Multi-Class Classifier of Accents for Audio Utterances in Spanish

Ricardo Zambrano

University of Maryland

rzambrano@gmail.com

***Abstract***

***This project sought to develop a multi-class classifier for Spanish accents. The proposed model may help improve the performance of speech activated systems operating in the Spanish language The Common Voice dataset was used as a source of audio utterances in Spanish. This dataset provides more than one million utterances recorded in mp3 format. Each speaker read a sentence and self-reported their demographics, including ‘accent’ label. The written text of each sentence are captured in the data. Because of the size of the dataset Databricks was the platform selected to train the classifier. MFCC features were used as an input for a logistic regression algorithm. The prediction accuracy of the algorithm was 0.54 in both the train set and test set. The model predicted the majority class. To improve the accuracy of the model, techniques to guarantee an independent and identically distributed sample of the utterance must be included. It is hypothesized that other machine learning algorithms might improve the accuracy of the prediction.***

**I. Introduction**

Spanish is the second language with most native speakers, about 485 million. Spanish is the official language of 20 countries and it is the third most used language on the internet. With the rise of speech activated systems, there is value in developing automatic speech recognition systems adapted specifically to the Spanish language.

Over time Spanish speakers have develop many distinct accents, with multiple accents occurring within the borders of a single country. As these accents developed the choice of words and sentences became endemic for individual communities. Furthermore, semantic ambiguity arose as words started to evoke different interpretation in different communities. For example, the word ‘marcha’ means to protest in Venezuela, whereas it means party in Spain.

When this semantic ambiguity arises during a conversation among human actors usually one of the speakers detects the meaning divergence between what the speaker is meaning and what the listeners are understanding. This leads to one of the speakers to confirm the meaning of the sentence or word. It is hypothesized that a non-human listener will have a hard time identifying the semantic ambiguity and may interpret sentences literally based on the dialect it was trained (or the majority class dialect present in the training dataset).

The main hypothesis of this project is that identifying the accent of a speaker might improve the performance of voice activated systems by removing the semantic ambiguity of words and commands uttered by the user of the system.

**II. Literature Survey**

In the section above, facts (2) and (3) might make the study of Spanish dialects and understudied area in the field of natural language processing. This hypothesis might be the reason why there are so few research articles in the ACL Anthology Journal. Upon a search of articles in the ACL Anthology website only two articles were found:

* Francom, J; et al. “ACTIV-ES: a comparable, cross-dialect corpus of ‘everyday’ Spanish from Argentina, Mexico, and Spain”
* Maier, W; Gomez-Rodriguez, C. “Language variety identification in Spanish tweets”. Language Technology for Closely Related Languages and Language Variants. October/2014. pages 25–35

No previous studies were found in ACL Anthology using spoken phrases recorded in audio files.

This section reviews existing literature relevant to your project.

It summarizes and analyzes previous research, theories, and studies related to your topic.

The literature survey helps identify gaps in knowledge, informs the research design, and provides a foundation for your project. Make sure to list all the other algorithms that you considered and the reason to go with the one you went with.

**III. Methodology**

*A. Common Voice Dataset*

The main objective of this project is to predicts the Spanish variant used by a speaker reading sentences recorded in an audio file. There are few audio datasets available for the Spanish language. For this project the Common Voice dataset was used.

Common Voice is a crowdsourcing project started by Mozilla to create a free database for speech recognition software. The project is supported by volunteers who record sample sentences with a microphone, and review recordings of other users. The review stage consists on a voting system that classifies recorded utterances into four quality categories:

* Validated: it has been upvoted enough
* Invalidated: it has been downvoted by several volunteers
* Other: it has not received enough up votes or down votes to fall into any of these two buckets
* Reported: the audio has been reported by other user(s)

Because this project had an emphasis on processing big data recordings on the ‘other’ category were selected. This is because most files in the Spanish language are still in the ‘other’ category. In particular, at the time of this writing the ‘other’ category had 1,150,345 files.

For this project Common Voice Corpus 15.0 [46.23 GB] for the Spanish language was used. It was published on Sep/13/2023 and it contains:

* 2,188 recorded hours
* 526 hours of validated audio
* 25,338 distinct speakers
* It is stored in MP3 format

*B. MFCC Features*

For this project two approaches were tried: (i) extracting features directly from the waveform using a Convolutional Neural Network, and (ii) extracting features of the audio recording using Mel-frequency cepstral coefficients (MFCC). Only the second approach was achieved.

Mel-frequency cepstrum (MFC) is a representation of the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear Mel scale of frequency. Taking as a group MFCCs capture the shape of the power spectrum of a sound signal. In brief, it is assumed the MFCC compress information about the audio signal into a small number of coefficients while discarding less relevant information. This compression makes the analysis of the entire waveform require less computation power.

*C. Data Load, Cleaning, and Encoding of Target labels*

The Common Voice zipped file (.tar.gz extension) was downloaded into a Databricks ‘Volume’. A Databricks ‘Volume’ is a Unity Catalog object representing a logical volume of storage in a cloud object storage location, in this case an AWS S3 Bucket. “Volumes provide capabilities for accessing, storing, governing, and organizing files.” Volumes were selected to store the Common Voice dataset because they can store and access files in any format, including: structured, semi-structured, and unstructured data.

The unzipped Comon Voice folder contains four tab-separated files listing the audio files that correspond each quality category. It also has a folder that contains the MP3 audio files of all quality categories. The tab-separated files list on each row the client ID of the collaborator/speaker, the recording’s file name as listed in the folder with the audio files, the text transcript of the read sentence, the number of upvotes/downvotes for the file, and demographic information of the speaker (including the accent/dialect used by the speaker).

The original dataset had 111 distinct accents. Several of these accents were duplicates, in the form of different names for the same accent. As an example of this situation, it was observed that the order of the words used in the name of the category appeared in different order. There were observations that reported two accents. In this latter case, the first self-reported accent was selected.

It was also observed that some utterances contained not information. These records were utterances reported with the ‘neutral’ label. These labels were re-named as ‘discard’ in order to remove them from the original dataset.

The rest of the accents were re-mapped using a dictionary. The goal of this mapping was to group accents that were one and the same in the same bucket. The map dictionary was built by hand using MS Excel as a tool to see side by side the original name of the accent and select the unique accent name from a group of preset categories. Afterwards this file was saved as a comma separated file and uploaded as a dataframe to Databricks. Finally, a dictionary was built from the dataframe. By using a user defined function and the accent dictionary, a crosswalk between ‘old accent label’ and ‘new accent label’ was applied to the rows in the ‘other’ dataset.

Next rows equal to ‘Discard’ in the accent column, as well as rows with missing values in both the accent and sentence columns were removed, the ‘other’ dataset had 932,533 rows.

The process of unzipping the MP3 audio files was taking more than 48 hours. Thus, it was decided to interrupt the unzipping process and work with the files available in the Volume. When the process was interrupted the Volume had about 765,000 audio files. Given that checking which files were available-or not- in the Volume had a runtime of order O(932,533 x ~765,000) and the S3 Bucket was slow (in contrast with the Volume, this computation would terminate in a commodity system); it was decided to take a random sample of the rows in the ‘other’ dataset and then only check if the file was in the Volume.

After running several tests, it was found that to meet the target of processing about 100,000 audio files, a random sample of 40% had to be taken from the rows in the ‘other’ dataset. This left 373,087 from which 105,996 had a matching audio file stored in the Volume.

Once the final sub-set was pinpointed, the accent column was encoded to integer labels.

*D. Lazy Evaluation and User Defined Functions (UDF)*

It was found that when running action commands in PySpark, the lazy evaluation would calculate all the operations and transformations recorded in the DAG since the loading of the dataframe. In particular, for this application PySpark’s lazy evaluation would prove impractical. It was observed that after including complex UDFs in the DAG (such as loading a waveform and calculating the MFCCs), action commands would crash the notebook every time they were run. Thereby, a strategy was employed to avoid PySpark’s lazy evaluation structure.

To avoid lazy evaluation, at given milestones the dataframe was saved as a csv file, to cache the transformations and operations applied to the dataframe so far. For example, once the final subset was loaded, cleaned, and proper encodings were implemented the dataframe was saved into the Volume as a csv file. Then the dataframe would be reloaded to continue with loading the waveform, extracting the MFCCs, and running the model.

The methodology describes the research design, approach, and techniques used to conduct the study.

It includes details about data collection methods, tools, and procedures.

This section should provide enough information for someone to replicate the study and assess its validity.

**IV. Results**

The results section presents the findings of your study.

It includes data, statistics, graphs, and any other relevant information that supports your conclusions.

Results should be presented objectively, often using visual aids to enhance understanding.

**V. Conclusion and Future Work**

The minimum requirement would be to build a classification model that predicts the dialect of a speaker reading sentences recorded in an audio file. If possible, this work will try to produce a transcription of the phrases uttered by the speaker. The challenge in the extension will be to capture the sentences uttered in Spanish dialects that pronounce words differently. For example, dialects that replace the letter ‘r’ with an ‘l’ (Puerto Rican) or that omit the letter ‘s’ at the end of a word (Venezuelan). It is possible that the extension of this work would be achievable only if a classifier model indicates the regional variation used by the speaker prior to encode-decode the audio file into text.

The conclusion summarizes the key findings of the study and interprets their implications.

It may discuss the project's contributions to the field, limitations, and potential areas for future research.

This section provides a final synthesis of the project and reinforces its significance.

**VI. References**

The references section lists all the sources cited in your report.

It follows a specific citation style (e.g., APA, MLA) and includes books, articles, and other academic materials.

Proper referencing is crucial for academic integrity and allows readers to locate the sources you consulted.