Multi-Class Classifier of Accents for Audio Utterances in Spanish

Ricardo Zambrano
University of Maryland
College Park, U.S.A.
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 recorded in the dataset read a sentence and self-reported their demographics, including 'accent' label. The written text of each sentence is 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 utterances must be included. It is hypothesized that other machine learning algorithms might improve the accuracy of the prediction.**

Keywords—automatic speech recognition, big data, Spanish language accents, common voice

I. Introduction

Spanish is the second language with most native speakers, about 485 million worldwide. Furthermore, 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.

Spanish speakers have developed many distinct accents over time, with multiple accents found within the borders of a single country. As these accents developed, the choice of words and sentences became a unique feature of each community. Furthermore, semantic ambiguity arose as words started to evoke different meanings in different communities. For example, in an informal conversational context, the word 'marcha' means 'to protest' in Venezuela, whereas in Spain it means 'party'.

When this kind of semantic ambiguity arises during a conversation among human actors, usually one of the speakers detects the divergence in meaning between what the speaker means to say and what the listener understands. This leads to one of the speakers confirming or clarifying 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 following the meanings encoded in the dialect the system was trained on (or the meanings in the majority dialect class present in the training dataset).

The main hypothesis this project holds is that identifying the accent of a speaker might improve the performance of voice-activated systems. This would be achieved by removing the semantic ambiguity of words and commands uttered by the users of such systems.

II. Literature Survey

Upon a review of research articles about classification of Spanish accents in the ACL Anthology website, only two articles were found:

- J. Francom, et al, ACTIV-ES: a comparable, cross-dialect corpus of 'everyday' Spanish from Argentina, Mexico, and Spain
- W. Maier and C. Gomez-Rodriguez, 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.

It is hypothesized that the study of Spanish dialects/accents might be an understudied area in the field of natural language processing. The causes may be varied. Among the potential causes, it might be worthwhile considering the fact that most relevant research in machine learning is taking place in the anglosphere. This naturally would lead to an emphasis in English language and languages spoken in regions considered as a security threat within the anglosphere (e.g. the Middle East and P.R. of China). Likewise, negative bias towards Spanish speakers in the U.S. might play a role in the lack of interest in developing technology tools in Spanish language. Last but not least, most of the Spanish-speaking world is located in a region with limited economic resources; and, given the steep cost of training language models, the pace of development is slower than in the anglosphere.

III. Methodology

A. Common Voice Dataset

The main objective of this project is to predict 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: when the recording 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 big data tools, 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 there were 1,150,345 files in the 'other' category.

The dataset version used for this project was the Common Voice Corpus 15.0 [46.23 GB] for the Spanish. 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

Two approaches were tried for this project: (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. Taken 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 with 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 a given category appeared in different order, but the same words were used and had the same meaning. There were observations that reported two accents. In this latter case, the first self-reported accent was selected. It is worthwhile mentioning that some of the observations with two categories had two categories that were in conflict with each other (e.g. Colombian Caribbean and Colombian Andean accent; these two categories are either/or and the speaker cannot use both simultaneously).

It was also observed that some utterances contained no information in their labels. These records were utterances reported with the overall 'neutral' label or a country-wide 'neutral' accent. 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 Python 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 the original name of the accent side by side and select the unique accent name from a group of pre-set categories. Afterwards this file was saved as a comma-separated file and uploaded as a dataframe to Databricks. Finally, the aforementioned dictionary was built from the dataframe. By using a user-defined function (UDF) 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. After this processing, the 'other' dataset had 932,533 rows left.

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 available- 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 to 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 observations from which 105,996 had a matching audio file stored in the Volume.

Once the final sub-set was loaded and cleaned, 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, the dataframe was saved as a csv file at given milestones, to cache the transformations and operations applied to the dataframe up until that point. 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 the processes of loading the waveform, extracting the MFCCs, and training the model.

One challenge faced in this project was posed by the fact that audio libraries in Python use Numpy arrays as an input and output. Meanwhile, PySpark is not compatible with Numpy data types. It was thought that using UDFs in combination with a casting command to transform the Numpy output as PySpark's DenseVector data type would be enough to go back and forth from the PySpark framework to the audio processing framework. However, this approach proved cumbersome.

To work around this challenge, a single UDF was designed. This UDF would load the waveform, calculate the MFCC features, either pad or cut the MFCC features to a standard dimension of 452 steps, then flatten the MFCC features matrix, and return it casted as a DenseVector data type.

Having a big data platform, such as Databricks, proved valuable in this step. Runing this UDF would not terminate in a commodity system. The command terminated in Databricks.

To train the model, lazy evaluation had to be avoided once more. Thereby, the UDF to load the waveforms and extract the MFCC features, the test-train split command, and the logistic regression model fit were run in a single line of the Databricks notebook. The true value as well as the predictions columns, on both the train and the test set, had to be saved in the Volume as csv files again, in order to avoid the lazy evaluation to crash the notebook. Accuracy calculations were done separately using MS Excel and a Jupyter notebook in the local commodity system.

IV. Results

A. *Model Accuracy*

Both in the test set as well as in the train set the accuracy of the model was 0.54. Although this is better than random selection, the model was predicting the majority class label. After inspecting the datasets, it was found the data was imbalanced. See figure 1 and figure 2.

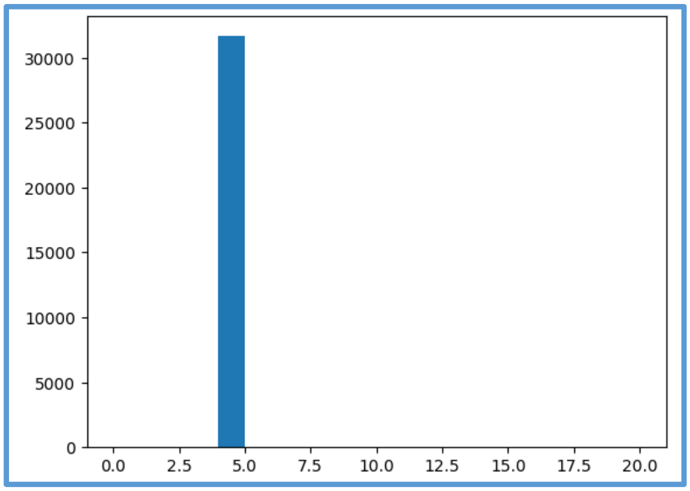

Fig. 1. Test Set Predicted Class

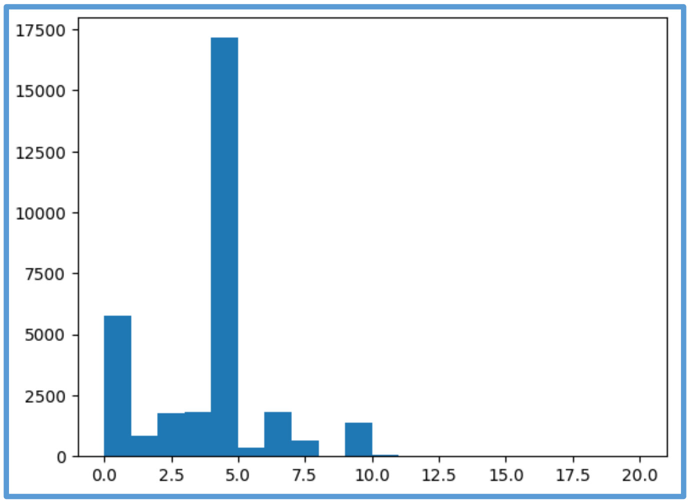

Fig. 2. Test Set Class Label True Value

It is hypothesized that the imbalanced dataset might have been a consequence of applying PySpark's 'sample' function. It seems that prior to randomly sampling a dataframe, this function orders the dataframe in accordance to values in one column, which might break independence and random sampling. Because of the nature of the dataset and due to having scarce computing capabilities, running tests to pinpoint whether the sample was truly random proved cumbersome. With more time to investigate, tests could be run on a simpler dataset. If this hypothesis is true, workarounds can be designed to guarantee true random sampling (e.g. generating a column with random numbers from the uniform distribution and then having the 'sample' function to organize the dataframe based on values of this column).

It is possible that the accuracy of the model can be improved by using other methodologies, such as: support vector machines or deep neural networks.

B. Improvements Required in the Dataset

It was noted that classes in the Common Voice dataset were not granular enough. Many of the classes in the Common Voice dataset confound accents that in reality are quite different, two clear examples are:

- Caribe: Cuba, Venezuela, Puerto Rico, República Dominicana, Panamá, Colombia caribeña, México caribeño, Costa del golfo de México
- Andino-Pacífico: Colombia, Perú, Ecuador, oeste de Bolivia y Venezuela andina

Even within the Andes region of Colombia, accents vary widely. The accent of a native from Medellín is different from the accent of someone from Bogotá. Likewise, within the Caribbean region of Venezuela, both the accent and the vocabulary of people from Maracaibo (western coast) and Cumaná (eastern coast) are quite different.

Furthermore, one of the challenges faced by future research is to catch sentences uttered in Spanish dialects that pronounce words differently. For example, dialects that replace the letter 'r' with an 'l' (some Puerto Rican accents) or that omit/deemphasize the letter 's' at the end of a word (Venezuelan). It can be seen in the examples above that in spite of the pronunciation differences, these two accents fall into the same category bucket within the Common Voice framework.

For these reasons, an improved version of the Common Voice dataset may be required to develop a fully reliable model. It might also be required a dataset with a larger proportion of observations reviewed so the 'validated' dataset could be used.

V. Conclusion and Future Work

This project set out to develop a classifier to predict the accent/dialect of a speaker reading a sentence in Spanish language recorded in an audio file. Because the focus of this project was in using big data platforms, only tools such as Dask or PySpark could be used. Furthermore, this project was self-financed, thereby storage units as well as computing resources in the cloud were limited (e.g. AWS S3 buckets were notoriously slower to read/write than commodity systems).

These limitations proved to be critical in pursuing the goal of this project in the following ways:

- PySpark is not compatible with Numpy data types, whereas most audio processing libraries rely heavily on Numpy. This imposed limitations to develop user defined functions (UDFs) in PySpark and on the machine learning methodologies available to develop the model
- The limited computing capability restricted the number of tests that could be run on the dataset as well as the amount of data that could be used. In turn this led to practical decisions that were aligned with the goal of meeting the semester's deadline (e.g. using a subset of the observations in the 'other' dataset instead of using all the observations available)

Taking into account that 21 accent labels were present in the usable sub-set of the Common Voice dataset, the achieved accuracy of 0.54 is well above random selection. However, the fact that the model was predicting the majority class label undermines the optimism derived from developing an accuracy above random classification.

It is assumed that future work on this problem will lead to better results given that the goal of using only big data platforms will be removed. It might be the case that big data platforms would still be useful (the loading of waveforms and subsequent preprocess terminated only in the Databricks platform). However, better results might be obtained by using the complete gamut of data analysis tools.

Among the approaches that should be tried in the future are: (i) implementing a Support Vector Machine model; and, (ii) implementing a transformer-like approach such as training the acoustic block of the HuBert model and connecting the output to a fully connected layer that outputs a probabilistic classifier.

Finally, it may be possible that an improved version of the Common Voice dataset for Spanish language is required to achieve the goal of having a useful and accurate model.

The main contribution of this work, if completed, would be to develop technology tools in the Spanish language. This project would also have commercial value as it would improve sound-activated systems by inserting an accent classifier module prior to the automatic speech recognition (ASR) module. The accent classifier would remove semantic ambiguity from the phrases uttered by the user and it is assumed this would lead to better understanding of the spoken commands received by the ASR system.

REFERENCES

[1] D. Jurafsky and J. H. Martin, Speech and Language Processing. 3rd ed. Draft, 2023. [Online]. Available: https://web.stanford.edu/~jurafsky/slp3/

[2] R. Ardila, M. Branson, K. Davis, et al, Common Voice: A Massively-Multilingual Speech Corpus. Presented at the proceedings of the 12th Conference on Language Resources and Evaluation, pages 4218–4222. Marseille, France 11–16 May 2020

[3] W.Hsu, B. Bolte, Y. Hubert Tsai, et al, HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units. [Online]. Available: https://arxiv.org/abs/2106.07447

[4] A. Baevski, H. Zhou, A. Mohamed, et al, wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. Print under review. [Online]. Available: https://arxiv.org/abs/2006.11477

[5] A. M. Ciobanu, S. Nisioi, L. P. Dinu, et al, Vanilla Classifiers for Distinguishing between Similar Languages. Presented at the proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects, pages 235–242. Osaka, Japan, 12 December 2016

[6] A. Guevara-Rukoz, I. Demirsahin, F. He, et al, Crowdsourcing Latin American Spanish for Low-Resource Text-to-Speech. Presented at the proceedings of the 12th Conference on Language Resources and Evaluation, pages 6504–6513. Marseille, France 11–16 May 2020