Speaker Fluency Level Classification Using Machine Learning Techniques

Alan Preciado Grijalva, Ramón F. Brena*

Grupo de Investigación en Sistemas Inteligentes, Tecnológico de Monterrey, México

*apreciado42@uabc.edu.mx

Abstract

Level assessment for foreign language students is necessary for putting them in the right learning group, but it is also a very time-consuming task, so we propose to automate the evaluation of speaker fluency level by implementing machine learning techniques. This work presents an audio processing system capable of classifying the level of fluency of non-native English speakers using five different machine learning models. As a first step, we have built our own dataset, which consists of labeled audio conversations in English between people ranging in different fluency domains/classes (low, intermediate, high). We segment the audio conversations into 5s non-overlapped audio segments and thereafter perform feature extraction on them. After this, we have tunned the appropriate number of Mel cepstral coefficients to be extracted extracted from the audios by evaluating accuracy performance. Thereafter, we have added the zero-crossing rate, root mean square energy and spectral flux features, proving that this improves model performance overall. Out of a total of 1424 audio segments, with 70% training data and 30% test data, our trained models have achieved a classification accuracy as high as 94.39% (support vector machine), the rest of our models have passed the 89% classification accuracy.

1. Introduction

The development of artificial intelligence (AI) - powered applications has been growing remarkably over the last decade [1]. In this context, mobile apps such as Cortana (Microsoft), Alexa (Amazon) or Siri (Apple) have proven to be a useful tool for daily human tasks. Other examples of AI-powered devices are autonomous cars [2], personal management, financial assistance and language learning apps. Most of these applications are integrating smoothly in our society. With regards to language learning apps, there are currently several software language companies that are employing AI techniques to improve user engagement and learning experience. The main promise of AI-powered language learning apps is that users will achieve basic proficiency in a foreign language as they progress through their lessons within a few months and with a small amount of time studying per day, all being guided by AI.

A slightly different language learning scenario is the one involving two or more (known or unknown) persons who are actively looking for tandem groups to improve their language skills. Currently, our group at ITESM is working on the construction of an AI-powered mobile app for language learning called *Avalinguo*. Avalinguo is an internet-based system, and it merges virtual reality with AI to create "digital classrooms" in which people, each one with a corresponding *avatar*, can practice a language [3].

Avalinguo has many benefits such as 1) users are not attached to fixed schedules, 2) it is portable and can be used anywhere (internet provided), 3) real-time real-person interaction, 4) user privacy is kept because of the use of avatars, 5) it clusters users based on profiles (target language, interests, etc.), 6) due to clustering, each login presents new possible matches with other users (recommendation system), 7) it contributes to a relaxed and casual participation by implementing fluency monitoring and topic recommendation during conversations and 8) its cost is inferior to particular online courses.

In this paper, we report work done related to point 7 by presenting the advancements corresponding fluency monitoring during a conversation between two or more persons. Our approach to this problem is based on audio analysis, starting with audio feature extraction and afterwards training machine learning (ML) models to perform classification of audio segments provided labeled target classes. Previously, there have been advancements in environmental sound classification [4] and real-time speech recognition based on neural networks [5]. In these cases, the audio sets have been environmental sounds (rain, cars, birds, etc.) and recorded speech, music and noise sounds, respectively. In our case, to approach the general problem of fluency level monitoring of each individual during a conversation, we have first proceeded to build our own audio set (*Avalinguo audio set*), the details of the audio set are presented in *section 2*. Thereafter, we have split each conversation in 5s non-overlapped segments, these segments have had some features extracted (mel coefficients + zero crossing rate + root-mean-square-energy + spectral flux). Later on, the feature vectors are fed into a classification model to train it and evaluate its performance using accuracy metrics. Our defined fluency classes are three: low fluency, intermediate fluency, and high fluency. We have compared five ML models, namely, multi-layer perceptron (MLP), support vector machines (SVM), random forest (RF), convolutional neural networks (CNN) and recurrent neu-

ral networks (RNN). The workflow described previously is the standard ML approach for the audio analysis of sound events [6].

The main hypotheses that we are trying to answer here is: *Given a labeled balanced audio set fulfilling predefined fluency metrics, can we construct a model capable of classifying the level of fluency of an audio segment?*

If so, this would allow us to determine whether a group in a conversation needs a recommended topic to keep it flowing. Also, we would be able to tell if Person A has a lower fluency than Person B and needs to be re-assigned to a lower fluency level group (same for the inverse case).

Our final classification results have achieved accuracies higher than 90% (except for one model), being the highest of up to 94.39% for an SVM. As a first step, we have determined the appropriate number of Mel coefficients (MFCCs) extracted to ensure high accuracies. Thereafter, we've proved that adding features to the baseline MFCCs such as zero-crossing-rate (ZCR), root-mean-squared-energy (RMSE) and spectral flux onset strength envelope (SF) increased overall model performance.

2. Avalinguo audio set

Having a clean-high quality data set is a must for any machine learning project. The main challenge for a model's architecture is to be able to grasp patterns among data so that it can be complex enough to perform accurate predictions/classifications. This can be affected if mislabeled or missing data is contained in the data set. On the other side, the technical limitations of a model have to do with the computational power available and the amount of labeled data required for appropriate training.

Unfortunately for us, there are no *publicly* available data sets that fulfill our research purposes. Whilst the community has been working extensively on the construction and support of audio sets [8], it has not been possible for us to address an audio set for audio-speech analysis composed of conversations by non-native English speakers. There are indeed audio corpora of people speaking English such as the UCSB and MUSAN [9-10], but no audio set of people who are actually learning the language (by this we mean people who hesitate when speaking, people who take long pauses when speaking, or also people who just speak too slowly).

We are mostly interested in recordings of this kind because the Avalinguo system will deal mainly with non-native English speakers who will present the speaking characteristics mentioned before. Due to this reason, we have decided to build our own audio set; *Avalinguo audio set* [11]. The Avalinguo audio set is a collection of audio recordings of people whose language fluency ranges from low to high. The audio recordings have the next common characteristics:

- Spontaneous (non-scripted) conversations
- Random conversation topics chosen by speakers
- Audios recorded with low-to-no background noise

The sources who provided the audio recordings are three: Friends/Family, Language Center (ITESM), and Youtube Audios. Each audio recording comprises a conversation that lasts around 10 minutes and has a wide range of topics from athletes speaking to physicists talking about science to sisters talking about their daily activities. All of these 10 minutes conversations were cut in 5s equally-sized segments and were thereafter manually labeled and assigned into one of the three fluency classes. Summarizing, the audio set consists of 1420 (5s duration) non-overlapped audio segments comprising three fluency classes (low, intermediate and high fluency). This is about 2 hours of recordings. The audios have sample rates ranging from 22050 Hz to 48000 Hz, are mono and multi-channel and were converted to MP3 format.

Figure 1 shows the class distribution of the audio set. The intermediate class has a higher percentage because, technically, it is easier to collect audios of people ranging in the intermediate level rather than the low level. Despite this percentage differences, we have kept all audio files to avoid reducing the size of the data set. Note that this can be considered a small audio set (ca. 2 hours) if it is compared with some public sets (such as Google AudioSet).

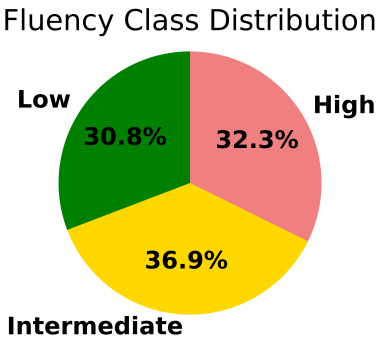

Figure 1: Avalinguo audio set class distribution. The set contains a total of 118.65 minutes of recorded audio.

2.1 Fluency metrics

Previously, we mentioned that each audio segment was labeled manually. In order to do this, we have defined baseline fluency levels definitions:

Low 0: Person uses very simple expressions and talks about things in a basic way. Speaks with unnatural pauses. Needs the other person to talk slowly to understand.

Low 1: Person can understand frequently used expressions and give basic personal information. Person can talk about simple things on familiar topics but still speaks with unnatural pauses.

Intermediate 2: Can deal with common situations, for example, traveling and restaurant ordering. Describes experiences and events and is capable of giving reasons, opinions or plans. Can still make some unnatural pauses.

Intermediate 3: Feels comfortable in most situations. Can interact spontaneously with native speakers but still makes prolonged pauses or incorrect use of some words. People can understand the person without putting too much effort.

High 4: Can speak without unnatural pauses (no hesitation), doesn't pause long to find expressions. Can use the language in a flexible way for social, academic, and professional

purposes.

High 5: Native-level speaker. Understands everything that reads and hears. Understand humor and subtle differences.

There is no single-universal definition for fluency. Actually, each language institution establishes a fluency metric for scoring based on their internal parameters. In our case, to score speaker fluency, we have taken the baseline definitions described above and have made specific emphasis on the next points regarding the concept:

1) Our metric is mainly sound based. With "fluent" meaning speaking without unnatural pauses.

2) If there is hesitation (slowness or pauses) when speaking, then that affects the fluency score of the speaker.

3) There is a distinction between fluency and proficiency. Meaning that fluency is someone able to feel comfortable, sound natural, and with the ability to manipulate all the parts of a sentence at will.

3. Experimental Framework

Our experimental procedure is the standard ML approach for analysis of sound events [12]. Our system consists of three main steps (see *Figure 2*): Feature extraction, classification and output of the predicted label with higher probability for individual segments (frames). First, we perform feature extraction on each audio frame. Thereafter, the feature vector is fed into a classifier which outputs the most probable class based on previous training.

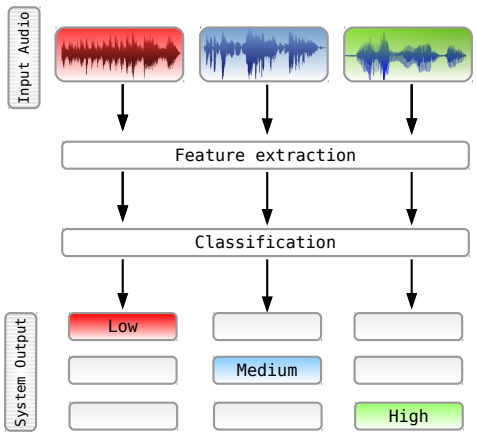

Figure 2: Pipeline of an audio classification system.

3.1 Audio segmentation

Audio segmentation is one of the most important preprocessing steps in most audio applications [13]. Research groups often propose novel segmentation frameworks in order to improve audio applications such as speech recognition [14]. In our case, our segmentation method simply consists on cutting the audio files into 5s non-overlapped segments and manually assign a frame to a single person (for example, Frame 1 belongs to person A). If a frame contains more than one person speaking, we assign it to the person that speaks more in it. After this, we proceed to label frames according to their fluency level. The segments were cut from the audios using a python module called Pydub [1].

Due to the way we proceeded, we didn't work with overlapped segments; since we are interested in differentiating persons among a conversation, this approach would have had ended as non-overlapped segments but with shorter time durations. The other approach that we explored was using voice activity detection (VAD), this approach creates audio segments when it detects a different voice or a pause during a conversation. However, the VAD interface [15] suppresses any possible silence within the conversation, that is, it only creates segments when people are speaking. Since we are interested in detecting possible silences and pauses in each audio segment, we have discarded this approach.

3.2 Feature Extraction

This is the step where features are extracted. We have written a python script to perform feature extraction for the created audio segments using a python package called Librosa [2] (among many functionalities, Librosa is commonly used for feature extraction, allowing to compute more than thirty audio features). The audio frames $f_0, ..., f_n$, have thus their corresponding feature vectors $p_0, ..., p_n$.

For the results presented in this paper, we have first varied the number of MFCCs extracted, this with the purpose to estimate the appropriate number of MFFCs to extract. Later on, we have added the ZCR, RMSE and SF to determine if this improves model performance.

Besides feature extraction, Librosa allows to plot audio spectrograms using its *display.specshow* function. For the sake of completeness, we show some commonly spectrograms for a single audio frame (see *Figure 3*).

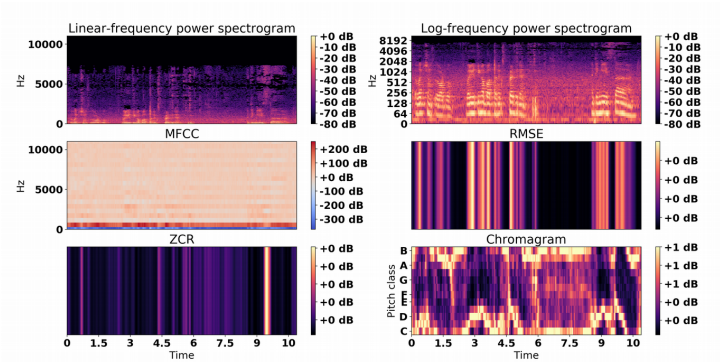

Figure 3: Spectral feature plots (single frame).

[1] Pydub: High level python interface for audio manipulation

[2] LibROSA: python package for music and audio analysis.

3.3 Audio Classification

In this work we are interested in comparing different classification frameworks, namely, multilayer perceptrons, convolutional neural networks, recurrent neural networks, support vector machines and random forest. The main goal here is to test each model classification accuracies as we vary the number of features extracted.

The proposed multilayer perceptron architecture has two hidden layers comprising 512 x 512 neurons followed by an output layer consisting of three neurons (one representing each fluency class). Each neuron within the hidden layers uses the relu function as the activation function. For the output layer, we implement the softmax function to convert the output into class probabilities. Finally, the predicted label is the class with the highest probability.

The convolutional neural network architecture has four hidden layers, the first two have 64 convolution filters and the following two have 32 convolution filters. The output layer consists of three neurons corresponding to our three classes and similarly to the multilayer perceptron, the activation function of the hidden layers is the relu and for the output layer is the softmax function.

The recurrent neural network architecture is a long-short-therm-memory (LSTM) [18]. It has two hidden layers comprising 256 x 32 neurons and three neurons corresponding to the output layer. This architecture also implements the same activation functions as the other two networks.

In *table 1* we summarize the architectures described above.

Neural Network	Hidden layers	Neurons	Activation
MLP	2	512x512x3	relu, softmax
CNN	4	64x64x32x32x3	relu, softmax
RNN	2	256x32x3	relu, softmax

Table 1: Neural networks architectures.

The other two models are traditional machine learning models. One is a support vector machine. This model has a basic construction (similar to the one proposed in the scikit-learn documentation). The main hyper-parameter of the estimator that we varied was the *regularization parameter C.* [3]

The other model is a random forest. This model also has a basic construction in the sense that our python script only initializes the model, trains it and then evaluates its performance. The single parameter we varied here was the *number of estimators* (number of trees).[4]

4. Experimental Results

In this section we present our feature extraction and classification results. The main data analysis tools that we have used for our experiments are Anaconda (under Python 2.7), Keras (backend Tensorflow), Scikit-learn, Librosa, Pydub and Pandas.

We have evaluated different model constructions, here, we report the model architectures (from *Table 1*) and hyper-parameters that achieved the highest classification accuracies.

[3]Link to SVM scikit-learn documentation.

[4]Link to RF scikit-learn documentation.

As a first step, we explore the effect of varying the number of Mel coefficients (N_{mel}) on the final accuracy. Increasing the value N_{mel} increases the complexity of a model, thus, it is our duty to find out the trade-off between the appropriate N_{mel} and the maximum achievable accuracy. There is a point in which adding more N_{mel} doesn't translate into considerable improvements (this can either increase accuracy by a small percentage or decrease it). With this experiment, we are able to get the appropriate N_{mel} for our data set.

As a second step, we show how adding features such as ZCR, RMSE and SF to the baseline chosen N_{mel} boosts accuracy in most of the cases.

4.1 Classification Experiments

We have chosen **SVM** and **RF** as our models based on research regarding the most appropriate ML approaches for audio classification. Both of these models have proven to be good candidates for the classification of sound events, such as the *ECS-50* audio set, as proposed by Piczak [17].

We are also comparing three different neural network models; the **MLP** architecture has proven to classify accurately speech, audio and noise audio of the *MUSAN* audio set as reported by Wetzel et al. [5], **CNNs** have also been used to classify the *ECS-50* audio set [4] and have also been used to classify audio without performing prior feature extraction, in the sense that the network itself extracts corresponding features from the waveform sample [12], lastly, we have also employed **RNNs** because according to Huy Phan et al., this type of neural networks achieved an accuracy of 97% in the classification of sound scenes from the *LITIS Rouen* dataset [19].

All our models were randomly sampled with 70% training data and 30% test data. Given the 1424 total audio frames of the Avalinguo audio set, this corresponds to 926 audio frames for training and 498 audio frames for testing.

In our first experiment, we have trained and evaluated accuracy performance with increasing values of N_{mel}. *Table 2* contains the achieved accuracies as we set $N_{mel} = 5, 10, 12, 20$. In each case, the accuracy improved considerably as the value N_{mel} increased. We obtained an accuracy as high as 94.39% with the SVM for $N_{mel} = 20$. We also trained our models with $N_{mel} = 30, 40$ but this only increased feature space dimensionality but not accuracy. From this experiment, we see that the commonly number of Mel coefficients used for audio-analysis (12 to 20) applies to our data as well.

Model	5	10	12	20
SVM	86.00%	89.49%	92.06%	94.39%
RF	84.80%	89.00%	90.42%	92.29%
MLP	78.00%	88.78%	89.01%	92.05%
CNN	80.00%	85.04%	87.61%	93.69%
RNN	78.90%	85.04%	86.44%	87.00%

Table 2: Accuracy performance of the classification models for different N_{mel} values.

The second stage consists in taking the $N_{mel} = 20$ as base-

line features and test extra spectral features to see the outcome. After doing feature exploration in our runs, we have ended up adding ZCR (as proposed in [5]), as well as RMSE and SF (as proposed in [7]). This translates to a 23-dimensional feature space (20 MFCCS + 1 ZCR + 1 RMSE + 1 SF).

The architecture of our models combined with the total final features yields a runtime of about 1 ms per classification on a 2.4 GHz single core CPU for the neural networks. For the SF and RF, it takes about two seconds to train completely.

Table 3 shows the obtained accuracies with the extra features. Once again, the SVM achieved the highest accuracy followed by the RF. The MLP and CNN obtained similar results, the one big difference is that the MLP outperformed the CNN in computing time; the former took about 1min to train completely, whereas the latter took about 5 mins to train completely. The RNN obtained the lowest accuracy and it took about 7 mins to completely train.

In contrast with the results from *table 2* (case $N_{mel} = 20$), the results in *table 3* ($N_{mel} = 20$ + extra features) show that the performance of the SVM remained equal, the CNN performance decreased by around 1% and the other three models increased their accuracy. We present graphically this comparison with a bar plot in *Figure 4*.

Features	SVM	RF	MLP	CNN	RNN
N_{mel}+ZCR+RMSE+SF	94.39%	93.45%	92.52%	92.75%	89.01%

Table 3: Accuracy performance of the classification models 20 MFCCs + extra features.

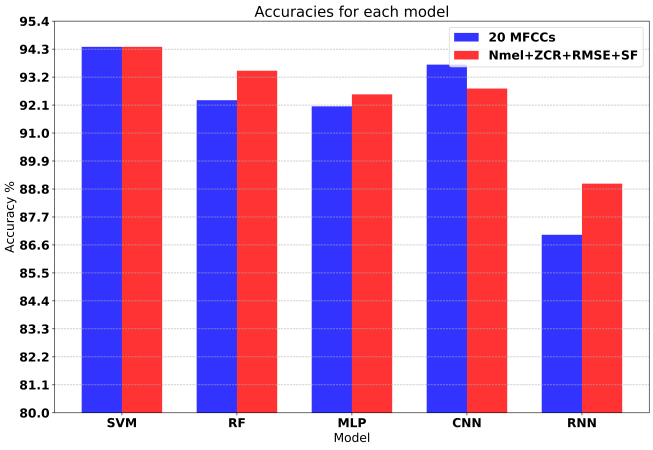

Figure 4: Accuracy comparison when using $N_{mel} = 20$ (blue bars) and with extra features (red bars). For sake of visualization, the accuracy of the plots starts at 80%.

With these results, we have that the SVM followed by the RF are the models that best classify our data set. The reason why the SVM couldn't improve further with the extra features can have to do with the design of the model itself. In this case, we varied the choice of the regularization parameter C but couldn't obtain any better performance. In the case of the RF, we increased the *number of trees* when we added the other features, gaining more than 1% accuracy.

The neural networks have slightly underperformed comparing them with the other two models. But they have achieved decent accuracies as well. The main difference between the deep learning and the traditional models has to do with the time it takes them to train, requiring the latter way less computational time. The under-performance of the neural networks does not exclude them from this analysis at all, it can be that a different architecture can boost their accuracies.

In order to evaluate the quality of the output of the classifiers we have used a confusion matrix. The corresponding map can be seen in *Figure 5* and it belongs to the SVM results, which trained to classify among our three fluency levels, has achieved a classification accuracy of 94.39%. The matrix was plotted using the *sklearn.metrics* module from scikit-learn [20]. From the plot, we see that the classifier actually predicted all the high labels (classes) correctly. For the intermediate label, it misclassified 15 audio frames either as a low or high label, this is understandable since, intuitively, it is harder to discriminate if a frame lies in the intermediate level or if it belongs to any of its "neighbors". For the low label results, the SVM predicted two audio frames as highly fluent whilst they belong to the low fluency class, this could be the "most critical" mistake our model has done by predicting that two audios whose fluency is low, actually has an almost native fluency. However, in the overall, the model has performed remarkably.

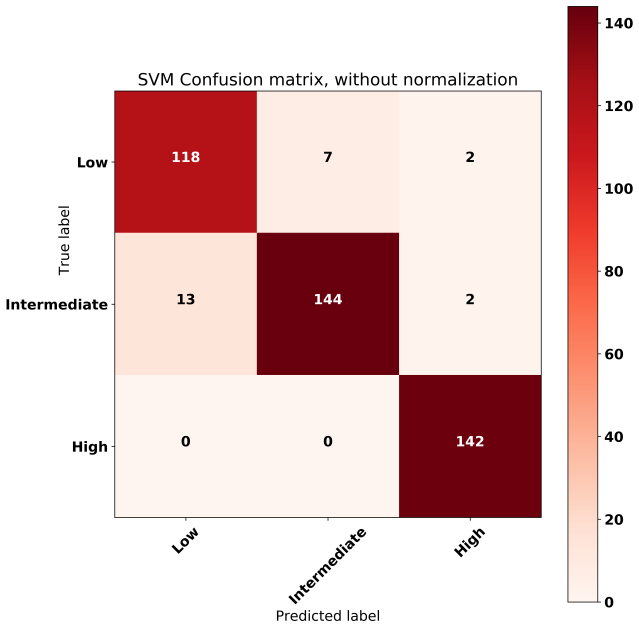

Figure 5: Confusion matrix for the SVM trained and tested with the Avalinguo audio set.

The link to the repository with the Python script to replicate this paper can be found in [16]. All the technical requirements to run the code are documented in the attached repository.

5. Conclusions

In this work, we have presented an audio processing system capable of determining the level of fluency of non-native English speakers, taken 5s non-overlapped audio segments from the Avalinguo audio set. We have used five different ML models to classify audio segments into low, intermediate or high fluency levels. Each model was capable of classifying audio frames with an accuracy of more than 90% (except one classifier that reached 89%).

As a first step, we have determined that the appropriate number of Mel cepstral coefficients for our data set is 20. Thereafter, with these baseline features, we have added zero-crossing rate, root mean square energy and spectral flux features to improve the accuracy of our models. The highest accuracies were reached by SVM and RF, with 94.39% and 93.45%, respectively. The neural networks achieved also remarkable accuracies (MLP 92.52%, CNN 92.75%, RNN 89.01%).

We have also reported the construction and details of the Avalinguo audio set, whose main characteristic is that it is composed by conversations of people who are learning the English language.

The accuracies that we have achieved can be considered high but nonetheless there is room for improvement. For example, tunning more precisely the hyper-parameters of the SVM and RF estimators by running grid searches. In the case of the neural networks architecture, we can still explore adding specific hidden layers and modifying the number of neurons per layer. Added to this, as other works have proposed, exploring with other audio features such as chromagram, Mel spectrograms or spectral contrast can improve accuracy. We must take into account that we are only defining three fluency classes, in order to make fluency levels more specific, we would have to define more fluency classes in between. This poses a challenge for the accuracy performance.

The main technical limitation that we have right now is that the Avalinguo audio set (with about 2 hrs of recordings) can be considered a small set. Part of our future work consists in maintaining and increasing the size of the audio set. Another technical challenge consists in automatically identifying persons within a conversation and at the same time, capturing the silences and pauses they can make.

This project will be integrated to the bigger Avalinguo system. Here, we will have to deal with the classification of live conversations, for example.

6. References

[1] David W. Cearley, Brian Burke, Samantha Searle and Mike J. Walker. *Top 10 Strategic Technology Trends for 2018* (2017). Gartner, Inc.

[2] Robert Spangenberg, Daniel Goehring and Raúl Rojas. *Pole-based Localization for Autonomous Vehicles in Urban Scenarios.* IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2016).

[3] Avalinguo Project Link

[4] Karol J. Piczak. *Environmental Sound Classification with Convolutional Neural Networks.* IEEE International Workshop On Machine Learning For Signal Processing (2015). Boston, USA.

[5] Micha Wetzel, Matthias Sperber and Alexander Waibel. *Audio Segmentation for Robust Real-Time Speech Recognition Based on Neural Networks* (2016).

[6] Toni Heittola, Emre Çakır and Tuomas Virtanen. *The Machine Learning Approach for Analysis of Sound Scenes and Events.* Springer International Publishing AG (2018).

[7] Ashutosh Kulkarni, Deepak Iyer and Srinivasa Rangan Sridharan. *Audio Segmentation.* Stanford University.

[8] Iulian Vlad Serban, Ryan Lowe, Peter Henderson, Laurent Charlin and Joelle Pineau. *A Survey of Available Corpora for Building Data-Driven Dialogue Systems.* arXiv:1512.05742v3 (2017).

[9] UC Santa Barbara Corpus of Spoken American English

[10] David Snyder, Guoguo Chen and Daniel Povey. *MUSAN: A Music, Speech, and Noise Corpus.* In: Computing Research Repository (2015).

[11] Github repository of the Avalinguo Audio Set

[12] Maxime Jumelle and Taqiyeddine Sakmeche. *Speaker Clustering With Neural Networks And Audio Processing.* arXiv: 1803.08276v1 (2018).

[13] David Rybach, Christian Gollan, Ralf Schlüter and Hermann Ney. *Audio Segmentation for Speech Recognition Using Segment Features.* ICASSP (2009).

[14] Manpreet Kaur and Amanpreet Kaur. *A Review: Different methods of segmenting a continuous speech signal into basic units.* International Journal Of Engineering And Computer Science (2013).

[15] Link to Python interface to the WebRTC Voice Activity Detector.

[16] Link to code for paper replication (github).

[17] Karol J. Piczak. *ESC: Dataset for Environmental Sound Classification.*

[18] Sepp Hochreiter and Jurgen Schmidhuber. *LONG SHORT-TERM MEMORY.* Neural Computation 9(8):1735-1780 (1997).

[19] Huy Phan, Philipp Koch, Fabrice Katzberg, Marco Maass, Radoslaw Mazur and Alfred Mertins. *Audio Scene Classification with Deep Recurrent Neural Networks.*

arXiv:1703.04770v2 (2017).

[20] Pedregosa et al., *Scikit-learn: Machine Learning in Python*, JMLR 12, pp. 2825-2830 (2011).

Acknowledgments

I would like to thank Prof. Brena from the Intelligent Systems Department at ITESM for me allowing to work in his group, for his guidance and support throughout my participation in this project. Special thanks to the people involved in the construction of the dataset and to the Language Center at ITESM for providing useful audio recordings. I'm thankful also with my cousin Eduardo who provided me with housing during my internship in Monterrey.