Error Correction in ASR using Sequence-to-Sequence Models

RENAUD Thomas

December 6, 2023

Abstract

Advancements in Automatic Speech Recognition (ASR) have been astonishing, yet challenges persist in achieving flawless transcriptions. This article explores a novel approach to implement an ASR error correction using Sequence-to-Sequence models. Inspired by Samrat Dutta's paper on "Error Correction in ASR using Sequence-to-Sequence Models", our goal is to fine-tune a Bidirectional Auto-Regressive Transformer (BART) to rectify errors introduced by an ASR model. The absence of a dedicated error dataset containing plausible ASR errors is a challenge. To address this, we propose an innovative solution: use a pre-existing ASR dataset to simulate errors through naive predictions based on audio records. This approach allows us to generate a tailored dataset for fine-tuning BART. This fine-tuned model, equipped with the ability to recognize and rectify errors, serves as an effective post-editing tool, contributing to the overall refinement of ASR transcriptions.

Keywords: ASR, Post-editing, Transformers, BART, Wav2Vec2

1 Introduction

Refining ASR models directly by improving their architecture can become a daunting task. ASR architecture, often highly complex and finely tuned, make it challenging to implement direct refinements that guarantee improved performance. Rather than attempting to overhaul the entire architecture, the article proposes a more pragmatic solution: integrating a model endowed with an inherent understanding or intuition about language. This heuristic approach increase awareness on common errors generated by ASR models, such as word boundary disambiguation, phonetic confusion, and spelling mistakes (see Table 1).

Table 1: ASR output for the corresponding speech, showing typical speech-errors in an ASR system. The various errors are as follows: Spelling mistake, Character dropped, word boundary error, Grammatical error

Speech	There is no alternative to that restaurant across the street that played jazz
ASR	There is no altnative to that restauran a cross the street that play jazz

*Table extracted from [2]

By introducing a model with linguistic intuition, the aim is to provide a supplementary layer of comprehension that can navigate and rectify the nuanced errors arising in ASR transcriptions. This approach acknowledges the complexity of language patterns, allowing the model to make informed decisions based on contextual understanding, thereby addressing specific error categories.

2 Our Approach

2.1 Baseline Strategy

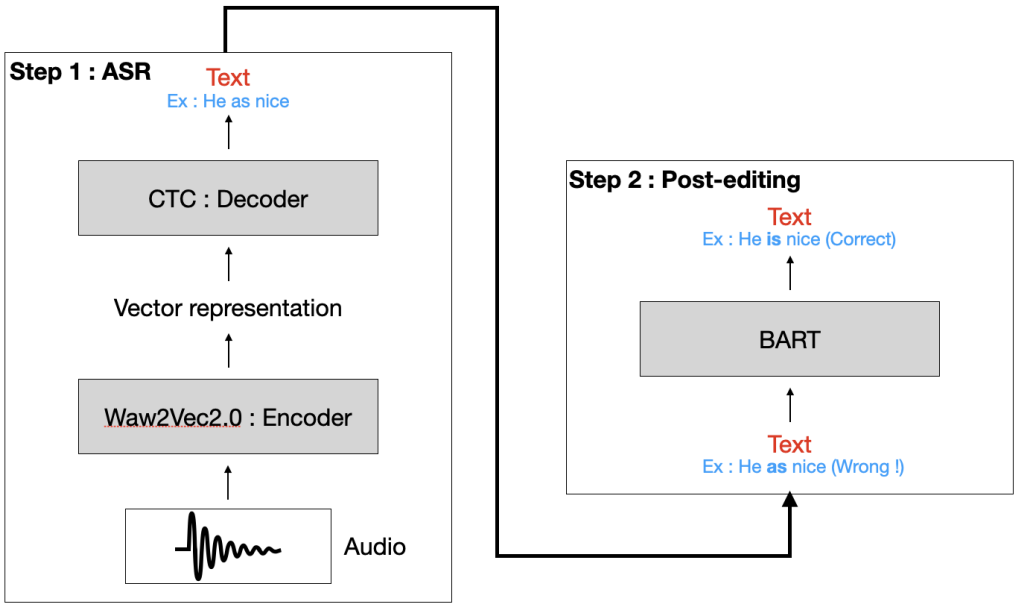

Figure 1: Schematic diagram illustrating the fine-tuning workflow of BART.

The lack of a dataset containing plausible errors from an ASR model sets the stage for a two-phase solution. The first phase involves the creation of a unique dataset, while the second phase focuses on fine-tuning BART using this dataset (see Figure 1).

2.2 Wav2Vec2

To generate potential common ASR errors independently of specific ASR architectures, we employ a pretrained wav2vec2 model [1] as the foundational ASR system. This model generates predictions for our chosen training set, Timit [3]. Given, wav2vec2 model is originally trained on LibriSpeech-960-hr, while all our samples are derived from the Timit corpus, finetuning is essential. This strategy enables to create naive prediction from audio records, forming the basis for the subsequent fine-tuning of BART.

2.3 Dataset Generation

To identify common ASR errors independently of the underlying ASR architecture, a fine-tuned wav2vec2 [1] model is employed as a base ASR system.

The dataset creation process deliberately produces errors through the utilization of a high word error rate ASR model. Through the concatenation of these intentionally corrupted predictions with the ground truth sourced from a comprehensive ASR dataset, our final dataset for fine-tuning BART is meticulously crafted. This hybrid dataset, encompassing both common errors and accurate transcriptions, serves as the training foundation to fine-tune BART for rectifying errors in ASR predictions.

2.4 Fine-tuning of BART

The fine-tuning of BART [4] involves employing the pretrained model as an ASR correction model. We use Hugging Face[1] implementation of 'bart-base' model. The ASR hypothesis is fed into BART,

[1]https://huggingface.co/facebook/ bart- base

and the output serves as the final transcription. The idea of Samrat Dutta behind this fine-tuning is to "leverage a sequence-to-sequence denoising autoencoder to correct outputs in ASR systems on English language" [2]. The fine-tuning workflow we use is a simplified version of the original proposal, we removed the data augmentation process and the addition on the phoneme sequence.

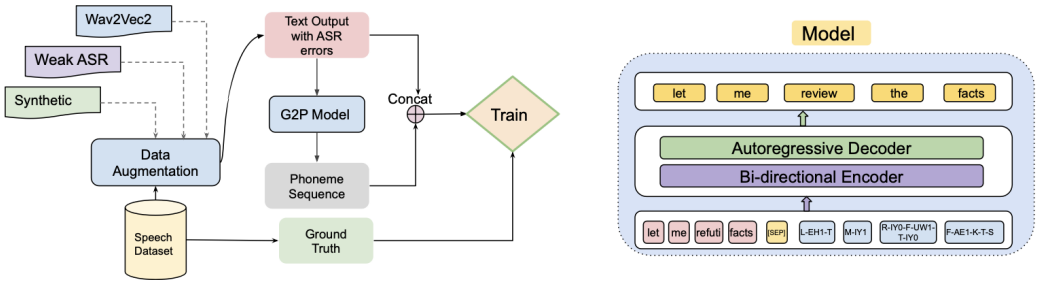

Figure 2: Schematic diagram illustrating the workflow of ROBART.
*Figure extracted from [2]

3 Experimental Setup

3.1 Dataset

TIMIT Acoustic-Phonetic Continuous Speech Corpus [3] is a standard dataset used for evaluation of automatic speech recognition systems. It consists of recordings of 630 speakers of 8 dialects of American English each reading 10 phonetically-rich sentences. It also comes with the word and phone-level transcriptions of the speech.

3.2 Implementation Details

Implementation details adhere to the recommendations of Samrat Dutta. This includes a "learning rate of 3×10^{-5}, AdamW optimizer, weight decay of 0.1, warm-up of 0.1, and a maximum sequence length of 35. During decoding, the beam size is set to 10. Fine-tuning involves randomly masking 15% of tokens to improve predictions within context." [2]. The models were trained for 10 epochs on Tesla K80 GPU.

3.3 Hardware Resources

The training process leverages Free Google Colab, a cloud-based platform providing access to high-performance Tesla K80 GPUs. The collaborative features and integration with Jupyter notebooks streamline code development, eliminating the need for local computational resources. With this configuration, we achieved an average training time of 15s per epoch.

4 Results and Analysis

The experimental results showcase instances where BART successfully corrects ASR predictions. In Table 2, we list a variety of errors that BART model is able to correct. This includes fixing phonetically confused word sequences in (2), identifying valid word boundaries in (2) & (4), fixing spelling errors in (1) & (3), etc.

We present our main results in Table 3. BART using the weak ASR predictions outperforms the weak ASR system. This improvement can be explained by the fact that ASR errors are speech-sensitive, and the stength of BART reside in its powerful denoising autoencoder. A significant reduction in Word Error Rates (WERs) is observed when combining ASR with BART compared to ASR predictions alone (see Table 3).

Table 2: ASR predictions perfectly corrected by fine-tuned BART.

ASR	(1a) destroy every file related to my audice
BART	(1b) destroy every file related to my audits
ASR	(2a) dissymber and jenuaerry are nice months to spind in my am
BART	(2b) december and january are nice months to spend in miami
ASR	(3a) the misquoat was retracted with an appoligy
BART	(3b) the misquote was retracted with an apology
ASR	(4a) youngsters love commoncandiy as treats
BART	(4b) youngsters love common candy as treats

Table 3: Word Error Rates of the proposed method compared to weak ASR

Model	**Word Error Rate (%)**
ASR Only	40.8
ASR + BART	9.9

5 Conclusion

In conclusion, this article presents a comprehensive approach to error correction in ASR using Sequence-to-Sequence Models. The methodology effectively addresses the challenge of limited supervised training data by employing realistic data generation techniques.

Experimental results demonstrate a substantial reduction in Word Error Rates (WERs), indicating the efficacy of the proposed approach. While the results may not directly replicate those of the original paper, the implementation achieves the overarching goal of error correction in ASR.

As the field of Natural Language Processing (NLP) continues to evolve, further exploration and understanding of model nuances are imperative. This project serves as a stepping stone, showcasing the potential for improving ASR accuracy through specialized error correction models.

The source code is available on my Github : here

References

[1] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. 2020.

[2] Samrat Dutta, Shreyansh Jain, Ayush Maheshwari, Souvik Pal, Ganesh Ramakrishnan, and Preethi Jyothi. Error correction in asr using sequence-to-sequence models. 2022.

[3] William M. Fisher Jonathan G. Fiscus David S. Pallett Nancy L. Dahlgren Victor Zue John S. Garofolo, Lori F. Lamel. Timit acoustic-phonetic continuous speech corpus. 1993.

[4] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. 2019.