Data Collection and Preparation for Global Speech Systems

Yocheved Levitan, Julia Hirschberg
Department of Computer Science, Columbia University
yocheved.levitan@gmail.com, julia@cs.columbia.edu

Abstract

In recent years speech technologies have advanced considerably. With all the progress that has been made, the question arises: how can we further develop these systems? To answer this question we must look critically at the status of speech systems today, to identify where they are lacking and discover how they can be improved. This paper explores two speech systems – Text to Speech (TTS), and Keyword Search, and offers suggestions and delineates steps towards their improvement. We hypothesize that TTS can be upgraded by implementing Data Selection methods.

1. Introduction

Text to Speech (TTS), the process of synthesizing speech from a series of written words, has existed for over one hundred years [0]. Keyword Searching, which involves spotting key phrases in a database, is a relatively newer project. Both tools are constantly evolving to become more relevant and useful for our world today. In the field of TTS, for example, researchers are working on building voices with a variety of speaking styles to reflect the context of the text that is being synthesized [1]. This is an important advancement from the baseline/average voice models that were previously available. Keyword Searching can now be implemented for databases of spoken as well as written language.

Despite their progress, these systems can be further developed. We propose two fundamental improvements. Firstly, we would like to implement data selection methods for TTS to increase its efficiency. Secondly, we aim to provide Keyword Search capabilities for low-resource languages. In the following sections we outline the steps involved in these two projects.

2. Text to Speech (TTS)

2.1 Methods

There are two popular ways to approach the task of speech synthesis. One method is concatenation – the process of stringing small audio segments together to form desired words. The other is known as HMM-based synthesis. This method uses Hidden Markov Models to generate speech.

Concatenative systems are effective in creating clear, natural-sounding voices. These systems are also relatively easy to implement. On the other hand, the systems that use HMM's are more complex, and the speech that they output can sound robotic. Yet the HMM-based synthesizers have one important advantage over other systems: they are flexible. In concatenative systems, once a voice is generated from a database of recorded speech, the results cannot be tampered with. However, because HMM-based synthesis is parametric in nature, the output can be modified simply by changing the relevant parameters. It is not necessary to record an entire new corpus if the resulting voice is not satisfactory or suited to the current task (as is the case with concatenation); rather, the parameters can be manipulated to produce a completely new speaking style.

This key feature allows us to conserve time, computer memory, and money, all of which are consumed in the project of recording audio data. The current project therefore focuses on improving HMM-based synthesis.

2.2 Process

We use is the HMM-based Speech Synthesis System (HTS) to implement an HMM speech synthesizer. Details of HTS, which is freely available for download from the web, are explored in depth in [2,3,4].

To synthesize speech with this system we begin by amassing a collection of audio and text (Section 2.3). Next, each audio segment is aligned with its text transcription. The HMM's are trained on the aligned data. They can then be used to generate new utterances.

2.3 Data Selection

When the input is predictable, the output is predictable as well. Data, however, is often imperfect. The format and quality of text and audio collected from sources such as blog posts, Wikipedia pages, and YouTube videos can vary widely and is often unknown.

To solve this dilemma, we consider data selection. Our goal is to develop an algorithm to detect and reject unwanted data before synthesis occurs. Previous work concerning data selection has been used to improve Automatic Speech Recognition (ASR) [5,6], but its application to TTS is novel.

2.4 Progress

We chose the Boston University Radio Speech Corpus as the model for an ideal data source for speech synthesis. This corpus was prepared in 1996, and was intended for TTS research. It includes cleanly recorded audio from radio segments, along with transcriptions and prosodic information. The files are divided by paragraph, but to more accurately identify problematic files, in accordance with our goal of improving data selection, we partitioned them further into sentences using the punctuation in the transcripts. We wrote a script in Python to create the text and audio sentence files. We then used the HTS Speaker Adaptive Demo to train a synthesized voice on this data. A few days later, the synthesis was complete and a coherent, intelligible voice was produced.

Now that we have a contrasting model to serve as a gold standard, we can make progress towards a data selection algorithm. As soon as it is developed, we can test it on our ideal data, analyze the results, and then compare it with the results that we get when it is applied to imperfect data.

3. Keyword Search (KWS)

3.1 Background

Familiarity with the concept of searching a database for a specific words or phrases has increased in recent years. Users are constantly searching Google and other sites for information on a variety of topics. Whereas in the past this task was restricted to text queries, with the advent of smartphones and tablets, which make typing inconvenient, voice search may become equally popular.

All webpages are indexed, and when a query is performed, the results are extracted from the index, and relevant material is displayed. Searching algorithms today produce results in a fraction of a second. KWS is an efficient, almost flawless tool. However, a major weakness does exist: it is not compatible with many languages. The term coined for the languages without a wide array of available tools is low-resource languages. They have a narrow scope of technologies at their disposal.

3.2 The Babel Program

The Babel Program is funded by the Intelligence Advanced Research Projects Activity (IARPA), organization. This organization promotes research competition with the objective of obtaining new insights and achieving revolutionary success. IARPA teams are comprised of diverse members from across the globe who brainstorm and collaborate to fulfill IARPA’s mission. The team that wins the competition receives a monetary award.

The challenge that the Babel program addresses is developing speech recognition technologies for low-resource languages in order to allow for KWS. The program involves different stages, and much progress has been documented already [7,8]. Columbia University is participating in this competition.

3.3 Low-Resource Languages

There are approximately 6,500 spoken languages that exist today. Some of these languages are considered endangered. Because of the diminishing population of speakers, these languages are at risk of becoming extinct. Another linguistic label is low-resource. A language tagged with this term can be spoken by a significant number of people, yet because of technical constraints, developers find it difficult to provide this language with essential speech tools. A major constraint is the lack of audio and text data that is necessary to build the proper technologies. If we could collect data for these languages, we would have the ability to create the appropriate systems.

3.4 Process

We began tackling this task by identifying the low-resource languages that we are interested in. The list includes: Telugu, Lithuanian, Tok Pisin, Kazakh, Cebuano, and Kurmanji Kurdish. These are the languages that were chosen by the administrators of the Babel program.

The Internet is the obvious resource for gathering large amounts of data in a variety of different contexts. Audio recordings, and text samples are free and are globally accessible. They can be found on a range of sites, from blogs to Wikipedia pages. The links on the webpages can be followed to collect even more useful information.

3.5 Progress

Telugu was the first language that we worked on. We discovered a site that contained a listing of about thirty blogs written in Telugu. Using the BeautifulSoup library, a Python library built for parsing HTML pages, we wrote a script to iterate through the archive pages for each blog and extract the text from the posts. Twelve blogs were scraped, resulting in over 400,000 words of Telugu text. In addition, the official Wikipedia page for the Telugu language was scraped, and the obtained text was added to the collection.

Lithuanian blogs were more sparse, but we found a cooking site with terms for Lithuanian foods and a beauty blog written in Lithuanian. We were also able to pull relevant text from the Lithuanian Wikipedia page.

The other languages have a slighter web presence, but we will explore all media to find as much data as possible.

4. Conclusion

The first problem for any research project in spoken language processing is acquiring suitable audio and text data. We have accomplished that task for two research projects: using data selection to improve HMM-based synthesis, and keyword search for low-research languages. Our work has provided a gold standard audio database for developing a data selection algorithm, and Lithuanian and Telugu data scraped from all over the Internet to serve as training data for the keyword search task. In future work, we plan to continue work on these two tasks.

5. Acknowledgements

I would like to thank Professor Julia Hirschberg, chair of the Computer Science department at Columbia University, for her guidance throughout my research. I am also indebted to Erica Cooper, a PhD student at Columbia, who has helped tremendously with my work. Finally, I would like to express my gratitude to the coordinators and donors of DREU (Distributed Research Experience for Undergraduates) for making this research possible.

6. References

[1] Degen, Wien, J. B. (1791). der menschlichen Sprache nebst der Beschreibung seiner sprechenden Maschine ("Mechanism of the human speech with description of its speaking machine,"). (German)

[2] Yamagishi, J., Usabaev, B., King, S., Watts, O., Dines, J., Tian, J., ... & Kurimo, M. (2010). Thousands of voices for HMM-based speech synthesis–Analysis and application of TTS systems built on various ASR corpora. *Audio, Speech, and Language Processing, IEEE Transactions on*, *18*(5), 984-1004.

[3] Yamagishi, J., Ling, Z., & King, S. (2008). Robustness of HMM-based speech synthesis.

[4] Zen, H., Tokuda, K., & Black, A. W. (2009). Statistical parametric speech synthesis. Speech Communication, 51(11), 1039-1064.

[5] Zen, H., Nose, T., Yamagishi23, J., Sako14, S., Masuko, T., Black, A. W., & Tokuda, K. The HMM-based Speech Synthesis System (HTS) Version 2.0.

[6] Wu, Y., Zhang, R., & Rudnicky, A. (2007, December). Data selection for speech recognition. In Automatic Speech Recognition & Understanding, 2007. ASRU. IEEE Workshop on (pp. 562-565). IEEE.

[7] Nagroski, A., Boves, L., & Steeneken, H. (2003, November). In search of optimal data selection for training of automatic speech recognition systems. In Automatic Speech Recognition and Understanding, 2003. ASRU'03. 2003 IEEE Workshop on (pp. 67-72). IEEE.

[8] Kingsbury, B., Cui, J., Cui, X., Gales, M. J., Knill, K., Mamou, J., ... & Woodland, P. C. (2013, May). A high-performance Cantonese keyword search system. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on* (pp. 8277-8281). IEEE.

[9] Mamou, J., Cui, J., Cui, X., Gales, M. J., Kingsbury, B., Knill, K., ... & Woodland, P. C. DEVELOPING KEYWORD SEARCH UNDER THE IARPA BABEL PROGRAM.