(a) UNIVERSITY OF LORRAINE

(b) IDMC

(c) LORIA

MSc Natural Language Processing - 2019/2020
UE 705 - Supervised Project

Acoustic scene classification for speaker diarization

Students :
Tahani FENNIR
Fatima HABIB
Cécile MACAIRE

Supervisors:
Md Sahidullah
Romain Serizel

May 2020

Abstract

Speaker diarization is the task of labelling segments of a long audio recording according to the speaker information. This work investigates the speaker diarization in the presence of different acoustic environments. In this report, we have used DIHARD II dataset to analyze speaker diarization performance for eleven different acoustic environments or scenes. We first introduced the basic concepts of speaker diarization and we reported the preliminary results in terms of *diarization error rate* (DER), *Jaccard error rate* (JER). Our preliminary results indicate that the speaker diarization performance can be substantially improved for several acoustic classes given that the acoustic condition is known apriori. We investigate two methods for categorizing the acoustic class. We explore supervised K-nearest neighbors and unsupersived K-means clustering on the DIHARD II challenge dataset. The results showed that applying a new classification to the data resulted in better diarization performance, with the best obtained with the K-mean method.

Contents

List of Figures

Chapter 1

Introduction

This work is focusing on speaker diarization. In order to fully understand what is it and what are the challenges, we first give some introductory elements about the speech basics, how the speech is collected and stored, and analyzed. Also, an overview of speech processing applications and different features and machine learning methods. Then, we explain, thanks to the DIHARD II [1] challenge and manipulations on specific data (calculations of the Diarization Error Rate (DER) and the Jaccard Error rate (JER), classification (t-SNE plots)), the problem statement regarding this speech recognition domain. Finally, we conclude our first step of work and we present our future steps to improve the results we got.

1.1 Speech basics

The human has the ability to produce hundred of sounds but not all of them are speech sounds, only the ones that we use in the spoken language are considered as such. Since sound is a wave, we can relate the properties of sound to the properties of a wave. We can therefore represent sound by a waveform, the vertical axes are the **Amplitude** which is the amplitude of sound pressure variations, measured from 0 [4]. The basic properties of sound are: **pitch**, **loudness** and **tone**.

- **Pitch**: is a perceptual property of sounds that allows their ordering on a frequency-related scale extending from low to high [5].The higher the pitch the higher the sound frequency.
- **Loudness**: The amplitude of a sound wave determines its loudness or volume. A larger amplitude means a louder sound, and a smaller amplitude means a softer sound [6].
- **Tone**: Tone, in acoustics, is a sound that can be recognized by its regularity of vibration. A simple tone has only one frequency, although its intensity may vary. A complex tone consists of two or more simple tones, called overtones. The tone of lowest frequency is called the fundamental; the others, overtones [7].

The figure 1.1 bellow captures the waveform of the English word "cold" with the software Praat [8]. The total duration of the waveform is of 0.19 seconds and the segmentation of the sound into phonetic transcription is visible: [kōld]. We can see here that the amplitude depends on the intensity of the sound. The [ō] of the word "cold" has a large amplitude, which means that it is a loud sound.

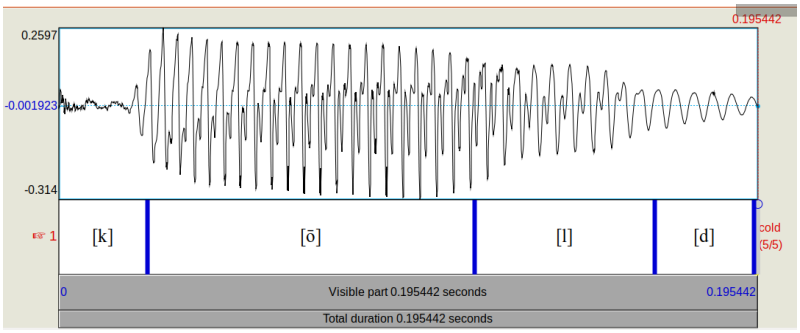

Figure 1.1: The waveform of the English word "cold" with the software Praat, and its phonetic segmentation. The amplitude of the waveform is from -0.31 to 0.25 and shows a high amplitude for the sound [ō].

1.2 Speech production & perception

Speech production is the process of producing speech sounds and occurs when air passes through the vocal system [9]. The sounds vary according to the position of the vocal parts during the air flow. We call these parts the **articulators** (see figure 1.2), and their study is called the **articular phonetics**. From figure 1.2, among other things, we observe the pharynx, a tube allowing swallowing and intervening in the respiratory and digestive system, the velum, depending on its position allows air to pass through the nose and mouth, or the tongue. On the other hand, **voice perception** [10] refers to how sounds are heard, understood and interpreted.

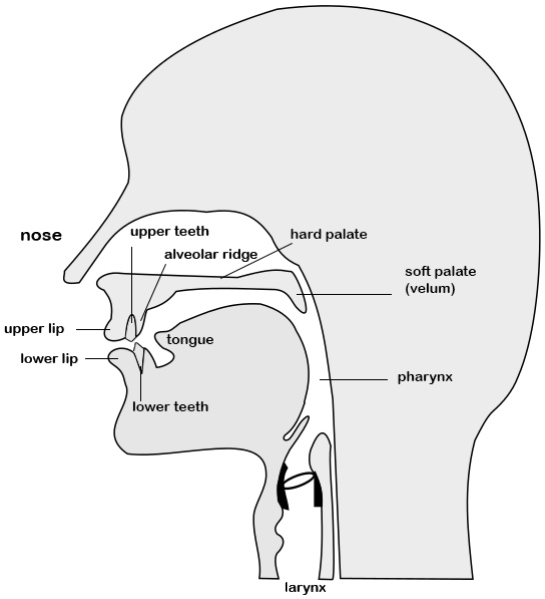

Figure 1.2: Representation of the articulators, with a human head seen from the side [2].

1.3 How the speech signal is collected & stored (sampling rate, formant, compression, etc.)

- **Sampling [11]**
  The signals that we use in the real world are analog signals like the voice. In order to process these signals, we need to convert them into digital signals so they can be understood by the computer. This converting process is called sampling.

- **Sampling rate [12]** is the number of samples per second. The rate of an analog signal is taken in order to be converted into digital form.
- **Formants [13]**
  A formant is an acoustic energy concentration in the speech wave around a specific frequency. There are several formants in every 1000Hz band, each at a different frequencies. In other words, formants occur at intervals of approximately 1000Hz. Every formant corresponds to a vocal tract resonance.

1.4 Analysis of speech (time-domain & spectrogram visualization)

The analysis of speech is conducted by **spectral analysis** [14]. Because the speech signals are time-varying, the analysis has to be a time-frequency analysis.

The analysis can be done thanks to spectrogram using the Fourier transform (correspond to the distribution study of energy along frequency). Spectrogram visualization shows a three-dimensional image of the evolution of the speech signal with time, frequency and intensity. In the figure 1.3, the energy is shown with black color. The energy and the characteristics of speech signal are parameters that are used to identify vowels or consonants in speech. Vowels are represented by a periodic signal and a significant amount of energy while the consonants are represented by a random signal and less energy.

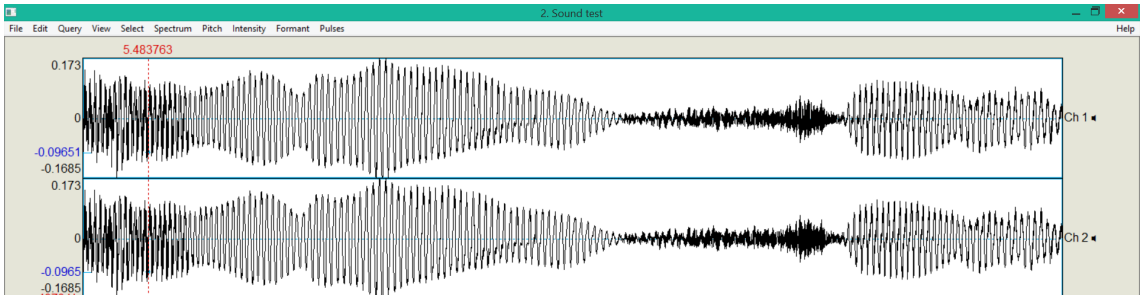

Figure 1.3: Spectrogram which represents the speech signal with Praat. The signal is black when there is a large amount of energy.

But before using the Fourier transform, the signal is splitted into windows. Small windows correspond to a wide range spectrograms and long windows to narrow band spectrograms [15]. Wideband spectrogram is used to observe the formant structure (dark bands which correspond to the peaks in the spectrum) while narrowband spectrogram informs of the harmonic structure. The figure 1.4 shows a wideband and narrowband spectrogram. This representation of a speech shows how the energy is distributed, and therefore, with the formant structure, identify to which vowel or consonant each band belongs. Pitch can be determined by finding inverse of the time duration after which the waveform repeats itself. Pitch is the difference in hertz between the harmonics. It is the fundamental frequency of vibration of the vocal folds, which are present at the top of one's trachea. The pitch signal is shown in blue in figure 1.4.

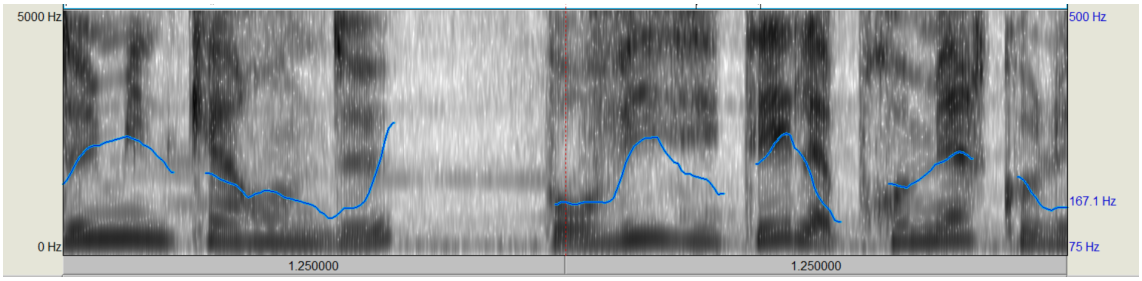

Figure 1.4: Wideband and narrowband spectrogram with Praat.

1.5 Overview of speech processing applications

This section introduces an overview of speech processing applications. These applications are used in a day-to-day basis.

These speech processing applications are divided in 4 categories:

- **speech recognition**
- **language recognition**
- **speaker recognition**
- **speaker diarization**

1.5.1 Speech Recognition

The capability of a machine or program to recognize words and phrases in spoken language and convert them to a machine-readable format is known as Speech Recognition [16]. The predominant use has been in the area of recognition and understanding of speech (voice dictation, natural language voice dialogues, voice dialing for smartphones, ...). An example of application is Voice-to-Text Apps. Google Assistant [17] is one of them. In more details, the user exploits his voice in order to look up information and tell to Google Assistant [17] what to do (send messages, write an email, add events to calendar, ...). Another interesting example is real-time translation. The principle is as follows: the user talks to his phone. After setting up two languages, the app will automatically translate the speech into the desired language. But again, this technology is going through several issues. It includes the difficulty of identifying certain words due to variations in pronunciation [18], the lack of languages support and the inability to override background noises [19].

1.5.2 Language recognition

Spoken language recognition refers to the ability to determine which is the language used in a speech sample [20]. This specific area has multiple application domains. The first domain concerns language translation which translated the speech to the target language (iTranslate in iOS [21], Google Translate [22], ...). Another area is in spoken document retrieval which is the process of indexing and then retrieving relevant items from a huge collection of registered speech audios when a user has a specific natural language query [23].

1.5.3 Speaker recognition

In an audio file, identify who is speaking is called speaker recognition. It is possible by extracting specific information about the speaker in speech waves [24]. Speaker recognition is used in several applications. For example, it cab be used as security in voice applications as well as in access control for both logical and physical access. Also, this technology is promising for criminal and investigations purposes which are based on voice samples records. It can also be used in identifying who is speaking at a conference, a meeting or during a dialogue.

1.5.4 Speaker diarization

An important task in audio retrieval and processing is speaker diarization or indexing which is the procedure to automatically divide a conversation involving numerous speakers into homogeneous segments and to group together all segments corresponding to the same speaker [25]. Speaker diarization mainly involves two steps: *speaker segmentation* and *speaker clustering*. We discuss this in more details in Section 1.7.

1.5.5 Other application areas

In addition to the above four areas related to recognition, other applications of speech processing technology are:

- **speech synthesis** which is related to the generation of speech signal for a given text [26]. Its application include interaction with voice assistants, helping disabled person, etc.
- **speech enhancement** which aims to improve the speech quality by reducing background noises [27, 28].

1.6 Overview of different features & machine learning methods for speech processing

Before giving an overview of the different features and machine learning methods for speech processing, let's set some elements of definition.

1.6.1 What is machine learning?

Machine learning is the science of automatic data pattern detection [29]. It is the science of getting computers to learn and act like humans do and improve their learning over time.

1.6.2 How does machine learning work?

Machine learning uses two techniques:

- **supervised learning** (training a model on data to predict the future) based on classification (into specific classes) and regression tasks.
- **unsupervised learning** (find patterns and structures in input data) based on clustering tasks. It will identify clusters in data based on similar characteristics.

In the field of acoustic speech recognition, we can describe the development step by step of different machine learning approaches chronologically:

- **Vector Quantization (VQ)** to classify audios. For example, several sounds are recorded in different places. This corresponds to the vector x to be classified. After determining the vectors that correspond to a typical sound in a specific environment, vector quantization will find the vector closest to the sound to be classified by calculating the distance between them [30].
- **Gaussian Mixture Model** contains gaussians, represented by $\kappa \in \{1, ..., K\}, K$ is the number of clusters from the dataset. Each κ is represented by the mean μ which is a dimensional vector and the covariance matrix, σ which define the width of the cluster. This model is efficient if there is one class in one classifier.
- **Support Vector Machine (SVM) [31]** uses a high-dimension space. In more details, the aim is to find a hyper-plane in N dimension spaces, thanks to binary data to, at the end, classify them. The hyper-plane dimension will be determined by the number of features.

- **Artificial Neural Network (ANN) [32]** is based on our human neuronal system. It consists of a set of neurons splitted into layers (input, hidden and output layers). Each neurons are connected to each other through weighted connections. A neuron value is the multiplication of the value of a connected neuron with a weight (set with the stochastic Gradient-descent algorithm). The weight can be computed with a bias. Thanks to an activation function $f(x)$, the bias value is transformed and attached to the neuron in the adjacent layer.
- **Deep Neural Networks (DNN)** consists of multiple machine learning algorithms in the form of multiple models [33]. Deep learning algorithms are used to enhance performance of computers in order to understand in a better way human capabilities. When the ANN uses 2 or 3 layers, Deep Neural Network can use more than 1000 layers.
- **Convolutional Neural Network (CNN) [34]** is a deep learning approach for images and spectrogram images. The algorithm will attribute weight and bias in distinct features inside the image and will output a model in order to differentiate between images.
- **Long Short-Time Memory (LSTM** is a Recurrent Neural Network (RNN) [35]. This approach takes into account, not only the current input, but also the previous one, hence the name recurrent.

1.6.3 Speech processing features

Machine learning methods extract features (energy, frequency, source, ...) from speech data to identify who is speaking [36]. Some interesting audio features extraction approaches for speaker recognition are:

- **Mel-Frequency Cepstral Coefficients (MFCC)** which extract features to represent the short-term power spectrum of a spectral envelop.
- **Principle Component Analysis (PCA)** which try to create the best data distribution representation by finding the most relevant combination of features.

Chapter 2

Speaker diarization

2.1 Definition & application

Speaker diarization is the operation of labeling a speech signal with labels corresponding to the identity of speakers [3]. It is the job of deciding "who spoke when?" in an audio or video recording involving an estimated number of speakers and an unspecified number of speakers. It has become a key technology for many tasks, including navigation, retrieval, or higher-level audio data inference.

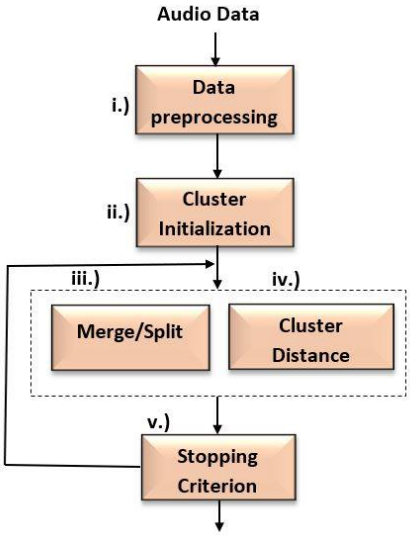

Figure 2.1: General speaker diarization architecture [3]

Fig. 2.1 represents a block diagram of the generic modules that build the most speaker diarization systems. The data preprocessing step (i) tends to be comparatively domain specific, for the data of meeting. Preprocessing typically involves noise reduction (e.g. Wiener filtering), multichannel acoustic beamforming, the parameterization of speech data into acoustic features (such as MFCC, PLP, etc.) and detects speech segments using the speech activity detection algorithm. Cluster initialization (ii) depends on the approach to diarization, i.e. the choice of an initial set of clusters in bottom-up clustering or a single segment in top-down clustering. Next, in Fig. 2.1 iii/iv, a distance between clusters and a split/merging mechanism is used to iteratively merge clusters or to introduce new ones. Optionally, data purification algorithms can be used to make clusters more discriminant. Finally, as illustrated in Fig. 2.1 (v), stopping criteria are used to determine when the optimum number of clusters has been reached.

Some of the applications for speaker diarization are [37]:

- Rich transcription: rich transcription (RT) [38] many metadata adds in spoken file, for example speaker identity and sentence boundaries.

- Automatic speech recognition (ASR) systems: segmentation algorithms could be used to divide audio into small segments for ASR processing systems.

- Audio archiving and monitoring: having archived meetings or conferences, they can be easily reached and monitored by interested persons who were unable to join such meetings.

- Audio indexing and retrieval: the speaker diarization system provides the automatic indexing of spoken audio files, enabling the end user to search the audio document by the identity of the speakers or their number.

- Speaker count: this application includes determining the number of speakers taking part in a conversation (most likely without having any previous information on either of the speakers).

There are three main application [38] domains for speaker diarization, as shown by Reynolds and Torres-Carrasquillo (2004):

- Broadcast news (BN): radio and TV programs with a variety of content, usually containing commercial breaks and music, on a single channel.

- Meetings: meetings or lectures in which multiple individuals communicate in the same room. Normally, recordings are made with a few microphones.

- Conversational telephone speech (CTS): single-channel recording of telephone conversations between two or more individuals.

Speech and speech indexing, document content structuring, speaker recognition (in the presence of multiple or competing speakers), speech-to-text transcription (i.e. speech-to-text-assigned speakers), are obvious examples of applications for speaker diarization algorithms.

2.2 Challenges & problem statement

A principal restriction of most current speaker diarization frameworks is that just one speaker is assigned to each segment. The existence of overlapped speech, though, is popular in multiparty meetings and, consequently, presents a significant challenge to automatic systems. Specifically, in regions in which more than one speaker is active, missed speech errors will be incurred and given the high performance of some state-of the-art systems, this can be a substantial fraction of the overall diarization error. The biggest single challenge is the handling of overlapping speech, which needs to be attributed to multiple speakers.

The fields of application, from broadcast news, to lectures and meetings, vary widely and pose different problems, such as access to multiple microphones, multimedia information, recognize the location and acoustic sign or overlapping speech. The detection and treatment of overlapping speech remains an unresolved problem.

The diarization can also be of poor quality when the speech is recording in a noisy environment. A poor diarization can conduct to misidentification of the speaker, or an absence of speech identification if the noise is higher than the speech of a person. As a matter of fact, the two spectrogram figures 2.2 and 2.3 bellow shows the differences between a noisy speech and a clear speech.

The figure 2.2 shows in a very clear way the formants (in red in the figure) and the intervals where there is speech and where there is not. On the contrary, the second spectrogram (figure 2.3 shows a very unclear spectrogram, as well as the formants, which makes very difficult to tell the difference between a speech and a sound that is not. A solution for this would be to clear in a first step the sound, in order to improve the identification of the speaker.

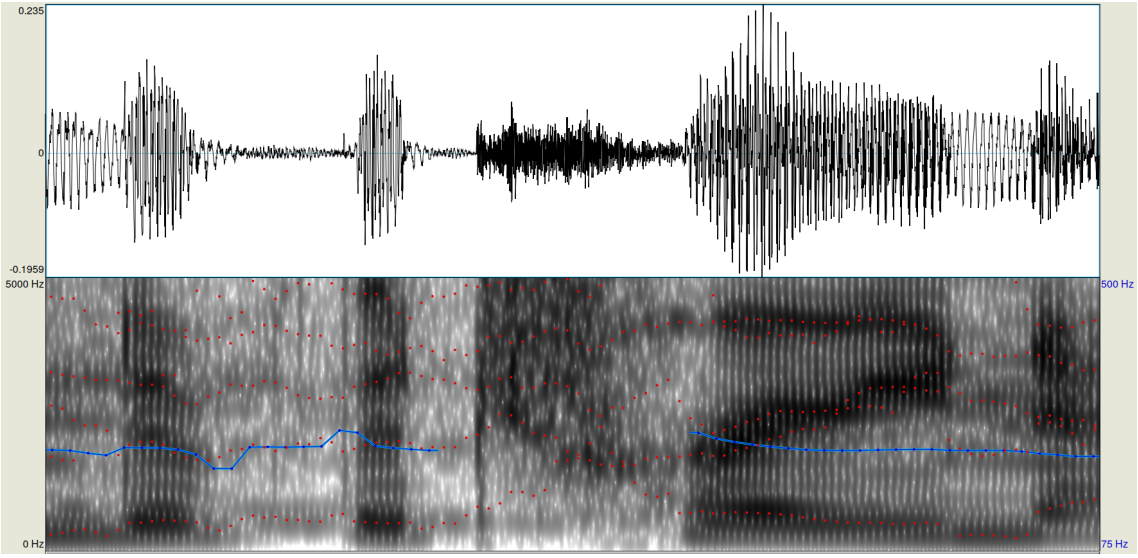

Figure 2.2: Spectrogram and Formants of a speech in a clear environment. The audio file is taken from the DIHARD II challenge in the category audiobook. Audiobook category is made of files with only one speaker and the speech does not include background noises. Praat is the software used to visualize the speech.

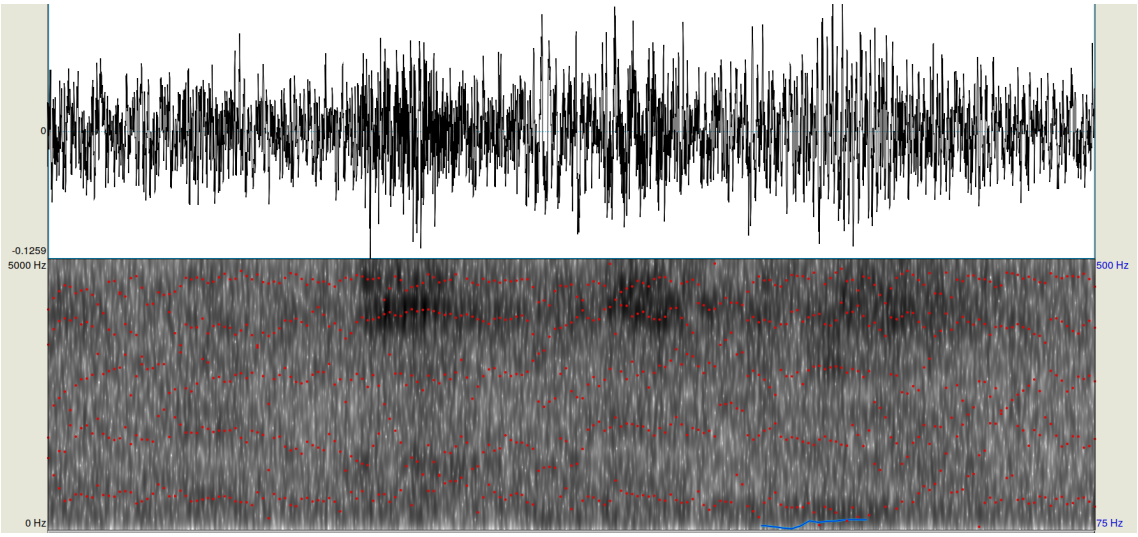

Figure 2.3: Spectrogram and Formants of a speech in a noisy environment. The audio file is taken from the DIHARD II challenge in the category restaurant. The restaurant category is made of files with background noises, and a large amount of different speakers. Praat is the software used to visualize the speech.

Finally, another important issue may be raised concerning the identification of the speaker. Sounds can be identified as speech when they're not. We can take the example of child noises, such as children babbling.

In the period of the spring 2018, the initial DIHARD challenge [39] ran and 20 teams registrations have been attracted of which 13 submitted systems. DIHARD I [39], contains a one channel input condition using wide band speech sample from 11 demanding fields, ranging from clean recordings of read audio books to very noisy high interactive recording of speech. And also offer multichannel input needs participants to perform diarization from farfield microphone. The DIHARD II challenge [1] is the second in a series of speaker diarization challenges and was designed to enhance the robustness of diarization systems to variation in recording equipment, noise levels

and conversational environments. Like its previous challenge, it examines the performance of the diarization system under two SAD conditions: diarization from the reference SAD supplied and diarization from scratch.

Showing in further details the problem statement of speaker diarization, we ran the experiment with the Kaldi project, a toolkit dedicated to speech recognition and computed several indicators.

For the next part, we are only going to talk about the DER (Diarization Error Rate) and the JER (Jaccard Error Rate) for each 11 categories which constitute the DIHARD II challenge dataset [1]. The 11 categories are as follow:

- audiobooks,
- broadcast interviews,
- child,
- clinical,
- court,
- map tasks,
- meeting,
- restaurant,
- socio-field (everyday conversations),
- socio-lab,
- webvideos.

DER is the overall percentage of the reference speaker time that is not accurately attributed to the speaker. The accurately attributed reference speaker time is described in terms of optimal mapping between the reference speakers and the system speakers. More specifically, the DER is equal to

$$DER = \frac{\text{FA + MISS + ERROR}}{\text{TOTAL}}$$

and each element refers to:

- TOTAL is the complete reference speaker time, therefore, the count of the periods of all the reference speaker segments.
- FA is the total time of the device speaker not related to the reference speaker.
- MISS is the overall reference time of the speaker not related to the device speaker.
- ERROR is the full reference time of the speaker assigned to the incorrect speaker. The higher the percentage, the more incorrect the diarization is.

The JER corresponds to the Jaccard Error Rate, a metric based on Jaccard Index and specially computed for the DIHARD II challenge [1]. The Jaccard Index is used to calculate the accuracy of a segmentation thanks to the ratio between two segmentations. The Jaccard Error rate is determined by the best mapping score between reference and system speakers and for each, the Jaccard index is calculated. The JER will finally result of a percentage which equals to 1 - average scores.

$$JER = \frac{\text{FA + MISS}}{\text{TOTAL}}$$

- TOTAL is the length of the combination of the reference and the system speaker segments; if the reference speaker was not combined with the system speaker, TOTAL is the duration of all the reference speaker segments.
- FA is the overall system speaker time not assigned to the reference speaker; if the reference speaker is not matched with the system speaker, FA is equal to 0.

- MISS is the overall reference speaker period not assigned to the system speaker; if the reference speaker has not been combined with the system speaker, MISS is equivalent to TOTAL.

The Kaldi tool computes a threshold for which the DER and JER scores will be the most efficient. We will show in section 2.2.1 and in section 2.2.2 the DER and JER scores obtained in 2 different configurations:

1. the experiment is run on the entire dataset. We calculate the JER and DER average by selecting the files that make up only a specific category. The threshold is the same for each category.

2. 11 different experiments are run. We first select the files by categories, then we start the experiment. Kaldi will inform us of the DER and JER scores for each category, and thus calculate the best threshold to get the best scores.

2.2.1 Results of the development dataset

The development dataset consists of 192 files. The files are splitted into different categories, described in the table 2.1.

Categories	**Average DER**	**Average JER**	**Number of files**
audiobooks	2.8	2.82	12
broadcast interviews	5.68	41.08	12
child	31.79	61.74	23
clinical	20.83	36.23	24
court	15.19	56.37	12
maptask	6.59	11.85	23
meeting	34.56	61.09	14
restaurant	49.77	79.03	12
socio field	14.84	40.19	12
socio lab	10.94	16.32	16
webvideo	37.85	62.15	32
TOTAL	**23.16**	**55.75**	**192**

Table 2.1: Table of average DER and JER for each category for global threshold (-0.3). Kaldi tool was run on the entire development dataset and the best threshold was computed. We then selected the files by category and the DER and JER scores of each files. Finally, the average DER and JER were calculated and described. The number of files per category is also filled in.

Table 2.1 shows, for each category, the average values of DER and JER, obtained thanks to the DER and JER values of all the files that make up this category. The threshold to get the best scores and computed by Kaldi tool is - 0.3. We will call it the global threshold.

For (all) the categories we notice that the best result is in the audiobooks category and the worst one is in the category restaurant.

Thanks to these results, we can also group some categories which have similar values of DER.

Audiobooks, broadcast interviews and maptasks can be group together with a DER value between 2.8 and 6.59. When listening to the specific audio files, the sound clarity is high (no background noises) and it is easy to understand and identify correctly who is speaking and when. Indeed, audio books files contain only one speaker and the record was made in a very clear environment. For the categories broadcast interviews and maptasks, the record was also made in the same environment but there were several speakers.

We can also group court, socio field and socio lab and clinical categories together. The DER values are comprised between 10.94 and 20.83. Even though the speech is easily understandable,

some differences are notable. For instance, there is small ambient noise which includes resonance noises, classroom noises, coughing noises, ... Also, there are several speakers and they sometimes cut each other off or speak at the same time. Specially for the socio lab and socio field categories, differences in speech intensity can be observed. This results in higher DER values compared to the first group previously identified.

The final group is made of child, meeting and restaurant categories with DER scores between 31.79 to 49.77. The quality of audio is poor and the background noises are high. It is sometimes difficult for the listener to understand what the speakers say. Specially for child audio files, in which we hear children chirping, but also noises coming from the water.

The web video category has a DER value of 37.85. It could be attached to the final group but when we listen to the audio files, we notice that these audios are really different between them. Indeed, some are recorded in a restaurant, or in an outdoor environment but some of them contain speech recorded in a very clear environment (really similar to audio books files).

The Jaccard Error Rate (JER) scores are highly correlated with the Diarization Error Rate (DER) scores. We see small differences between specific categories such as audiobooks, maptask, or socio lab. But others have a higher value. For example, broadcast interviews category JER is 35 points higher than the DER score. The difference is explained when the audio files contain dominant speaker(s).

Table 2.2 shows the DER and JER scores in the second configuration. 11 experiments are run, for each category, and the threshold is set on a scale between -2.0 to 0.5. Kaldi tool will compute the best threshold for each category, and not for the entire development dataset.

Categories	**DER**	**JER**	**Threshold**
audiobooks	0.00	0.00	- 2.0
broadcast interviews	7.07	47.47	- 0.8
child	36.10	67.05	- 0.4
clinical	17.80	28.76	- 0.3
court	13.23	47,54	0.0
maptask	8.58	15.20	- 0.4
meeting	33.11	60.53	- 0.4
restaurant	52.00	77.82	- 0.3
socio field	24.48	55.66	- 0.5
socio lab	11.26	18.75	- 0.6
webvideo	39.57	77.74	- 0.5
TOTAL	**22.10**	**45.13**	-

Table 2.2: Table with the DER and JER scores for each category with the best threshold computed by Kaldi tool for each. The number of files is the same.

DER scores of the restaurant or child categories in table 2.2 are high compared to the audio books, which is very logic sense because the audio books are recorded in a very clear environment. Other categories have good DER results such as broadcast interview and maptasks (results bellow 10).

We saw that the results when setting the same threshold for each category (described in table 2.1 are similar but are less relevant. For example, a noticeable difference is observed for the category socio field (DER score in table 2.1 is 14.84 and DER score in table 2.2 is 24.48).

Finally, the TOTAL calculated by taking the average of DER and JER scores is lower in the table 2.2 than in table 2.1. Running the experiment for groups of files with similar characteristics and thus for a specific threshold results in better DER and JER scores.

2.2.2 Results of the evaluation dataset

The evaluation data of the DIHARD II Challenge [1] are composed, as for the development dataset, of audio files recorded during different situations (maptask, monologues, broadcast interviews, ...). For almost all domains, the same sources are used but the files can come from different sources in between, especially for the meeting (domain draws from ROAR instead of RT04) and sociolinguistic fields (domain draws from DASS instead of SLX) domains. Some of the values obtained for DER and JER in the tables described below differ significantly from the values obtained when the development data were used.

This section is made of two tables which contain the Diarization Error Rate (DER) and Jaccard Error Rate (JER) scores of the evaluation dataset under the two configurations described in the previous section.

Categories	**Average DER**	**Average JER**
audiobooks	1.84	1.84
broadcast interviews	6.49	33.39
child	33.83	68.37
clinical	21.58	35.38
court	20.45	64.64
maptask	5.29	11.35
meeting	42.44	61.22
restaurant	55.37	80.37
socio field	16.13	36
socio lab	9.74	19.06
webvideo	41.25	73.02
TOTAL	**25.71**	**58.96**

Table 2.3: Table of average DER and JER of each category for global threshold (- 0.3).

Table 2.3 has higher average DER and JER scores in general compared to table 2.1. The scores show a accurate performance of diarization of 75%, and a JER score of 58 %. The first configuration of experiment, which is to take the entire files and compute a global threshold, used for all categories is clearly under-performing.

Categories	**DER**	**JER**	**Threshold**
audiobooks	0.01	0.01	- 2.0
broadcast interviews	21.94	71.55	- 0.8
child	37.99	76.96	- 0.4
clinical	23.52	38.32	- 0.3
court	13.27	47.50	0.00
maptask	9.05	18.94	- 0.4
meeting	47.72	68.35	- 0.4
restaurant	51.33	77.67	- 0.3
socio field	22.65	53.46	- 0.5
socio lab	12.52	27.07	- 0.6
webvideo	45.05	83.05	- 0.5
TOTAL	**25.91**	**51.17**	-

Table 2.4: Table which contains DER and JER of each category from the evaluation set (specific threshold).

Table 2.4 shows closed DER and JER scores to 2.2) for categories audiobooks, child, maptask, restaurant, socio lab and webvideo. Some categories, for example, broadcast interviews, have higher DER and JER scores. In a global view, the total DER and JER scores are higher than in table 2.2.

To improve diarization performance, a new classification may be considered. Indeed, by classifying some files differently, such as grouping files with similarities, we could improve the scores previously described.

To better understand this problem, we visualize the data using a specific algorithm, t-SNE.

2.3 t-SNE visualization

t-Distributed Stochastic Neighbor Embedding (t-SNE) is a non-linear dimensionality reduction algorithm used for exploring high-dimensional datasets [40]. It maps multi-dimensional data to two or more dimensions suitable for human observation.

To represent the data from the DIHARD II challenge [1], we performed two t-SNE plots. These two visualizations uses two types of data : x-vectors and i-vectors. X-vectors and i-vectors are extracted thanks to a deep neural network and represent, in a compact form, speech utterances. In more details, deep neural network [41] maps sequences of features of each speech to fixed-dimensional embeddings.

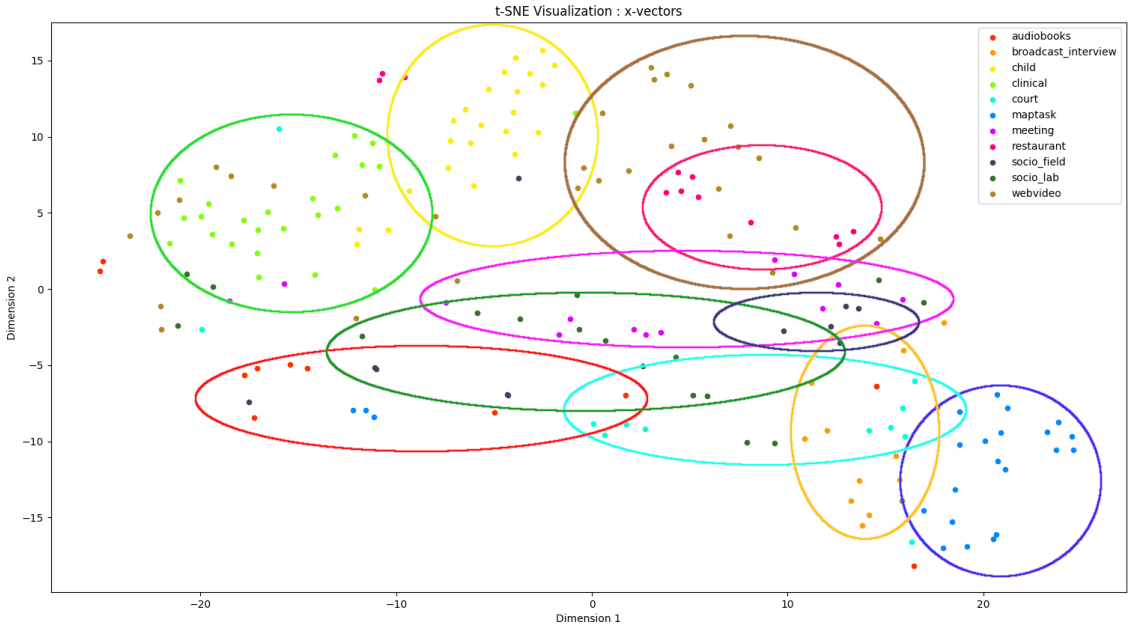

Figure 2.4: t-SNE visualization of audio recordings of DIHARD II dataset [1] with x-vectors. The different categories have a specific color and each point is assigned to a category. Circles are an approximation of the clusters (one per category). Some clusters overlap or contains point which are not originally classified into the category. Two dimensions were used to reduce the x-vector of each point into two coordinates x and y.

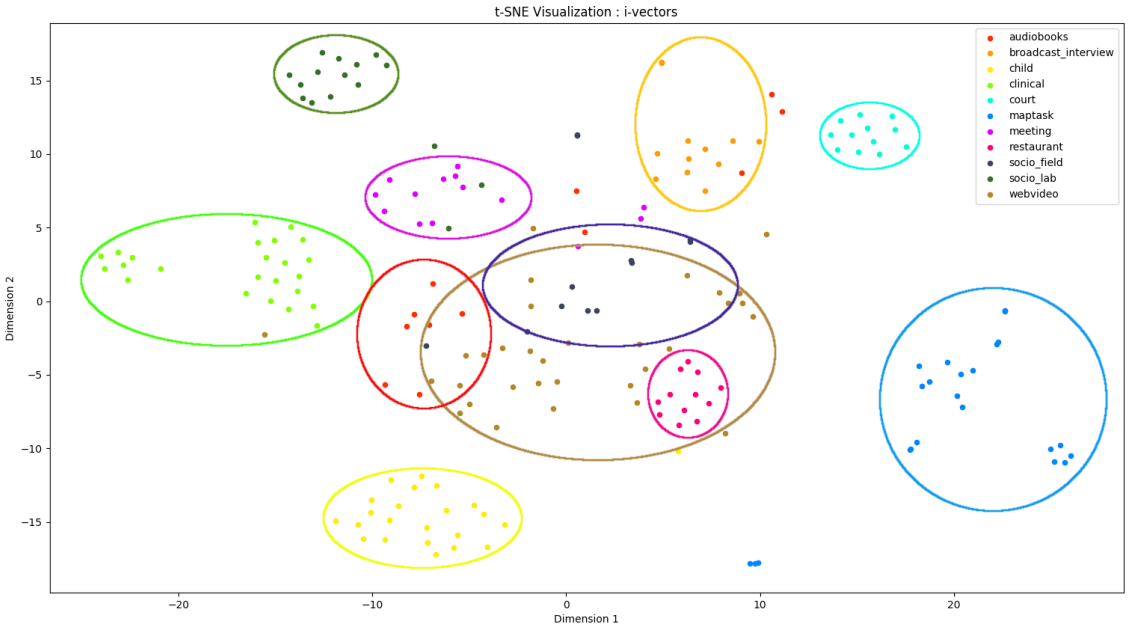

Figure 2.5: t-SNE visualization of audio recordings of DIHARD II dataset [1] with i-vectors. Circles are an approximation of the clusters (one per category). Clusters are relatively well defined because they contained only files classified into the category but we still notice overlapped clusters. t-SNE use a two dimensions scale with coordinate x in abscissa and coordinate y in ordinate.

Figure 3.2 and figure 2.5 displays a t-SNE visualization of the DIHARD II dataset [1] in two dimensions. Between the two figures, figure 2.5 which uses i-vectors creates a better visualization the clusters.

In figure 2.5, the files from the categories clinical, child, maptask and court are well clustered. The others have their data very scattered, such as the categories web video or audiobooks and their data are mixed between them. The webvideo data are not well represented in contract with child category. It means that some files initially labeled in a category present dimension similarities to files which belongs to another category. For example, a web video can contain speech of a child, therefore this file is not well categorized.

To face this issue, we decided to search and then implement relevant classification methods, from two different approaches (supervised and unsupervised methods), in order to get, at the end, better results in the task of categorizing audio files, and therefore to get higher diarization performances.

Chapter 3

Acoustic scene classification

The purpose of acoustic scene classification is to label a test recording into one of the given predefined classes that characterizes the environment in which it was recorded.

Acoustic scene classification is an important task in improving the reliability of speech application, and in our case, speaker diarization [42]. This is a challenging problem due to the variability in different classes. A great deal of research is therefore needed to identify sound scenes in a precise manner and classify them, especially when these scenes contain overlapping sounds or sounds whose quality is diminished due to the environment. Over the years, various approaches are proposed for categorizing the acoustic scenes.

In this study, we explore two approaches and apply it for categorizing DIHARD II Challenge dataset. One of the approach is *unsupervised* where we do not use the class labels for audio data classification and the other approach is *supervised* where we do not use the class label.

Since our focus is on speaker diarization, we use the same audio data used for speaker diarization, i.e., the DIHARD II dataset [1], consisting of recordings from different acoustic scenes. Recordings were made at environments for each group of scenes. The available recording information includes: class of acoustic scene (audiobooks, broadcast interviews, child, ...), duration, number of recordings, and sources. We use the x-vectors and i-vectors embedding of the dataset for acoustic scene classification. We compute the embeddings using Kaldi toolkit [43]. We use the VoxCeleb (v2) recipe for this embedding computation task.

3.1 Unsupervised method: Clustering

Unsupervised machine learning algorithms derive patterns from a dataset without reference to known, or labeled, outcomes [44]. Unlike supervised machine learning, unsupervised machine learning methods cannot be directly applied to a regression or a classification problem because the information about the labels are not known, and therefore the algorithm model cannot be trained. Unsupervised machine learning algorithms discover by itself the relations between the data points in the data set.

In this work, we use k-means algorithm which is an unsupervised clustering algorithm. It takes a number of unlabeled points and attempts to group them into a k number of clusters [45]. The k in k-means shows the number of clusters to have in the end (5 clusters if $k = 5$). The method is unsupervised since the points have no class labels (there are not labeled).

k-means principle

To process the data, the k-means algorithm in data collection starts from the first group of randomly picked centroids that are used as starting points for each cluster, and then conducts iterative (repetitive) calculations to optimize the position of the centroids [46].

It stops creating and optimizing clusters when either:

- the centroids have stabilized — their values have not changed because the clustering has been efficient. OR

- the number of iterations defined has been accomplished.

There is no single classification for the same dataset. The difficulty in implementing this method results in the choice of the number of cluster k, which will allow to identify interesting patterns between the data. The choice of k is not automated but methods exist to get required number.

The method that will be used in the following is the so-called *elbow method* [47]. Elbow method calculates the variance of the different clusters as a function of several k-values. The variance corresponds to the sum of the distances between each centroid of a cluster and the different observations within the same cluster. Here we try to identify the k value that minimizes the distance between the centroids of the clusters and the observations within the same cluster.

The variance is computed as follows:

$$V = \sum_{j} \sum_{x_i \to c_j} D(c_j, x_i)^2$$

where c_j is the centroid of a cluster, x_i is the ith observation in the cluster having for centroid c_j and $D(c_j, x_i)$ is the Euclidean distance between the centroid and the point x_i.

The optimal number of clusters is the point representing the elbow, the point of the elbow being the number of clusters from which the variance no longer reduces significantly.

k-means method is easy to implement, and when data have a large number of attributes, it can be computationally faster compared to hierarchical clustering only if k is low [48]. Moreover, the algorithm is able to identify the non-linear structures. The k value (number of clusters) needs to be defined manually when property of the data is known.

3.1.1 Application of the k-means algorithm on the DIHARD II challenge dataset

First, we apply the k-means algorithm on the x-vectors from our development data. The idea here is to group the entire dataset consisting 11 classes into a fewer categories. Our assumption is that different audio classes are similar. Moreover, pre-assigned hard labels do not correspond to the actual quality of the audio-data. For example, the audio files in web-videos have large within class variability. Therefore, grouping them into limited number of classes will help to set the threshold more effectively.

The construction of the k-means model is divided into four steps:

1. Initialize k clusters with k centers.

2. Fit the data to the model. We use the entire development data.

3. Evaluate the model and compute the new centers.

4. Repeat the steps 2 and 3 until the algorithm find the best k value.

We used the x-vectors development data and we clustered them using the k-means algorithm. In order to choose the best K value, we applied the **Elbow method** to determine the optimal value of K to perform the K-Means Clustering Algorithm.

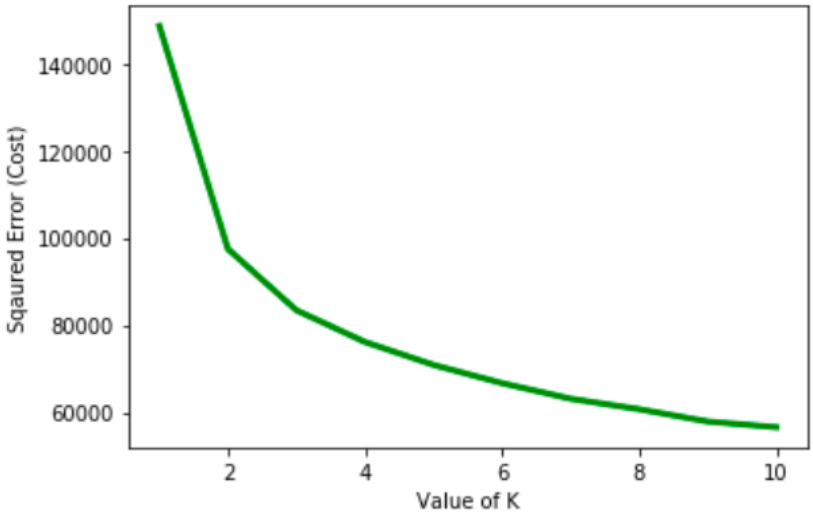

Figure 3.1: The squared error (Cost) for different values of K (from 1 to 10).

From figure 3.1 which displays the squared error for different values of K, the elbow is forming when K is equal to 4. The optimal value is 4 for performing K-Means with the x-vectors development dataset.

Thanks to the identification of the best K number of clusters, we created the K-means model with a K value of 4. The development dataset from x-vectors are clustered.

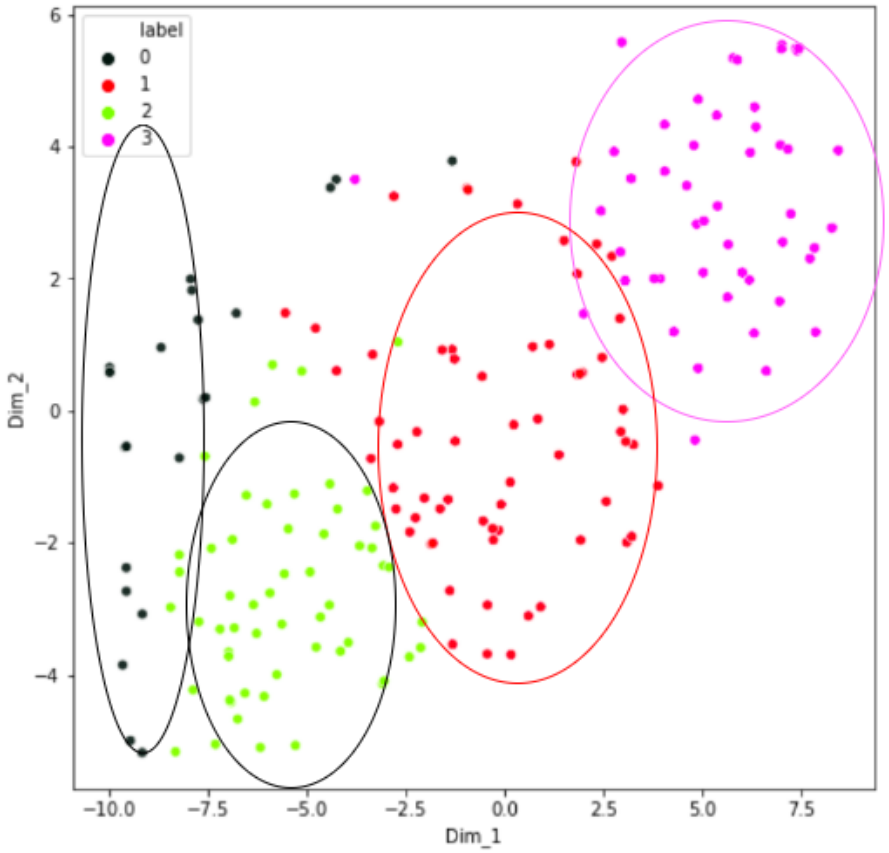

Figure 3.2: t-SNE visualization of data grouped in four clusters defined by the K-means algorithm from x-vectors development dataset. t-SNE method reduces the data into two dimensions. The circles represent the four clusters, labeled between 0 to 3.

Figure 3.2 is the t-SNE visualization of the 4 clusters identified by the K-means algorithm. The data are reduced into 2 dimensions. The distribution of data in each cluster is very clearly identified. Clusters do not overlap.

To understand the distribution of each category in each cluster, we calculated the percentage in table 3.1.

Categories	**Cluster 0**	**Cluster 1**	**Cluster 2**	**Cluster 3**	**Number of files**
audiobooks	66.7 %	8.3 %	0	25 %	12
broadcast interviews	0	0	0	100 %	12
child	0	47.83 %	52.17 %	0	23
clinical	0	4.2 %	95.83 %	0	24
court	33.33 %	0	0	66.66 %	12
maptask	13 %	0	0	87 %	23
meeting	0	57.14 %	28.57 %	14.28 %	14
restaurant	0	75 %	25 %	0	12
socio field	25 %	58.33 %	0	16.67 %	12
socio lab	12.5 %	56.25 %	18.75 %	12.5 %	16
webvideo	6.25 %	53.125 %	34.375 %	12.50 %	32
TOTAL	**22**	**63**	**56**	**51**	**192**

Table 3.1: Table of the percentage of files by category in each cluster defined by the K-means model on x-vectors development dataset. The last line of the table refers to the number of files in each cluster.

66.7 % of audiobook files are clustered in the cluster 0, 8.3 % in cluster 1 and 25 % in cluster 3. Cluster 2 does not contain any audiobook files. Moreover, cluster 0 does not contain files from categories broadcast interviews, child, clinical, meeting and restaurant. Based on these characteristics, we deduce that the files grouped in cluster 0 are audio files with a clean sound, few background noises or overlap sounds and few speakers because of the categories present in this cluster. TOTAL row refers to the number of files in a cluster. Cluster 0 compared to the others have few files inside, which is logic because few audio files have clear sound, few speakers, etc.

Evaluate the model

The model is evaluate with two scores type:

- **Completeness Score** is satisfied if all data points in a class are grouped in the same cluster [49]. This measure is not dependent on absolute label values, i.e. a change in label values will not affect the completeness score.
- **Homogeneity Score** is an indicator to check whether a cluster contains only samples from a single class [50].

Both Completeness and Homogeneity scores are in the range [0,1], the bigger the value the better the model.

K value	**Homogeneity Score**	**Completeness Score**
2	0.12	0.41
3	0.20	0.41
4	0.30	0.53
5	0.35	0.52
6	0.33	0.45
7	0.37	0.46
8	0.42	0.51

Table 3.2: **Homogeneity Score** and **Completeness Score** of K-means models with different k values on x-vectors development dataset.

In table 3.2, when the K value is set to 4, the Homogeneity score is 0.3. The higher the K value, the higher the homogeneity score. However, the best completeness score is obtained when K is equal to 4 (0.53). It means that in 53 % of the case, the points from a given class are grouped in the same cluster. Through these different indicators, our initial hypothesis that a value of K defined at 4 will provide the best classification model is verified.

Applying diarization on the new classification

The K-means algorithm clusters the files in 4 clusters. We applied the diarization on each cluster with the files it contains.

Clusters	**DER**	**JER**	**Threshold**
0	6.33	42.23	- 0.6
1	34.30	63.30	- 0.2
2	31.85	63.13	- 0.5
3	10.53	38.19	- 0.4
TOTAL	**20.75**	**51.63**	-

Table 3.3: DER and JER scores of each K-means clusters on the 192 files (development dataset).

Table 3.3 shows the DER and JER scores obtained for each cluster with the K-means classification on x-vectors development dataset. As an overall view, the average DER (TOTAL row) computed with the DER score of each cluster are better (20.75) than these in table 2.2 (22.10). Applying the K-mean clustering algorithm concludes in better diarization performance than with the initial classification (in 11 categories).

Applying K-means algorithm on i-vectors development dataset

We applied the K-means algorithm on i-vectors dataset. We followed the same steps than in the previous section with x-vectors dataset. The K-means model uses the same k value.

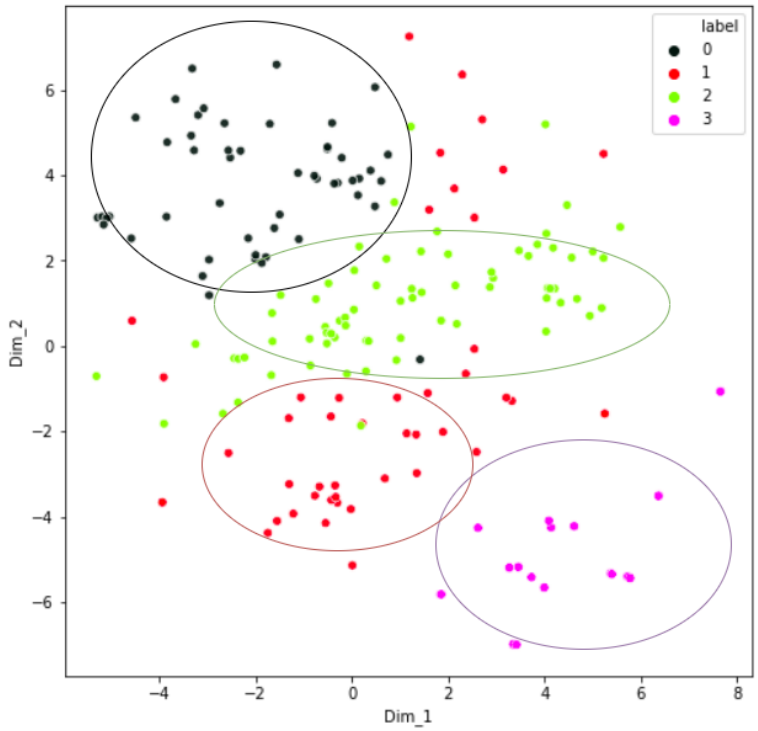

Figure 3.3: t-SNE visualization of data grouped in four clusters defined by the K-means algorithm from i-vectors development dataset.

Figure 3.3 shows the t-SNE visualization of the clusters obtained with the K-means algorithm on i-vectors development dataset. The 4 clusters are well recognizable. However, some data points from cluster 1 could be clustered in cluster 0 or 2 according to their position in the figure. The classification could be improved in this case.

Table 3.4 shows the percentage of distribution of each category in each cluster.

Categories	**Cluster 0**	**Cluster 1**	**Cluster 2**	**Cluster 3**	**Number of files**
audiobooks	8.33 %	91.67 %	0	0	12
broadcast interviews	100 %	0	0	0	12
child	0	0	100 %	0	23
clinical	95.83 %	0	4.17 %	0	24
court	0	100 %	0	0	12
maptask	0	0	0	100 %	23
meeting	71.42 %	0	0	28.57 %	14
restaurant	0	75 %	25 %	0	12
socio field	0	0	0	100 %	12
socio lab	100 %	0	0	0	16
webvideo	3.13 %	6.25 %	90.62 %	0	32
TOTAL	**51**	**46**	**72**	**23**	**192**

Table 3.4: Table of the percentage of files by category in each cluster defined by the K-means model on i-vectors development dataset. The last line of the table refers to the number of files in each cluster.

When comparing table 3.1 with table 3.4, we notice that the K-mean model built from i-vectors dataset gives better results in clustering the audio files. From the table 3.4, 100 % of the broadcast interviews files are clustered in cluster 0, the same for the child audio files which are clustered in cluster1. The same pattern is seen also for maptask, socio field and socio-lab categories with 100 % of their files clustered in only one cluster. The files of other categories are clustered in several clusters but the majority of them are clustered in one. For example, 95.83 % of the clinical files are clustered in cluster 0, 75 % of the restaurant files are clustered in cluster 1, etc.

Evaluate the model

The final step is to evaluate the model. Table 3.5 shows the Homogeneity and Completeness scores for a k-value between 2 to 8.

K value	**Homogeneity Score**	**Completeness Score**
2	0.16	1
3	0.35	0.93
4	0.48	0.86
5	0.45	0.81
6	0.85	0.85
7	0.63	0.86
8	0.65	0.80

Table 3.5: **Homogeneity Score** and **Completeness Score** of k-means model computed with different k values.

When the model is built with a k value of 4, the Homogeneity score is 0.48 and the Completeness score is 0.86. Looking at the values obtained by the other models, these two scores are not the best. Indeed, the higher Completeness score is obtained with a k value of 2, and the best Homogeneity score is seen with a k value of 6. In general, the model with k = 4 is a compromise between the best values of the two indicators.

Applying diarization on the new classification

The K-means algorithm clusters the files in 4 clusters. We applied the diarization on each cluster with the new classification of files.

Clusters	DER	JER	Threshold
0	21.09	41.97	- 0.6
1	9.77	47.88	- 0.4
2	41.13	71.31	- 0.3
3	7.62	14.21	- 0.4
TOTAL	**19.9**	**43.84**	-

Table 3.6: DER and JER scores of each K-means clusters on the 192 files (development dataset).

The TOTAL row in table 3.6 refers to the average DER and JER scores obtained with the DER and JER scores of the 4 clusters. As a global overview, applying the diarization on this new classification concludes in better DER scores compared to table 2.2 with the files classified in 11 categories. Moreover, the DER and JER scores are better than in table 3.3 (19.9 against 20.75 for DER and 43.84 against 51.63 for JER). We conclude in this case that K means clustering algorithm is efficient on i-vectors dataset and get better diarization performance when the files are classified in 4 clusters.

3.2 Supervised method: Classification

A supervised classification method (as contrary to an unsupervised classification method) is one which depends on labeled input data to discover a function that produces a suitable output when new unlabeled data is given [51].

3.2.1 The k-nearest neighbors (KNN)

KNN is an algorithm for clustering. It groups large amount of data into smaller groups together which have similar properties. By the increasing volume of data since few years, KNN algorithm has become a widely clustering technique to meet the global demand for data processing, [45].

K-Nearest Neighbors principle

KNN algorithm is based on supervised learning. Training samples are composed of n-dimensional numerical data. Each data point is therefore represented by one sample in a defined n-dimensional space [52]. The KNN algorithm will calculate the distance between each point and the centroid of each cluster [53]. Each point will be assigned to the cluster for which the distance is the smallest, i.e. the closest. When a new point is affiliated to a cluster, the centroid of each group is recalculated by taking the average.
KNN algorithm uses the k value (the k number of clusters) and the distance metric (Euclidean distance) to calculate the distance between the new points and the nearest neighbors [54].
The Euclidean distance is defined as follow:

$$D_e(x, y) = \sqrt{\sum_{j=1}^{n} (x_i - y_j)^2}$$

where D is the euclidean distance, x and y are the coordinates of the ith point.

How to choose the k value?
The neighbours are defined by a set of data which are already correctly classified [52]. The k value depends on the input data set. Constraints are observable:

- if the number of neighbors is low, the more underfitting will be observed,
- but if the k value is closed to N number of observations, we will be in a overfitting situation.

As a result, the resulting model will generalize poorly on observations it has not yet seen. In order to get the best accuracy for a model, we calculate n times the model at k values and look at the best score obtained with the dataset.

What are the advantages and disadvantages?

KNN classification algorithm is labeled as lazy learner because the algorithm doesn't build a new model unless a sample with unlabeled data has to be classified. Lazy learners can therefore have very high computational costs if there are a large number of neighbours [52].

Each attribute has the same weight. When some attributes are not important to the classification task in the data, the model may be biased.

As summarized by Bhatia, Nitin & al. in the "Survey of Nearest Neighbor Techniques", the table 3.7 describes advantages and disadvantages of KNN algorithm.

Advantages	**Disadvantages**
Fast training	Biased k value
Simple and easy implementation	High complexity
Robustness / Effectiveness in large training data	Computational costs
	Biased model by irrelevant attributes
	Lazy learner (runs slowly)

Table 3.7: Advantages and Disadvantages of KNN algorithm.

3.2.2 Application of the KNN algorithm on the DIHARD II challenge dataset [1]

Applying KNN algorithm on x-vectors dataset

We first applied the KNN algorithm on x-vectors data. In the x-vectors data, there are the development dataset and the evaluation dataset. The data contain, for each audio files, a vector and a label which corresponds to the one of the 11 categories previously described.

The construction of the model is splitted into 3 different steps:

1. Train and test the x-vectors development data on several KNN model - it will determine the model with the best k-value to get the best accuracy score possible for the data.
2. Create the model with the previously set k-value.
3. Evaluate the model.

The first step is to split x-vectors development data into training and test data. We train different KNN model (with different k values) on the training data and then we test the model with the test data. The model will compute the accuracy score of the predicted labels on the test data. The best accuracy score will therefore be the best KNN model. The table 3.8 bellow displays the accuracy scores of KNN models according to the k value. The best model accuracy obtained with KNN algorithm is with a k value of 3. This means that the vote chose by the 3 nearest neighbours to attribute the class of a point is the most accurate in this case.

K value	Accuracy score
1	66.6 %
2	66.6 %
3	**70.8 %**
4	64.5 %
5	66.6 %

Table 3.8: Accuracy score (in percentage) of KNN model with different k values. The KNN model is trained with a sample of 144 data and is tested with a sample of 48 data. The line in bold shows the best accuracy score with a k value of 3.

The figure 3.4 displays the error rate according to the k value. The error rate is computed with the arithmetic mean of the difference between the predicted value of the labels and the real value of labels. We observe that the lowest error rate value is obtained with a k value of 3. This verifies the accuracy score obtained in the table 3.8.

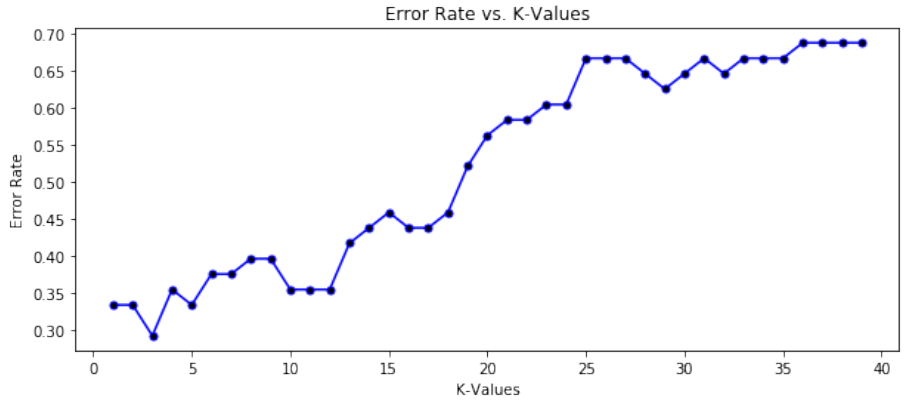

Figure 3.4: Error rate according to the k value. The lowest error rate is with a k value of 3. The higher the k value, the higher the error rate.

After defining the best KNN model, we create it using the training data and a k-value of 3.

Thanks to the construction of the model, we can now predict the class on a dataset. We decide to predict the class on the set of development data from the x-vectors. The accuracy score is 79.6 % in this case. To see how much audio files are well classified, we display the confusion matrix in table 3.9. The confusion matrix referred to as an error matrix, is a specific table layout that we used to visualize the performance of our algorithm. The rows correspond to the initial labels and the columns to the predicted labels.

	1	2	3	4	5	6	7	8	9	10	11
1	11	0	0	0	1	0	0	0	0	0	0
2	0	10	0	0	2	0	0	0	0	0	0
3	0	0	21	2	0	0	0	0	0	0	0
4	0	0	1	23	0	0	0	0	0	0	0
5	0	0	0	0	12	0	0	0	0	0	0
6	0	0	0	0	0	23	0	0	0	0	0
7	0	0	0	2	0	0	6	2	3	1	0
8	0	0	0	0	0	0	0	12	0	0	0
9	0	0	1	0	1	0	0	0	9	1	0
10	0	0	1	2	0	0	5	0	0	8	0
11	0	1	1	5	0	0	1	6	0	0	18

Table 3.9: Confusion matrix computed on the KNN model on the 192 files (development dataset).

The numbers from 1 to 11 correspond to the 11 categories. The files from the categories court, maptask are well classified (the predicted class is the same). The categories audiobook and clinical have 1 audio file which is not classified in the same category. Finally, the categories broadcast interviews, child, meeting, socio field, socio lab, and webvideo have several audio files which are classified in other categories by the KNN classifier.

From the confusion matrix, a table can be generated showing the percentage distribution of one category within another category relative to predictions. For example, in table 3.10, the new class clinical (column number 4) contains 5.8 % of files originally classified as child (row number 3), 67.6 files from the same class, 5.8 % of files originally classified as meeting (row number 7), 5.8 % of files originally classified as socio lab (row number 10) and 14.7 % of files originally classified as webvideo (row number 11). We also see that audiobook, maptask and webvideo categories contain only files initially classified in the same category.

	1	2	3	4	5	6	7	8	9	10	11
1	100	0	0	0	6.25	0	0	0	0	0	0
2	0	90.9	0	0	12.5	0	0	0	0	0	0
3	0	0	84	5.8	0	0	0	0	0	0	0
4	0	0	4	67.6	0	0	0	0	0	0	0
5	0	0	0	0	75	0	0	0	0	0	0
6	0	0	0	0	0	100	0	0	0	0	0
7	0	0	0	5.8	0	0	50	10	25	10	0
8	0	0	0	0	0	0	0	60	0	0	0
9	0	0	4	0	6.25	0	0	0	75	10	0
10	0	0	4	5.8	0	0	41.6	0	0	80	0
11	0	9	4	14.7	0	0	8.3	30	0	0	100

Table 3.10: Percentage distribution of one category within another category relative to predictions.

We will see later, by applying the diarization on the categories with this new classification if we get better DER and JER scores.

The final step consists to evaluate the classifier. We apply the evaluation dataset from x-vectors to the model and compute the accuracy score and the confusion matrix. Evaluation dataset contains 194 files, also classified in the same 11 categories.

The accuracy score with the KNN model is 57.2 %.

Table 3.11 shows the files which are well classified or not, thanks to the confusion matrix.

	1	2	3	4	5	6	7	8	9	10	11
1	3	1	0	1	3	2	0	0	0	0	2
2	0	4	0	0	4	1	1	0	2	0	0
3	0	0	17	2	0	0	0	2	0	0	2
4	0	0	0	24	0	0	0	0	0	0	0
5	0	0	0	0	11	0	0	0	0	1	0
6	0	0	0	0	0	19	0	0	0	0	0
7	0	0	2	3	0	0	2	3	1	0	0
8	0	0	0	0	0	0	0	12	0	0	0
9	0	1	1	6	0	0	8	2	2	0	2
10	0	0	0	0	1	0	6	0	0	4	1
11	0	0	6	4	0	0	1	9	2	0	13

Table 3.11: Confusion matrix computed with the KNN model on the 194 files (evaluation dataset).

The categories clinical, maptask and restaurant have all the files well classified. The others have a misclassification of the files, which is logic because the accuracy score is of 57.2 %.
To understand in a better way why certain files are not classified in the category initially assigned,

we check the recording of the files. The audio files 26 is labelled as a webvideo. From the classification report, this file has a predicted label socio field. When we listen to it, there is only one speaker and the sound is clear despite some background noises. Files from socio field class have also these characteristics. it is therefore logic that this file is classified in another category. Another example is with the file 8, classified as webvideo and with a prediction class child. The sound is barely audible, moreover many people are talking at the same time and background noises are present. The overall quality of the speech is poor, which is close to the files from the child category. On the other hand, for some files, we notice that the classification is not correct, especially for the audiobook category. Some files have another class assigned, which does not correspond. For example an audiobook file is classified in the clinical category, this category including audio files with several speakers, background noises.

Applying diarization with the new classification

KNN model tells us the new class of each file. The final step is to apply the diarization on each class with the new classified files. We compute only the diarization of the development dataset.

Categories	**DER**	**JER**	**Threshold**
audiobooks	0.0	0.0	- 2.0
broadcast interviews	7.14	46.90	- 0.7
child	33.92	64.12	- 0.4
clinical	22.51	42.53	- 0.4
court	12.23	45.60	- 0.2
maptask	8.46	14.12	- 0.4
meeting	28.38	55.65	- 0.5
restaurant	47.00	75.27	- 0.3
socio field	21.23	48.03	- 0.5
socio lab	10.73	33.42	- 1.0
webvideo	47.08	82.56	- 0.7
TOTAL	**21.69**	**46.2**	-

Table 3.12: Table which contains DER and JER of each category from the development dataset (specific threshold) with the new classification.

Comparing the table 3.12 with table 2.2, the DER average obtained by the control classification is slightly higher (22.10) than that obtained by the new classification (21.69). Audiobook category has the same DER and JER scores. The categories child, court, maptask meeting, restaurant, socio field and socio lab have their DER and JER scores lower in table 3.12. Files in these categories provide better DER and JER scores for diarization. Finally, broadcast interviews, clinical and webvideo categories have lower DER and JER scores. The new classification for these categories does not result in better performance for diarization.

Applying KNN algorithm on i-vectors dataset

We also used the same algorithm on the i-vectors dataset. The steps are the same than for the x-vectors dataset.

We first train and test KNN models with different k-values. The development x-vectors dataset are splitted into training and test data. Table 3.13 displays the accuracy score obtained according to different k-value. The KNN model with the highest accuracy score is with a k-value of 1.

K value	Accuracy score
1	**83.3 %**
2	62.5 %
3	60.4 %
4	58.3 %
5	58.3 %

Table 3.13: Accuracy score (in percentage) of KNN model with different k values. The KNN model is trained with a sample of 144 data and is tested with a sample of 48 data.

We then run the second step which consists of creating the KNN model with the k-value previously chosen. The KNN model can now predict the class on a dataset. On the development data as a whole, the predicted labels are almost identical to the initial labels (95 % the same). We decide to predict the class on the set of evaluation data from the x-vectors. The accuracy score is 62.8 % in this case.

	1	2	3	4	5	6	7	8	9	10	11
1	0	0	1	1	3	0	0	3	2	0	2
2	0	0	0	0	1	0	1	9	1	0	0
3	0	0	18	0	0	0	0	5	0	0	0
4	0	0	0	24	0	0	0	0	0	0	0
5	0	0	0	0	12	0	0	0	0	0	0
6	0	0	0	0	0	19	0	0	0	0	0
7	0	0	0	0	0	0	11	0	0	0	0
8	0	0	0	0	0	0	0	12	0	0	0
9	0	0	1	2	0	0	5	8	6	0	0
10	0	0	0	0	0	0	2	0	0	9	1
11	0	0	5	1	0	0	0	13	5	0	11

Table 3.14: Confusion matrix computed on the KNN model on the 194 files (evaluation dataset).

Table 3.14 displays the confusion matrix computed by the KNN model with a k-value of 1 on the evaluation i-vectors dataset. It is interesting to see the differences between this confusion matrix and the confusion matrix in table 3.11. Indeed, when we decide to build a KNN model from x-vectors or i-vectors, we do not first get the same k-value and the predicted classes on an identical dataset are not the same. For example, in table 3.14, the files from category broadcast interviews are not classified in this category in opposite in table 3.11 where 4 of them are classified in the same category.

Applying diarization on this dataset is not interesting here because when the model is used on the set of development data from i-vectors, the predicted labels are almost identical to the initial labels. In this case, the DER and JER diarization scores will be similar.

This KNN model does not allow a better diarization when working on i-vectors dataset.

3.3 Conclusion of the two methods applied on the DIHARD II challenge dataset [1]

The diarization performance results obtained on the DIHARD II challenge dataset are better than those obtained from the initial 11-category classification of these audio files. In more detail, the unsupervised algorithm, the K-mean, obtains slightly better performance compared to the supervised algorithm, the KNN (about 1 point difference for the DER score). There is also a slight difference in the DER and JER scores depending on the data source. Indeed, the scores are better when i-vector data are used. We recall here that i-vectors and x-vectors are extracted from audio files using a deep neural network process.

Chapter 4

Conclusion & Future Work

The aim of the project is to improve the performance of the identification of a speaker in challenging acoustic scenes as restaurant or meetings by defining the acoustic conditions of the record environment towards improving speaker diarization performance.

In experiments with the DIHARD II challenge [1] dataset which contains audio files which are categorized in 11 classes (from very clear environment to very noisy one), we identified several issues. First of all, the diarization performance is different for different categories of audio files when computed with a global threshold. In experiments with category-specific speaker diarization where the thresholds are separately optimized, we have found that the optimized thresholds are not necessarily same over all the categories. This motivates us to further optimize the speaker recognition component.

In t-SNE visualization of the speech embeddings, we have observed that i-vector based embeddings are better than x-vector based embeddings. We also notice that some of the audio categories are nicely clustered and some of those clusters are somewhat closer. This indicates that some of the categories can be grouped together. On the other hand, the audio recordings of highly scattered audio categories (i.e., web videos) can be assigned to a different category indicated by the cluster closest to that file in terms of embedding similarity.

We have done the classification on i-vectors and x-vectors using the supervised machine learning algorithm K-NN and the unsupervised approach K-means. We notice that the diarization performance computed on the new classification by the two algorithms are better when applying on the 11 acoustic scenes classification. In addition, classification from the i-vector dataset gives better diarization performance compared to classification from the x-vector dataset.

The future prospects of this project are:

- We were limited to the dataset that we worked on. An interesting things would be to record new data in different recording environments to see what results we can get, and how we can improve them.
- Use neural network algorithms in the classification of the data set files.

Bibliography

[1] Neville Ryant, Kenneth Church, Christopher Cieri, Alejandrina Cristia, Jun Du, Sriram Ganapathy, and Mark Liberman. Second dihard challenge evaluation plan. *Linguistic Data Consortium, Tech. Rep*, 2019.

[2] The Production of Speech Sounds. `http://www.personal.rdg.ac.uk/~llsroach/phon2/artic-basics.htm`, 2019.

[3] Xavier Anguera Miro, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland, and O. Vinyals. Speaker diarization: A review of recent research. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(2):356–370, February 2012.

[4] Barry Truax. Handbook for acoustic ecology [cd-rom]. *Burnaby, BC: Cambridge Street Publishing*, 1999.

[5] Anssi Klapuri and Manuel Davy. *Signal processing methods for music transcription*. Springer science Business media, 2009.

[6] Characteristics of a Sound Wave. `https://www.siyavula.com/read/science/grade-10/sound/10-sound-03`, 2015.

[7] Encyclopædia Britannica, Tone. `https://www.britannica.com/science/tone-sound`, Apr 2019.

[8] Paul Boersma and David Weenink. Praat: doing phonetics by computer (version 5.1.13), 2009.

[9] Catherine Anderson. *Essentials of Linguistics*. 2018.

[10] Brian Moore, Lorraine Tyler, and William Marslen-Wilson. Introduction. the perception of speech: From sound to meaning. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 363:917–21, 04 2008.

[11] Thomas Zawistowski and Paras Shah. An introduction to sampling theory. *Ejemplo de muestreo y sobremuestreo*, 2010.

[12] Term: Sampling rate (audio). `http://www.digitizationguidelines.gov/term.php?term=samplingrateaudio`, 2019.

[13] Praat for beginners - sidney wood. `https://person2.sol.lu.se/SidneyWood/praate/frames.html`, 2005.

[14] Praat for beginners - tutorial: Spectral analysis. `https://swphonetics.com/praat/tutorials/spectral-analysis/`, 1994-2019.

[15] The spectrograph spectral analysis. `http://www.ncvs.org/ncvs/tutorials/voiceprod/tutorial/spectral.html`, 2020.

[16] Speech recognition. `https://searchcustomerexperience.techtarget.com/definition/speech-recognition`, 2016.

[17] Speech-to-text - transcriptions audio basées sur le machine learning. `https://cloud.google.com/speech-to-text?hl=fr`, 2020.

[18] R Golda Brunet and Hema A Murthy. Impact of pronunciation variation in speech recognition. In *2012 International Conference on Signal Processing and Communications (SPCOM)*, pages 1–5. IEEE, 2012.

[19] Colleen G. [Le Prell] and Odile H. Clavier. Effects of noise on speech recognition: Challenges for communication by service members. *Hearing Research*, 349:76 – 89, 2017. Noise in the Military.

[20] Haizhou Li, Bin Ma, and Kong Aik Lee. Spoken language recognition: from fundamentals to practice. *Proceedings of the IEEE*, 101(5):1136–1159, 2013.

[21] itranslate voice. `https://www.itranslate.com/voice`, 2020.

[22] Google translate. `https://translate.google.com/`, 2020.

[23] Kenney Ng and Victor W Zue. Subword-based approaches for spoken document retrieval. *Speech Communication*, 32(3):157–186, 2000.

[24] Sadaoki Furui. *Chapter 7 - Speaker Recognition in Smart Environments*. Academic Press, Oxford, 2010.

[25] Sylvain Meignier, Daniel Moraru, Corinne Fredouille, Jean-François Bonastre, and Laurent Besacier. Step-by-step and integrated approaches in broadcast news speaker diarization. *Computer Speech and Language*, 20(2-3):303–330, April 2006.

[26] Thierry Dutoit. *An introduction to text-to-speech synthesis*, volume 3. Springer Science & Business Media, 1997.

[27] Patrick A. Naylor. Chapter 33 - introduction to speech processing. In Joel Trussell, Anuj Srivastava, Amit K. Roy-Chowdhury, Ankur Srivastava, Patrick A. Naylor, Rama Chellappa, and Sergios Theodoridis, editors, *Academic Press Library in Signal Processing: Volume 4*, volume 4 of *Academic Press Library in Signal Processing*, pages 983 – 984. Elsevier, 2014.

[28] Philipos C Loizou. *Speech enhancement: theory and practice*. CRC press, 2013.

[29] Shalev-Shwartz Shai and Ben-David Shai. *Understanding machine learning: from theory to algorithms*. Cambridge University Press, 2016.

[30] Aalto University Wiki, Vector Quantization (VQ). `https://wiki.aalto.fi/pages/viewpage.action?pageId=149883153`, 2019.

[31] Aravind Ganapathiraju. Support vector machines for speech recognition. 2002.

[32] Gülin Dede and Murat Hüsnü Sazlı. Speech recognition with artificial neural networks. *Digital Signal Processing*, 20(3):763 – 768, 2010.

[33] A. B. Nassif, I. Shahin, I. Attili, M. Azzeh, and K. Shaalan. Speech recognition using deep neural networks: A systematic review. *IEEE Access*, 7:19143–19165, 2019.

[34] Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu. Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on audio, speech, and language processing*, 22(10):1533–1545, 2014.

[35] Haşim Sak, Andrew Senior, and Françoise Beaufays. Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition. *arXiv preprint arXiv:1402.1128*, 2014.

[36] Namrata Dave. Feature extraction methods lpc, plp and mfcc in speech recognition. *International journal for advance research in engineering and technology*, 1(6):1–4, 2013.

[37] M.h. Moattar and M.m. Homayounpour. A review on speaker diarization systems and approaches. *Speech Communication*, 54(10):1065–1103, 2012.

[38] C. Barras, Xuan Zhu, S. Meignier, and J.-L. Gauvain. Multistage speaker diarization of broadcast news. *IEEE Transactions on Audio, Speech and Language Processing*, 14(5):1505–1512, 2006.

[39] Neville Ryant, Kenneth Church, Christopher Cieri, Alejandrina Cristia, Jun Du, Sriram Ganapathy, and Mark Liberman. First dihard challenge evaluation plan. 2018.

[40] Jinlong Wu, Jianxun Wang, Heng Xiao, and Julia Ling. Visualization of high dimensional turbulence simulation data using t-sne. *19th AIAA Non-Deterministic Approaches Conference*, May 2017.

[41] Yung-Yao Chen, Yu-Hsiu Lin, Chia-Ching Kung, Ming-Han Chung, and I-Hsuan Yen. Design and implementation of cloud analytics-assisted smart power meters considering advanced artificial intelligence as edge analytics in demand-side management for smart homes. *Sensors*, 19(9):2047, Feb 2019.

[42] Daniele Barchiesi, Dimitrios Giannoulis, Dan Stowell, and Mark D Plumbley. Acoustic scene classification: Classifying environments from the sounds they produce. *IEEE Signal Processing Magazine*, 32(3):16–34, 2015.

[43] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. The kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*, number CONF. IEEE Signal Processing Society, 2011.

[44] Geoffrey E. Hinton and Terrence J. Sejnowski. Unsupervised learning. In *Encyclopedia of Machine Learning and Data Mining*, 1999.

[45] Abhishekkumar K and Sadhana. Survey report on k-means clustering algorithm. *International Journal of Modern Trends in Engineering Research*, 4(4):218–221, Apr 2017.

[46] Dr. Michael J. Garbade. Understanding k-means clustering in machine learning. `https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1`, 2018.

[47] MA Syakur, BK Khotimah, EMS Rochman, and BD Satoto. Integration k-means clustering method and elbow method for identification of the best customer profile cluster. In *IOP Conference Series: Materials Science and Engineering*, volume 336, page 012017. IOP Publishing, 2018.

[48] Tung-Shou Chen, Tzu-Hsin Tsai, Yi-Tzu Chen, Chin-Chiang Lin, Rong-Chang Chen, Shuan-Yow Li, and Hsin-Yi Chen. A combined k-means and hierarchical clustering method for improving the clustering efficiency of microarray. In *2005 International Symposium on Intelligent Signal Processing and Communication Systems*, pages 405–408, 2005.

[49] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[50] Homogeneity score - mastering machine learning algorithms by giuseppe bonaccorso. `https://www.oreilly.com/library/view/mastering-machine-learning/9781788621113/2830f738-c6a5-460a-b518-23ecd3745c2d.xhtml`, 2020.

[51] Machine learning challenge winning solutions. `https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761`, 2018.

[52] Thair Nu Phyu. Survey of classification techniques in data mining. In *Proceedings of the International MultiConference of Engineers and Computer Scientists*, volume 1, pages 18–20, 2009.

[53] Phillip A Laplante. *Encyclopedia of Information Systems and Technology-Two Volume Set.* CRC Press, 2015.

[54] Liangxiao Jiang, Zhihua Cai, Dianhong Wang, and Siwei Jiang. Survey of improving k-nearest-neighbor for classification. *Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007)*, 2007.