MASTER THESIS

Automatic speech recognition system for neurological tests

Institut de Recherche en Informatique de Toulouse,
Université Paul Sabatier,
118 Route de Narbonne, Toulouse France

From 2020-01-20 to 2020-07-17

Written by:

Lila GRAVELLIER

Supervisors:

Julien PINQUIER, Maître de Conférences
Jérôme FARINAS, Enseignant chercheur

Academic Supervisors:

NTNU: Torbjørn SVENDSEN
Phelma: Cornel IOANA

Grenoble INP Phelma, 2019-2020
NTNU, 2019-2020

Signal Image Communication Multimédia Promo 2020
Electronic System Design Master of Science

Abstract

Abstract The Evolex project consists of developing automated tools for medical tests. For a lot of neurological tests, doctor and speech therapists have to write down the answers given by their patients and time them. A software could replace this tedious task and open up new perspectives in terms of analysis. Medical professionals gathered many responses to three different tests, to help the IRIT to develop a performant software. The goal of this master thesis is to build the Automatic Speech Recognition system for these three tests using State of The Art techniques. After a study of the specific requirements from the doctors, we create the different blocks of an ASR system on the Kaldi toolkit: the acoustic models, the language model and the lexicon. By testing on the Evolex corpus, we optimize the system and adapt it to the target use. We obtain satisfactory results and implant the three systems in a user-friendly software for the medical professionals.

Résumé Le projet Evolex est né pour rendre service aux docteurs et orthophonistes. En effet, de nombreux tests pour détecter des pathologies neurologiques nécessitent d'écouter de nombreuses réponses de patients, de les noter et de les chronométrer. Ces tâches difficiles peuvent être remplacées par un logiciel et ouvrir des perspectives en terme d'analyse. Des médecins ont monté le projet avec l'IRIT, et leur ont fourni de nombreux enregistrements de sujets aux différents tests. L'objectif de ce stage est de créer un système automatique de reconnaissance de la parole utilisant des techniques issues de l'état de l'art actuel, pour transcrire directement les enregistrements des patients, et analyser leurs résultats. Pour cela nous avons étudié les réponses au test, et nous avons ainsi pu créer les différents blocs d'un système de reconnaissance vocale : les modèles acoustiques, le modèle de langage et le lexique. Nous avons pu tester notre système sur le corpus créé par les médecins, et ainsi le valider. La reconnaissance vocale est ainsi intégrée dans un logiciel plus complet de passation des tests destiné aux professionnels de la santé.

Acknowledgements

I would like to express my sincere thanks to my supervisors Julien Pinquier and Jérôme Farinas, and also Jim Petiot, who has worked with me on the project, for this great internship. I have been only two months at the institute before the Covid19 crisis, but I would really like to stay with them after my internship. They are welcoming, supporting and motivating people, and I really enjoyed working with them. Thanks to all the other members of the SAMoVA Team, for the global atmosphere, the games at lunch breaks, and all the good times. I would also like to thanks my supervisor at NTNU, Torbjørn Svendsen, who led me to discover the speech recognition in practice, and made me use Kaldi for the first time. I think I owe him my recruitment for this internship !

Introduction

In the medical sector, for neurological conditions, a part of the evaluation consists of testing the lexical access of patients. Thanks to data gathering for years and years, psycho-linguistics have obtained standards for some of these tests, and now can evaluate the patients on their answers to the different tests. For instance, one of the well-known tests is the verbal fluency [4], which asks the patient to name as many words as possible in a limited time that respects one rule, for example, words which begin with the letter V. The downside of these kinds of tests is that they are tedious tasks for doctors, they have to write down every word said by the patient for a long period, and also obtain different times about the patient's reactions. This difficulty leads to approximate results, and sometimes a loss of time for the doctors in hospitals. These two main remarks have led to a common conclusion: if the test could be automatised, the results would be more accurate, with more information and could show patients progress from one test to another [3]. Moreover, it could turn this tedious task in a simpler one, and leave doctors able to focus on the patient. Another interesting aspect of the automation of these tests would be to define more precisely the standards and then having the opportunity to measure the illness of patients, which is not obvious in some cases. For example, after strong treatment like chemo against cancer, concerned individuals encountered troubles in their daily life conversation, but their issues could not be identified as a disease, while it can be very difficult to live with. The automation of the tests could measure and quantify the lack of neurological capacities of patients.

In 2013, the project Evolex is born with the collaboration of the Toulouse NeuroImaging Center (ToNIC) which represents the medical part of the project and the team Structuration, Analysis, Modeling of Video and Audio documents (SAMoVA) of the Institut de Recherche en Informatique de Toulouse charged with the automation and processing of the tests (see figure 1). A private partner, COVIRTUA, has joined the project recently to develop a real product usable on a large scale, and possibly marketable.

Figure 1: Three partners of the Evolex Project

When I arrived in the IRIT institute, the project has been through two versions [1], [2]. The SAMoVA team has first developed a version with a simple automatic speech recognition (ASR) system able to give the transcription of the speech and to give information about the reaction time of patients. As the system was not optimal, the speech scientists or doctors who used the tool had to correct the automatic transcription. However, a web interface has been created to do the correction task, to help them.

This first version has allowed the health professionals to record the answers of many subjects with no language disorders for different tests. This first experiment enabled us

to gather the data about the different possible answers to the tests. All these information are crucial first to improve the automatic transcription of the recordings, and then to establish the standards of the tests. Moreover, the decisions about what answer is judged as correct or incorrect were taken by the health professional.

In this context, I was recruited to develop a new automatic speech recognition system for the tool with better performances, by using the gathered data from the first versions. Knowing the characteristics of the tests, which means the possible answers, the noise, the difficulties, mistakes and so on, enables us to create systems specifically adapted to each different test.

After giving some precision about the task and the needs of the project in the first section, the tools and method accordingly chosen to build the new automatic speech recognition systems are detailed in the second one. Then, the last section is an evaluation of the work, with a discussion between the previous versions and the one resulting from this internship.

Contents

1 Specifications and needs for the project **7**
1.1 Description of the selected tests . . . 7
1.2 User interface . . . 7
1.3 Requirements . . . 8
1.4 Project management . . . 8

2 Tools and methods **10**
2.1 Speech recognition toolkit: Kaldi . . . 10
2.2 Acoustic models . . . 10
2.2.1 Training Corpus: Common Voice . . . 11
2.2.2 Training process: Kaldi recipe . . . 13
2.2.3 Model choice: TDNN Chain . . . 14
2.3 Lexicon . . . 15
2.3.1 MHATLex Tool . . . 15
2.3.2 Use and adaptation . . . 15
2.4 Language Model . . . 16
2.4.1 N-gram language models . . . 17
2.4.2 Unigram and optimization . . . 17
2.4.3 Language models for the fluency task . . . 17
2.4.4 Language models for the denomination and the generation tasks 18
2.5 Output processing . . . 18
2.5.1 Confidence score . . . 18
2.5.2 Validity . . . 19

3 Evaluation and results **20**
3.1 Corpus . . . 20
3.2 Results . . . 20
3.2.1 Word Error Rate . . . 20
3.2.2 Fluency . . . 20
3.2.3 Denomination . . . 22
3.2.4 Generation . . . 23
3.3 Discussion . . . 23

4 Conclusions **25**

List of Figures

1 Three partners of the Evolex Project . . . 3
2 Main page of the Evolex tool inside the Covirtua interface . . . 8
3 Global view of the Evolex equipment and functionning . . . 8
4 Gantt Diagram. The "O" mark the expected work over time, whereas the colors indicate the executed work. . . . 9
5 The automatic speech recognition system for the Evolex project . . . 10
6 Gender and accent statistics of the French Common Voice corpus . . . 11
7 Fundamental frequency statistics depending on the gender . . . 13
8 The word "cat" described with monophones . . . 14
9 The word "cat" described with triphones . . . 14
10 Lexgen execution file . . . 16
11 Example of graph resulting from the decoding of the sentence "relax rich rage reality" . . . 19
12 Example of output for the denomination task. Columns: audio path, beginning of the word in seconds, duration in seconds, transcription, validity . . . 19
13 WER of the four fluency task depending on the minimum confidence score, all the words below the minimum confidence score are removed . 21
14 Evolution of inserted words and WER without insertions depending on the minimum confidence score . . . 21
15 WER results for each stimuli of the denomination task with insertion deletion and substitution repartition . . . 22
16 WER results for each stimuli of the generation task with insertion deletion and substitution repartition . . . 23

List of Tables

1 Statistics on the French Common Voice corpus fr_412h_20191210 . . 12
2 Acoustic models comparison on the Common Voice testing data . . . 15
3 WER of the four fluency tasks depending on the language model . . . 20
4 WER of the four fluency tasks with minimum confidence score of 0.59 and with te Evolex_LM . . . 22
5 WER of the four fluency tasks with Evolex 1 and the new Evolex system 24

1 Specifications and needs for the project

1.1 Description of the selected tests

Three main tests have been selected by the doctors to have the first feedback of the software. The three tests are called generation, denomination and verbal fluency. All these tests are shortly described below.

Word Generation: The patient receives a spoken word as a stimulus, for example, the word "cat" is pronounced, then the patient has to say a word linked to the given one, but not the same. The patient could say "dog" after hearing the word "cat". This exercise is repeated several times, and the patient has to answer as quickly as possible to each stimulus.

Picture Denomination: Patients receive a visual stimulus, a candid picture with only one or two objects, and have to name what they see on it. For example, if a picture of a cat appears on the screen, the patient should say "cat" as quickly as possible .

Verbal fluency: For this task, the patient has to name as many words as possible that he knows to respect a given rule. Two kinds of rules can be given: a semantic one, consist of naming words of a category like animals, sports and fruits, for example. The other kind of rule is a phonemic one, and consist of giving words which begin with the same letter. For example, if the rule is "words beginning with B" the patient could answer "Boat, Bell, Beer, Blood...". The most prevalent version of this task is to give words for one minute or two minutes.

For the Evolex project, the three tests need an automatic speech recognition system. There are 60 audio stimuli for the word generation test, 60 visual stimuli for the denomination test, and four different verbal fluency which are: words which begin by "V", words which begin by "R", fruits and animals. All along with the report, we will talk about denomination, generation, R fluency, V fluency, animals fluency and fruits fluency.

Important remark: All the project is in French, and exclusively for the French language.

1.2 User interface

The private partner of the project, Covirtua, has already developed different online tests for medical purposes, but without speech recognition. They created a user-friendly interface for the doctors and their patients. For the Evolex project, they gave to the IRIT an empty code block inside their tool to easily access the functionalities such as the profile creation, the different modules and visuals for the tasks and others utilities.

Functioning The figure 2 shows the main page of the Evolex tool in the Covirtua software. The tool needs two screens: one for the doctor, and the other for the patient. The doctor creates a profile for the patient and then can choose one of the three tests available: fluency, denomination or generation.

For example, in the case of the denomination, the doctor can choose the list of stimuli he wants. Then, he launches the task, and the first picture appears. The recording starts at the same time, and once the patient answers, the doctor can click to get to the second stimuli. The recording is then ended, and a new one starts. At

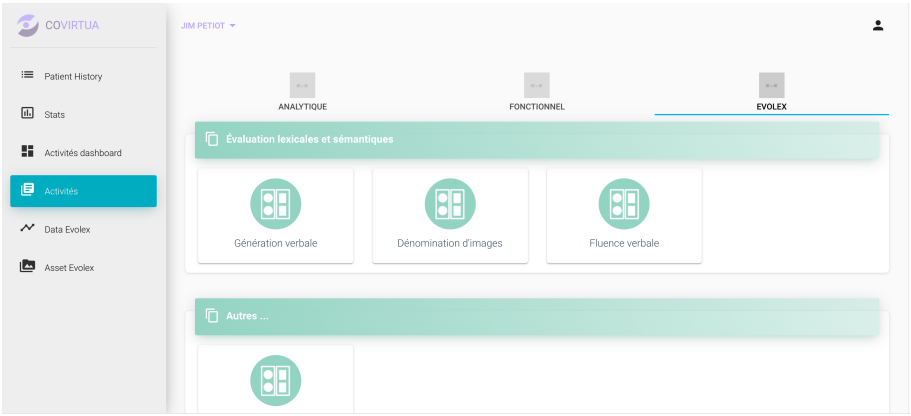

Figure 2: Main page of the Evolex tool inside the Covirtua interface

the end of the test, the results are given in a synthesis page and saved in the database. The other tasks works quite the same way.

1.3 Requirements

The medical professionals of the project established the different needs for their diagnosis. The transcription of the answers and the reaction time are the most important ones. Then, the software has to distinguish valid answers to the wrong ones. Further analysis has to be done on the answers, but they are out of my work. The medical context of this project implies to respect confidentiality. All the data is stored in a dedicated confidential server, hosted at the IRIT. During the tasks, the recordings of the answers are sent to the server where they are processed to give the summarized results on the user interface. The automatic speech recognition system and the data analyser must process in a reasonable time, and be able to work with many users at the same time. The figure 3 describes the global system of the Evolex project.

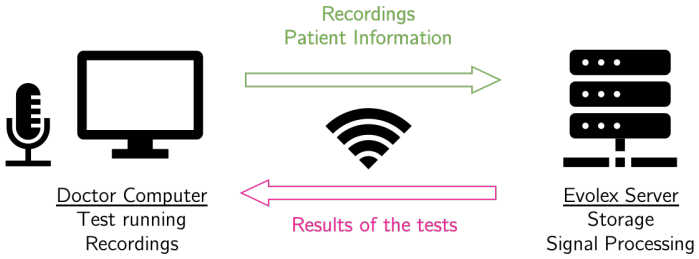

Figure 3: Global view of the Evolex equipment and functionning

1.4 Project management

To meet these goals, I had 25 weeks. Once a week I had a meeting with my supervisors, to submit my work of the week. For each meeting, I had to prepare a Google slide that they could access whenever they want to comment on it. I have now a huge google slide which describes every week of my internship. By keeping track of my work, I could compare old results, be sure of what I had already done, and what I should do. Naturally, it was even more helpful for the writing part.

The Figure 4 describes the expected and executed word over time. The main tasks are coloured in the diagram. Most of these tasks are described below in the report. The interesting point of this diagram is that the acoustic model and the language model took the same time. In the expected plan, the acoustic model should have taken three more weeks. The acoustic may seem the most important thing in speech recognition because we do audio processing. Actually, the language model is crucial. A bad language model can lead to 100% of errors, even if we obtain the best acoustic models. All the other tasks were on time. However, we have to keep in mind that many wrong ways have to be taken to find the good one, and this is why all the tasks take much time.

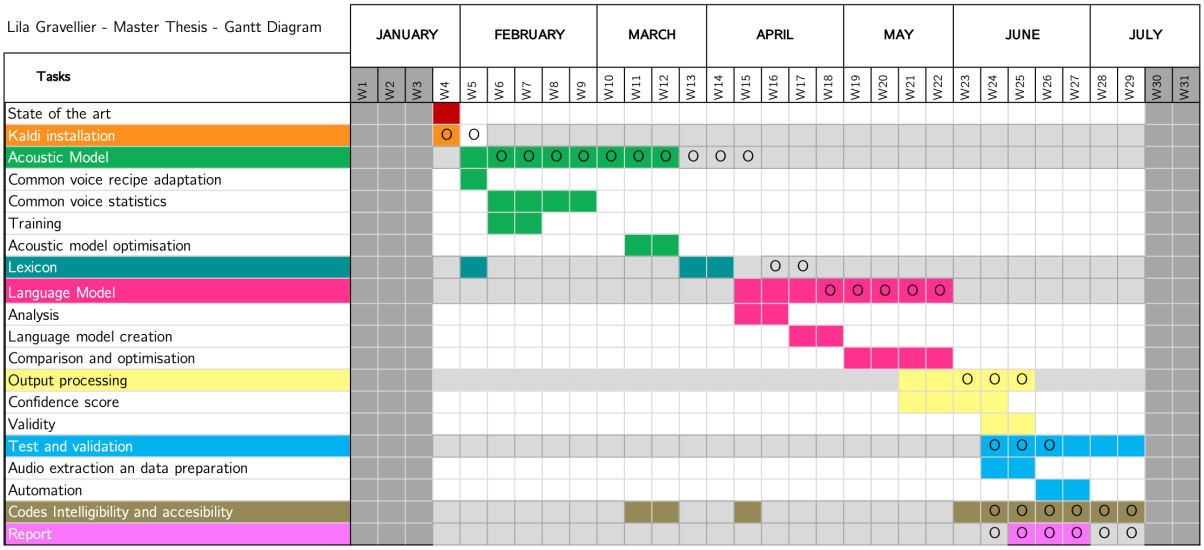

Figure 4: Gantt Diagram. The "O" mark the expected work over time, whereas the colors indicate the executed work.

Remark In this project, I was alone for the automatic speech recognition system building. My supervisors were advising me every week on my work, but **I am the author of all the work described below**. The web development and formatting was not included in my goals, all these part has been done by Jim Petiot, an IRIT engineer.

2 Tools and methods

This section describes the tools and methods used to fulfil the requirements explained in the last one and more precisely the built of the Automatic Speech Recognition (ASR) system inside the Evolex server of the figure 3. The input of an ASR system is an audio recording, and in our context, the output is the transcription of the speech. The first step in speech recognition is to extract some features from the audio recording. Then, the features are analysed to extract a hypothetical transcription. This process of decoding requires three main blocks: acoustic models, a lexicon and a language model. These different steps are summarised in the block diagram 5. This section will detail the function of each block and how they were built in our work.

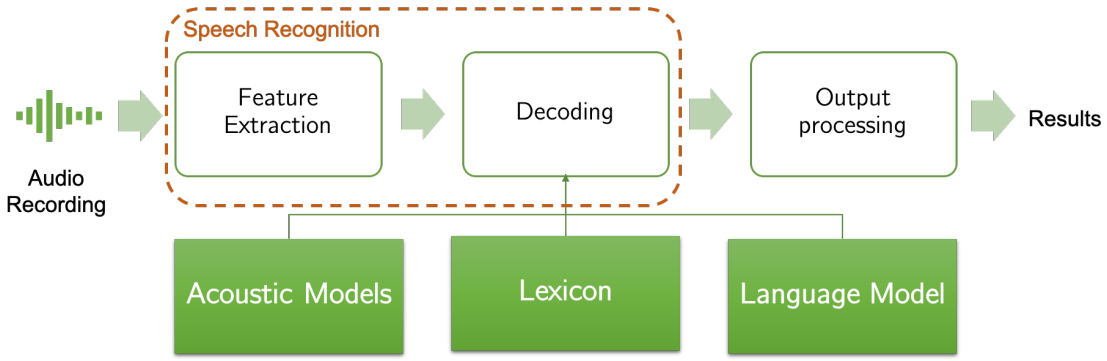

Figure 5: The automatic speech recognition system for the Evolex project

2.1 Speech recognition toolkit: Kaldi

Kaldi is an open source toolkit for speech recognition mainly written by Daniel Povey [6]. It contains a lot of speech tools written in C++, and it's a completely modular toolkit, designed for researchers. A great advantage of this toolkit is that it offers a huge list of recipes to use the famous corpus from the Linguistic Data Consortium. A recipe is a folder with all the scripts needed to create an automatic speech recognition system based on a specific corpus. These examples enable to better understand the possibilities of the toolkit, and to globally comprehend how it works, even if there are a huge number of codes inside. We chose this toolkit because we wanted to use a corpus named Common Voice which has its own recipe on Kaldi. Moreover, Kaldi is in constant evolution and offer the last speech recognition finding, such as Deep Neural Networks models.

2.2 Acoustic models

Acoustic models enable the system to recognise one phoneme from another given a feature vector. A phoneme is the smallest unit of sound to distinguish one word to another. There are 36 phonemes in the French language. For example, the phoneme transcription of the word "café" in French would be "/k/ /a/ /f/ /e/", with four different phonemes. To build an acoustic model for our task, we need to learn the

characteristics of each of these 36 phonemes. An analysis of many occurrences of the same phoneme enables us to create a model of the phoneme. The occurrences have to be in a different context, or from different speakers, in order to minimise the bias. There are several ways to learn the characteristics of phonemes, we will describe the main principle of the learning process further below. First, we need many occurrences thanks to a transcribed corpus, and then we need to train a machine learning system on the corpus. We finally test the acoustic model and adjust the parameters for the best performances.

2.2.1 Training Corpus: Common Voice

The Common Voice corpus is an open-source data set of voices gathered thanks to crowdsourcing. This is Mozilla's initiative `https://voice.mozilla.org/fr`. Everyone can participate by going on the website and recording their voice or listening to other people's voices. Mozilla give the speaker some sentences to say, and the audio recordings are judged by other users of the website, two downvotes lead to the deletion of the record, whereas two upvotes lead to its incorporation in the corpus. This system enables the academic word to gather a lot of free data in many languages. Even if the biggest data set is the English one from the beginning, other languages start to reach many hours of speech, for the happiness of international researchers.

The interesting aspect of the Common Voice corpus for our context is the audio quality of the recordings. The use of our ASR system will be in a medical environment, with the computer equipment in situ. This implies a lot of variation among microphones, background noise, distance speaker-microphone. As the Common Voice corpus has been created with at least as many microphones as speakers, we will get an acoustic models close to our target use of the ASR system. As we are working on a system for French tasks, we will use the French Common Voice corpus.

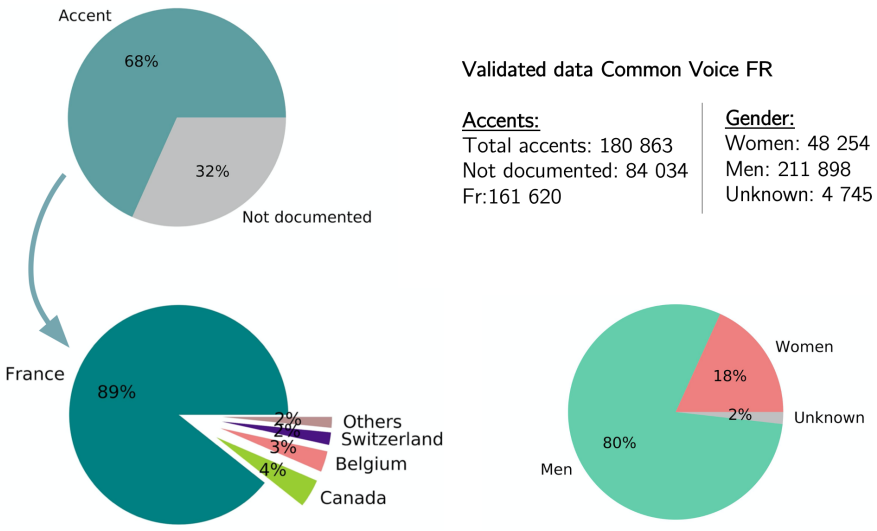

Figure 6: Gender and accent statistics of the French Common Voice corpus

To take part in the Common Voice project, web users can create an account and give some information about their gender, age and accent. But if they don't create an account, they still can record their voices and this lead to data without any information about the speaker. The following part is a statistic analysis about the Common Voice

corpus when we started to use it. The version is $fr\_412h\_20191210$ with 350 validated hours of audio recordings.
We want to characterise better the corpus. All the results are in the figure 6 and the table 1. The following paragraphs explain the procedure to obtain them.

Table 1: Statistics on the French Common Voice corpus fr_412h_20191210

Validated data	Mean	Std	Min	Max
Speech duration by utterance (s)	3.13	1.43	0.28	14.58
Speech duration by speaker (s)	347	1411 (23min)	0.74	3080 (8.5h)
SNR (dB)	53.8	14.4	2.26	91.1

Speech Duration As the speakers are recording themselves, they have to press the stop button when they are done, and this leads to a lot of useless recording time without speech. To evaluate the real time of speech, we analysed the corpus with the tool Web RTC Voice Activity Detection (VAD) from Google `https://pypi.org/project/webrtcvad/`.The tool is free, precise and really easy to use with a Python script. We get the time of speech in each utterance and can calculate the real speech duration of the corpus.

Speech Duration by Speaker Another interesting aspect is the speech duration by speaker. In fact, a big standard deviation means that the corpus is unbalanced, because some speakers are representing too much of the data from the corpus. This is calculated by adding the audio duration for each speaker, if we know the speaker (non anonymous).

Signal to Noise Ratio We used the SOX Linux tool to determine the SNR of each utterance `https://doc.ubuntu-fr.org/sox`. The command "sox FICHIER.wav -n stats" gives a lot information on a audio file. The SNR is obtained by doing the following calculation from the sox data: SNR = |RMS Tr dB - RMS Pk dB| .

Gender For all the validated data, 53 846/ 263 796 utterances are defined as unknown gender, thus 20% of the corpus. In order to have a greater idea of the gender distribution, I decided to create a gender classifier based on the fundamental frequency of speech. We extracted the fundamental frequency of speech of all the recordings thanks to the python library adapted from Praat: Parselmouth. The figure 7 shows the distributions of the six statistics for male (M) and female (F). We deduce from these graphs that the mean, the median and first quantile are good statistics to separate the two genders. We separated the data into train and test sets to build a neural network. The training part is done thanks to the six different characteristics: mean, median, standard deviation, first quantile minimum and maximum. The precision of the classifier is 93%. With this classifier, we obtained a better idea of the gender distribution.

Accent Accents of speakers have been listed and shaped in a concise graph. The French accent from France represents a large proportion which is better for our target use of the ASR system. However, it is interesting to train our model with variability

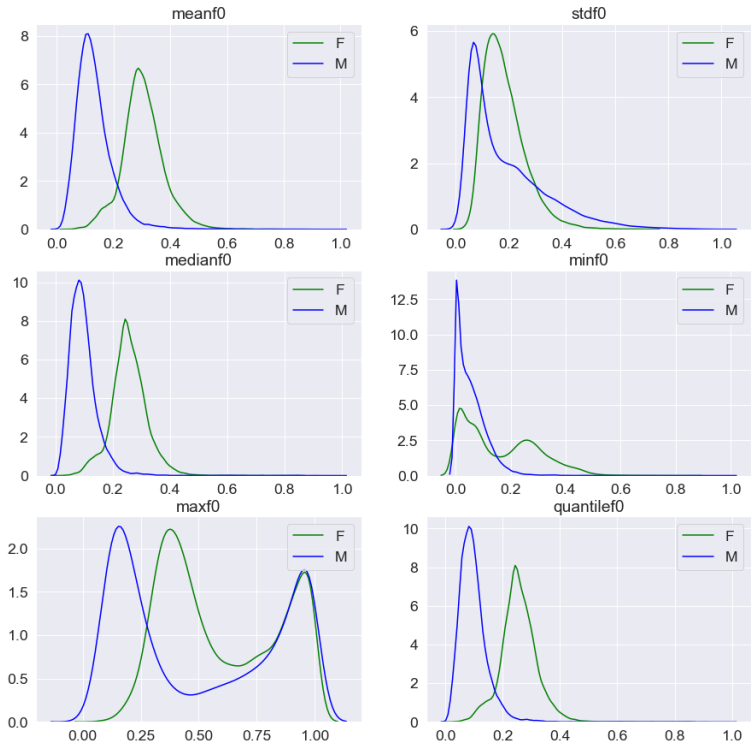

Figure 7: Fundamental frequency statistics depending on the gender

within the accents, to avoid a complete failure with the patients speaking with strong local accents.

2.2.2 Training process: Kaldi recipe

We will not detail the functioning of each speech tools used in this process because it needs a lot of background and context unnecessary in this report.

Feature extraction We want to learn the characteristics of phonemes to create an acoustic model. We first need to convert our audio recordings from the Common Voice corpus into features which give precision about the phonemes and can distinguish them from one and another. The most characteristic features in speech processing are MFCC, Mel Frequency Cepstral coefficients.

Once we extract the features, the Kaldi recipe script offers us several training processes with Gaussian Mixture Model/Hidden Markov Model framework. At the beginning of the training, random Gaussians are chosen to estimate each phoneme. During the training process, the transcription is aligned with the audio to determine the position of the phoneme in the audio. Then the Gaussians are corrected to better fit with the real phonemes.

Monophone training In this training process, each phoneme is independent to the context, which means that the phoneme /ʌ/ is evaluated with the same way in a word like "mum" (mʌm) than "up" (ʌp). The phoneme is separated into three parts, beginning, middle and end of the phonemes as depicted in the Figure 8

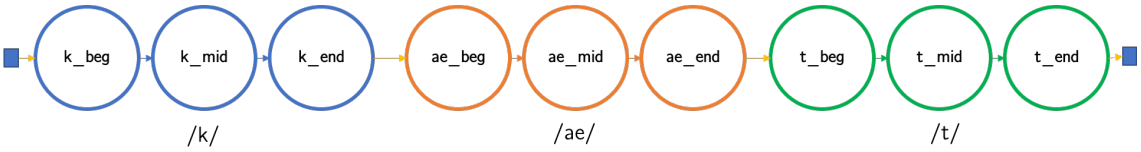

Figure 8: The word "cat" described with monophones

Triphone training Unlike monophones, the triphones take the context into account. The phoneme of mum /ʌ/ will also be divided in three parts but, this time the left and right sides of the phonemes will include the closest phoneme. In the Figure 9 we can now see that some blocks have changed, including the right and left phonemes.

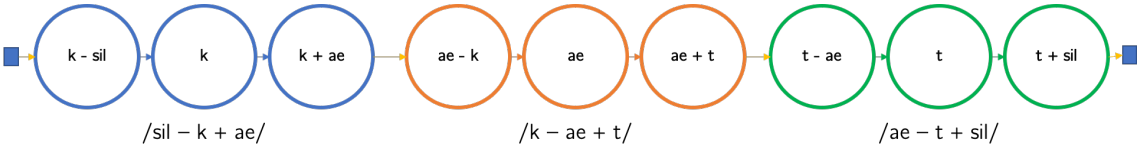

Figure 9: The word "cat" described with triphones

Additional triphone trainings We can improve the training process with some speech tools. The $\Delta + \Delta\Delta$ training uses more features than the basic model. It takes the delta and double delta features derivatives of the signal) in addition to the MFCC. The LDA-MLLT (Linear Discriminant Analysis – Maximum Likelihood Linear Transform) takes into account the speakers. It adapts the models with parameters estimated on each speaker. The SAT (Speaker Adaptive Training) tries to normalise the features depending on the speaker to get more standardised data.

TDNN Chain Training Deep neural networks are increasingly used for speech recognition. In fact, they give outstanding results, and the existing computer equipment enable researchers to use them with short calculation time. A Time Delay Neural Network (TDNN) model has a specific topology, and has the particularity to take time dependency into account. The network keeps in memory the context (before and after the current input) of the feature to better model the phonemes. The "Time Delay" is the size of the context taken into account for each input.

2.2.3 Model choice: TDNN Chain

After training the different models on the training data, we can test them with the testing data, already split by Common Voice. Table 2 gives the results of all the models and the parameters selected for the training. The score Word Error Rate (WER) corresponds to the percentage of wrongly recognised words on the total number of words in the real transcription. The TDNN Chain model is the best model, with only 14.93% of errors. We will take this model for the Evolex project. As we don't need anymore to split the data of Common voice, for tests and train, we learn a new model on all the validated data of the corpus.

Table 2: Acoustic models comparison on the Common Voice testing data

Training Method	Iterations	Nb Gaussians	Leaves	WER (%)
Monophone	40	2000	-	41.89
Triphone $\Delta + \Delta\Delta$	35	30 000	2000	26.53
Triphone LDA+MLLT	35	35 000	2500	25.71
Triphone LDA+MLLT+SAT	35	50 000	2500	25.59
TDNN Chain	-	-	-	**14.93**

2.3 Lexicon

A lexicon is a pronunciation dictionary. It lists all the words and the corresponding sequences of phonemes for each one of them. Its first role in the automatic speech recognition system is to give all the phonemes available in the task. During the training process, it converts the word transcription of the training corpus in phoneme transcription in order to evaluate a model for each phoneme and not for each word. Finally, during the decoding phase, when the ASR system is used for the selected task, it enables the reconstitution of the words from the hypothetical sequences of phonemes.

2.3.1 MHATLex Tool

There are several ways to obtain a lexicon. Automatic generator of lexicons can convert grapheme (how the word is spelt) to phonemes. They are trained to learn the pronunciation rules of a specific language. This method is advantageous to generate a lexicon from any list of words but has a non-zero probability of error. Another way is to manually create a small lexicon with only the specific words of the task. However, in our case, the fluency tasks require a too extensive dictionary to do this by hand. The last option is to obtain a vast vocabulary lexicon already verified and corrected. The IRIT has contributed to a lexical resource project in 2001, so we have free access to the MHATLex lexicons [5], which are all verified and corrected. MHATLex contains many lexicons, but not only with the pronunciation of the words, but also a lot of characteristics fields, like gender, number, verb time, root of words and so on. The Lexgen tool enables us to extract from the existing lexicons all the information we need, which in our case are the spell of words and the phonological representation.

2.3.2 Use and adaptation

The Lexgen tool is a Perl script. The execution of the script needs a file with all the precision on the needed output. The figure 10 details the file to generate the lexicon we used for the ASR system. This execution creates 28 files: 26 files for words beginning by each letter of the alphabet, and 2 more files for words finishing with an apostrophe like "aujourd' " (beginning of today in French) and for those needing a dash like for the euphonic t "-t-elle" in the question "aime-t-elle?" (does she like?).

```
################################################################################
##
## Exemple de génération phonétique avec supression d'attributs
##
## Fichier paramètres MHATLex :
## * Génération d'un lexique phonétique       sous ./Tmp4
##   via le modèle MHAT simplifié           (sous ../MHATPhonRes/MHATSimp)
##   à partir du lexique phonotypique contenu sous ../Lexicons/MHATLexPhy
##
## NB1. Le lexique phonétique hérite du champ graphie (GRAPHIE),
##      du lexique phonotypique
## NB2. Le répertoire 'Tmp4' contenant le lexique en sortie doit exister
## NB3. L'option 'TriLexiqueOut: Oui' permet de trier (en éliminant les doublons)
##      chaque fichier constituant le lexique en sortie
##
################################################################################
RegleGPC:
RegleGPM:   ../MHATPhonRes/MHATSimp/mhatregle.simp.gpm
DirLexiqueIn:   ../Lexicons/MHATLexPhy/
LexiqueIn:  ?.M.phy,apostrophe.M.phy,clitic.M.phy
FormatIn:   GRAPHIE;CTXG;PHON;CTXD;CS
AlphabetIn: ../MHATPhonRes/MHATSimp/phy.simp.alf
DirLexiqueOut:  ./Tmp4/
LexiqueOut:
FormatOut:  GRAPHIE;PHON
AlphabetOut:    ../MHATPhonRes/MHATSimp/pro.simp.alf
TriLexiqueOut:  Oui
```

Figure 10: Lexgen execution file

To obtain a lexicon usable in Kaldi, we need to merge all the files in a single one. Moreover, in the MHATLex lexicons, the accents are replaced by numbers, so we decided to re-establish the accents to simplify the use in Kaldi and the coherence with the language model.

The obtained lexicon contains around 400 000 words. It may seem sufficient at first sight but many answers from the subjects we gathered are out of the lexicon. In fact, proper nouns are not in the MHATLex lexicon, nor than brand names, or name of new technologies in the last twenty years. As the system can't recognise a word out of lexicon, we need to update it to get the best performances. During the development process, encountered out-of-lexicon words have been added. However, to ensure a great evolution of this system with the next corpus, we need to easily add some words to the lexicon. Nevertheless, any format errors or use of the wrong phoneme in the phonological representation can break the whole process of recognition. To avoid any sabotage, we wrote a script which enables the administrator to add some words to the lexicon. The words and their phonological representation s are automatically checked and validated or not before their addition to the lexicon.

2.4 Language Model

The language model gives to the recognition system the occurrence probabilities of words or sequence of words. For example, in a more general context, if the ASR system has to recognise conversation speech, and the speaker says "My house has three floors". The acoustic model first analyses the audio data, and because this model is not perfect, two possible sentences compete: "My house has three floors"and "My mouse has three floors". To us, it seems evident that the first sentence is the good one, but it is because we have experienced in the English language, and we understand the meaning. Here we expect the language model to indicate that the second sentence is **less probable** than the first one and then decide to keep the first one. There are several ways to build such a language model, but we will explain only the most famous one: n-gram language models.

2.4.1 N-gram language models

The N-gram language model calculates the occurrence probability of a word given its history of $N-1$ previous words [7]. Let us say we have the sequence of words "I watched a" and we want to know what is the probability that the next word is "movie". We want to calculate the probability P(movie| I watched a); the history has three words, so this is a 4-gram probability. To obtain this probability, we need a big corpus of the English language, and we count how many times the sequence "I watched a movie" appears in the corpus. Then we count how many times the sequence "I watched a" appears in the same corpus. The result of:

$$\frac{Count(I\ watched\ a\ movie)}{Count(I\ watched\ a)} \tag{1}$$

will give the probability of this 4-gram. We can do the same operation for each 4-gram existing in English. In practice, the most common N-gram used is the trigram (3-gram). We will thus consider that a word depends only on the two previous words in a sentence. To calculate the global probability of a whole sentence, we multiply the probabilities of each trigram of the sentence. However, the first word of a sentence has no history, so we will also need to calculate bigram and unigram to do the whole calculation. For example, to compute the probability of the whole sentence "I watched a movie", we will do the following calculation:

$$P(I)P(watched|I)P(a|I\ watched)P(movie|watched\ a) = P(sentence) \tag{2}$$

Remark: If one probability within the multiplication has a zero value, the sequence probability is leading to zero. Thus, we must be careful and give a non-zero value to the unfounded sequences in the corpus.

2.4.2 Unigram and optimization

In our context, the answers of the subjects to the different tests are only isolated words. There is no such dependence between the words and their history. Thus, we cannot use trigrams or bigrams, but only unigrams. The probability of a word with the unigram method is given by:

$$\frac{Count(word)}{Count(vocabulary\ size)} \tag{3}$$

We cannot take an extensive vocabulary for a unigram language model. If we take the whole dictionary and calculate the unigram probability of each word, we will get a uniform law so that every word will have the same occurrence probability. Such a language model has zero effect on speech recognition. The objective is to select the possible words of the tasks, and then to give the best probability distribution between the selected words, to help the ASR system. To optimise the language model, we took different lists of words and probability distribution and compared them.

2.4.3 Language models for the fluency task

For the fluency tasks, the possible words are limited: animals, fruits, R words and V words. The first idea, to get an initial score, is to take only the allowed words and to give

them a uniform distribution, we will call it **Uniform_LM**. However, we have access to the answers given by around a hundred subjects during a data-gathering campaign from 2012 to 2014. We can use the answers to create a new language model with the words obtained in this campaign in addition to the words allowed by the task but not given by the subjects. We give to each word the probability calculated as following:

$$\frac{Count(occurrence\ word)}{Count(all\ answers + not\ given\ allowed\ words)} \tag{4}$$

This is the **Evolex_LM**. We also want to know if we can get better results with out-of-context probabilities. BRULEX is a lexical database which gives word frequency in movies and books. We used the word frequencies of the allowed words of the task to build the last language model: **Brulex_LM**.

2.4.4 Language models for the denomination and the generation tasks

For the denomination task, patients have only to give the name of the object in the picture. The possible words are very limited. For a few pictures, several answers are considered as correct. For example, in the picture "finger" ("doigt" in French), if the patient says forefinger ("index" in French), the answer is correct. The first thing to do is to put the correct answers in the language model. Then, as we have examples of wrong answers, we can add them to the language model to recognise them. If the word is not correct, it will be recognised or not but the system will indicate an invalid answer (see section 2.5.2).
For the generation task, the model language is more difficult to obtain. The patients indeed have to generate a linked word. However some links are complex to identify. The idea is to include all the answers of the subjects we have in the language model and to add the new answers in future versions. In the same way as for the evolving lexicon in the section 2.3.2, we enable the administrator to add some words to the language mode if new correct answers are given.

2.5 Output processing

With the acoustic model, the lexicon and the language model ready, we can analyse and decode the audio recordings and get the transcription and the time. But some output processing must be done to complete the result.

2.5.1 Confidence score

The tool will be used for a medical test; thus, the ASR system must be reliable not to corrupt the data and state nonexistent disease for the patient. A way to avoid this is to score the confidence of the system in its transcription. If the score is very low, the word detected is probably wrong, and then the doctor can listen to the audio and correct the transcription. To understand how a confidence score can be extracted from the ASR system, we must describe more precisely how the decoding process is done in Kaldi.

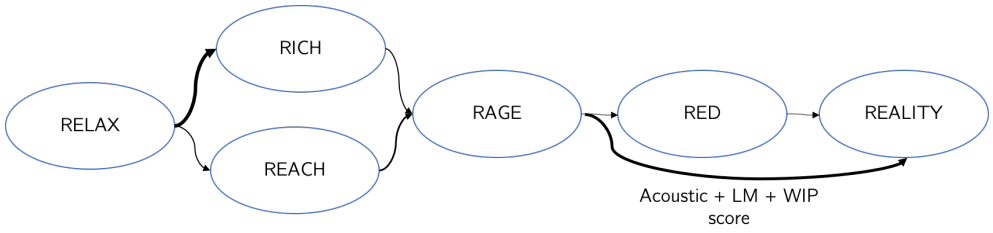

Figure 11: Example of graph resulting from the decoding of the sentence "relax rich rage reality"

The decoding process During the decoding phase, the trained acoustic model analyses the extracted features and find the phonemes models with the closest features. However, for one observed sound, several phonemes can compete because of speech variability (noise, speaker voice, intonation, accent and so on). Thus, at the end of the decoding phase, different sequences of phonemes, and so different sequence of words are given by the system. All the sequences are shaped in a graph. For example, in the figure 11, we can see the decoding result of the real sequence of words: " relax rich rage reality". Each arrow corresponds to an acoustic score. This graph is called "lattice" in the Kaldi Tool. To extract the final transcription, we must go through the graph and select what we define as the best path. In fact, we can adjust the weight of the language model inside the graph to obtain new scores for each arrow. Moreover, we can set up a word insertion penalty which eliminates the words with deficient transition scores. This parameter is handy to avoid insertions of words when babble noise appears in the recordings. With the parameters of language model weight and word insertion penalty set, we can extract the best path using Minimum Bayes Risk method all along the graph. We obtain the transcription with the time of each word, and their confidence score between 0 and 1. When the score is close to 1, there is a high confidence in the transcription, and the more it goes down, the worse it becomes. The words with a deficient confidence score can be removed or replaced by "word??" to show the uncertainty of the system. An analysis of the system performance depending on the minimum confidence score taken to keep the word will be done in the last section to determine the uncertainty and certainty limits.

2.5.2 Validity

The medical professionals have studied the answers of many subjects with the first version of Evolex. They defined valid answers and the wrong ones. So, in addition to the transcription and the time of the answers, we need to categorise them as valid or invalid. All the results are then shaped in simple and summarised graphics inside the Evolex application. The figure 12 shows an example of a result file, the final output of the system inside the Evolex server.

```
/audio_path/abricot.wav 0.93 1.14 ??? invalide
/audio_path/abricot.wav 2.10 0.72 fruit  invalide
/audio_path/abricot.wav 3.45 0.51 abricot valide
```

Figure 12: Example of output for the denomination task. Columns: audio path, beginning of the word in seconds, duration in seconds, transcription, validity

3 Evaluation and results

3.1 Corpus

To test the ASR system, we use the Evolex corpus. From 2012 to 2014, speech therapists and doctors have recorded around 100 subjects on the different tasks of fluency, denomination and generation. All the audio recordings and the subject information have been classified in a SQL database. We could not access the age of subjects, but they tried to obtain recordings of all age group from 18 to 90 years old. We test the systems described in the last section with this corpus. The tests enable us to adjust some parameters, and to define the strengths and vulnerabilities of the system.

3.2 Results

3.2.1 Word Error Rate

To measure the performance of an ASR system, the common unit used is the Word Error Rate (WER). The WER calculates the percentage of wrong words of the output transcription on the total number of words of the true transcription. There are three types of mistakes: word insertion, word deletion and word substitution.The WER is calculated as follow:

$$WER = \frac{S + D + I}{N} \tag{5}$$

As we count the number of insertions in the number of wrong words, the addition of all mistakes can be higher than the total number of words. This situation leads to WER higher than 100%.

Remark: The WER calculation compares the real transcription to the hypothetical transcription. This is a strict calculation, thus, if a word has been wrongly spelled in the real transcription, the correct and well spelled hypothetical transcription will be considered as incorrect. Moreover, if the doctor has only written the correct answer of the subject without all the other answers he gave by hesitating, the ASR system will give many words instead of the only one given in the real transcription. This leads to a lot of insertions mistakes, whereas the words have been said. We have to remind that the WER is worse than it should be because of these uncertainties.

3.2.2 Fluency

Language Model The comparison between **Evolex_LM**, **Brulex_LM** and **Uniform_LM** in the table 3 shows that the best language model for the four tasks is the **Evolex_LM**.

Table 3: WER of the four fluency tasks depending on the language model

Fluency	Uniform_LM	**Evolex_LM**	Brulex_LM
R WER (%)	34.47	**23.70**	31.40
V WER (%)	32.59	**25.54**	33.98
Animals WER (%)	31.03	**25.13**	31.43
Fruits WER (%)	30.80	**29.03**	30.85

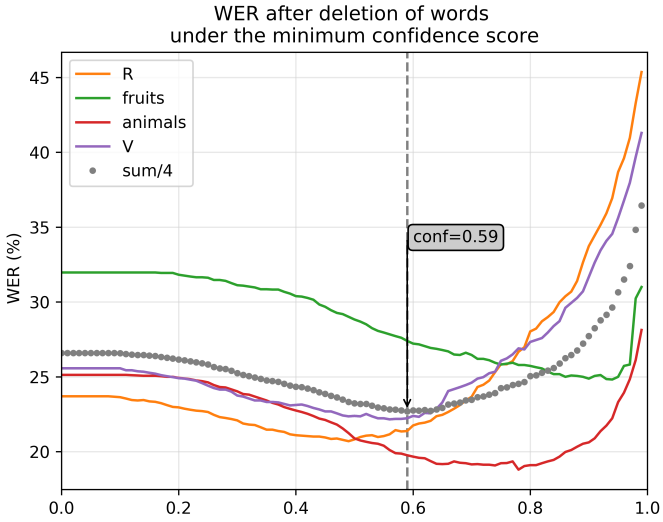

Figure 13: WER of the four fluency task depending on the minimum confidence score, all the words below the minimum confidence score are removed

Confidence Score We want to find the uncertainty and certainty limits of the confidence score. The minimised sum of the four WER leads to an optimal certainty limit at a confidence score equals to 0.59. Above this limit, the system is sure about the transcription. Under this limit, we can say that the words recognised by the system are probably wrong, and so we decide to keep the transcription but to indicate the uncertainty with two interrogation points. To determine the uncertainty limit, which will divide the uncertain words to the aberrant ones, we must analyse the result with a different point of view. A qualitative analysis of the results showed us that the wrong words were most of the time insertions due to background noise, babble noise or even sentences said by the subject between the words of the task. The deletion and the substitution mistakes are considered as worse errors, and we need to avoid them. The figure 14 shows the evolution of the WER without insertions and the number of word insertions depending on the minimum confidence score, which means we only took deletion and substitution as mistakes.

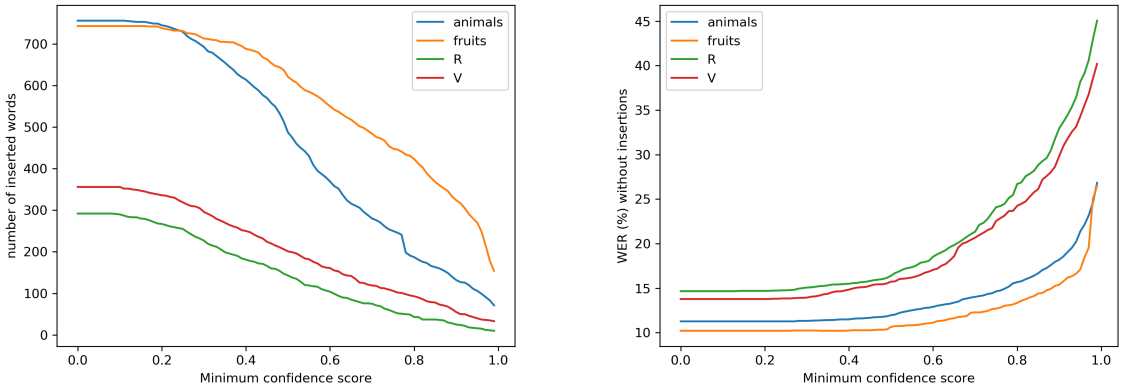

Figure 14: Evolution of inserted words and WER without insertions depending on the minimum confidence score

This graph highlights that the insertions go down with the removal of words with a confidence score between 0 and 0.3 and after this limit, the number of deletions and substitutions climb quickly. Remove the words below 0.3 of confidence score will not affect the real words said by the subject but only remove inserted words probably due to noise. Thus, we can define the uncertainty limit to 0.3. Instead of giving the transcription of these words, we will only give an interrogation point, to indicate that a word has been said, but there are probably not in the task.
In the table 5, the results with the removal of the words under the certainty limit and with the Evolex_LM.

Table 4: WER of the four fluency tasks with minimum confidence score of 0.59 and with te Evolex_LM

Fluency	WER (%)	Insertion	Deletion	Substitution	Nb words
R	21.38	107	378	206	3232
V	22.23	162	279	230	3019
Animals	19.76	379	444	255	5456
Fruits	27.40	558	278	100	3416

3.2.3 Denomination

For the denomination task we obtain WER between 0 and 15%. This excellent rate comes from the fact that in the denomination task, there are very few possible answers. The language model is this way precisely adapted to the task. The worst score are for the pictures with a bigger uncertainty : for example, a picture of wizard can lead to many answers like "sorcerer", "magician", "prestidigitator".Moreover, subjects tend to describe the pictures with a lot of word like "a wizard which is waiving a magic wand". In this case, all the words "which is waiving a magic wand" are not in the language model, and are not recognized.

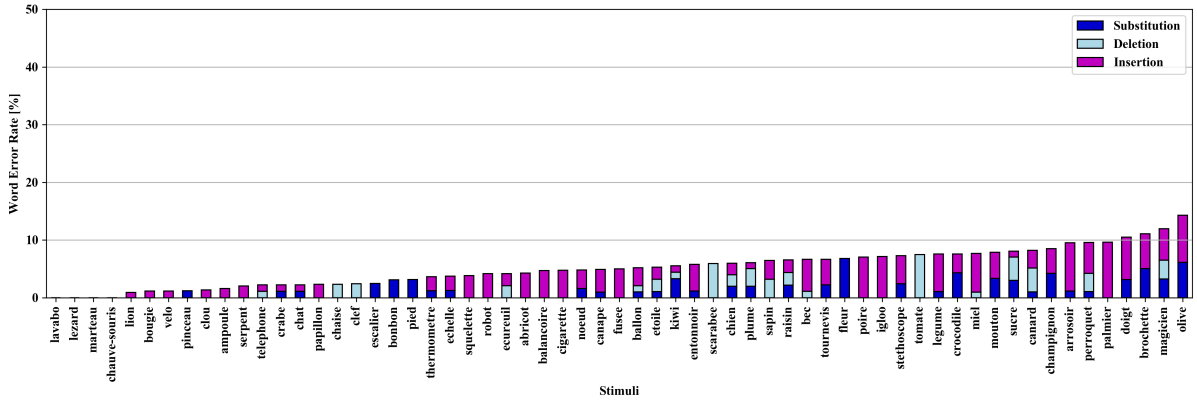

Figure 15: WER results for each stimuli of the denomination task with insertion deletion and substitution repartition

3.2.4 Generation

For the generation task, we obtain a WER between 0 and 40% depending on the stimuli. We had to add a new output processing for this task. In fact, in this task the patients have to generate a word after hearing one. We found out that many subjects were repeating the word heard before answering the test. We removed all the words identical to the stimulus. We noticed very bad rates for some words. In fact, the generation task is way harder to set up than the denomination one. The language model needs to be expanded because the answer depends on the subject imagination. Moreover, the stimulus is an audio, leading to a lot of test failures : subjects misunderstand the word they heard, and answer to a complete other word. In other case, they just do not hear anything at all and say "I didn't understand". These kind of difficulties lead to bad scores for many words.

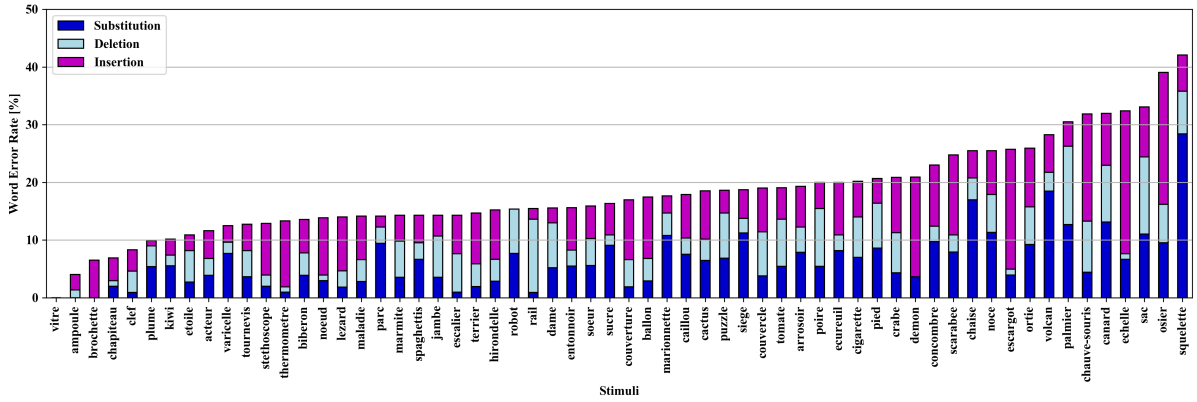

Figure 16: WER results for each stimuli of the generation task with insertion deletion and substitution repartition

3.3 Discussion

We have to keep in mind the remark of section 3.2.1. The WER obtained can be worse than the reality due to errors in the transcriptions or different spellings between the language model and the transcription. Moreover, the subjects were not perfect, and many recordings contain a lot of "filling blank" words, like "I don't know, did I say this already?". In these cases, the system becomes weaker and make a mistake.

An example We took an audio example and extracted the real transcription of the new system transcription and the one from Evolex 1. The speaker is an 80-year-old lady with an accent from Toulouse. During the test, the doctor says "It can be any word", the lady answers "yes but with an R". This kind of interaction between the doctor and the patient happens and leads to errors in the system. This example is interesting to see the reaction of our new system to these challenges. The well-transcribed words are in green, and the wrong ones are in red.

Real transcription: roitelet raton rat rhinocéros ragondin [...conversation] rare raison [je sais pas] ras-de-cou rire râler râteau ratisser rebut râler redire roucouler refaire

New system transcription: roitelet raton rat rhinocéros ragondin ruiner?? rer rare raisonneuse ??? rire râler râteau ratisser rebut râler redire roucouler refaire

Evolex 1 transcription: râleur rebeller rapport rein rhinocéros raccorder roux rouget rivière rare revendeur repasser rabais roux rire rire rite retour ratisser

Two observations. When the lady talks between the words, the system recognises wrong words such as "ruiner??" and "rer" But the two-interrogation points show that the system was not confident in its transcription. Another problem depicted in this example is the "ras-de-cou". The word is not in the language model and so cannot be adequately recognised; this can be solved with the addition of words in the language model. If the word is under the uncertainty limit, it is replaced by three interrogations points. These interrogation points indicate that a word has been said, but it was not recognised at all (deficient confidence score). By comparison, the Evolex 1 transcription has only two correct words. This low score may be because the system used was not trained with a lot of variability, and no language model was optimised for the tasks. The Table 5 gives the WER of the first version of Evolex and the new version for the four tasks of fluency.

Fluency	Evolex 1 WER (%)	New Evolex WER (%)
R	66	21.38
V	78	22.23
Animals	67	19.76
Fruits	71	27.40

Table 5: WER of the four fluency tasks with Evolex 1 and the new Evolex system

We can conclude that a new version was necessary, and we succeeded in improving the system with at least 43% fewer errors than before.

4 Conclusions

To automatise some tedious tests, speech therapists and doctors have asked the IRIT and other partners to develop a software using speech recognition. A first version was made at the beginning of the project but was not adapted to the tasks. The goal of my master thesis was to start from scratch and build a new automatic speech recognition system with better performances. After a study on the tools and methods needed for the project, I built the automatic speech recognition system required by the medical professional. Each block of the ASR system has been specifically made for the generation, denomination and fluency tasks. The Evolex corpus enabled us to judge its performance. We can conclude that my new version is way better than the first one, with at least a diminution of 40% for the word error rate . It has been implemented into the user interface and will enter a test phase in the next few months, to evaluate precisely its performances, and to analyse the vulnerabilities. I made possible the configuration of some blocks such as the lexicon and language models, to upgrade the system and make it evolve. On the same scheme, other systems can be added to the server, and the project can grow and be generalised for many medical tests.
Personally, this master thesis has been an excellent opportunity to understand the speech recognition science better, and to practise a lot with speech tools like Kaldi, which is essential in this field of study. Moreover, I discovered a new face of the research world, working with companies and hospitals. I have been pleasantly surprised to know that public researchers often work with companies on common projects. This "research and development" aspect seduced me. I enjoyed a lot my internship at the IRIT, even if the Covid19 has drastically reduced my attendance time in situ. Finally, this internship carried out my professional project. In fact, I am passionate about speech processing, and I look forward to continuing to work on this field.

References

[1] Bruno Gaume et al. "Automatic analysis of word association data from the Evolex psycholinguistic tasks using computational lexical semantic similarity measures". In: September (2018). URL: https://hal.archives-ouvertes.fr/hal-01881336.

[2] Bruno Gaume et al. "Toward a Computational Multidimensional Lexical Similarity Measure for Modeling Word Association Tasks in Psycholinguistics". In: 2019. DOI: 10.18653/v1/w19-2908.

[3] Balland Jouanicq Courtade. "Etude de faisabilité d'un logiciel de reconnaissance vocale adapté à des tâches d'évocation lexicale". In: (2015).

[4] Gitit Kavé et al. "Which verbal fluency measure is most useful in demonstrating executive deficits after traumatic brain injury?" In: *Journal of Clinical and Experimental Neuropsychology* (2011). ISSN: 13803395. DOI: 10.1080/13803395.2010.518703.

[5] Guy Pérennou and Martine De Calmès. "MHATLex: Lexical resources for modelling the French pronunciation". In: *2nd International Conference on Language Resources and Evaluation, LREC 2000*. 2000.

[6] Daniel Povey et al. "Sensory-perception testing box." In: *Canadian Journal of Occupational Therapy* 35.4 (1968), p. 140. ISSN: 00084174.

[7] O Stanislas. "Modèles de langage ad hoc pour la reconnaissance automatique de la parole". In: (2014). URL: http://tel.archives-ouvertes.fr/docs/00/95/42/20/PDF/These%7B%5C_%7DStanislas%7B%5C_%7DOger.pdf.