EPFL

CHILI
computer-human interaction
in learning and instruction

LEARN
Center for Learning Sciences

Classification of Classroom Discourse towards a Reflective Teacher Dashboard

Author:
Léo Claude Hauser

Supervisor(s):
Sina Shahmoradi

Professor:
Pierre Dillenbourg

April 8, 2023

Contents

1 Introduction

During the last few years, the existing ways of teaching have considerably changed. New technologies have provided new tools to the teachers. They were first considered as a help for the them, but then it appears that using and teaching with these new technologies is crucial to have an educational background that is coherent in today's world. The Automatic Teacher-Reflection Dashboard that I worked on with this project is relevant in the actual educational context because it provides the teacher a modern tool to get a feedback of the lessons. It is also a useful tool when it comes to teacher training.
The main idea is to give the teacher an almost immediate timeline of the course content, where he/she can see when he/she was talking alone, when there was teacher-student interaction, when the students were speaking alone or when the class was silent. This speech segmentation is a precious tool for the teachers, especially in the lessons where interaction is key, like the ones using robots as we will see later. These highly interactive courses were my main focus in this project, because it is in this context where this dashboard is the most useful. For example, the teacher can see very easily if there was enough interaction with the students or not, and it gives him/her hints to eventually improve a particular section of the lesson. But this kind of course is also the most challenging one, because we have to deal with a lot of different voices at different intensities at the same time.
Using machine learning models to extract useful information of a class discourse is a research topic that exists, but is still quite unexplored. This paper [5] shows that an algorithm is able to identify teacher questions in a classroom and was a reference for this project.
Overall the goal of this project is to provide the teachers a quick and easy tool that gives a global overview of the lesson content that can be used to improve the way of teaching.

2 Data and Methods

2.1 Data

As explained in the introduction, this project relies on classroom recordings for its usage, but also for its training and testing.

2.1.1 Raw Data

To do so, I worked on audio records of mathematical lessons where the Cellulo Robots were used. They are designed to be very simple to use and their main purpose is to be used on paper sheets that contain learning activities. They are also designed to be used as swarm robots, but not in our case. The children, aged between 8 and 12, used the robots to learn the concept of axis and coordinates. This kind of lesson is divided in three different activities where they work in groups. In these groups, one student holds a tablet and the others work with the robot. Their goal is to find a particular location that is shown on the tablet, and they should slowly understand by themselves that the coordinate system is the most efficient way to find a point.
These lessons were recorded using a microphone that the teacher wears during the session. No other additional microphones were present in the classroom.
In this project, I worked on 3 different class recordings. The teacher and the students are different in every file, and the lessons were taught in different languages. The gender of the teachers is also different. These recordings were provided by the laboratory and information about each file can be found in table 1.
If we look at the file in a more technical point of view, they all use the .wav format. This format is commonly use in speech analysis algorithms because its structure is particularly adapted for signal processing, especially in Python.

	E_1.wav	**E_2.wav**	**F_1.wav**
Language	English	English	French
Duration	1:23:23	1:14:53	1:26:27
Gender of teacher	Male	Female	Male

Table 1: Audio files details

2.1.2 Data Labelling

As introduced before, the idea is to create an algorithm that generates a content feedback of the lesson. To do so, we need to define categories that are relevant for the teacher. Another important constraint is that every audio segment in a given time window must be classifiable into one and only one category. Here are the 4 categories that have been chosen for this project:

1. **Interaction**: as explained in the introduction, the interaction detection was focused in this project because it represents the essence of these robotic classes. In this project, interaction is defined as teacher and student(s) speaking together in a given time window. It does not matter how long every participant is talking, as long as he/she is talking. For example, if in a 10 seconds time window the teacher is speaking 1 second and a student 9 seconds or vice-versa, it is still considered as interaction.

2. **Teacher only**: this category represents a time window where the teacher is the only person that is talking. The speech length does not matter, as long as no student is speaking. We have two main cases in this category: the teacher is speaking in the front of the class (the speech is for all students) or the teacher is talking individually with a specific student/group. This distinction is important for the algorithm as we will see later in this report.

3. **Student(s) only**: it is the same idea as the previous category. One or many students are speaking without any teacher voice in it. We also have two main cases here: a student is explaining something to the entire classroom (when the teacher asks an open question for example) or when one or many students are talking to the teacher in a more individual/group-level way.

4. **Silence**: it is simply defined as a time window where no one is speaking. Obviously, a classroom is never completely silent. So the definition of a "silent" classroom has to be made with a noise tolerance as we will see later.

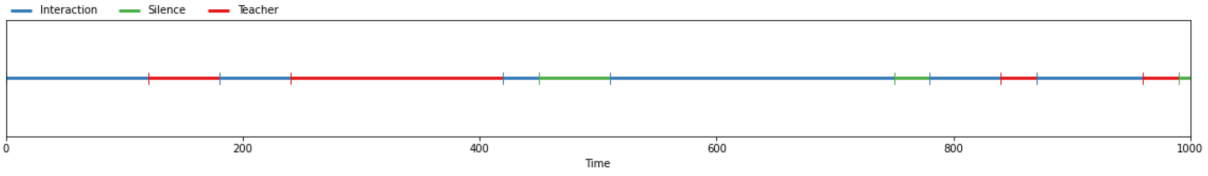


Figure 1: Possible timeline output (time in seconds)

As introduced above, I had to listen to the 3 provided class recordings and manually label them. To do that, I used a spreadsheet with 2 columns: the first one for the time window index and the second one for the label. The categories were respectively named **I**, **T**, **S** and **Sil**. Then, the spreadsheet is converted into a .csv file. This format is used all over the place for labelling because it is easy to upload and work with in a Python program, especially with the commonly used libraries.

2.1.3 Window Size Selection

Obviously, these categories cannot be defined without a time window. For example, it is impossible to classify a sound as "Interaction" if it is only based on a given time t. We need to define a time window size that is large enough to contain enough sound to do a proper classification. But on the other hand, it cannot be too large because at some point, the classification does not make sense anymore. For example, if we take a 1 minute window size, we have a high probability to have 3 or even 4 categories in it.
Another reason to have a quite small time window size is because it is more relevant for the teacher. If he or she wants to see when the students were speaking and the window size is too large, the teacher would have to search manually over the recording window when the students were talking. In practice, it is even worse than that, because the moment when they were talking has high probability to be ignored or not classified correctly if the other sounds in this window belong to another category.
After many tests, I have chosen a fixed 10 seconds time window. With this size, the teacher will have a precise timeline where a specific lesson segment can be easily isolated. I also noticed that a "typical" teacher-student interaction (at least in these robotic sessions) is about 10 seconds. In a more practical point of view, the labelling that I had to do manually with these recordings was easier for me with a fixed and rounded time value.

2.1.4 Segments Distribution

The total number of each label for every audio file is shown in table 2. As we can see in this table, we have a high number of "Interaction" segments in every audio file. It can be easily explained by the nature of the lesson. We also have a decent number of "Teacher Only" speech segments for every file. On the other hand, we do not have a lot of "Silence" and "Student(s) Only" speeches. This difference has an impact on how the algorithm is constructed as we will see later.

	E_1.wav	**E_2.wav**	**F_1.wav**
Interaction (I)	346	274	350
Teacher Only (T)	141	134	105
Student(s) Only (S)	13	41	39
Silence (Sil)	1	0	19
Total	501	449	513

Table 2: Number of labeled segments of each category

2.2 Methodology

As shown in figure 2, we have a linear methodology for this project. For every audio file (every teacher), I had to first convert every recording to the right format (.wav). Then, I labelled the data independently according to the method explained in section 2.1.2. After that, the data is uploaded into the Python algorithm and goes through different analysis and pre-processing steps. These steps will be explained in details in the next sections.
The processed data is then passed to the threshold based algorithm which relies on the pre-processing. The data will be segmented and, based on certain conditions that will be explained in the corresponding section, will be classified directly or still have to go through a machine learning model. The purpose of this model is to classify the segments that the previous algorithm was not able to classify.
Finally, the user gets a classification output for every time window of the file (or a selected part of it) in the form of an array that can be easily converted into a timeline if necessary.

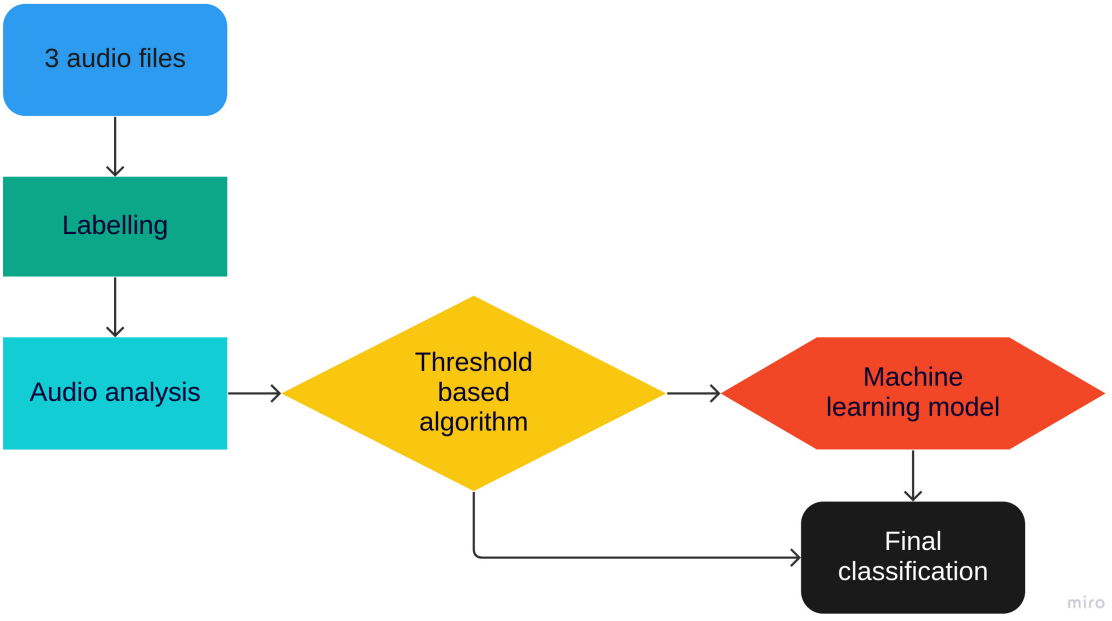


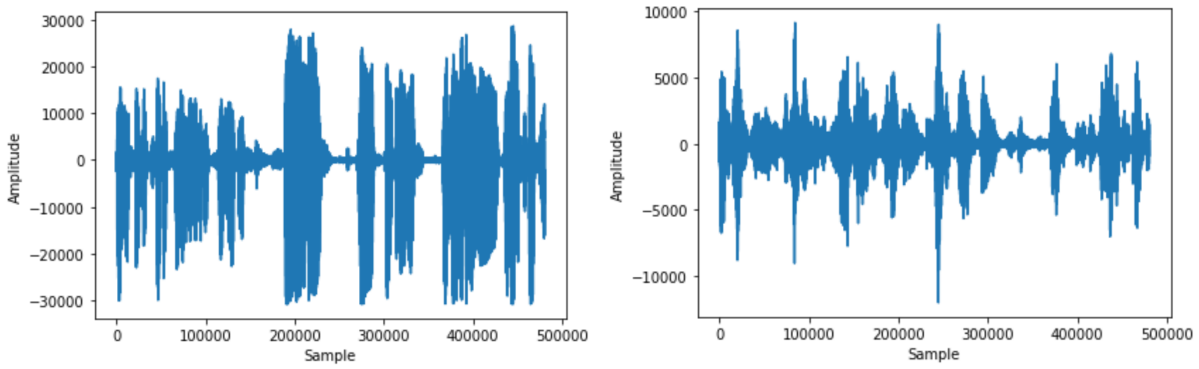
Figure 2: Project methodology diagram

2.3 Audio Analysis

As introduced in the methodology section, it is necessary to do an audio analysis and pre-processing before executing any classification algorithm. In this section, we will explore the different signal processing techniques that are used in the classification algorithm. Note that these processing methods are totally independent from each other.

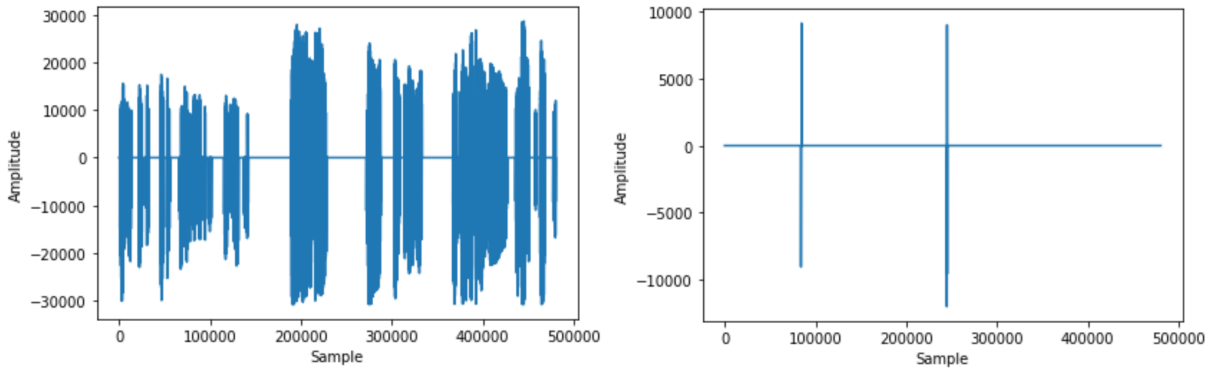
2.3.1 Amplitude

The amplitude analysis is probably one of the most basic techniques that we will see in this section, but still has a key role in this project. Note that since we use .wav files in this project, the amplitude definition is described accordingly.
The audio file can be described as a table containing an amplitude for every time unit and a rate. If we are in the case of a stereo file, it is the same but with 2 tables. The rate [Hz] determines how many amplitude measures (samples) are made in 1 second. In this project, the files have a 48kHz rate and 16 bits amplitude. Thus, the amplitude is in a range from -32768 to +32767.



(a) "Teacher Only" segment (b) "Student(s) Only" segment

Figure 3: Amplitude of every sample in a 10 seconds interval for 2 different categories



(a) "Teacher Only" segment (b) "Student(s) Only" segment

Figure 4: Amplitude of every sample in a 10 seconds interval for 2 different categories after applying the threshold

In the context of this project, the teacher always has a wearable microphone. It is a key parameter, because the amplitude analysis of the signal relies on this parameter to be as accurate as possible. As we can see in figure 3, the amplitude of the teacher's voice can reach 30000 (edge case). For a segment of type "Student(s) Only", the peaks are close to 10000. This difference is the first key element of the classification algorithm because we can use a threshold to detect if the teacher is talking or not. If we apply a function that keeps only amplitudes that are above a certain threshold in absolute value (9000 was chosen in this example) and put the others to 0 otherwise, we obtain the graphs shown in figure 4.
After applying the threshold, we can clearly see that all the content of the "Teacher Only" segment remains and almost nothing in the "Student(s) Only" one. The first part of the algorithm, namely the *Threshold Based Algorithm*, uses this observation to detect if the teacher is speaking. The same principle can also be used to detect if the class is silent if another threshold is applied. The eventual remaining peaks like the ones in the "Student(s) Only" segment are generally not taken into account by the algorithm because the detection is based on a ratio calculation over the entire time window as we will see later. So an isolated peak will be simply ignored, because it often corresponds to noise or a quick shout in the classroom.

2.3.2 MFCC

MFCCs stands for Mel-frequency cepstral coefficients. It is a commonly used signal processing technique for speech detection/recognition. To understand what it is, we first need to focus on the Fast Fourier Transform.
If we take a signal that is in the time domain, or simply an amplitude measure at a given time, the Fast Fourier Transform (FFT) allows us to transform it to the frequency domain. This kind of graph is called a Fourier spectrum and shows the frequency amplitude at a given time interval. An example is shown in figure 5.

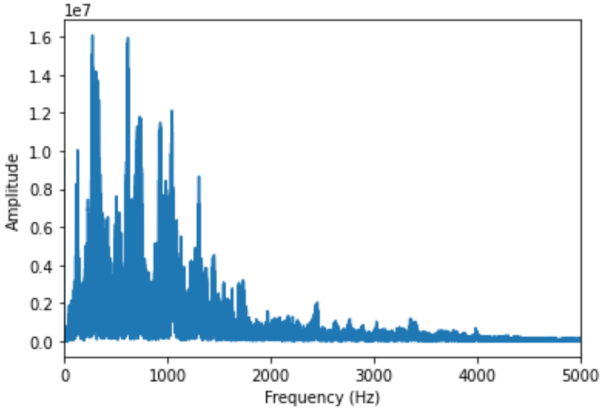


Figure 5: Applying the FFT to "Student(s) Only" segment

The FFT is the first step of the MFCC computation. After that, we must compute the log of the magnitude of this Fourier spectrum, and then again compute the FFT spectrum of this log by a cosine transformation. But in practice, these computations can be done quite easily using libraries. The log computation is done by using the *Mel scale*. This scale contains coefficients that have been determined experimentally to capture as close as possible the frequency variations a human being perceives. Figures 6 and 7 represent the obtained *cepstrums* after the MFCCs extraction. Note that in these figures, the computed magnitudes have been transformed to [dB] for a better understanding, but is not done in practice.
It is hard to interpret the value of the MFCCs as a human (because it is not made for), but we still can make some observations. In figure 6, we can see that low coefficients have a higher value between 10 and 15 seconds. This can be explained by the high amplitude of low frequencies generated by the teacher's speech. The same observation can be made around 25 seconds. On the other hand, the lack of high amplitude of low frequencies is clearly visible in figure 7.

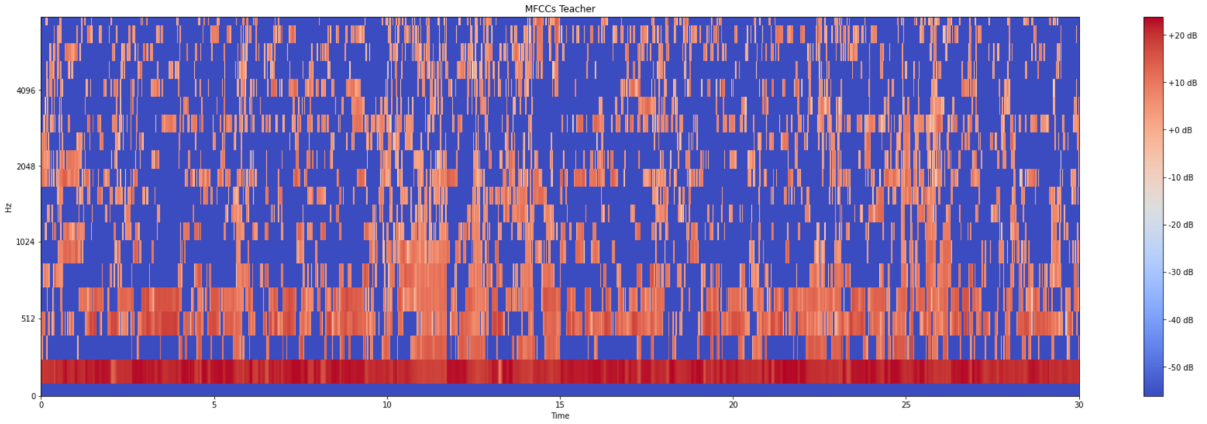


Figure 6: Extracting MFCCs from a "Teacher Only" segment

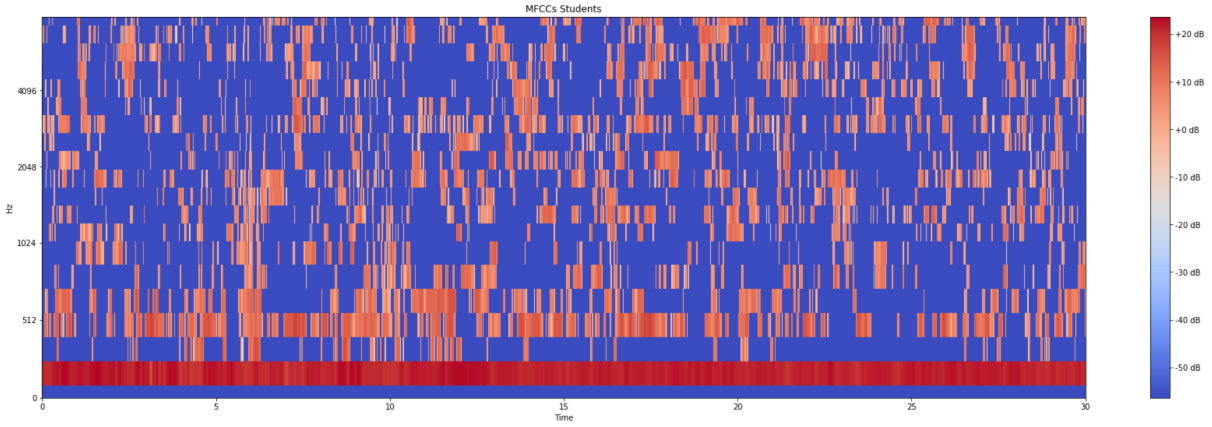


Figure 7: Extracting MFCCs from a "Student(s) Only" segment

When computing the MFCCs, it is necessary to choose how many coefficients we need. This number represents how many frequency envelopes we have for each time unit. In other words, in how many elements the frequency axis is divided (the default number is 20 as shown in the figures). The coefficients are used as *feature vectors* in the machine learning model. In this project, a number of 26 coefficients has been chosen based on this paper [4].
It is relevant to use this paper as a basis because it is close to this project by construction. Their goal was to improve a speech recognition system using a machine learning model trained by MFCCs and PCA (see section 2.5.1) segments as done here.

2.4 Threshold Based Algorithm

We can now focus on the first part of the classification algorithm. We will go over the main structure of the code. The threshold based algorithm is part of the segment_signal function that can be found here [8].

2.4.1 Thresholds Application

The first step of the algorithm is to load the audio file according to the path and time interval given by the user. If the file is stereo, on channel is automatically selected. Then, two thresholds are applied to the whole file based on the amplitude as explained previously in section 2.3.1. The purpose of the first applied threshold is to detect if the class is silent. It means that it has to be just a little bit lower than any student or teacher's speech amplitude. In a segment that is silent, the amplitude will be 0 at any time after application. After that, another threshold is applied to the file and isolates the teacher's speech. After both applications, the new generated audio files are saved as new .wav files to be able to relax the RAM if necessary.
It is clear that the 2 thresholds are different for every teacher and class (other people and room), so I had to determine them with an audio analysis for every class recording. In the code, they are defined as constants.

2.4.2 Segments Analysis

The goal of this step is to compute the ratio of every category inside a segment according to the threshold. To do that, we divide the 2 files where a threshold was applied into segments. These audio segments have a fixed size of 10 seconds. For every segment, a teacher, student and a silence counter is set up. Then, the algorithm iterates over *samples envelopes* (250ms) inside the segment. If it detects that the envelope contains amplitudes that are not null (according to the different thresholds), the corresponding segment counter is incremented. The idea behind *samples envelopes* is that after a threshold, it is often the case that we just have a thin peak in the amplitude when a word is said (almost no duration). To have an exhaustive category ratio inside a segment, we need to approximate typical word duration. After readings and observations, it was fixed at 250ms and is represented by a *samples envelope*.

After analyzing every envelope, the duration ratio of every category is computed for every segment using the counters (how many samples envelope belonging to a given category over all samples envelopes in the segment).

2.4.3 Segments Classification

The final step of this algorithm is to decide which category a segment belongs to. To be able to do that, we need to define minimal ratio constants for "Teacher Only", "Student(s) Only" and "Silence" categories. This part is a branch where the algorithm checks if the segment has the minimal teacher speech ratio. If it is the case, this segment will be temporarily classified as "Segment containing teacher speech", so it can also be "Interaction". If it does not contain teacher speech, it checks with the same logic if the segment is a "Student(s) Only" or "Silence" segment.
These two last categories are definitive for the segment, but the "Segment containing teacher speech" still has to go through a machine learning model that we will see in the next section.

2.5 Machine Learning Based Algorithm

We have seen that the execution of the *Threshold Based Algorithm* is good but not enough to do a proper classification. It is still necessary to make a distinction between "Teacher Only" and "Interaction" when the last algorithm has chosen the "Segment containing teacher speech" option.

2.5.1 PCA

Before doing any machine learning (ML) training, it is necessary to perform some data pre-processing. According to the paper [4] mentioned before, a good solution for this kind of classification problem is to apply the *Principal Component Analysis* (PCA) to the MFCC features. The suggested values are n=26 for the MFCC features and n=18 for the number of features after PCA.
The idea behind it is to capture enough details in the MFCCs extraction, and to reduce them to less but more relevant features for the ML algorithm. In this project, the suggested values were indeed the best that were found as we will see later in section 3. Figure 8 shows what we get after applying a 2-dimensional PCA on MFCCs of 2 different kinds of segments. Note that the MFCCs are normalized before doing the PCA.
As we can see, there is a quite strong overlap between the 2 categories. "Interaction" segments contain teacher speech, so it can be a possible explanation. Another hypothesis is that we have only 2 principal components in this example (instead of 16 in the project), so it is more difficult to capture the small differences between the 2 categories. This is a good illustration why this classification is challenging.

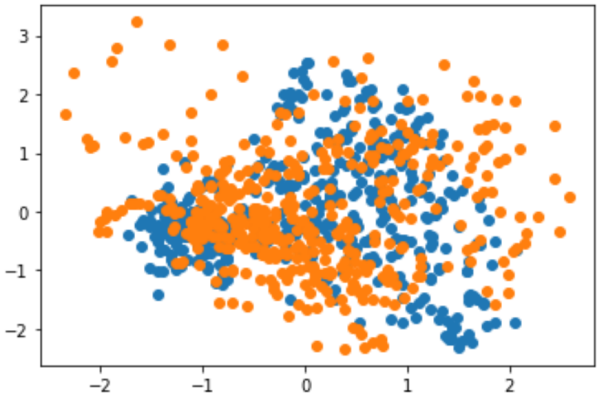


Figure 8: Performing PCA (n=2) with MFCC (n=26) on 2 segments: "Teacher Only" in blue and "Interaction" in orange

2.5.2 Neural Network

Now, let's talk about the ML model, which is the last piece of the final algorithm. As explained above, its purpose is to decide if a "Segment containing teacher speech" is a "Teacher Only" or "Interaction" segment. A different model has been trained for every single audio file (teacher), so every class recording has its own testing and training data. This choice has been made for accuracy and long-term update purposes. These models are saved in the repository [8] and the exact training/testing ratio are detailed in section 3.
The neural network (NN) is a Multi-Layer Perceptron (MLP) model that is an effective technique in speech recognition algorithms [3]. The training has been done using exclusively "Teacher Only" and "Interaction" segments because they are the 2 categories that the model is supposed to classify. I used *grid searching* to find the best parameters for each model (learning rate, layers structure, regularization term). An example of an hidden layer structure that was chosen for an English teacher can be found in figure 9.
The MLP model is commonly used for speech detection/recognition systems. This paper [2] shows that this kind of model works well with MFCCs and references multiple examples where this technique is used.

2.5.3 Other Tested Models

Based on these two papers [6] [1], other models than MLP were also tested in this project: KNN and SVM. The KNN model uses its training set during testing, so no training execution is required. Few nearest neighbors numbers were tested and k=3 provides the best accuracy. The detailed results can be found in section 3.2.

2.6 Final Segmentation Algorithm

The final structure of the classification algorithm can be found in figure 10.

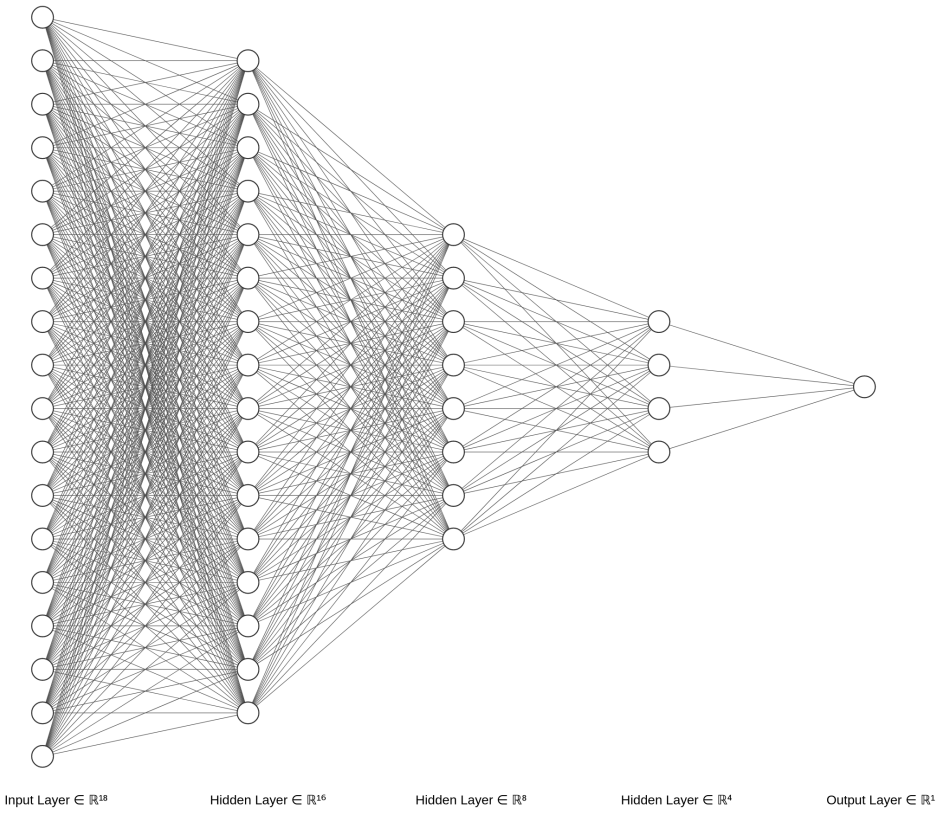


Figure 9: The hidden layers structure of the MLP model

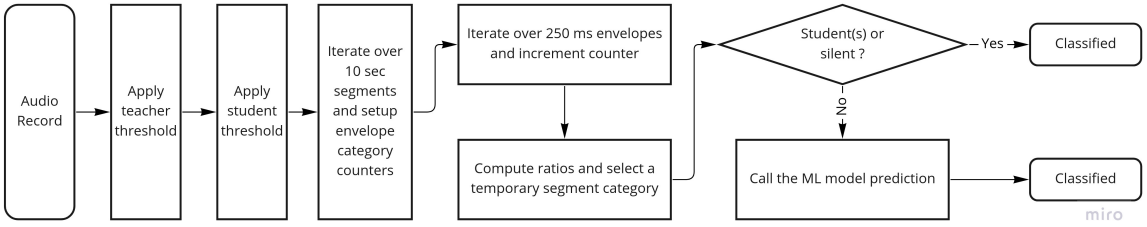


Figure 10: The final structure of the algorithm

3 Results

For the MLP models training, I made the decision to train and test them using always the same number of segments of each category for accuracy purposes. For every file, I tried to find a compromise between having enough training segments of each category (there is always less "Teacher Only" segments than "Interaction" in the audio files) and having a relevant testing set. I always tried to use as many segments as possible for the training. The ratios are shown in table 3. Note that I decided to add the "Teacher Only" training set of E_1.wav to F_1.wav because 90 segments were not enough to have sufficient training.

	E_1.wav	**E_2.wav**	**F_1.wav**
Training Set Interaction (I)	110	110	200
Training Set Teacher Only (T)	110	110	200
Test Set Interaction (I)	30	20	15
Test Set Teacher Only (T)	30	20	15
Training Set Ratio	79%	85%	93%
Test Set Ratio	21%	15%	7%

Table 3: Training/test sets ratios

3.1 Accuracy

Accuracy was the main focus in this project, rather than training time for example. This choice was made because we can assume that the teacher wants feedback that is as close as possible to what happened in class, and not how much time he/she has to wait for the results. We can make a distinction between the accuracy of the ML model (2 categories) and the final algorithm (4 categories). Their accuracy for each file can be found in table 4.

	E_1.wav	**E_2.wav**	**F_1.wav**
MLP model accuracy	85%	90%	87%
Overall accuracy	81%	80%	70%

Table 4: Accuracy of the MLP model and final algorithm

3.2 Performance

The performance is almost similar between the 3 best MLP models that were chosen for the final algorithm, simply because their structure is basically the same. To make a more relevant performance comparison between the models, only one file (E_1.wav) was chosen. Note that many parameters were tested for every model, so only the best ones were selected for the models represented in table 5.
Finally, we can measure the computation time of the final classification algorithm for every audio file. These results can be found in table 6.

	n **MFCC**	n **PCA**	**Training time [s]**	**Testing time [s]**	**Accuracy**
SVM	26	18	524	102	77%
KNN	26	18	-	44	68%
MLP	26	21	31.9	0.0356	83%
MLP	26	18	29.3	0.0253	85%
MLP	26	16	62.3	0.0316	83%
MLP	26	10	55.8	0.0403	82%

Table 5: Performance and accuracy of different tested models

	E_1.wav	**E_2.wav**	**F_1.wav**
Overall computation time [s]	260	206	257

Table 6: Overall execution time for each file

4 Reflection

4.1 Limits and Models Comparison

As we can see in table 5, we have quite significant accuracy (and execution time) differences between the models. KNN and SVM appear not to be the best choices for this classification problem. According to this paper [7], these 2 models are supposed to perform a decent classification in speech recognition systems, but not in all cases. To really improve their accuracy, I should have tried to pre-process the data in different manners. In other words, to reduce the number of dimensions in an optimized way for these 2 models. It is also possible that KNN could have maybe done better with a different distance measure technique, which can leads to important accuracy differences as explained in the same paper [7].

On the other hand, MLP model has good accuracy (for speech detection/recognition problems) and a really fast testing time. The difference between MLP models depending on the number of PCA features is not very significant as we can see in table 5. The MLP model seems to work well with the MFCC/PCA method and appears to be better and more performant than any other model. My hypothesis is that the high number of features and the quite small and embedded nuances between the different segments can be detected more easily with 3 hidden layers NN as we have here. Note that best accuracy that we have with a $n = 18$ PCA also corresponds to the best number of features found in this paper [4].
A major challenge in this project was the small number of training samples, especially for the "Teacher Only" and "Student(s) Only" ones. They have on average respectively 3 and 10 times less available segments than the "Interaction" category, and even up to 20 times less for one file. It is hard to train a model with quite small and unbalanced sets. The same problem appears for the thresholds calibration. For example, one record contains 13 "Student(s) Only" segments. These segments are generally quite heterogeneous (open question answer, group discussion, individual question, ...), so a precise calibration is not something that is easy to do.
During my audio analysis, I have encountered a few perturbations: saturation, clothes touching the microphone, phone ringing near it, etc. This situation causes a significant number of "Silence" misclassifications because the algorithm relies on a low threshold.

4.2 Scalability and Possible Improvements

In this project, we have seen that by construction, the algorithm is trained and calibrated individually for each teacher. At least one file has to be labeled manually, and the algorithm should provide good results. In my opinion, with a little bit more labeled lessons and most importantly more varied ones, the accuracy of the algorithm should increase. It means that if the number of teachers that are using it is quite small it can be used and eventually trained as it is now. On the other hand, if the number of teachers that are using it becomes significant, it will be necessary to adapt and train the model to be teacher-independent (at least by language and/or gender). It should be doable because the training set will also increase a lot.
Another improvement that could be done is to improve the quality of the recording, because it was a limitation as explained in the previous section. An idea to solve this problem could be to use a high-range microphone in the front of the classroom and add table microphones near the different groups in the case of an interactive activity. It was sometimes hard to understand the voice of the student voices (some of them speak very quietly), so this kind of setup should solve this problem.
This kind of setup implies that the threshold system has to be abandoned. But if we assume that we have enough training recordings, a NN could be enough (and more efficient) to classify all the segments since the students will be understood more clearly. It also offers a lot of new possibilities for the algorithm.

5 Conclusion and Future Work

As a conclusion, I think that this kind of reflective dashboard can be a very useful tool for the teacher, especially in lessons that are supposed to be interactive. During teaching, it can be hard for the teacher to remember if the class was receptive to a specific question and when a student said something interesting. It can also be useful to do comparisons between lessons and eventually improvements.
Even though the results are good for this kind of classification, I think that it can still be improved with techniques that were mentioned before. The algorithm was quite challenging to create due to limited training sets, but gives a solid basis for further improvements and extended training. The algorithm was also designed to be as generic as possible, in the sense that it is possible to change the time window and other values easily.
It is also important to take the data labelling into consideration if this algorithm is used on a larger scale. Depending on the complexity of the record, it can take twice the time of the file to label it. It is important to stick to the rules that have been defined to have a consistent labelling across the files. Using a dedicated software could help to reduce the labelling time.
In future projects, an idea could be to implement real-time feedback for the teacher. We have seen that the category prediction is done quickly. So if the segments have an acceptable size, the prediction could be done almost instantly. Exploring more advanced models could be something that can lead to better accuracy. I was thinking about a convolutional neural network (CNN) that uses MFCCs as images and not feature vectors for example. Modifying the algorithm to have a teacher-independent prediction could also be something very interesting and useful to do for the teaching community.

References

[1] Yusuf Yaslan and Zehra Cataltepe. "Audio Music Genre Classification Using Different Classifiers and Feature Selection Methods". In: vol. 2. Jan. 2006, pp. 573–576. DOI: 10.1109/ICPR.2006.282.

[2] Wouter Gevaert, Georgi Tsenov, and Valeri Mladenov. "Neural networks used for speech recognition". In: *Journal of Automatic control* 20.1 (2010), pp. 1–7.

[3] Pialy Barua et al. "Neural network based recognition of speech using MFCC features". In: *2014 International Conference on Informatics, Electronics and Vision (ICIEV)*. 2014, pp. 1–6. DOI: 10.1109/ICIEV.2014.6850680.

[4] Hoang Trang, Loc Tran, and Huynh Nam. "Proposed combination of PCA and MFCC feature extraction in speech recognition system". In: 2015 (Feb. 2015), pp. 697–702. DOI: 10.1109/ATC.2014.7043477.

[5] Nathaniel Blanchard et al. "Identifying Teacher Questions Using Automatic Speech Recognition in Classrooms". In: Jan. 2016, pp. 191–201. DOI: 10.18653/v1/W16-3623.

[6] Hadhami Aouani and Yassine Ben Ayed. "Emotion recognition in speech using MFCC with SVM, DSVM and auto-encoder". In: *2018 4th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*. 2018, pp. 1–5. DOI: 10.1109/ATSIP.2018.8364518.

[7] Bhagyalaxmi Jena, Anita Mohanty, and Subrat Mohanty. "Gender Recognition of Speech Signal using KNN and SVM". In: *SSRN Electronic Journal* (Jan. 2021). DOI: 10.2139/ssrn.3769786.

[8] Léo Claude Hauser. *Automated Class Discourse*. 2022. URL: https://c4science.ch/diffusion/12312/repository/master/.