OTTO VON GUERICKE
UNIVERSITÄT
MAGDEBURG

EIT

FAKULTÄT FÜR
ELEKTROTECHNIK UND
INFORMATIONSTECHNIK

Otto-von-Guericke-Universität Magdeburg

Fakultät für Elektro- und Informationstechnik

Institut für Informations- und Kommunikationstechnik (IIKT)

MASTER THESIS

Analysis of acoustic Features and automatic Recognition Experiments for Conversation Addressee Detection

First Examiner:	Prof. Dr. rer. nat. A. Wendemuth
Second Examiner:	Dr. -Ing. R. Böck
Supervisor:	Dr.-Ing. Ingo Siegert
Name:	Shuran Tang
Study Course:	Elec. Engin.& Inf. Tech (EEIT)
Matrikelnummer:	205980
Date of Submission:	19.06.2017

OTTO VON GUERICKE
UNIVERSITÄT
MAGDEBURG

EIT

FAKULTÄT FÜR
ELEKTROTECHNIK UND
INFORMATIONSTECHNIK

MASTER THESIS

für Herrn Tang Shuran

Thema

Analysis of acoustic Features and automatic Recognition
Experiments for Conversation Addressee Detection

Background:

Humans are able to detect whether a person is speaking to them or to another person by visually observing the person. Furthermore, the way people are talking is dependent of their situation, which results in slightly different ways of talking. Technical speech interaction systems do not have this opportunity to detect the conversation addressee and are bound to compromises. The person has to initiate a conversation with the system by using a keyword or by pushing a button.

Objective of this thesis:

The master thesis will investigate this issue. Therefore, in a first step given speech material (LAST MINUTE CORPUS and LMC INTERVIEWS) of the same persons talking to a machine and to another person, has to be prepared by removing samples with unwanted background noise and the voices of the system or other persons. Afterwards, acoustic differences should be analyzed using appropriate statistical tests based on the distributions of the extracted commonly used features. The outcome has to be judged regarding the question whether persons are talking differently to humans or technical systems. If this can be affirmed, recognition experiments to distinguish the conversation addressee have to be conducted with both validation strategies: speaker dependent and speaker-independent.

Magdeburg, 16.01.2017

Ausgabe am: 20.01.2017
Abgabe bis: 19.06.2017

Prof. Dr. rer. nat. A. Wendemuth
Aufgabensteller

Erstprüfer: Prof. Dr. A. Wendemuth
Zweitprüfer: Dr. -Ing. R. Böck

Prof. Dr. rer. nat. C. Hoeschen
Vorsitzender des Prüfungsausschusses

Betreuer: Dr. -Ing, I. Siegert

Declaration by the candidate

I hereby declare that this thesis is my own work and effort and that it has not been submitted anywhere for any award. Where other sources of information have been used, they have been marked.

The work has not been presented in the same or a similar form to any other testing authority and has not been made public.

Magdeburg, Jun 19, 2017

Abstract

There have been a lot of applications developed to help people interacting with the computers through acoustic signal. But the traditional methods have some limits in certain scenarios, such as driving car and noisy party, and some methods may lead to confusion of the computer if the instruction signal is mixed with the background voice. In these circumstances, the computer system can’t detect the addressee of the instruction.

Therefore, in this thesis, a new method is developed to help companion computers to automatically detect the conversation addressee from the acoustic signal. Specifically, a feature set is extracted from the original dataset. Then various machine learning algorithm is applied to train a classifier model, which is used to classify the input signals and decide the addressee of it. At last, a comparison between different algorithms and samples is made.

The result of the experiments has proved the validity of this method. However, a further experiment should be conducted afterwards to check its robustness in subjects with different languages and cultures.

Table of Contents

1 Introduction..5
1.1 General Meaning of Conversation Addressee Detection..5
1.2 The Problems for Conversation Addressee Detection...7
1.3 Literature Review..9
2 Methods..11
2.1 General Approach...11
2.2 Dataset Introduction..13
2.3 Data Preparation...14
2.4 Feature Extraction...17
2.4.1 Low-level Descriptors...18
2.4.2 Functionals...22
2.5 Statistical Tests..23
2.5.1 Test design..24
2.5.2 Test calculation...25

2.5.3 Effect Size.....26
2.6 Ranking Score.....27
2.7 Pattern Recognition.....28
2.7.1 Classifier Algorithms.....28
2.7.2 Experiment Groups.....30
2.7.3 Performance Evaluation.....31
3 Results.....34
3.1 Data preparation.....35
3.2 Feature Analysis.....38
3.2.1 Extracted Features.....38
3.2.2 Test Results.....39
3.2.3 Ranking Score.....41
3.2.4 Selected Features.....42
3.3 Recognition.....44
3.3.1 Initial Experiments.....44
3.3.2 Classifier Selection.....44
3.3.3 Performance Comparison for Gender.....49
3.3.4 Performance Comparison for Age.....50
3.3.5 Performance Comparison for Normalization.....52
4 Conclusion.....53
5 References.....55

List of Symbols

b	Spectral band index $b \in [0, B]$
B	Number of spectral bands
m	Discrete frequency bin index $m \in [0, M]$
M	Number of discrete frequency bins
n	Discrete time index $m \in [0, N]$
N	Number of samples
$X_p(m)$	Power spectrum
$\Phi(m)$	Spectral filter shape, discretised to bins
$C(i)$	ith cepstral coefficient
Θ	Frequency scale transformation function

List of Tables

Table 1 The Functionals of the Emobase Feature set in OpenSMILE
Table 2 Speakers Number of each Group
Table 3 Confusion Matrix
Table 4 Explanation of Labels and Amounts of Corresponding Segments
Table 5 Number of Features for Corresponding LLD
Table 6 Comparison of LLD Number between All and Selected Features
Table 7 Comparison of Functionals Number between All and Selected Features
Table 8 Confusion Matrix for Random Forest
Table 9 Performance Matrix for Random Forest
Table 10 Confusion Matrix for Support Vector Machine
Table 11 Performance Matrix for Support Vector Machine
Table 12 Comparison of Average Performance Metrics of All Algorithms
Table 13 Performance Matrix for Gender Comparison
Table 14 Performance Matrix for Normalization Comparison

List of Figures

Figure 1 Scheme of the Derived Method
Figure 2 The Interface of ikannotate2
Figure 3 Significance Level of All Features
Figure 4 Distribution of Features’s Significance Level
Figure 5 The Distribution of Ranking Score for SDF

1 Introduction

1.1 General Meaning of Conversation Addressee Detection

With the development of modern technology, people trend to spend more time on interaction with computers or machines. But for the interactions between human and computer, there is one problem. And it is caused by the capability difference between human receiver and computer receiver. Humans are able to detect the conversation addressee by visually observing the speaker. In contrast, a computer can’t combine the direction of the acoustic signal source and the speaker, because its lack of a vision system. So it doesn’t know if it is the addressee of the message.

Matured automatic speech recognition (ASR) systems, such as “Siri”, “OK, google” or “Alexa”, use a keyword or a button to initiate a conversation, so that the system could confirm that they are the addressee of the following utterance [1]. Only after this confirmation, the system would take the following utterances as input. Afterwards, related information is extracted from the input signal, and corresponding instructions or requests are constructed to meet the requirements of the speaker. However, the confirmation process takes time to conduct, and its success rate is severely influenced by the situational noise and accent of speaker. Sometimes one speaker may just say out the keyword or push the button by accident, which would result into inconvenience. For example, as reported in recent news, a family in California bought an Amazon Echo, and wrote code for their daughter to help her buy dollhouse. The local media reported this, but some viewer of this report found that their devices happened to order a dollhouse because of mistaking remark from the TV as commands

[1].

Are there any other methods could help the ASR system to figure out the conversation addressee without those keywords or button? It would be an opportunity to improve user experience and reduce mistakes for our human computer interaction device.

In this master thesis, I would try to detect the conversation addressee with the analysis of acoustic features and automatic recognition experiments.

1.2 The Problems for Conversation Addressee Detection

The conversation addressee detection is challenging, because the speaker is broadcasting his voice to all the objects in the conversation scenario, but he may change his intended addressee at any time. In the real world conversation, the addressee detection is not only achieved by speech recognition, but also the combination of deictic references, gaze and gesture. However, in this scenario, it is assumed that only the speech signal is used for the addressee detection. As known to us, most modern conversation interaction system combined the usage of both voice and button, such as “Siri” or “OK, google”. However, it is not convenient or even dangerous in some scenario, such as driving or working. If the users can interact with the system without seeing or touching, this problem would not be under consideration anymore.

As the input of a ASR system, acoustic signal is 1-dimensional continuous signal. The ASR should decide the time point when the signal source begin to take it as the addressee. Usually, people would not make a pause when they change their addressee. So there is always no distinguished indication when people talk in the real world, which increase the difficulty to prepare for the training dataset. However, in these experiments, the voice from same subject would be recorded separately when he talked with human and computer. Therefore, this problem would be resolved.

Secondly, in a conversation, there maybe not just one speaker but multi-speakers at present. Overlaps always occur in this situation. The signal segments classification become more complicated. In these experiments, the overlaps would be manually labelled as noise, and would not be used for training and test.

Additionally, as illustrated before, the addressee detection is a multi-modal task for human. The information contained in the utterance sometimes plays as an essential role when someone is trying to confirm who is addressed by the speaker. But if an ASR is applied in the addressee detection system, the error rate of the ASR would considerably influence the result of addressee detection. Can we find a method to detect the addressee without considering the semantic meaning of the speech? Instead of the speech content, can we use the acoustic characteristic to distinguish the particular addressee from others?

1.3 Literature Review

Since the human-computer interaction become more and more widely applied to our daily life, some researches about addressee identification has been done for the past years.

In 2004, researchers take the power of acoustic cues, visual cues, and their combination, to identify the addressee in HHI and HCI [2]. Three cues were separately used to identify addressee in scenario experiments. The aim of the experiments is to evaluate the performance of each cues independently. In the first experiment, they used both the low level and high level features for the identification. The features, including number of imperatives and sentence length, are used for classifier training. The result shows that only 49% utterance could be correctly classified by the model. For the visual experiment, the human's head pose is visually estimated by researchers, and the result shows that it is a reliable cue for addressee identification. At last, they combined the first method and the visual method together. The combined method shows a significant performance improvement. The average correct classification rate reaches as high as

0.92.

However, for some real working environments, the visual system is not available. Therefore, the head poses of the subjects would not be known.

The former experiments were conducted in a face–to–face scenario. For non face–to–face scenario, for example communication through radio, the identities of both the speaker and addressee are not clear. Therefore, in a paper [3], the researchers tried to understand and predict the identities of the speakers and addressees when they organize their work through radio. The data set used in this research consists of multi–party dialogues in a military simulation exercise. According to several levels of interaction structures, the data was segmented. The segments were labelled with various factors, such as location of the speaker, or role of the addressee. Then the segments were classified into different patterns based on those features.

The result shows that, the exchanges between different task team are more formal, and 75% of them include an identification call for the speaker or addressee. In contrary, only 50% of the exchanges within the same team include an identification call. Some factors, including roles, ranks and relationships between speakers and addressees, have strong impact on whether the exchange include identity information.

Although, people don’t include an identity information when they speak in daily life, the method to use various labelled factors to find the communication pattern is enlightening. In this thesis, the data set is also segmented and labelled before classification process. However, instead of semantic features, the acoustic characteristic of the speech is used as the features to detect the speech addressee. The detail of this method is illustrated in the following section.

2 Methods

2.1 General Approach

In this thesis, a new method for the addressee detection is developed. This approach is based on an assumption that, the characteristics of the speaker’s voice are different when they are addressed to computer or human.

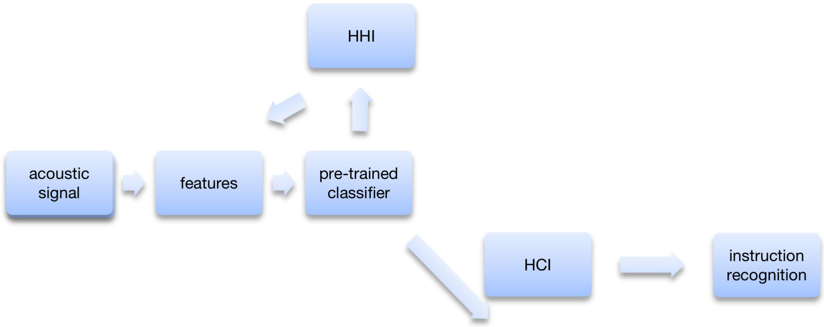

Figure 1: Scheme of the Derived Method

As shown above, instead of using deictic words or a button, the speech recognition system would take the acoustic signal as input. Then, a set of acoustic features is extracted from the input, which is taken as one observed instance. The instance is classified by a pre-trained model as Human-Human Interaction (HHI) or Human-Computer Interaction (HCI). If it was classified as HCI, the system would process it into the speech recognition stage, and the instruction or command would be constructed from it. However, if it was classified as HHI, the system would realize that it is not the addressee of this sentence. Then it drops this observation and wait for the next input.

One issue of this method is selection of the features. Which feature should be used to represent the difference of the speech characteristic for the model training? In order to find the answer, a statistical test is designed and performed between the HCI group and samples from HHI group. Based on the test result, an influence score is constructed for each feature. The score shows to which degree the feature is different between the two groups. Afterwards, the features with score higher than a certain threshold, are selected from the whole set according to the score. The selected feature set is then used to train the model. Afterwards, the trained model can be used in a computer system to perform the addressee detection.

2.2 Dataset Introduction

The dataset used for this thesis is the LAST MINUTE corpus, conducted as a Wizard of Oz (WoZ) experiment. This dataset aims to help researchers to investigate the performance of users when they interact with a companion system [3]. The data set does not only contain audio and video information, but also the transcript of all interactions. Additionally, all subjects have filled several psychometric questionnaires. Therefore, this data set includes all necessary information for the investigation of this thesis.

In LAST MINUTE corpus, the subject is expected to plan their luggage for a 2 weeks’ holiday. So they pack their own suitcase by choosing items from 12 categories. For each category, the subjects get a menu of options. The subjects should interact with the system to make their selection. For the system, it sets a bunch of rules for the suitcase, such as weight limit, size restriction, risk level etc.

133 subjects, balanced by their age, gender and education level, have performed the experiment. After the experiment, an additional interview is performed for selected subjects. The audio bandwidth is 44kHz. Biopsychological data, such as heart beat, respiration, as well a video data is also recorded and available. The experiments were conducted in German.

In this thesis, only the audio part of the data set is used. The recorded audio signal would be segmented into pieces for easier labelling process and further analysis. The recorded interactions between the subjects and computer systems are labelled as Human Computer Interaction (HCI), and the interview parts are labelled as Human Human Interaction (HHI). For each subject, the segmentation number of HHI samples range from 1200 to 2400, and the segmentation number of HCI samples range from 60 to 220. The duration of each segmentation ranges from 0.35 second to 30 seconds. The experiment date and time are also labelled as part of the file name for each segmentation. The linguistic meaning of the audio segmentations is not concerned in this experiment.

2.3 Data Preparation

Data preparation is essential step for the whole experiment in the thesis, because the real world experiment data is a mixture of noise, silence, unexpected speech and expected clean speech. The unwanted samples should be removed from the interested part of the dataset, so a training dataset without unwanted information can be formed. In order to achieve this goal, manual labelling is necessary for the samples.

Data preparation is a considerately time consuming process. For each segmentation of the audio recordings, one needs to play the file, listen to the content, choose right label, and then process the next one. Afterwards, the right labels should be recorded and linked to corresponding segments. Sometimes one needs to re–listen to the audio if the choice can’t be made. This process should be repeated for more than 80000 segmentations, and every choice should be inputted and saved in certain file. So it is quite time consuming.

Therefore, an annotation tool is required to automate the whole process. In ikannotate2, the configuration for the labels could be easily modified with certain format. The tool also integrates the inspection and labelling together to form a more fluent work flow.

The labelling tool “ikannotate2” was developed by the Cognitive System Group of the OVGU [5]. It is implemented with the C++ application framework Qt5, therefore it could play various format media files and run natively under Windows, Linux and macOS.

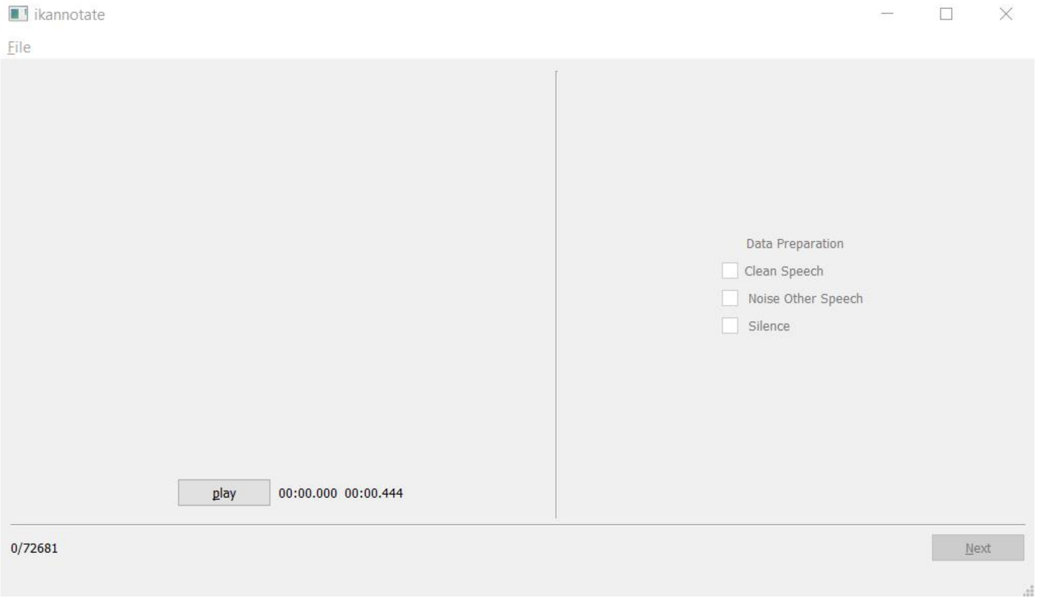

Figure 2: The Interface of ikannotate2

As shown above, the number of imported audio file list is shown at left bottom. The time tag besides the “play” button indicates the total duration of the current audio file. The labels are set at the right side of the interface, and the check box would indicate the choice. Whenever the correct labels were chosen, the user should click the “next” button at the right bottom, then the correct labels are saved in the background and the next audio file is automatically loaded.

Besides the manual labelling with ikannotate2 in this thesis, some automatic filters were developed using python. For a lot of the samples, the utterance duration is too short to make any sense. Therefore, a program is implemented to label all samples with duration less than 0.35s as unwanted

“Too Short” samples. Another implementation is used to automatically filter out the silent samples. The program is based on the assumption that, for a silent sample, the signal energy of every segmentation from this sample should not exceed a certain limit. If a sample complies with this criterion, it would be labelled as unwanted “Auto Silence”. The threshold for the energy limit was manually set by the pre-test. After the automatic labelling, the result is validated by listening to examples of the filtered segments.

2.4 Feature Extraction

By physical explanation, speech is a micro variation of pressure in the physical world. Acoustic signals are discrete digital numbers sampled from the continuous pressure variation. However, most signals in the audio samples are unnecessary or redundant information. Usually we would use digital signal processing techniques to discard the unnecessary information, and keep the interested part. This process is called feature extraction [5].

In this thesis, the feature extraction is implemented with the open source software “openSMILE”. The openSMILE is a modular and flexible feature extractor toolkit for signal processing and machine learning [7]. It is written in C++, and has fast efficient and flexible performance across the platforms, such as Linux, Windows and MacOS. In this thesis, the feature set “emobase.conf” is used, which is ongoing designed for emotion recognition, but also used for different approaches. It consists of 988 acoustic features. They are constructed from 26 low-level descriptors and 21 different basic functionals.

2.4.1 Low-level Descriptors

The low-level descriptors (LLD) include: intensity, loudness, 12 Mel-Frequency Cepstral Coefficients (MFCC), pitch (F0), probability of voicing, F0 envelope, 8 Line Spectral Frequencies (LSF), Zero-Crossing Rate. They are shortly described in the following.

Intensity: intensity is defined as the sum of squared amplitudes of the signal x(n), where x(n) is usually weighted by a Hamming window function [7]. The formula is shown below:

$$I = \sum_{n=0}^{N-1} x^2(n) \tag{2.1}$$

Loudness: Because loudness is a concept related to human’s understanding about acoustic signals, not only the energy of the signal should be considered, but also the human’ hearing or perception should also be taken into account [7]. Loudness is a concept related to the energy of the signal, but also considering the human perception. It could be calculated as below:

$$E_{I,approx} = (\frac{I}{I_0})^{0.3} \qquad (2.2)$$

In the formula, it refers to the intensity of the signal. Then $I_0=10^{-6}$ is introduced as reference intensity when the signal amplitude |x(n)| = 1.0 and Sound Pressure Level at 60 dB [8].

MFCC: MFCCs have been a very successful feature set in the field of speech recognition since it was first introduced in 1993 [8]. Because they effectively simplify the representation of the speech amplitude spectrum. In the following part, I would introduce the process to compute MFCC step by step.

The first step is to convert the signals into frames by applying window functions. Typically, a hamming window is used to avoid edge effect. Then the Discrete Fourier Transformation(DFT) is applied to each frame as following:

$$X(m) = \sum_{n=0}^{N-1} x(n) * e^{\frac{-j2\pi mn}{N}} \qquad (2.3)$$

In order to model the masking effect of human hearing in both time and frequency domain, a set of frequency bins is always combined together to form a reduced number of bands [8]. Due to the fact that lower frequencies are perceived more importantly than higher frequency, the central frequency of each band is scaled according to Mel Frequency Scale [9].

$$f^{(mel)} = \Theta(f) = 1127 * \log\left(1 + \frac{f}{700}\right) \quad (2.4)$$

Therefore, the band spectrum X(b) would be converted into band spectrum in Mel Scale X(mel)(b). Afterwards, a logarithm of the amplitude spectrum is derived from the former result.

In the last the step to construct MFCC feature set, the components of the Mel-spectral vectors should be decorrelated. Usually it is approximated with the Discrete Cosine Transform(DCT), resulting into Mel-cepstral coefficients:

$$C^{(mel)}(k) = \sqrt{\frac{2}{B}} \sum_{b=0}^{B-1} X_P^{(log,mel)}(b) \cos\left(\frac{\pi k}{B}\left(b + \frac{1}{2}\right)\right) \quad (2.5)$$

The MFCC features would be achieved after the lower order coefficients of the Cepstral coefficients $C^{(mel)}(k)$ are emphasized.

Pitch: Pitch is the perceived frequency of a tone. It is related to pitch(F0), the lowest frequency in harmonic series of the tone, but not identical. It can be detected by Pitch Detection Algorithms. The pitch of a tone can be detected in time domain or frequency domain. For time domain, an estimate of the fundamental frequency for each small frame is constructed. The pitch is therefore considered as the average of the estimates. For frequency domain, the acoustic signals are converted into frequency spectrum. Then various methods can be applied to detect the peaks of the harmonic series.

Probability of voicing: The Probability of voicing can be calculated by formula below,

$$P_v = \frac{ACF_{max}}{ACF_0} \tag{2.6}$$

The short-time Autocorrelation function (ACF) describes the signal's self-similarity at given discrete time lags. As shown in the equation above, the primary factor for Probability of voicing is the energy of the signal. It is also influenced severely by the Zero Crossing Rate of the signal.

F_0 envelope: F_0 is also known as the fundamental frequency, it is the lowest frequency of the periodic waveform for the tone. It is a property of the vibration source. Some information about the state of the vocal tract is also contained in the amplitudes of the F_0 harmonics.

Line Spectral Frequencies: Line Spectral Frequencies (LSP) can be considered as the boundaries besides the formants. The LSPs have important quantization properties in the transmission of vocal tract information from speech encoder to decoder [10]. There are various methods to generate LSPs parameters, but the core problem is to find the roots of the line spectrum polynomial pair.

Zero-Crossing Rate: Zero-Crossing Rate (ZCR) describes the frequency of sign changes of the audio signal. A sign change is defined to occur when:

$$x(n-1) \cdot x(n) < 0$$

(2.7)

A high ZCR usually represents that the signal has a high frequency, but it is also related to the content of the signal. Gaussian noise always has a rather high ZCR than harmonic signals. ZCR is also used to tell the difference between voiced signal and unvoiced signal.

2.4.2 Functionals

For the configuration of the feature extraction, multiple statistical functionals could be applied to each LLD. The functionals include: Max./Min. value and respective relative position within input, range, arithmetic mean, 2 linear regression coefficients and linear and quadratic error, standard deviation, skewness, kurtosis, quartile 1–3, and 3 inter-quartile ranges.

The configuration for each feature is shown in the output file. The functionals are expressed as suffix appended to the name of each LLD. For example, "_sma" indicates that the descriptor is smoothed by moving average filter with a window length 3. The other functionals are explained in the table below:

Functional Name	Functional Intro

max	The maximum value in frames
min	The minimum value in frames
range	The difference between the max and min
maxPos	The absolute position of the maximum value in frames
minPos	The absolute position of the minimum value in frames
amean	The arithmetic mean of the frames
linregc1	The slope of a linear approximation of the contour
linregc2	The offset of a linear approximation of the contour
stddev	The standard deviation of the values in the frames
skewness	3rd order moment
kurtosis	4th order moment

Table 1: The Functionals of the Emobase Feature set in OpenSMILE

2.5 Statistical Tests

The aim of this thesis is to develop a new method to automatically identify the addressee. This method is based on the assumption that the character of the speech would differ when a speaker speaks to different addressees. Especially, when the difference addressees refer to human and computer, this method would be used to vastly optimize our interaction experience with a companion system.

But how can we determine whether there is a significant difference between the acoustic signals for different addressees? Therefore, it is assumed that the mean of one feature in HHI samples should be significantly different with it in HCI samples of the same subject. In order to quantitatively examine this assumption, a t-test should be designed and performed for the dataset after feature extraction.

2.5.1 Test design

The first step for test design is to form a hypothesis about test result. In this situation, the hypothesis is shown as below [11]:

Null hypothesis H0: There is no difference between the population mean of the features extracted from the HHI group and HCI group for each subject, therefore:

$$\mathrm{u}\boldsymbol{I}_1 = \mathrm{u}\boldsymbol{I}_2 \qquad (2.8)$$

Alternative hypothesis H1: The difference between the population mean of the features extracted from the HHI group and HCI group for each subject is significant.

Therefore:

$$\mathrm{u}\boldsymbol{I}_1 \neq \mathrm{u}\boldsymbol{I}_2 \qquad (2.9)$$

The Welch's t–test, which is also called unequal variances t–test, is considered to be the best choice, because:

a) It is assumed that the features are of Gaussian distributions.

b) The variance of the population is unknown

2.5.2 Test calculation

Before the actual calculation, a group is sampled from the HHI data set of the corresponding subject. The sample size is equal to the size of subject’s HCI data set.

Then at first, for each subjects, the mean $\overline{x}$ 1 , $\overline{x}$ 2 and variance s_1^2, s_2^2 of every feature are calculated for both HHI group and HCI group.

Secondly, the statistic t could be calculated by the following formula:

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{s_1^2}{N_1} + \frac{s_2^2}{N_2}}} \qquad (2.10)$$

The degree of freedom should also be calculated v=N−1, where N represent the sample size for each group. While a confidence level of 5% was chosen and it’s two sides test, the reference t distribution value $t_{v,0.025}$ could be calculated.

Afterwards, the calculated t would be compared to $t_{v,0.025}$.

If,

$$-t_{v,0.025} < t < t_{v,0.025} \qquad (2.11)$$

As the conclusion, there is no significant difference between the means of these two groups. In other words, the null hypothesis is accepted.

However, if,

$$|t| > |t_{v,0.025}| \qquad (2.12)$$

As the conclusion, there is significant difference between the means of these two groups. Therefore, the null hypothesis is rejected.

The calculation is implemented with python, the mainly used libraries are Numpy, Scipy and Math. Details of the implementation can be found in the attachment.

2.5.3 Effect Size

After the implementation of the statistic test, the significant different features are distinguished from others. But in order to understand the strength of each test, the effect size for every test result should be calculated.

The effect size is a quantitative estimation for the magnitude of effect statistics [12]. The Cohen's d is one well used type of the effect size. It is based on the difference between the means of the sample groups. It can be calculated as below:

$$d = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{s_1^2 + s_2^2}{2}}} \tag{2.13}$$

As suggested by Cohen [16], 0.2 should be considered as a small effect size, 0.8 as a large size, and 0.5 as a medium size. If a large effect size is used to set the threshold for feature selection, some significant features would be missed and not taken into account. Otherwise, if a small effect size is used, some non-significant features would be selected. Therefore, a medium effect size d=0.4 is chosen for the experiments.

2.6 Ranking Score

After the statistic test, some features are considered significantly different between HCI samples and HHI samples. But it is still a question that to which degree is the difference. Therefore, in order to distinguish the significantly different features (SDF) from others, it is necessary to construct a ranking score for the features. Then the features can be sorted and selected according to their score.

As the result of the statistical tests, the Boolean matrix only show the qualitative answer. In order to find the quantitative strength of the test result, the effect size of each test should be calculated and taken into account as discussed before.

If the null hypothesis of the test was rejected, a larger effect size indicates a stronger effect of the test result. In contrary, it would get close to zero if the null hypothesis was not rejected. Therefore, the ranking score matrix is defined as:

$$R = T \cdot D \tag{2.14}$$

T represents the Boolean matrix of the test results. D is the effect size matrix of the tests. They are all 30*988. As the result, the ranking score of each feature is the arithmetic mean of each column for matrix R.

2.7 Pattern Recognition

When the appropriate feature set is extracted and selected from audio files, in order to decide the addressee for each of them, a classifier model is used. The corresponding audio file would be correctly classified into HCI group if it is spoken to the computer. A lot of algorithms have been used in solving classification task. In this thesis, 3 most well-known and performance-proved algorithms are used for the classification. Then they are tested and evaluated over the dataset.

2.7.1 Classifier Algorithms

The aim of a classifier is to assign class labels to the objects based on their observed features vector. The training process of a classifier is to construct the rule of assigning labels according to the past observations.

Among the classifier algorithms, Naive Bayes is distinguished by its simplicity and robustness [13]. The principle can be easily understood, and complicated parameter estimation is not needed. It can be easily and rapidly applied to huge amount of data. As the dataset used in this thesis include more than 90000 observations, Naive Bayes is a good choice for the classification.

Another appropriate choice are Support Vector Machines (SVMs). The core idea of SVM is to build binary classifier, optimized to separate observed vectors in a constructed feature space. The best possible separation rule is the decision boundary between the classes. Usually, the so called “best possible” is achieved by maximizing the margin between the two classes in the feature space. As the SVM has already been proven to be successful in speech analysis [14], it is also applied in the recognition experiments of this thesis.

Another method used in classification tasks is Decision Tree. In the trained model, it sets decision thresholds for features based on the observed samples in the training set. In other words, a decision tree is formed in the training process, and it is used to assign class labels in the tests. Though this method is easy to understand and computational efficient, it is not robust enough and always led to overfitting. So as to avoid these problems, the Random Forest algorithm is used. Random forest is actually a weighted combination of multi decision trees. But instead of the whole training dataset, decision tree in random forest is trained by random samples from the dataset. This randomness provides robustness and complexity for the model. Therefore, although random forest is more computational expensive than decision tree, it always provides better accuracy.

In this thesis, these three algorithms are used to build the classifier and compared with their performance.

2.7.2 Experiment Groups

How does the age or gender influence the classifier performance? As discussed before, the dataset also includes the information about the subjects in a wide range, such as gender and age. For the sake of answering the question, the subjects are separated into different groups according to their age and gender. The table below shows the number of speakers belongs to each group:

		Age		Sum of Age
		Young	Old	
Gender	Male	35	28	63
	Female	37	33	70
Sum of Gender		72	61	133

Table 2: Speakers Number of each Group

If the training dataset is composed of only elderly female subjects, how is the performance of the classifier on a test set composed of male and young subjects? What if the test set is composed of just one subject while a group of subjects for the training set? Corresponding methods, such as 10-fold cross-validation and Leave-One-Subject-Out, are used to evaluate the related influence.

Traditionally, features in training dataset are normalized before the training process, because it eliminates the scale difference between the features. But in these experiments, how would feature normalization influence the performance of classifier? In order to answer this question, the feature set of all subjects is normalized for both training and test dataset. The calculation process is shown below:

$$x' = \frac{x - min\ (x)}{max(x) - min\ (x)} \qquad (2.15)$$

The x represents the original feature value, and x' is the normalized feature. After the normalization, the classifier trained with normalized features is compared with the one trained with the original dataset for their performance. The result is shown in latter sections.

2.7.3 Performance Evaluation

After the training of a classifier model, it is important to estimate how the model will perform when given a new instance. It must be known to which extent the classifier makes a correct prediction.

The most basic evaluation metrics for a model is misclassification rate.

Misclassification rate is calculated as below:

$$\text{Misclassification Rate} = \frac{\textit{Number of incorrect Predictions}}{\textit{Total Predictions}} \qquad (2.16)$$

Misclassification rate makes a general evaluation about the classifier model, but it is just a one–dimensional metric. For sake of more detailed evaluation, the Confusion Matrix should be used for further analysis. In a confusion matrix, every prediction is labelled as True Positive (TP), True Negative (TN), False Positive (FP) or False Negative (FN). The structure of a confusion matrix can be shown as a table below:

		Prediction	
		Positive	Negative
Target	Positive	True Positive	False Negative
	Negative	False Positive	True Negative

Table 3: Confusion Matrix

If a prediction is labelled as True Positive, it means the real class label of the target instance is Positive, and the predicted class label is also Positive.

If one prediction is labelled as True Negative, it means the real class label of the target instance is Negative, and the predicted class label is also Negative.

If one prediction is labelled as False Positive, it means the real class label of the target instance is Negative, and the predicted class label is also Positive.

If one prediction is labelled as False Negative, it means the real class label of the target instance is Positive, and the predicted class label is also Negative.

To summary, the TP and TN are correct predictions, but FP and FN are incorrect predictions. For a better understanding of the classifier model, 4 helpful metrics are constructed based on the Confusion Matrix. They are Recall, Precision, F-measure, Correctness rate.

The recall tells, how confident we are about the ability of the classifier model, to find all the instances with positive class labels. It can be defined as follows:

$$Recall = \frac{TP}{(TP+FN)} \tag{2.17}$$

The precision tells, how confident we are about the prediction correctness whenever the classifier makes a positive prediction. It can be defined as follows:

$$Precision = \frac{TP}{(TP+FP)} \tag{2.18}$$

The F-measure is the combination of Recall and Precision. It is actually a weighted harmonic mean of them. It can be defined as follows:

$$F_1 \, measure = 2 * \frac{Recall*Precision}{Recall+Precision} \tag{2.19}$$

The Correctness rate is the overall evaluation for a model's performance. It is shown as follows:

$$Correctness \, rate = \frac{TP+TN}{TP+TN+FP+FN} \tag{2.20}$$

3 Results

In this section, the results of the different phases of the experiment are presented. First, the result of the data preparation is illustrated. The labels used for the labelling process, the methods used to increase the labelling process and the problems encountered during the labelling process are explained. Second, the results of the feature analysis, including the feature set configuration, and the definition of the ranking score are shown. Finally, the result of the recognition phase is shown. It includes the implementation of the initial experiment, the selection of the algorithm model, comparison between genders and ages, and the evaluation of the recognition performance.

3.1 Data preparation

Although the information about the subjects is well preserved, the audio segments are not labelled. In order to train the classifier model, those audio files have to been manually labelled about their contents. For convenience, as discussed before, the labelling tool “ikannotate2” is used for the procedure.

In summary, 6 different labels were used in the labelling process. They are composed of 'Clean Speech', 'Silence', 'Noise Other Speech', 'Auto Silence', 'Too Short', 'Error File'. The explanation of the labels and the amounts of segments with corresponding label are shown in the Table 4.

Label Name	Label Explanation	Segments Amount
Clean Speech	There is clear and meaningful utterance from the subjects in this audio file, and the background noise is within an acceptable level.	25369
Silence	Silence means that, there is neither meaningful utterance nor much background noise in this file. 5044 files were labelled as “Silence”.	5044
Noise	this file is either filled with non-sense background	20763

Other Speech	noise or the utterance from the experimenter.	
Auto Silence	Auto Silence means that, this file is labelled as “Silence” automatically by program. 5163 files were labelled as “Auto Silence”.	5163
Too Short	Too Short means that, the duration of this file is too short, so it is impossible to contain any meaningful utterance.	11004
Error File	There is internal damage of this file, and openSMILE is not able to read it and extract feature from it	2

Table 4: Explanation of Labels and Amounts of Corresponding Segments

The labelling process require a lot of patience even with the help of the labelling tool. However, some methods have still further been used to increase the labelling efficiency.

Among all the audio files, some of them actually contain no information because their durations are too short. After running the script, it was found that the duration of 11004 files is shorter than 0.35 second. Therefore, those files are automatically labelled as “Too Short”.

Besides, the content of some files are actually silence. In other words, there is neither utterance nor noise in them. To automatically label these files, a rectangular window function was multiplied along the signal series. If the sum of the products keeps under a certain threshold, it would be said that, this audio file is filled with silence. Consequently, 5163 files were classified as “Auto Silence” in this way. The python function is attached below:

```
Def sil_detect(file_path,window_size=0.025,step=0.025,
         threshold=16,endurance=3):
    temp= AudioSegment.from_mp3(file_path)
    temp.export("temp.wav",format="wav")
    rate, data= wavfile.read("temp.wav")
    frames_num=rate*window_size
    steps_num=rate*step
    windows_num=int((len(data)–frames_num)/float(steps_num))+1
    energy_list=[]
    for i in range(windows_num):
        window=data[i*steps_num:i*steps_num+frames_num]
        energy_list.append(get_energy(window))
    energy_list=np.array(energy_list)
    energy_list=energy_list[~np.isnan(energy_list)]
    exceeds=len([x for x in energy_list if x >= threshold])

    return (exceeds<=endurance,energy_list)
```

Listing 1: Code to automatically Detect Silence

After an audio files is played, there is no prompt tone to remind the listener whether the file is finished or not. In order to solve this problem, I sorted the audio files path list by their durations, so the ikannotate would play the files by sorted duration. Then the label would be decided after nearly appropriate time.

3.2 Feature Analysis

3.2.1 Extracted Features

The emobase features set is composed of 998 features. Among them, 114 features are composed from Pulse-code modulation (PCM) information. There are also 456 Mel-frequency cepstral coefficients (MFCC) based features. In total, 304 of them are related to line spectral pair frequencies (lspFreq) LLDs, and 38 are related to voicing probability (voiceProb), 76 are related to the fundamental frequency (F0).

PCM	MFCC	lspFreq	voiceProb	F0	sum
114	456	304	38	76	988

Table 5: Number of Features for Corresponding LLD

3.2.2 Test Results

As mentioned before, the HHI group should be sampled for each test. The reason is that, the average ratio of the number of HHI files to the number of HCI is about 10:1. Because the minimum number of samples in HCI group is 55, it is reasonable to set the sample size as half of it avoid fortuity. Therefore, for each subject, both the HCI and HHI groups are randomly sampled 30 times for the test. Then the test result is a Boolean matrix of 30*998 dimensions, representing whether the corresponding features are significantly different between HHI and HCI. If a feature is significantly different between HCI and HHI ($\alpha<0.05$), the corresponding position in the matrix would be filled with "1", otherwise it is filled with "0".

To evaluate each feature, a sum function is applied to each column of the matrix. After performing the same function to each subjects, an average significance level would be gained for each feature. If a feature is completely different between HHI and HCI, the significance level will reach as high as 30. However, if it is completely same between the two groups, the significance level will be 0.

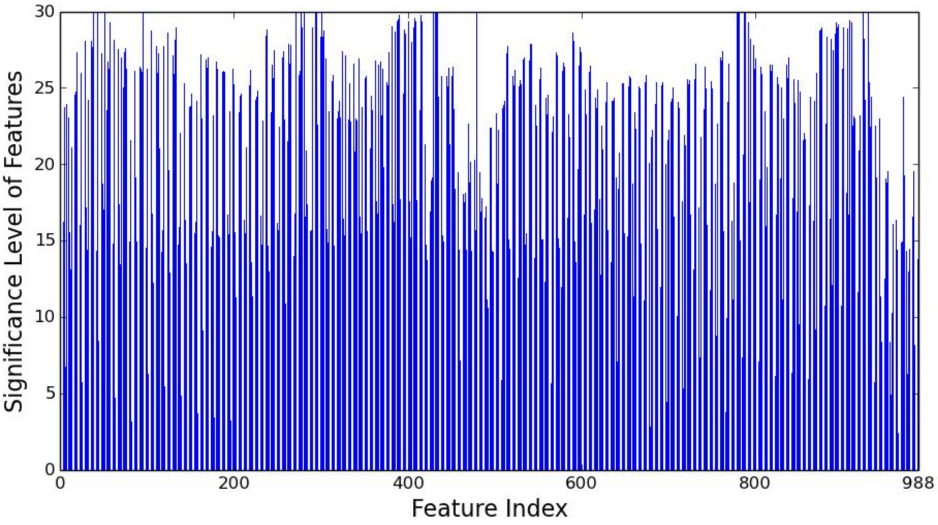

Figure 3: Significance Level of All Features

For better visualization, the distribution of the significance level is shown below:

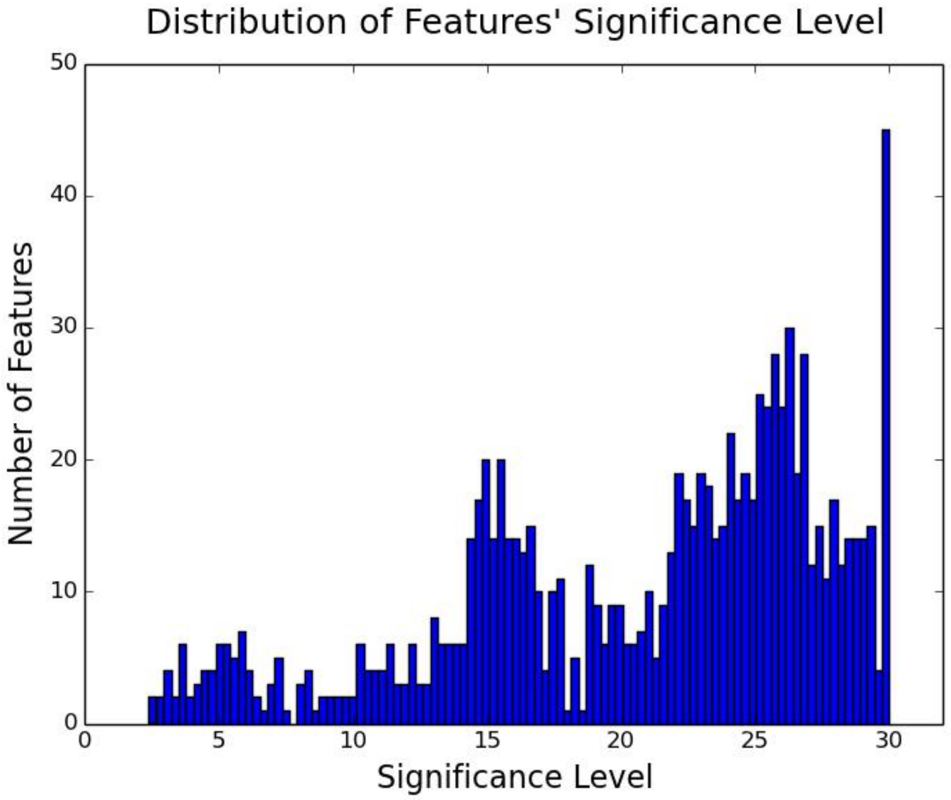

Figure 4: Distribution of Features's Significance Level

It is shown that, there is a gap at 18 for significance level of the features. Therefore, the feature set can be separated by significance level 18. At the same time, only a few features have their significance level below 12. It means that, for only a small number of features, the difference between HHI and HCI is not significant.

The purpose of this figure is to show that, for most subjects, the number of significantly different features is around 750, and the standard deviation is about 25. In other words, 95% observations have significantly different features number between 700 and 800.

3.2.3 Ranking Score

The construction process of the Ranking Score has been discussed in former section. It is product of the t-test result (1 for true and 0 for false) and the effect size of this test. Generally speaking, a higher ranking score means the feature is more likely to be significant between HHI and HCI samples, and it also has stronger effect size for the test. After the calculation, the distribution of the scores is shown below:

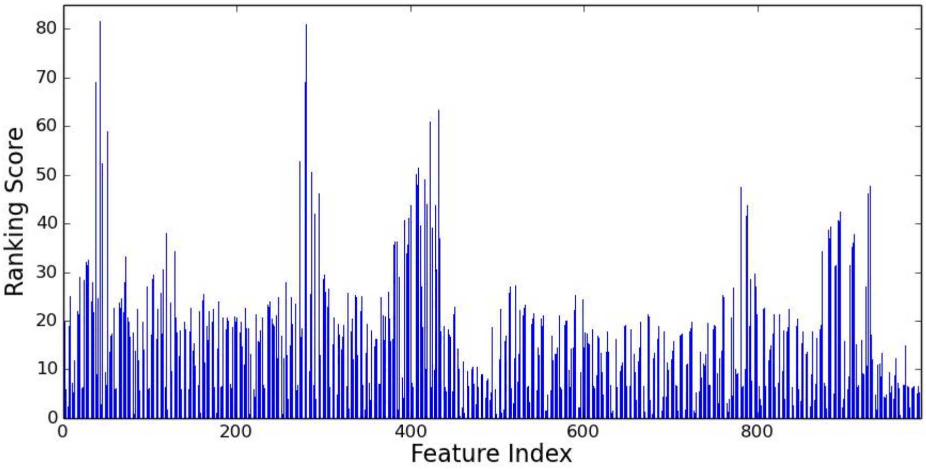

Figure 5: The Distribution of Ranking Score for SDF

As shown in Figure 5, the score of most features is lower than 30. But the highest score can reach as high as 80. A score of 10 is also considered as one of the boundaries, and 80% features have a score higher than 10.

It has been found that there is a gap around 18 for the significance level. Additionally, as discussed before, d=0.4 represents a relatively medium effect size. The product of the significance level of the gap and medium effect size is 7.2. When a threshold of 7.2 is set for the ranking score, 30% of all features are below it. Therefore, the other 700 features are selected as the most efficient features, and used for the next training process.

3.2.4 Selected Features

According to the ranking score, 700 features are selected from the others. But what is the difference between the selected features and unselected ones in their in their composition? The difference is explored in two aspects, the LLD and the functionals.

Here is the table to compare the difference in their LLD.

LLD Name	PCM	MFCC	lspFreq	voiceProb	F0
Number in All	114 (100%)	456 (100%)	304 (100%)	38 (100%)	76 (100%)
Number in Selected	28 (24.56%)	128 (28%)	76 (25%)	15 (39.47%)	41 (53.95%)

Table 6: Comparison of LLD Number between All and Selected Features

As shown in Table 6, about 25% features composed of PCM, MFCC and lspFreq are selected as the training features, but more than 45% features composed of voiceProb and F0 are selected.

It means the features related to voice probability and fundamental frequency are more likely to be different between HHI and HCI samples.

Here is the table to compare the difference in their functionals.

Functional Name	max	min	range	maxPos	minPos	amean	linregc1	linregc2	stddev	skewness	kurtosis
Number in All	104 (100%)	104 (100%)	52 (100%)	52 (100%)	52 (100%)	52 (100%)	52 (100%)	52 (100%)	52 (100%)	52 (100%)	52 (100%)
Number in Selected	54 (51.92%)	48 (46.15%)	52 (100%)	3 (5.77%)	0 (0%)	26 (50%)	1 (1.92%)	40 (76.92%)	51 (98.08%)	27 (51.92%)	26 (50%)

Table 7: Comparison of Functionals Number between All and Selected Features

As shown in Table 7, 50% features composed of max, min, amean, skewness and kurtosis are selected as the training features. More than 80% features composed of range,stddev and linregc2 are selected. When it comes to maxPos, minPos and linregc1, almost no features related to them is selected.

Therefore, it means features composed of range,stddev and linregc2 have high probability to be different between HHI and HCI samples. However, there is no significant difference for features related to maxPos, minPos and linregc1.

3.3 Recognition

3.3.1 Initial Experiments

As discussed before, the key step of addressee identification is actually a classification of the utterance. For these experiments, the computer would be considered as the addressee if the audio file is classified as HCI. In contrary, the computer would not be considered as the addressee if the audio file is classified as HHI.

3.3.2 Classifier Selection

At first, the Naive Bayes algorithm is used to build the classifier. A dataset consisted of clean speech from all instances is used for training and test. For an overall understanding of the performance, the test dataset is extracted from it by 10-fold cross-validation. As discussed before, the features used for the training is composed of 700 features, which are selected according to their ranking score. After training and test with WEKA, the result confusion matrix is shown below:

		Prediction	
		HHI	HCI
Target	HHI	20256	959
	HCI	216	3938

Table 6: Confusion Matrix for Naive Bayes

From the confusion matrix, the Recall, Precision and F1-measure are calculated, and shown as below:

Class	Precision	Recall	F1-measure
HHI	0.989	0.955	0.972
HCI	0.804	0.948	0.870
Weighted Avg.	0.959	0.954	0.955

Table 7: Performance Matrix for Naive Bayes

Secondly, the Random Forest algorithm is used to build the classifier. All instances are also used for training and test with 10-fold cross-validation. The confusion matrix is shown below:

		Prediction	
		HHI	HCI
Target	HHI	21202	13
	HCI	72	4082

Table 8: Confusion Matrix for Random Forest

From the confusion matrix, the Recall, Precision and F1-measure would be calculated, and shown as below:

Class	Precision	Recall	F1-measure
HHI	0.997	0.999	0.998
HCI	0.997	0.983	0.990
Weighted Avg.	0.997	0.997	0.997

Table 9: Performance Matrix for Random Forest

At last, the Support Vector Machine algorithm is used to build the classifier. All instances are also used for training and test with 10-fold cross-validation. The confusion matrix is shown as below:

		Prediction	
		HHI	HCI

Target	HHI	20966	249
	HCI	39	4115

Table 10: Confusion Matrix for Support Vector Machine

From the confusion matrix, the Recall, Precision and F1-measure would be calculated, and shown as below:

Class	Precision	Recall	F1-measure
HHI	0.998	0.988	0.993
HCI	0.943	0.991	0.966
Weighted Avg.	0.989	0.989	0.989

Table 11: Performance Matrix for Support Vector Machine

With the SVM classifier, 25081 instances are correctly classified, and 288 instances are incorrectly classified. Therefore, the overall correctness rate of the model is 98.86%.

For a better comparison, the average performance shown below:

Class	Naive Bayes	Random Forest	SVM
Precision	0.959	0.997	0.989
Recall	0.954	0.997	0.989

F1-measure	0.955	0.997	0.989

Table 12: Comparison of Average Performance Metrics of All Algorithms

It should be mentioned that, the metrics are generated by WEKA, and they represent the average number if cross-validation is used. WEKA does not report the standard validation of the number, therefore it is not attached.

From the performance matrix above, it is clearly shown that, Random Forest outperform Naive Bayes and SVM in every metrics. The reason can be derived from the characteristic of both the training dataset and the algorithm itself.

The features dimensionality and sample size of dataset play as an important role in the determination of classifier performance [16]. It has been shown that, if the features dimension is much larger than sample size, the SVM is found to be a better choice. In contrary, Random Forest usually outperforms SVM in face of dataset with large sample size because of its higher complexity [17]. In this case, the sample size is much larger than the features dimensions of the training dataset. Therefore, random forest is a better choice.

Naive Bayes Classifier by its nature has high bias and low variance. It can serve as the baseline accuracy score when working with limited data

[18]. However, when the dataset turns to be large, its performance is not guaranteed. Therefore, it is not as powerful as random forest in this case. So, I will choose random forest algorithm in the following experiments.

3.3.3 Performance Comparison for Gender

In order to find the characteristic difference between two genders and their influence over the model's performance, various experiments are implemented.

At first, the model is trained by instances from all subjects, and the instances are grouped by the subjects' gender, and taken as test set respectively. Secondly, the grouped instances are used as both training set and test set for 10-fold cross-validation. Thirdly, the model trained with instances of one gender is tested with the opposite gender.

Training Set	Test Set	Precision	Recall	F1-Measure	Correctness
all	male	1	1	1	1
all	female	1	1	1	1
male	cross validation	0.996	0.996	0.996	0.9959
female	cross validation	0.997	0.997	0.997	0.9965

male	female	0.997	0.997	0.997	0.997
female	male	0.997	0.997	0.997	0.9965

Table 13: Performance Matrix for Gender Comparison

The Table 13 has shown that, the performance of the classifier is competitive. All instances in the test set for different gender have been correctly classified. When the test set is constructed by the Cross-Validation, some error was made by the model. It is because the test set is not included in the training dataset.

It has been found that male and female have different vocal fold length and larynx source strength. As a result, female radiate at higher frequencies [19]. For most frequency related features, there is no neglectable difference between male and female. Therefore, the performance of classifier trained with instances of single gender dropped when test against another gender.

3.3.4 Performance Comparison for Age

In order to find the characteristic difference between youngsters and elderly and their influence over the model’s performance, various experiments are implemented.

At first, the model is trained by instances from all subjects, and the instances are grouped by the subjects’ age, and taken as test set respectively. Secondly, the grouped instances are used as both training set and test set for 10-fold cross-validation. Thirdly, the model trained with instances of the same age is tested with the others.

Training Set	Test Set	Precision	Recall	F1-Measure	Correctness
all	old	1	1	1	1
all	young	1	1	1	1
old	cross validation	0.997	0.997	0.997	0.997
young	cross validation	0.996	0.996	0.995	0.996
young	old	0.991	0.991	0.991	0.998
old	young	0.992	0.992	0.992	0.992

Table 13: Performance Matrix for Age Comparison

As shown in the table above, all instances in the test set for different age have been correctly classified. When the test set is constructed by the Cross-Validation, some error was made by the model. It is because the test set is not included in the training dataset.

When comparing the voice from young and old subjects, former research [20] has shown that, of all spectral characteristics, the first and second harmonic differ significantly between them. Therefore, the performance of classifier trained with instances of old/young dropped when test against the other group.

3.3.5 Performance Comparison for Normalization

Experiment is implemented to find the influence of the feature normalization over the model’s performance.

Training Set	Test Set	Precision	Recall	F1-Measure	Correctness
norm all	cross validation	0.998	0.998	0.998	0.9975
all	cross validation	0.997	0.997	0.997	0.9966

Table 14: Performance Matrix for Normalization Comparison

Usually, data normalization increases the performance of the model by reducing the scale difference of the features. However, the performance doesn’t improve significantly in this circumstance. Because there is not so much difference in the scale of the original features this time. Therefore, both the model showed similar outcome in the final result.

4 Conclusion

In this thesis, a new method to detect the conversation addressee is developed and validated. First, at the data preparation phase, the instances are manually labelled with help of labelling tool “ikannotate2”. Then a set of low level features is extracted for each instance of the dataset with the help of OpenSMILE. In order to distinguish the acoustic difference between HHI samples and HCI samples, a statistic test is designed and performed on the extracted features. In the next step, a ranking score is constructed for each feature, only the most significant features are afterwards used for the recognition experiments. Afterwards, various machine learning algorithms are used to train the classifier model with the selected features. At last, comparisons between different sample groups are performed and analyzed. Additionally, the impact of the feature normalization is also analyzed.

The result of statistic test has shown that, the features composed of LLDs like PCM, MFCC and line spectral frequencies, are very likely to be different between HHI and HCI samples, and the features composed of functionals like range and stddev, are almost certain significant different between them. Based on the test result and effect size of the test, a ranking score of 7.2 is set as the threshold. Then 700 features are selected as the training set for the later usage.

The recognition experiments take random forest as the classifier algorithm. Samples from subjects, who is various in age and gender, have been used in the training and test process. The experiment results have shown that, this new developed method has pretty good performance on the given dataset. The overall correctness rate reaches 0.9975. When it is compared with former work [1], which has 0.92 as its correctness rate, this method shows significant improvement. However, its performance is based on limited dataset. If the dataset is more filled with noise, or if the speed of the speech goes faster, how will this method perform? Therefore, a further research should be done to examine its performance when applied in more generalized circumstances.

5 References

[1]. Andrew Liptak. "Amazon’s Alexa started ordering people dollhouses after hearing its name on TV",The Verge. Retrieved from:https://www.theverge.com/2017/1/7/14200210/

[2]. Katzenmaier, Michael, Rainer Stiefelhagen, and Tanja Schultz. "Identifying the addressee in human-human-robot interactions based on head pose and speech." *Proceedings of the 6th international conference on Multimodal interfaces*. ACM, 2004: 144-151.

[3]. Martinovski, B., Traum, D., Robinson, S., & Garg, S. "Functions and patterns of speaker and addressee identifications in distributed complex organizational tasks over radio." *Diabruck: seventh workshop on semantics and pragmatics of dialogue*. 2003:7-8 .

[4]. Rösner, Dietmar, Jörg Frommer, Rico Andrich, Rafael Friesen, Matthias Haase, Manuela Kunze, Julia Lange, and Mirko Otto. "LAST MINUTE: a novel corpus to support emotion, sentiment and social signal processing." *4th International workshop on corpora for research on emotion sentiment and social signals—ES3. ELRA*. 2012:82-89.

[5]. Ingo Siegert, Andreas Wendemuth. "ikannotate2 – A Tool Supporting Annotation of Emotions in Audio-Visual Data*".* Elektronische Sprachsignalverarbeitung 2017

[6]. Saunders, John. "Real-time discrimination of broadcast speech/music." *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*. Vol. 2. IEEE, 1996: 993-996.

[7]. Eyben, Florian, Martin Wöllmer, and Björn Schuller. "Opensmile: the munich versatile and fast open-source audio feature extractor". *Proceedings of the 18th ACM international conference on Multimedia*. ACM, 2010: 1459-1462.

[8]. Young, Steve J., and Sj Young. “The HTK hidden Markov model toolkit: Design and philosophy”. University of Cambridge, Department of Engineering, 1993.

[9]. Logan, Beth. "Mel Frequency Cepstral Coefficients for Music Modeling*". ISMIR.* 2000:12-13.

[10]. McLoughlin, Ian Vince. "Line spectral pairs*". Signal processing* 88.3 (2008): 448-467.

[11]. Soong, Tsu T. “Fundamentals of probability and statistics for engineers”. John Wiley & Sons, 2004: 139-148.

[12]. Nakagawa, Shinichi, and Innes C. Cuthill. "Effect size, confidence interval and statistical significance: a practical guide for biologists." *Biological Reviews* 82.4 (2007): 591-605.

[13]. Wu, Xindong, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J. McLachla. "Top 10 algorithms in data mining." *Knowledge and information systems* 14.1 (2008): 1-37.

[14]. Schuller, Björn, and Anton Batliner. *Computational paralinguistics: emotion, affect and personality in speech and language processing*. John Wiley & Sons, 2013.

[15]. Cohen, Jacob. "Statistical power analysis for the behavioral sciences Lawrence Earlbaum Associates." *Hillsdale, NJ* (1988): 20-26.

[16]. Statnikov, Alexander, Lily Wang, and Constantin F. Aliferis. "A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification." *BMC bioinformatics* 9.1 (2008): 319.

[17]. Wu, Xindong, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh. "Top 10 algorithms in data mining." *Knowledge and information systems* 14.1 (2008): 1-37.

[18]. Kelleher, John D., Brian Mac Namee, and Aoife D'Arcy. "Fundamentals of Machine Learning for Predictive Data Analytics." (2015): 322.

[19]. Titze, Ingo R. "Physiologic and acoustic differences between male and female voices." *The Journal of the Acoustical Society of America* 85.4 (1989): 1699-1707.

[20]. Decoster, Wivine, and Frans Debruyne. "The ageing voice: changes in fundamental frequency, waveform stability and spectrum." *Acta oto-rhino-laryngologica belgica* 51.2 (1996): 105-112.