformly labeled as *negative*.

We train a BERT model on this dataset, and it reached 85.01% classification test accuracy. The precision, recall and F1-score of each class is given in Table 2.

Sample Class	Precision	Recall	F1 Score	Support
Complete Sentence	0.87	0.94	0.90	109857
False Sentence Boundary	0.83	0.81	0.82	42677
False Word Recognition	0.84	0.70	0.77	21897
Missing Words	0.64	0.50	0.56	21711
Repeating Words	0.96	0.99	0.98	21781

Table 2
Precision, Recall and F1-Score of each sample class

From the result, we can see that the simplest task is to identify repeated words in the sentences (F1-score near 0.98). Identifying complete sentences is also a relatively easy task, with a F1-score of 0.90. The hardest task for the model is detecting whether there are missing words in the sentence. It achieves only 64% precision and 50% recall on this task.

The confusion matrix is drawn in Fig. 2. From this figure, we can further see that the classifier finds it difficult to classify between complete sentences and sentences

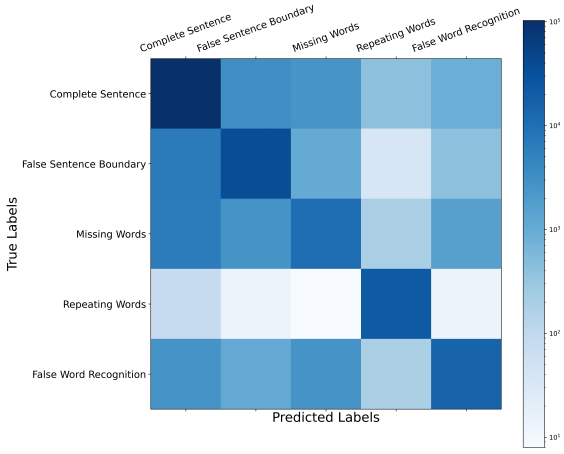

Figure 2: Confusion matrix for BERT trained on Multi-Label Dataset

with missing words, even though in most of the cases more than one word is missing in the erroneous sentences. This is understandable because in most cases, not every word is indispensable, even we lose some words, and maybe the meaning is not exactly the same but the sentence still makes sense grammatically.

4.5. Result on real-world ASR outputs

Finally we test our trained multi-modal BERT model on the real-world ASR outputs from CEASR corpus [15]. The predictions are shown in Fig. 3, where we can see the model is able to capture real-world ASR errors correctly, while we also provide an example where the model fails.

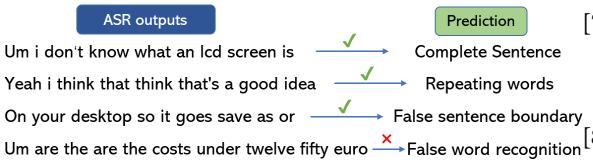

Figure 3: Prediction on real-world ASR outputs

5. Conclusion

In this paper, a dataset for detecting speech recognition errors was created, where four different types of typical speech recognition errors were taken into account. Experimental results show that transformer models are capable of providing good performance on classification of the constructed dataset for speech recognition error, reporting approximately 90% accuracy for BERT, GPT2 and BIG-BIRD. A Random Forest was trained based on the five models, and further improved the test accuracy to over 90.51%. Overall, the results suggest that using state-of-art transformer models can provide good quality for detecting the errors in speech recognition systems, and provide feedback on further improvements of speech recognition systems. In our future works, special adjustments might be needed to better cope with identifying missing words in recognized sentences.

References

[1] D. Yu, L. Deng, Automatic speech recognition, volume 1, Springer, 2016.

[2] N. Agarwal, M. A. Wani, P. Bours, Lex-pos feature-based grammar error detection system for the English language, Electronics 9 (2020) 1686.

[3] Z. He, English grammar error detection using recurrent neural networks, Scientific Programming 2021 (2021).

[4] Y. Liu, E. Shriberg, A. Stolcke, D. Hillard, M. Ostendorf, M. Harper, Enriching speech recognition with automatic detection of sentence boundaries and disfluencies, IEEE Transactions on Audio, Speech, and Language Processing 14 (2006) 1526–1540. doi:10.1109/TASL.2006.878255.

[5] Y. Liu, A. Stolcke, E. Shriberg, M. Harper, Using conditional random fields for sentence boundary detection in speech, in: Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05), 2005, pp. 451–458.

[6] D. Tuggener, A. Aghaebrahimian, The Sentence End and Punctuation Prediction in NLG text (SEPP-NLG) shared task 2021, in: Swiss Text Analytics Conference–SwissText 2021, Online, 14-16 June 2021, CEUR Workshop Proceedings, 2021.

[7] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[8] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al., Language models are unsupervised multitask learners, OpenAI blog 1 (2019) 9.

[9] M. Zaheer, G. Guruganesh, K. A. Dubey, J. Ainslie, C. Alberti, S. Ontanon, P. Pham, A. Ravula, Q. Wang, L. Yang, et al., Big Bird: Transformers for Longer Sequences., in: NeurIPS, 2020.

[10] F. A. Gers, J. Schmidhuber, F. Cummins, Learning to forget: Continual prediction with lstm, Neural computation 12 (2000) 2451–2471.

[11] A. Thompson, All the news: 143,000 articles from 15 American publications, =https://www.kaggle.com/snapcrack/all-the-news, 2017.

[12] N. Reimers, I. Gurevych, Making monolingual sentence embeddings multilingual using knowledge distillation, arXiv preprint arXiv:2004.09813 (2020).

[13] W. Foundation, Wikimedia downloads, ???? URL: https://dumps.wikimedia.org.

[14] K. Gopalakrishnan, B. Hedayatnia, Q. Chen, A. Gottardi, S. Kwatra, A. Venkatesh, R. Gabriel, D. Hakkani-Tür, A. A. AI, Topical-chat: Towards knowledge-grounded open-domain conversations., in: INTERSPEECH, 2019, pp. 1891–1895.

[15] M. A. Ulasik, M. Hürlimann, F. Germann, E. Gedik, F. Benites de Azevedo e Souza, M. Cieliebak, Ceasr: a corpus for evaluating automatic speech recognition, in: 12th Language Resources and Evaluation Conference (LREC) 2020, European Language Resources Association, 2020, pp. 6477–6485.