A Framework for Punctuation Restoration Benchmarking

Viet Dac Lai, Franck Dernoncourt, Thien Huu Nguyen

University of Oregon, Adobe Research, USA

{vietl}@uoregon.edu

Abstract

As a result of the considerable improvement of automated speech recognition (ASR) over the past few years, there exist many online APIs for ASR. In addition to transcribing the audio recordings, these APIs typically also perform punctuation restoration, which is the task of inserting punctuation marks into a non-punctuated text. It is often time-consuming to evaluate how the performance of the punctuation restoration provided by these API compare with each other, and against newly proposed algorithms. In this paper, we present a lightweight framework that allows users to easily benchmark punctuation restoration APIs on several corpora.

Index Terms: punctuation restoration, speech recognition, benchmark

1. Introduction

Punctuation restoration is the task of restoring fundamental text structures, such as sentences and phrases by inserting punctuation marks into non-punctuated texts, e.g. texts generated by an automatic speech recognition (ASR) system. Punctuation restoration (PR) is an important post-processing step to improve the readability of texts from ASR systems. Moreover, in natural language processing (NLP), PR is even more important as it enables the use of advanced techniques to process texts at sentence level to achieve optimal performance for various tasks, e.g., part-of-speech tagging and dependency parsing. Prior studies have shown that with proper sentence split and punctuation, a downstream application can tolerate the word error rate of 25%, which is extremely high compared to the current state-of-the-art ASR. Figure 1 demonstrates how punctuation restoration improves the readability of ASR-generated texts.

The performance of automated speech recognition (ASR) systems has drastically improved over the past few years, to the point that some studies report performance results that equal or outperform humans [1, 2, 3, 4]. These systems allow users to interact with machines by voice, and be more efficient than when typing [5], for example. As a result, the use of ASR is becoming increasingly commonplace and the number of ASR APIs has significantly increased. Aside from performing ASR, the APIs also typically add punctuation to the text, i.e. perform PR.

These ASR APIs are used by three categories of users: researchers, developers, and end-users. Researchers may use these APIs to obtain performance baselines for their new ASR algorithms. Developers and end-users want to select the API that satisfies their requirements (e.g., in terms of accuracy, language, latency, privacy, customization, or price).

In this paper, we present a lightweight framework that allows these three categories of users to easily benchmark the PR ability of these ASR APIs on the corpora of their choice.

use the marquee tool to draw a selection around the empty space on one side then hold shift and add the other areas to the selection too go to edit and fill then change the drop down menu to content aware photoshop should automatically generate a completely new background but it might make a couple of small mistakes these can be easily fixed with the patch tool
Use the marquee tool to draw a selection around the empty space on one side. Then hold shift and add the other areas to the selection too. Go to edit and fill, then change the drop down menu to content aware. Photoshop should automatically generate a completely new background. But it might make a couple of small mistakes. These can be easily fixed with the patch tool.

Figure 1: *Punctuation restoration improves the readability of a text generated by an ASR system.*

2. The Punctuation Restoration Benchmark Framework

2.1. Overview

The framework is written in Python 3, and runs on Linux, macOS, and Microsoft Windows. It currently supports the following ASR APIs: Google Speech Recognition [6], Microsoft Bing Speech-to-Text [7], and Speechmatics [8]. The framework is easily extendable to more APIs.

The required format for corpora is a list of pairs of speech files and gold transcriptions with punctuation. The framework comes with an example corpus as well as scripts to convert well-known speech corpora into this format. Speech files may be FLAC, Ogg, MP3, or WAV files.

Figure 2 presents an overview of the system. Listing 1 gives an overview of the configuration file.

2.2. Performance Metrics

The framework is provided with a performance assessment script that computes punctuation restoration metrics comparing the predicted punctuated transcriptions with the reference punctuated transcriptions. Table 1 presents some performance metrics of several ASR APIs on the publicly and freely available TED corpus [9] and the BehancePR corpus. We only use the official test set for each corpus.

We wish to emphasize that the results we present do not aim at ranking existing ASR APIs, since the performance may be affected by whether the corpus was used as part of the training set. Also, different APIs may differ on how well they handle languages other than English, speaker accents, background

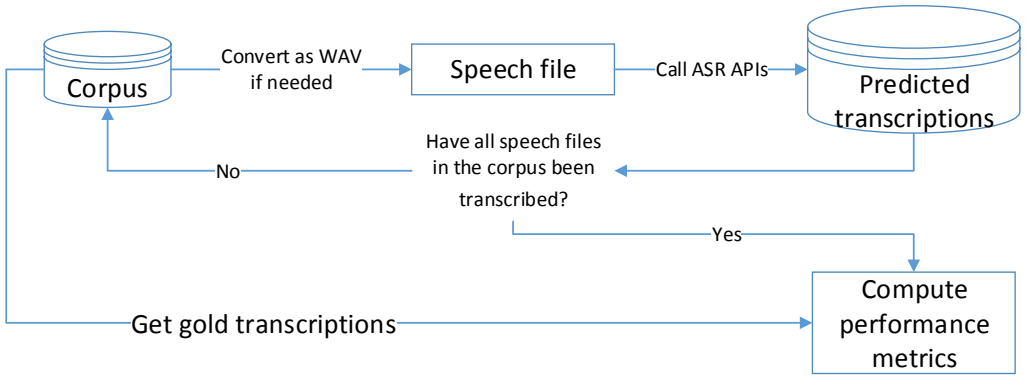

Figure 2: *Overview of the ASR benchmarking framework. First, the user has to provides a corpus that contains speech files with their reference (gold) transcriptions. The framework then converts each file to WAV format if needed, and calls the ASR APIs. When all speech files have been transcribed, the framework computes a set of performance metrics (e.g., word error rate) by comparing the predicted transcriptions with the gold transcriptions.*

noise, etc. Instead, the results we present aim at demonstrating the use of the benchmarking framework.

```
[general]
data_folder              =  ../example_dataset
transcribe               = true
asr_systems              = google,bing
overwrite_transcriptions = false
evaluate_transcriptions  = true
speech_file_type         = wav
delay_between_transcript = 0
speech_language          = en-US
transcription_encoding   = UTF-8

[credentials]
bing_key                 = [removed]
google_credentials       = [removed]
```

Listing 1: *Configuration file used to define a benchmark in the framework. This is the only file the user has to modify. The* `dataset_folder` *defines the location of the folder,* `transcribe` *indicates whether the speech files should be transcribed,* `asr_systems` *lists which ASR API(s) should be called,* `overwrite_transcriptions` *specifies whether a speech file that has already been transcribed should be transcribed again, and* `evaluate_transcriptions` *indicates whether the framework should compute performance metrics once the predicted transcriptions have been collected.*

Table 1: *Benchmark results presenting the word error rates expressed in percentage for several ASR APIs on the following 2 corpora: BehancePR and TED corpus. Please refer to the GitHub repository (see footnote on page 1) for the most up-to-date and comprehensive benchmarks.*

API	BehancePR	TED
Google	70.1	81.5
Microsoft	62.2	79.8
Speechmatics	67.9	79.2

3. Conclusion and Future Work

In this article we have presented a framework to benchmark the punctuation restoration abilities of ASR APIs. The framework is lightweight and easy to use: we hope it will make it more convenient for developers, end-users, and researchers to decide which ASR API or off-line model to use for their punctuation restoration needs and quickly compute some baseline performance for existing or new punctuation restoration corpora.

4. References

[1] W. Xiong, J. Droppo, X. Huang, F. Seide, M. Seltzer, A. Stolcke, D. Yu, and G. Zweig, "Achieving human parity in conversational speech recognition," *arXiv preprint arXiv:1610.05256*, 2016.

[2] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen *et al.*, "Deep speech 2: End-to-end speech recognition in english and mandarin," in *International Conference on Machine Learning*, 2016, pp. 173–182.

[3] D. Amodei, R. Anubhai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, J. Chen, M. Chrzanowski, A. Coates, G. Diamos *et al.*, "End to end speech recognition in english and mandarin," 2016.

[4] E. Edwards, W. Salloum, G. P. Finley, J. Fone, G. Cardiff, M. Miller, and D. Suendermann-Oeft, "Medical speech recognition: reaching parity with humans," in *International Conference on Speech and Computer*. Springer, 2017, pp. 512–524.

[5] S. Ruan, J. O. Wobbrock, K. Liou, A. Ng, and J. Landay, "Speech is 3x faster than typing for english and mandarin text entry on mobile devices," *arXiv preprint arXiv:1608.07323*, 2016.

[6] "Google Chrome's Speech API," https://www.chromium.org/developers/how-tos/api-keys, Accessed: March 3, 2018.

[7] "Microsoft Bing Speech-to-Text API," https://www.microsoft.com/cognitive-services/en-us/speech-api, Accessed: March 3, 2018.

[8] "Speechmatics API," https://speechmatics.com, Accessed: March 3, 2018.

[9] M. Federico, S. Stüker, L. Bentivogli, M. Paul, M. Cettolo, T. Herrmann, J. Niehues, and G. Moretti, "The iwslt 2011 evaluation campaign on automatic talk translation," in *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*. Istanbul, Turkey: European Language Resources Association (ELRA), may 2012. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2012/pdf/1126_Paper.pdf