INTERNATIONAL INSTITUTE OF
INFORMATION TECHNOLOGY
HYDERABAD

ASR Data Pruning

Automatic and semi-automatic data collection
for domain-adapted ASR

BTP-1

Spring'23

Team:
Harshita Gupta (2020101078)
Karmanjyot Singh (2020101062)

Report

Abstract

This project aims to collect health domain data in English using a combination of automatic and semi-automatic data collection methods, including ASRs, data collection platforms, text scraping, and open-source videos. The collected data will be used to develop domain-adapted ASRs for improved accuracy in healthcare settings. ASR technology automatically transcribes spoken language into written text, without the need for human intervention, and has become increasingly popular in recent years. The use of ASR in healthcare can improve accessibility for individuals with hearing or speech impairments, increase efficiency in transcription and translation tasks, and enable new forms of human-computer interaction.

Keywords: ASR, Audio, Transcript, Web Scraping, Web Crawler

Contents

1. Introduction

Automatic Speech Recognition (ASR) technology has revolutionized the way we interact with computers and devices. With the ability to transcribe spoken language into written text automatically, ASR has enabled new forms of human-computer interaction and opened up opportunities for improved accessibility, efficiency, and productivity. In the healthcare domain, ASR has the potential to facilitate the accurate and efficient transcription of medical information, contributing to improved healthcare outcomes and patient care. However, ASR performance can be limited by the use of domain-specific language, vocabulary, and accents, which can lead to errors and inaccuracies in transcription. Our BTP-1 project aims to address this limitation by collecting health domain data in English using a variety of automatic and semi-automatic data collection methods.

2. Methodology

2.1 Web Scraping

Much of the data is scattered across various sources, such as medical journals, articles, and online forums,etc. Hence, we employ web scraping, to make use of this huge chunk of data. In this project, we will use web scraping techniques to gather medical data from various sources, such as online medical forums, social media platforms.

To accomplish this, we first collected a source repository, containing potential websites to scrape data from, we filtered the selection further based on the website reviews and the authenticity of the sources, and the broad range of topics covered by the websites. We shortlisted `https://www.medicalnewstoday.com/` and `https://www.edutopia.org/`. These are websites, had data divided over speciific domains, categorised by disease/domain, and thus provide a rich source of data for scraping.

For web-scraping task, we used libraries such as beautifulsoup, selenium, for automated text/data extraction from the sites.

The web-scraping task was divided into two sections:

1. **Web Scrapping**

   In this specific phase, we utilized the methods provided by the selenium library, which offers dynamic control that allows us to load and hold for a web page until all the jsx modules are loaded, thus waiting until the site data is fully available. Unlike beautifulsoup, this provided us with the ability to collect more data. Selenium also offers additional functionalities like scrolling, making it a better choice than beautifulsoup. In this phase, we extracted data from the selected websites mentioned earlier, using different scripts for each website.

2. **Web Crawler**

   In the latter part of the website scraping process, we aimed to create a generic solution that could be extended to a larger set of websites. Previous scripts were designed specifically for the structure of particular websites, limiting their flexibility and ability to work on other sites. The main objective was to extract most of the relevant data from any given link and store it in a suitable data format. To accomplish this, we primarily used beautifulsoup to develop the initial version of the crawler. The crawler navigated through the website, collecting and appending all hyperlinks into a storage array. It iterated through each of them until the list was finished or the maximum tree depth was reached, collecting and storing relevant data in a suitable data format.

2.2 Youtube Videos

To collect speech data from YouTube videos for our health domain project, we initially identified relevant channels and videos with built-in captions to obtain accurate transcript files. We utilized Python libraries such as pytube and youtube-dl to extract audio from the videos.

Pytube is a Python library that provides an easy-to-use interface for downloading YouTube videos. It allows you to fetch video and audio streams of a YouTube video and download them as separate files. Pytube can handle various YouTube video formats and resolutions and offers features like video and audio merging, video metadata extraction, and video thumbnail generation.

Youtube-dl is another popular Python library used for downloading videos from various websites, including YouTube. It also provides features like video and audio merging and subtitles extraction.

Additionally, we used the YouTubeTranscriptAPI to obtain transcript files. The YouTube Transcript API is a web service provided by YouTube that allows you to retrieve transcripts and captions of YouTube videos programmatically. It provides a simple REST API that returns the transcript text along with timestamp information. The API can be used to obtain transcripts in various languages, including English, Spanish, French, German, Italian, and many others. It can also handle automatic captions, which are generated by YouTube's speech recognition technology, and manual captions, which are added by video creators.

```
{'text': 'At a Maryland country fair in 2017,', 'start': 7.505, 'duration': 3.31}
{'text': 'the prize pigs were not \nlooking their best.', 'start': 10.815, 'duration': 3.24}
{'text': 'Farmers reported feverish hogs with \ninflamed eyes and running snouts.', 'start': 14.055, 'duration': 5.09}
{'text': 'But while fair officials worried \nabout the pigs,', 'start': 19.145, 'duration': 3.001}
{'text': 'the Maryland department of health was \nconcerned about a group of sick fairgoers.', 'start': 22.146, 'duration': 5.6}
{'text': 'Some had pet the pigs, while others had \nmerely been near their barns;', 'start': 27.746, 'duration': 4.648}
{'text': 'but soon, 40 of these attendees would \nbe diagnosed with swine flu.', 'start': 32.394, 'duration': 5.149}
{'text': 'More often than not, sick animals \ndon’t infect humans.', 'start': 37.543, 'duration': 3.74}
{'text': 'But when they do, these \ncross-species infections,', 'start': 41.283, 'duration': 3.43}
{'text': 'or viral host jumps,', 'start': 44.713, 'duration': 1.9}
{'text': 'have the potential to produce \ndeadly epidemics.', 'start': 46.613, 'duration': 3.45}
{'text': 'So how can pathogens from one species \ninfect another,', 'start': 50.063, 'duration': 3.85}
{'text': 'and what makes host jumps so dangerous?', 'start': 53.913, 'duration': 3.71}
{'text': 'Viruses are a type of organic parasite \ninfecting nearly all forms of life.', 'start': 57.623, 'duration': 5.51}
{'text': 'To survive and reproduce, they must move\nthrough three stages:', 'start': 63.133, 'duration': 4.147}
{'text': 'contact with a susceptible host, \ninfection and replication,', 'start': 67.28, 'duration': 4.056}
{'text': 'and transmission to other individuals.', 'start': 71.336, 'duration': 3.39}
{'text': 'As an example, let’s look \nat human influenza.', 'start': 74.726, 'duration': 3.57}
{'text': 'First, the flu virus encounters \na new host', 'start': 78.296, 'duration': 2.8}
{'text': 'and makes its way into \ntheir respiratory tract.', 'start': 81.096, 'duration': 2.97}
{'text': 'This isn’t so difficult, but to survive \nin this new body,', 'start': 84.066, 'duration': 3.42}
{'text': 'the virus must mount a successful \ninfection', 'start': 87.486, 'duration': 3.019}
{'text': 'before it’s caught and broken down \nby an immune response.', 'start': 90.505, 'duration': 3.86}
{'text': 'To accomplish this task,', 'start': 94.365, 'duration': 1.47}
{'text': 'viruses have evolved specific interactions\nwith their host species.', 'start': 95.835, 'duration': 4.37}
{'text': 'Human flu viruses are covered in proteins', 'start': 100.205, 'duration': 3.175}
{'text': 'adapted to bind with matching receptors \non human respiratory cells.', 'start': 103.38, 'duration': 5.45}
{'text': 'Once inside a cell, the virus employs \nadditional adaptations', 'start': 108.83, 'duration': 4.67}
{'text': 'to hijack the host cell’s reproductive \nmachinery', 'start': 113.5, 'duration': 2.959}
{'text': 'and replicate its own genetic material.', 'start': 116.459, 'duration': 2.91}
{'text': 'Now the virus only needs to suppress\nor evade the host’s immune system', 'start': 119.369, 'duration': 4.82}
{'text': 'long enough to replicate to sufficient \nlevels and infect more cells.', 'start': 124.189, 'duration': 4.137}
{'text': 'At this point, the flu can be passed on to\nits next victim', 'start': 128.326, 'duration': 3.84}
{'text': 'via any transmission \nof infected bodily fluid.', 'start': 132.166, 'duration': 3.75}
{'text': 'However, this simple sneeze also brings \nthe virus in contact with pets,', 'start': 135.916, 'duration': 4.78}
{'text': 'plants, or even your lunch.', 'start': 140.696, 'duration': 2.45}
{'text': 'Viruses are constantly encountering \nnew species and attempting to infect them.', 'start': 143.146, 'duration': 5.54}
{'text': 'More often than not, this ends in failure.', 'start': 148.686, 'duration': 2.499}
```

Figure 1: *An example transcript generated using the YouTubeTranscriptAPI for a video*

To ensure accurate and efficient transcription of the audio data, we segmented each audio file into 15-second clips and edited the corresponding srt files to match the clip lengths. This decision was based on our experience working with the Google speech-to-text API in the preliminary task, where we encountered inaccuracies and missing sentences in the transcriptions generated from larger audio files. By segmenting the audio data into smaller chunks, we were able to improve the accuracy of the transcription and process the data more efficiently. We found that this approach was particularly effective when dealing with longer audio files and challenging audio conditions. By segmenting the audio into smaller, more manageable chunks we reduced the amount of data the API had to process at once. Also, smaller segments can be processed faster and in parallel.

The source code for our project can be found on GitHub. [1]

[1] https://github.com/KarmanjyotSingh/BTP-1

3. Data

3.1 Without Audio

We collected text data, covering various domains pertaining to medicine, from the sites https://www.medicalnewstoday.com/, https://www.healthline.com/ and https://www.edutopia.org/, we collected around 8000 files of data, with each text file containing around 3-4 sentences. The data was checked and verified by the guiding mentors, due to limited storage constraints, we constrained ourselves to these many files, however, the given script could be employed to extracts many folds of data from the site.

When data is collected from websites, it is often in a raw and unstructured format. The crawler used in this case helps in collecting data by navigating through the website and generating JSON files for each section of the website. JSON or JavaScript Object Notation is a popular format for storing and exchanging data, and is widely used in web development.

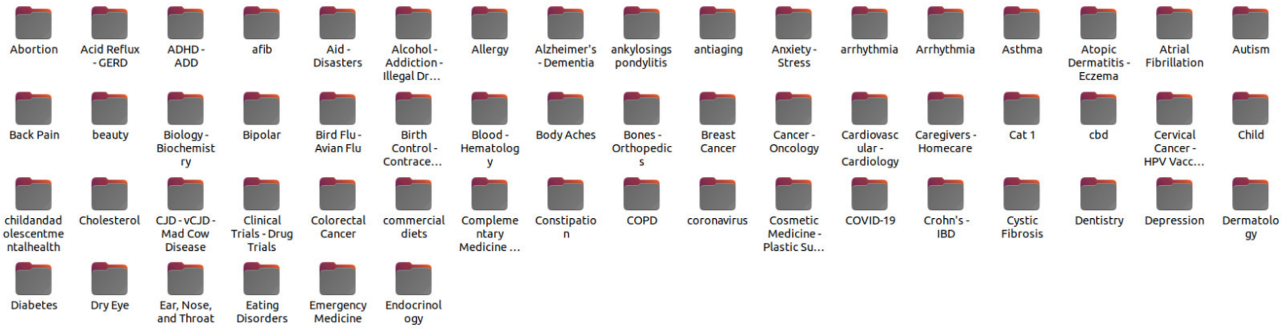
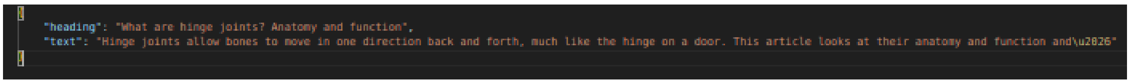


Figure 2: *Sample Data Sorted into Categories*

The data collected from the websites is quite general and needs to be further processed to extract insights and useful information. This is where the JSON format provides great flexibility, as it allows for easy parsing of the data, manipulation of the data structure, and extraction of relevant information. A part of the data could be found at : https://drive.google.com/drive/folders/1EizJyA6CBcW90YtU32p80-3Ilw0-J5-r?usp=sharing

Figure 3: *Folder View of Collected Data*

Each JSON file contains two important attributes, namely ’heading’ and ’text’, which are key to extracting useful information from the data. The ’heading’ attribute stores the main section heading of the data, while the ’text’ attribute contains the corresponding text data. This enables efficient and accurate identification and extraction of specific data points from the large amount of data collected, making it easier to analyze and draw insights from the data.



```
{
    "heading": "What are hinge joints? Anatomy and function",
    "text": "Hinge joints allow bones to move in one direction back and forth, much like the hinge on a door. This article looks at their anatomy and function and\u2026"
}
```

Figure 4: *Sample JSON File*

3.2 With Audio

For our study, we collected audio and transcript data from various YouTube channels related to health and medicine, with no restrictions on the accent. We have audio data in American, British, and Indian English, with a total of over 3000 audio-transcript pairs.

The YouTube channels we collected data from include Kenhub - Learn Human Anatomy, JJ Medicine, EZmed, Medical Dialogues, Osmosis from Elsevier, Global Health Media Project, and TED-Ed.

In total, we extracted approximately 8 hours of audio data from the videos. The average duration of each audio file is 8.43 seconds.

We obtained 3362 text files containing transcripts for the audio data, with a total of 3940 sentences. The average number of sentences in each text file was 1.17.

The total number of words in 3289 text files was 76075, with an average of 23.13 words per file.

This data provides insights into the size and composition of the speech data we collected for our project, and serves as a basis for further analysis and modeling.

The video links for the collected data are available here [2] , and the data itself is stored in [3].

4. Results

We conducted a word cloud analysis to gain insights into the topics and diseases covered in the audio-transcript data we collected. The word cloud revealed that the most common words and phrases in the data included ”sciatic”, ”blood”, ”nerve”, ”cells”, ”diabetes”, ”glucose”, and ”symptoms”, among others. This suggests that the data covers a wide range of topics related to health and medicine.

While more trustable datasets like SNOMED CT and ICD10 had restricted access, we could leverage open-source solutions like Scapy. Scapy provides access to a trained ML model that categorizes words related to the biomedical domain.

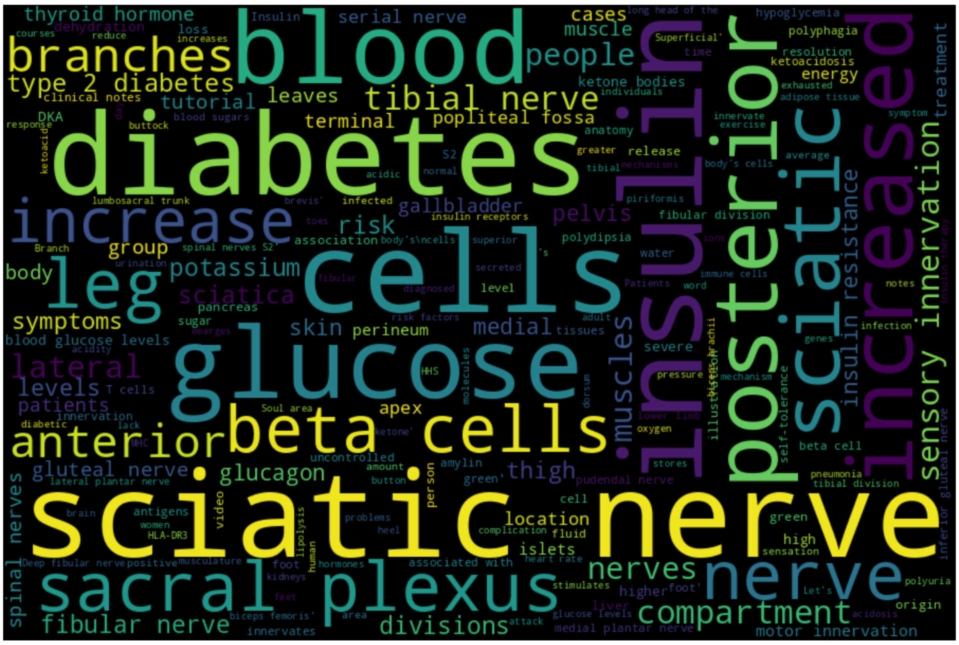


Figure 5: *Wordcloud representation of topics covered*

5. Future Work

This project has laid the groundwork for further exploration of speech data in the context of health and medicine. Moving forward, there are several avenues for future

[2] https://docs.google.com/document/d/121zj8116w-knNowS0pW-6IbZzBYcFXf965u_RmnlnwA/edit?usp=sharing

[3] https://drive.google.com/drive/folders/1YwjZMVcjws00UJjoYQ8DiQoI0G0pbWAz?usp=sharing

work that we plan to pursue:

Collection of data in multiple Indian languages: We plan to expand our data collection efforts to include audio and text data in 2-3 Indian languages, in addition to English.

Data annotation for integration with Chatbot: We also plan to annotate the data to facilitate the integration of our speech data with chatbot technology.

ASR model accuracy: We plan to use automatic speech recognition (ASR) models to generate text files corresponding to our audio data and compare these transcripts to our manually obtained transcripts to estimate the model's accuracy. This will enable us to identify areas for improvement and fine-tune our ASR models for greater accuracy and reliability.

Preprocessing of data collected using the generalised web crawler: We plan to improve the quality and consistency of the data collected using our generalized web crawler.

Copyright issues: We will work to verify and address any copyright issues related to our use of data from YouTube channels or other websites. We recognize the importance of respecting copyright and intellectual property and will take all necessary steps to ensure that our research is conducted in an ethical and responsible manner.

Analyze the dataset to identify the number and distribution of male and female speakers. Preprocess the audio files to remove noisy data with multiple speakers to improve dataset quality.

6. References

- https://github.com/jiaaro/pydub
- https://github.com/ytdl-org/youtube-dl/blob/master/README.md
- https://pypi.org/project/youtube-transcript-api/
- https://www.youtube.com/@Kenhub
- https://www.youtube.com/@osmosis
- https://www.youtube.com/@TEDEd