Speech recognition

Wiktor Reczek
Faculty of Computer Science, Electronics and Telecommunications
AGH University of Science and Technology

Abstract

We present speech recognition model based on audio files recorded by smartphones using *python_speech_features* library to extract features out of signals. This model based on *MFCC* and *MFCC delta features* has achieved accuracy at 96% when trained and evaluated on dataset from 16 different people with 13 spoken commands(audio).

Keywords: speech recognition, feature engineering, classifier, scaling

1. Introduction

Speech recognition is the ability of the program to recognize spoken language and convert it into written language. Simpler software has limited amount of words, which can be recognized and more complex software is able to recognize every word in given language, although with limited accuracy.

In this paper we describe the model, which recognizes spoken words in *wav* format, recorded by smartphones. Ultimately, the model can be used as a *Smart-Home Speech Recognition System*.

2. Methodology

We started with 68 audio files in *wav* format and corresponding 68 files containing time tags for different commands spoken in audio file. These files come from 16 different people at the age of 22-25 including 6 men. Every person recorded 4 audio files (excluding one person who recorded 8 files, with two different devices) with spoken commands. Some of them differed in sample rate, other differed in phonic sound reproduction(mono vs stereo), but all of them differed in quality(because of different recording devices). Our goal was to create a model that would recognize these commands.
For this purpose we read the data and split it into two sets: *training set* and *validation set,* convenient for use when trained/evaluated. Train size is dependent on dataset. For single subject we use 50-50 (i.e., 2 training files and 2 evaluating file) train-test-split and for all subjects we use 75-25 (i.e., 3 training files and 1 evaluating file) split. Then we extract the useful features out of signals*. We* used here *Mel-Frequency Cepstral Coefficients* from *python_speech_features* library. It produces MFCC matrix with 26 features(13 from MFCC and 13 from MFCC deltas). After that*,* we run the script to fit our model and then we can evaluate the model. Hyper-parameters used in our model are optimal as they were found by cross-validation.

MFCC matrix was calculated with 25ms length of the analysis window, 15ms length of the step between successive windows, 1700 FFT size and N=2 additional preceding and following frames.

Our model, which is RandomForestClassifier was trained with 500 estimators and default values for other hyper-parameters. It was trained to recognize commands, which were given as MFCC matrix, which were computed on trimmed piece of file containing only one command. Every row of this matrix where classified apart, then mode was calculated on them all and returned as a result of classification. That way we trained model to recognize commands for all subjects.

For one subject we separated commands into three subcommands: <COMMAND_TOP>, <COMMAND_MID>, <COMMAND_BOT>, which were the new labels for our rows in MFCC matrix (30% first rows – COMMAND_TOP, next 40% of rows – COMMAND_MID, and the rest 30% of rows - COMMAND_BOT), then where returning the result of classification, only the root of the comment was distinguished. That way we achieved about 8 percentage points results better than in previous approach(without dividing the command into subcommands), but as we see in the table below, that didn't work for every student.

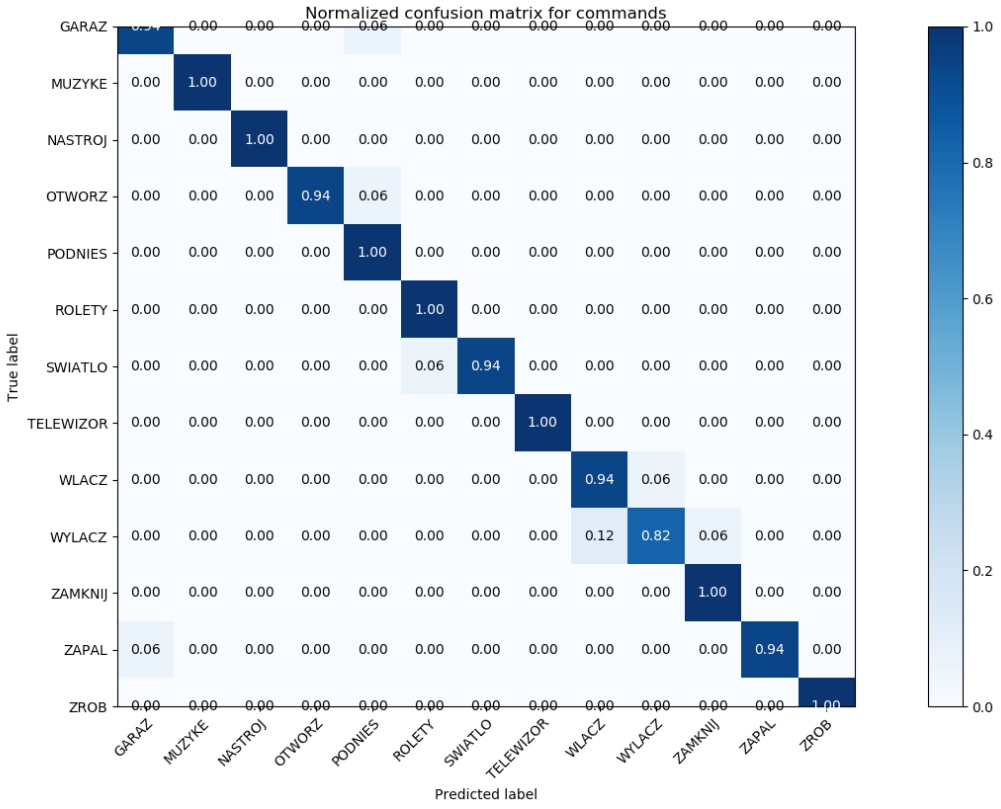
3. Results

For one subject:

Student ID	Accuracy [splitted / not splitted]
258118	88.46% / 92.31%
258126	61.54% / 73.08%
258135	84.61% / 88.46%
266701	96.15% / 84.62%
266702	92.31% / 92.31%
266708	69.23% / 69.23%
266710	76.92% / 57.69%
266711	96.15% / 100%
266712	100% / 92.31%
266723	80.77% / 73.08%
266725	96.15% / 96.15%
266753	88.46% / 96.15%
266761	100% / 100%
273352	96.15% / 90.38%
273356	92.31% / 84.62%
282075	96.15% / 96.15%

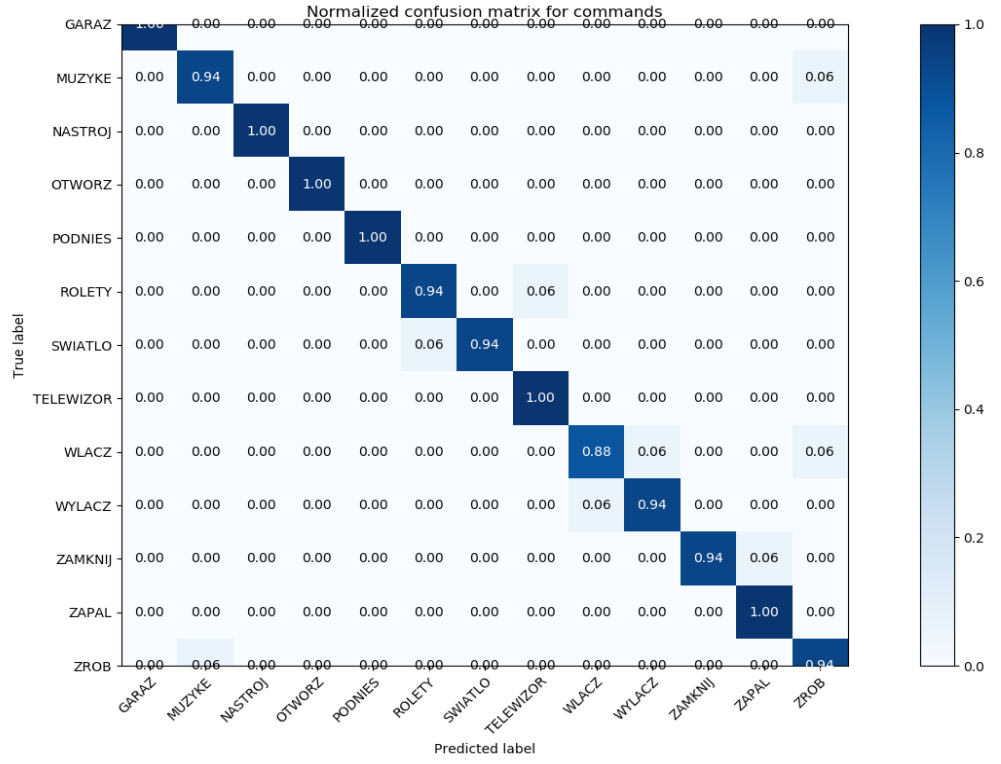
Table 1. Results of classification on one subject

As mentioned earlier, for all subjects we achieved accuracy of 96%.
Here is a *confusion matrix for* not splitted commands*:*



which gives 96.38% accuracy.

And *confusion matrix* for splitted commands:



which also gives 96.38% accuracy.

4. Conclusion

Our model based on RandomForestClassifier has very high accuracy in offline speech recognition, given that some of the given files were very poor quality and some of them were tagged incorrectly, then it seems to be promising in online speech recognition.

Sources:

[1] What is speech recognition

[2] Mel-Frequency Cepstral Coefficients