University “Politehnica” of Bucharest

Faculty of Electronics, Telecommunications and Information Technology

Investigation on Language Identification Methods

Diploma Thesis

submitted in partial fulfillment of the requirements for the Degree of Engineer in the domain *Electronics and Telecommunications,* study program *Technologies and Communication Systems*

Thesis Advisors

Ş.l. Dr. Ing. Andi BUZO

Ș.l. Dr. Ing. Horia CUCU

Student

Mihai DOGARIU

2014

University “Politehnica” of Bucharest
Faculty of Electronics, Telecommunications and Information Technology
Department Telecommunications

Department Director’s Approval:

Prof. Dr. Ing. Silviu CIOCHINĂ

[illegible]

DIPLOMA THESIS
of student *Dogariu Mihai, group 441G*

1. Thesis title: *Investigation on Language Identification Methods*

2. The student’s original contribution will consist of (not including the documentation part): *Design of the language identification system; collecting the necessary resources for creating a comprehensive database: building the acoustic, language and phonetic models for 2 to 3 different languages and improve the recognition accuracy at an acceptable level; testing the system under different conditions; configuration tuning (each configuration depends on a series of parameters; the best setup of these parameters is desired); study of several contemporary methods used for language identification; defining the requirements of the system (regarding both functionality and performance); conceptual analysis (schematic representation of the main procedures); implementation of the system; in the end, comparative study of the experimental results; drawing the appropriate conclusions regarding the advantages and drawbacks of the chosen method.*

3. The project is based on knowledge mainly from the following 3-4 courses: *Digital Signal Processing, Decision and Estimation in Information Processing, Object Oriented Programming*

4. The construction part of the project remains in the property of: *UPB*

5. The Intellectual Property upon the project belongs to: *UPB*

6. The research is performed at the following location: *UPB*

7. The thesis project was issued at the date: 15.10.2013

Thesis Advisor:

Ș.l. Dr. Ing. Horia CUCU

[illegible]

STUDENT:

Mihai DOGARIU

[illegible]

Statement of Academic Honesty

I hereby declare that the thesis *"Investigation on Language Identification Methods"*, submitted to the Faculty of Electronics, Telecommunications and Information Technology in partial fulfillment of the requirements for the degree of Engineer in the domain *Electronics and Telecommunications,* study program *Technologies and Communication Systems*, is written by myself and was never before submitted to any other faculty or higher learning institution in Romania or any other country.

I declare that all information sources I used, including the ones I found on the Internet, are properly cited in the thesis as bibliographical references. Text fragments cited "as is" or translated from other languages are written between quotes and are referenced to the source. Reformulation using different words of a certain text is also properly referenced. I understand plagiarism constitutes an offence punishable by law.

I declare that all the results I present as coming from simulations and measurements I performed, together with the procedures used to obtain them, are real and indeed come from the respective simulations and measurements. I understand that data faking is an offence punishable according to the University regulations.

Bucharest, *29.06.2014*

Mihai DOGARIU

[illegible]

Table of Contents

List of Figures

List of Tables

List of Acronyms

ASR	Automatic Speech Recognition
LID	Language Identification
HMM	Hidden Markov Model
MFCC	Mel-Frequency Cepstrum Coefficient
FFT	Fast Fourier Transform
DCT	Discrete Cosine Transform
PLP	Perceptual Linear Prediction
SDC	Shifted Delta Cepstra
LM	Language Model
LPCC	Linear Prediction Cepstral Coefficients
GMM	Gaussian Mixture Model
UBM	Universal Background Model
UBM-GMM	Universal Background Model Gaussian Mixture Model
EM	Expectation Maximization
MAP	Maximum Aposteriori
PRLM	Phoneme Recognition followed by Language Modeling
PPRLM	Parallel Phoneme Recognition followed by Language Modeling
CMU	Carnegie Mellon University
SpeeD	Speech and Dialogue

Introduction

Ever since prehistoric times speech has been the most used means of communications. This was the case thousands of years ago and it still is the dominant way of communicating in the modern society. In the early days of humanity the only possibility for 2 persons to communicate was to address each other from a short distance. With the progress of technology human speech was allocated a greater deal of importance and fast enough people started to take interest in how speech is created and interpreted. At first these ideas were approached by sciences such as anatomy but soon enough engineers started to investigate the methods in which they can synthesize speech or even transport it over long distances.

A major technological breakthrough was the invention of the telephone in 1876 by Graham Bell, which allowed people to send informational messages in the form of speech over great distances. These devices quickly became popular and are now amongst the most used small appliances [1]. Even though many other forms of communication emerged (telegraphy, written messages etc.) speech remained the most important and used means of communication.

With the ongoing technological progresses throughout the last century computers evolved at an astonishing rate. They gained high computational power and numerous capabilities which people tend to exploit to the fullest, but all these led to more complex systems and harder to comprehend for the regular user. As a response to this need certain interfaces were developed in order to ease the communication between humans and machines. Up to this day this is still a highly debated problem on which many organizations and researchers are focusing their attention. Given the fact that technology is now addressed to regular users and not to power users, it is of great importance for everyday users to operate machines without being previously trained for this specific task. It is obviously simpler for a regular user to express his desire to a machine by simply talking to it instead of using peripherals such as a keyboard or a mouse. Taking for example the case in which a user wants to mute the speakers of a computer, it is easier for him to say *mute sound* instead of accessing several command panels, using the mouse or keyboard in order to tick a box that performs the desired task, all while paying attention to the display so he can be sure that he is not making any mistakes.

The advantages of a world in which machines can be voice controlled are obvious, in terms of security, processing speed and user's multitasking ability given by the possibility to work on multiple tasks with parts of their bodies which would be otherwise involved in controlling the machine. It would also offer more interaction between users and machines which, in term, would lead to a greater degree of satisfaction of the end-user.

In order to implement a system that can interface between humans and machines, based on human speech, it is necessary to build a spoken language system which deals with speech recognition and speech synthesis. But these 2 abilities are not enough if we want to have a versatile spoken language system, because it would lack speech interpretation. An understanding component is necessary in order to interpret the input speech, transform it in basic instructions for the machine, execute them, give a reply and then interpret the response in such a way that the average user understands it. After this it is the speech synthesis system's duty to output the reply of the machine in the form of speech.

This thesis approaches 2 fields of interest out of which one is the first part of the spoken language system, i.e. the Automatic Speech Recognition (which will further be called ASR) system. The second part of the thesis regards Language IDentification (which will further be called LID) systems.

ASR systems' objective is to transform spoken utterances into text. The performance of an ASR system is strongly related to the targeted language, the number of speakers used for training, the context in which the system was trained, the number of words in that language's dictionary, the level of the recording's environmental noise etc. Regarding the targeted language a distinction can be made between languages for which there is a small amount of resources and languages for which there is a high amount of resources. A *low-resourced language* is a language spoken by a large number of people but for which significant research work hasn't been done. These are languages that lack a significant amount of text resources and organized speech. For such languages the ASR system creation is a daunting process because it involves not only the training and testing of the system but also the resource accumulation. This is the case for Romanian and Albanian, two of the languages targeted by this thesis. On the other hand, languages which presented interest for previous researches have ASR systems with better performances and are more robust.

Regarding language's morphology a distinction between *rich-morphology languages* (e.g. Romanian) and *poor-morphology languages* (e.g. English) can be made. This term refers to the different morphological forms of words, mainly because of declination. For example, in English, the present tense of the verb *to write* has only 2 morphological forms: "write" and "writes", while in Romanian it has 5 different morphological forms: "scriu", "scrii", "scrie", "scriem", "scrieți", "scriu". In addition to this characteristic there is another aspect that must be taken into consideration when analyzing a certain language and that is the agglutination. In an agglutinative language most of the words are formed by joining morphemes together [2]. Each of these morphemes expresses a definite meaning. Agglutinative languages like Turkish present composed words, such as *taam-uk-ul-igw-aasy-an-il-a* meaning "cause each other to be unseated for (someone)" [3]. All these particularities influence the size of the dictionary for a specific language and that is why English has less words than German, for instance, where a great number of compound words can be found.

Even though some ASR systems can be implemented with a higher number of words in the dictionary it does not account for an easier task for the system. The meaning of the spoken utterances is the one that makes the difference between two ASR systems trained with an equal number of words in the dictionary. This is the case when we are to compare between an airplane ticket disposer which is supposed to have as input parameters mostly proper names, i.e. the names of the towns to which someone can travel and a system designed to recognize spontaneous speech between more individuals. Given these two cases a better result would be for the airplane ticket disposer because it has a lower linguistic uncertainty.

In the second part of this thesis more attention is paid to the language identification technique and its utility. LID systems are of great importance in countries which have more than one official language in order to discriminate between them. In these countries more than one of the official languages can be mixed inside the same phrase resulting in a code-switched speech. For example, in South Africa there are up to 11 official languages.

A LID system can be successfully implemented in many applications, especially with the ongoing tendency towards globalization. It is easy to remark that societies have started to use terms from foreign countries and it is not a peculiar habit anymore. Many societies use English words and it has started to shift from a trend to a normal situation. Furthermore, the ease of traveling brings more and more strangers to every country, strangers who do not particularly know the mother tongue of that country. LID systems come to the aid of these people helping them to cope with the language barriers. They can be a valuable asset to countries with more than one official language and in countries with strong external language influences. Languages such as English and Mandarin are already being given the proper attention but there are not enough resources to develop similar systems for *low-resourced languages*. Even though there might be enough researchers interested in developing such systems they are still confronted with the dismaying task of gathering a proper amount of text and audio corpus in order to train a robust enough system.

This thesis aims to take on the challenge of proving that with the proper information one can collect both text and speech in order to build an ASR system to which a LID system will be added with satisfying results. Given the fact that Romanian and Albanian are both low-resourced languages it makes it even harder to find appropriate text and speech. Furthermore, with a very low number of people in our country understanding Albanian it will be shown that it is still possible to build ASR and LID systems for a language of which one knows very little about, but with some additional help.

Chapter 1. Automatic Speech Recognition – Theoretical Aspects

1.1. Introduction to Automatic Speech Recognition

The aim of an ASR system, also known as a speech-to-text system is to translate an audio signal, a speech signal, into a written sentence. The speech signal is presumed to be a sequence of words, naturally uttered by a person. In the modern society spoken language is the most common means of communication and people make use of it on a daily basis with very few exceptions.

An ASR system is designed with the purpose to aid the interaction between humans and machines. An ideal ASR system would be so trust-worthy that humans could rely on it to make only voice interaction with machines and reduce any other sort of interaction to a minimum or even to totally dispose of it. As an example, an ASR system can be imagined to help surgeries, that would be so precise and with such a strong understanding capability that a doctor could rely on it to perform a surgery without checking the displays of the machines and suffering less from human error factors. This is an ideal case in which the ASR would be very robust, would answer in the same manner no matter who the speaker is and would be independent of environmental noise. This is not yet the case but researchers are making progresses in this direction in order to create new ASRs and optimize the ones that already exist.

ASR systems are taking on the challenging task to implement hands-free and eyes-free interactions between humans and machines and thus helping many industries such as previously mentioned, medicine, journalism, where journalists could easily write an entire article in a matter of minutes, literature, where authors could make a better use of the time spent on writing and many others. A natural expansion of ASR systems is the reverse process, i.e. the text to speech system, together with which a speech to speech assembly can be implemented. This is a very modern subject with many applications mostly in the entertainment industry where the end users show greater satisfaction than for previous interaction systems. Speech to speech systems often require an algorithm to provide meaning to the recognized words and interpret them in such a manner that the response, which is the responsibility of the text to speech system, provides a logical sequence of words that best fits the context of the input speech. This strongly depends on the message that the intermediate assembly between the speech to text and text to speech systems receives at its input. Therefore, a simple translation from speech to text is not sufficient for an ASR system as it is also supposed to give a minimum meaning to the text that it outputs. Having this in mind, it is desired to have a logical succession of words in the text form as close as possible, if not identical, to the succession of words contained in the input speech.

The above stated problem helps us to accommodate a better interpretation for the ASR taking into account the fact that it is amongst the first fields in which statistical modeling of large data quantities became a standard. In a probabilistic manner, the problem of ASR can be put this way: *What is the most likely sequence of words W* in the language L, given the speech utterance X?* [4]

In a formal representation it results:

$$\boldsymbol{W}^* = \mathbf{arg}\max_{\boldsymbol{W}} \boldsymbol{p}(\boldsymbol{W} \mid \boldsymbol{X}) \tag{1.1}$$

This equation states **W*** is the most probable word sequence with the highest posterior probability, given the speech utterance. According to the Bayes rule one gets the posterior probability and the most probable word sequence:

$$\boldsymbol{W}^* = \mathbf{arg}\max_{\boldsymbol{W}} \frac{\boldsymbol{p}(\boldsymbol{X} \mid \boldsymbol{W})\boldsymbol{p}(\boldsymbol{W})}{\boldsymbol{p}(\boldsymbol{X})} \tag{1.2}$$

Given the fact that **p(X)**, the probability of the utterance to be stated is independent of the sequence of words **W**, we can ignore it, leaving us with:

$$\boldsymbol{W}^* = \mathbf{arg}\max_{\boldsymbol{W}} \boldsymbol{p}(\boldsymbol{X} \mid \boldsymbol{W})\boldsymbol{p}(\boldsymbol{W}) \tag{1.3}$$

At this moment the problem of giving an estimate of the sequence of words having given the speech utterance can now be interpreted as two distinct problems with a lower degree of complexity:

1) estimating the prior probability of the word sequence **p(W)**
2) estimating the likelihood of the acoustic data given the word sequence **p(X | W)** [4]

Each of the two problems can be solved by means of building a language model to compute the first probability and an acoustic model to compute the second probability, respectively. Both of these models can be built independently of each other but they will only be used together to make a valid decoder as stated in Equation 1.3 [4]. Therefore, the practical challenge of speech recognition is how to build accurate language models **p(W)**, and acoustic models **p(X | W)**. Improving the accuracy of the acoustic model poses great difficulties because for a language with large dictionary, as it is in this case, the acoustic model cannot cover the entire range of spoken words. That is why it is necessary to train the system based on sub-words units, meaning phonemes. Creating a robust phonetic model for a language gives the possibility to recognize words that the system did not encounter in the training process, similar to a person decoding a message heard for the first time in his/her life. Usually, the phonetic model consists of giving a phonetic representation for every word found in the vocabulary of that language. Having a phonetic transcription of every word, the decoder will try to map every spoken phoneme in such a way that it will form a meaningful entity found in the vocabulary taking into account the probability for that word to occur in a certain context, probability given by the language model.

Another important aspect regarding the decoder is that it does not operate with the direct wave form of the input speech, but it models it first, extracting a series of parameters that will be later discussed in Section 1.4.1. Having these in mind we present a general block diagram of the decoding part of the ASR system in Figure 1.1.

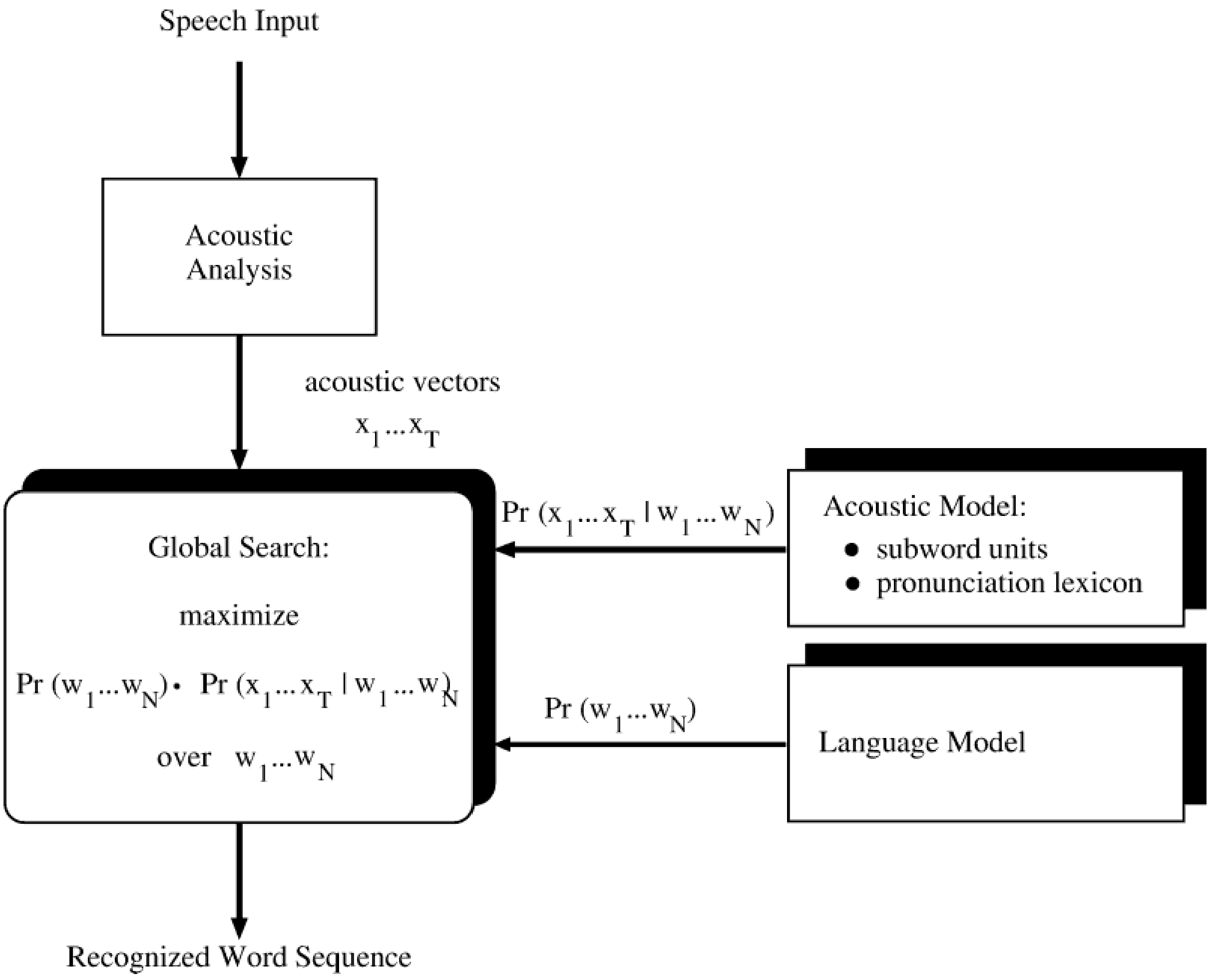


Figure 1.1 General Block Diagram of an ASR Decoding System [5]

Each of the main blocks in Figure 1.1 will be treated separately in the following part of this paper and so will be the training part of the speech recognizer in which language modeling and acoustic modeling will be debated.

1.2 Language modeling

A good way of judging a speech recognizer is to see how well it responds to speech which it hadn't had the chance to train its models on. This is called managing sparse data, because it is impossible to keep records of every possible word sequence that can exist in a certain language. Language modeling is one of the two most important features when talking about speech recognizing in that it gives relevant information about recognizing and understanding natural speech.

The goal of spoken language identification can only be achieved if certain parameters regarding language modeling are taken into account. Such parameters are the syntax and semantics of the language (which determine the correct sequence of words from the grammatical point of view and an evaluation of that sequence's meaning) and knowledge of that language's pragmatics (what a speaker is most likely to say in different contexts, given what they already said). A clear differentiation between these two is impossible because they often interleave when creating a language model [6]. Syntax, semantics and pragmatics offer redundancy from the information point of view. Such redundancy can and is exploited with the use of statistical modeling concepts on which speech recognition heavily relies.

The Language Model (LM) is what gives a best estimation of the likelihood of the sequence of words **W = w1, w2, w3, …** to appear in the form of a sentence in the source language [4]. The LM comes to help the acoustic model in the decision making process of decoding a speech input. For example, the two fragments "compact would author's " and "contact with others" are very similar to one another from the acoustic point of view, which makes the task of the acoustic model very difficult in taking a decision on what was spoken without having any other information regarding the message that these two fragments contain. However, the language model offers this much needed information giving different likelihood probabilities to the two fragments in question. This way, the LM will give "compact would author's" a much lower probability to be part of natural speech, if not even null, than the one assigned to "contact with others". It is obvious that the first syntax is almost impossible to be encountered in natural free speech while the second one could be uttered in many situations. There are numerous other examples of such confusions that the acoustic model could be faced with. Many of these can be solved by statistical decisions provided by the LM of that language. LM lets the recognizer make the right decision when two different sentences sound the same [7].

The problem that LM responds to can be rephrased in mathematical terms as the chain rule of probability:

$$\boldsymbol{p(W) = p(w_1, w_2, \dots, w_n) = p(w_1)p(w_2|w_1) \dots p(w_n|w_1, w_2, \dots, w_{n-1})} \qquad (1.4)$$

As it can be seen from Equation 1.4 the probability of a sequence of words $\mathbf{W}$ to appear in the recognized speech is a product of the probabilities assigned to words to appear after a certain sequence of words. Equation 1.4 can also be written as:

$$\boldsymbol{p(W) = \prod_{i=1}^{n} p(w_i|w_1, w_2, \dots, w_{i-1})} \qquad (1.5)$$

The probability $\boldsymbol{p(w_i|w_1, w_2, \dots, w_{i-1})}$ is interpreted as follows: "The probability of the word $\boldsymbol{w_i}$ to appear after the sequence of words $\boldsymbol{w_1, w_2, \dots, w_{i-1}}$ in this precise order". It can be observed from this that the probability to obtain a word $\boldsymbol{w_i}$ depends on the history of its predecessors. However, it is not recommended to take into consideration the entire history of the utterance because some histories may appear only once or have a large number of words in the entire sequence so that it is not worth the amount of resources that need to be allocated in order to record such a history. This can be resumed to a simple balance between number of terms in the history and memory allocation. Therefore, it is necessary to limit the number of preceding words from the history to $\mathbf{m}$ or to put it in another manner, only the last $\mathbf{m}$ words contribute to the probability of choosing the next word. This method is a Markov assumption, stating that:

$$\boldsymbol{p(W) = \prod_i p(w_i|w_{i-m}, w_{i-m+1}, \dots, w_{i-1})} \tag{1.6}$$

This leads to applying the N-gram language model, which is the most common practice in speech recognition techniques used for language modeling. N-grams represent a LM built while taking into consideration only the last N words in a sequence in order to determine the probability of the next word. The most used are single words (unigrams), 2-grams (bigrams) and 3-grams (trigrams) with more attention given to trigrams. The amount of text data on which the training of such a language model is performed is called *text corpus.* For a LM of good performances it is necessary to use a large enough text corpus which comes back to the initial problem or resource gathering. The number of words required for such a case is of many millions, reaching even billions.

Considering for example the case of bigrams we must use the maximum likelihood estimate and count how often the word $\boldsymbol{w_{i-1}}$ precedes the word $\boldsymbol{w_i}$ [9], or the probability to come upon the pair of words $(\boldsymbol{w_{i-1}}, \boldsymbol{w_i})$, in this order. The probability thus can be calculated with the formula:

$$\boldsymbol{p(w_i|w_{i-1}) = \frac{count(w_{i-1}, w_i)}{count(w_{i-1})}} \tag{1.7}$$

Equation 1.7 states that we can find the probability of the pair of words $(\boldsymbol{w_{i-1}}, \boldsymbol{w_i})$ to appear by counting the number of this pair's occurrences and dividing it by the number of occurrences of the word $\boldsymbol{w_{i-1}}$. It can be easily seen that according to the meaning of such sequences higher or lower probabilities will be assigned. For example, we expect the sequence "I would" to have a much higher probability than the sequence "Chinese pizza", being very context dependent. In a similar manner trigrams are approached, their probabilities being calculated according to:

$$\boldsymbol{p(w_i|w_{i-2}, w_{i-i}) = \frac{count(w_{i-2}, w_{i-1}, w_i)}{count(w_{i-2}, w_{i-1})}} \tag{1.8}$$

The larger the number of words in the history of a sequence is to be considered the better the results will be, providing better accuracy and more robustness. However, this comes at the price of needing even larger text corpus with the number of words in the language model growing exponentially. It has been a common practice to choose 3-grams to obtain satisfactory result for a speech recognizer.

1.3 Language Model Evaluation

As it has been seen in the previous section the language model's purpose is to try to determine the most probable sequence of words in a speech signal. Consequently, a means of evaluating such language models is necessary. There are more ways in which an objective evaluation can be made but we will approach the most common two of them.

1.3.1 Perplexity

LM's evaluation is most commonly done by means of Word Error Rate (WER), which is basically an alignment between the correct speech transcript, also called reference file and the decoded speech transcript, also called hypothesis file. This method can be successfully used if we have access to the reference files, which is not generally the case. It also depends strongly on the ASR in question and is computationally expensive [13]. Thus, perplexity was taken into account, regarded as the main LM evaluation metric, between ASRs with the same vocabulary size.
Perplexity can be calculated starting from the formula:

$$\boldsymbol{PP_T(p_M) = \left(\prod_{i=1}^{t} p_M(w_i|w_1 \dots w_{i-1})\right)^{-\frac{1}{t}}} \tag{1.8}$$

The above equation can also be interpreted as the inverse of the geometric average probability assigned to each word in the test set by the model [13]. Computing its logarithm gives the upper bound of the number of bits expected in compressing. Generally, the lower the perplexity is the better the estimation of the ASR regarding the sequence of words is.

1.3.2 Out Of Vocabulary words

The second metric that we discuss is the Out Of Vocabulary (OOV) words. As a language model is trained on a finite text corpus it is obvious that it cannot contain all possible words of that language. Consequently, it is possible to come upon words that the system has not been trained with during the decoding process. These words are considered OOVs. The higher the number of OOV words is, the worse the result of the recognizer will be, giving false hypothesis for each of these words. The perplexity for such a word is considered infinite so it must not be taken into account when computing the sum of all perplexities for the sequence of words in which the OOV word is encountered. In this case the following evaluation will be made and taken into consideration [4]:

$$\boldsymbol{OOV}[\%] = \frac{\boldsymbol{\#OOVs}}{\boldsymbol{\#words}} * \mathbf{100} \tag{1.9}$$

1.4 Phonetic modeling

Having a large vocabulary to work with for designing a speech recognition system it is impossible to use words as basic speech units because there would be too many words outside the training corpus that the implemented system has never seen to have good results. Also, different tasks may require from an ASR to mold itself to a specific context for which there has been no training data available. For example, if an ASR is asked to decode utterances linked to quantum Physics and the training corpus consisted mainly of news and political terms there would be a very high probability of error if the ASR were to try and decode the speech with words as basic speech units. For easier adaptation of the ASR it is required to build a phonetic model as well. This serves as a link between the acoustic model and the language model, between the likelihood of the acoustic data and the probability of the word sequence.

The phonetic model usually consists of a pronunciation dictionary, linking each word in the dictionary to its respective phonetic transcription, written as a sequence of phones. Again, the phonetic model is linking the acoustic model which uses phones to the language model which uses words [4], thus being an interface between the two.

It is good to know that for phonetic languages the list of phones will contain less phonemes than in the case of languages which are not phonetic. Thus, it is expected to see more phonemes for English than for Albanian and Romanian. The lower the number of phonemes, the easier it is for the ASR to map them in meaningful sequences in order to form words.

1.5 Acoustic modeling

After years of research and debate in ASR domain the greatest challenge remains that of obtaining a good accuracy. Factors such as variations in context, in speaker and in environmental noise have a strong impact on the ASR’s accuracy [6]. The acoustic modeling of an ASR is arguably the most important part of the system and it must be treated likewise. As stated in Section 1.3 ASR systems do not estimate the likelihood of entire sequences of words but of phonemes which are smaller speech units. As a consequence, the decoding process implies estimating the likelihood of these small speech units linked in such a way that they form word models and eventually word sequences models [4]. It has been proved that for this kind of approach the Hidden Markov Model (HMM) is best suited. It is a powerful tool used for segmentation, time warping, pattern matching and integrating context knowledge in a unified manner [6].

1.5.1 Acoustic features

As it can be observed in Figure 1.1 a speech recognizer does not decode the time-domain waveform of a speech signal. Instead it performs and acoustic analysis which in term gives at the output some acoustic features which will aid the acoustic model in the decoding process. These acoustic features will be used, more precise, by the HMM.

It is known that speech signal is a quasi-stationary signal, presenting stationary properties on small frames, of 20ms to 30ms. This frames are generated every 10ms, thus resulting in a partial overlapping. In order to smooth the edges of these frames they will be multiplied with a window function. Usually, the Hamming window is chosen for this multiplication as it gives the smoothest and least distorted spectrum that typical framing windows can give. Thus, the initial speech signal's time-domain waveform is now translated into a sequence of time-domain sequence of quasi-stationary frames [4].

The main parameters extracted during this stage are the Mel-Frequency Cepstrum Coefficients (MFCC) and the Perceptual Linear Prediction (PLP). Cepstral coefficients are preferred above spectral coefficients because they are de-correlated with the help of the Discrete Cosine Transform (DCT) as opposed to the second type, which possess great correlation between adjacent spectral coefficients [4]. MFCCs are obtained according to the following block scheme:

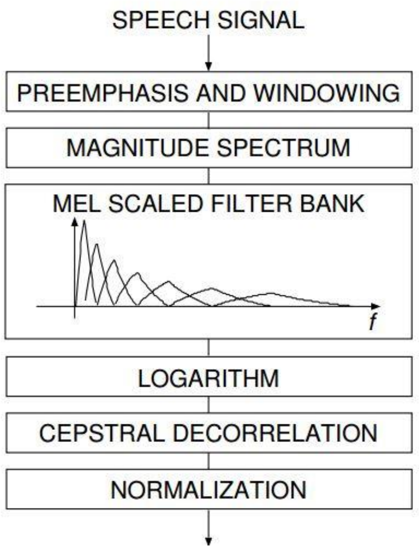


Figure 1.2 MFCC generation block scheme [10]

As it can be seen in Figure 1.2 there are several steps required to obtain the MFC coefficients [10]:

- preemphasizing the original samples and applying a Hamming window of length around 25ms
- the magnitude spectrum is obtained by computing the Fast Fourier Transform (FFT) of the windowed signal
- the magnitude spectrum is compressed by a Mel scaled filter bank (filters of triangular frequency response, equally distanced regarding their central frequency)
- compute logarithm of the power of each of the Mel frequencies
- perform the DCT in order to de-correlate the signal
- normalize the obtained terms in order to account for different audio channels

The PLP acoustic features are obtained according to the following block diagram:

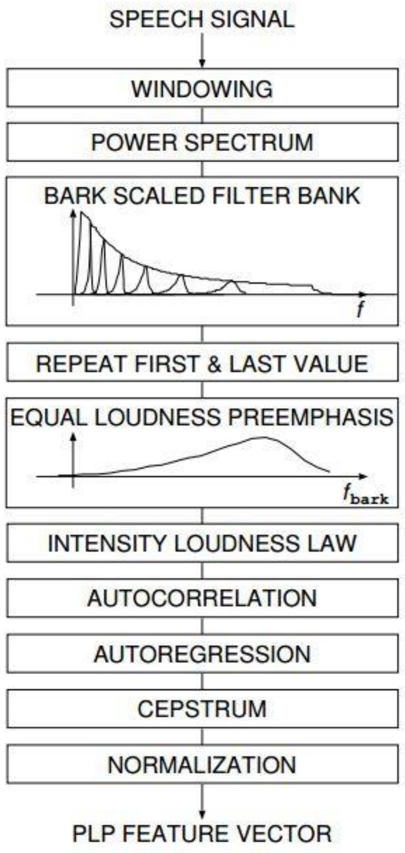


Figure 1.3 PLP generation block scheme [10]

As it can be seen in Figure 1.3 there are several steps required to obtain the PLP acoustic coefficients [10]:

- a Hamming window is applied to the speech signal without pre-emphasizing it. This window is slightly narrower than in the case of MFCC, with a length of 20ms.
- the power spectrum is obtained with the use of FFT
- a filter bank composed of trapezoidal frequency response filters is now applied to the power spectrum
- the first and last value are repeated because the filters reach beyond the valid frequency range and their output is discarded and replaced with the value of the right or left neighbor
- the equal loudness pre-emphasis is used to compensate the non-equal perception of loudness at different frequencies [11], having as output the intensity of the speech
- the intensity-loudness law gives an approximation of the loudness perceived by the human hearing as a function of the intensity
- autocorrelation coefficients are calculated with Inverse Discrete Fourier Transform (IDFT)
- the autocorrelation coefficients are then transformed into autoregressive coefficients by using the Levinson-Durbin recursion [10]
- finally, these autoregressive coefficients are transformed into cepstral coefficients and are optionally normalized

These two kinds of acoustic features are computed on a relatively small window length of 20-25ms with MFCC being usually preferred to PLP. Even so, they both give plenty of information regarding speech models and information about the coefficients' dynamics. The second type of information has been found to be useful as well because it contains important data about the coefficients' variation rate, adding to the local temporal dynamics of the speech signal [4]. Together they give a comprehensive description of the speech signal which will further be used. It is worth mentioning that these acoustic features are used not only in the training phase of the system, but also in the decoding process, the entire system relying on these coefficients instead of the actual speech signal.

1.5.2 Hidden Markov Model

As it was stated in the previous section speech signal is completely characterized by the acoustic features for automatic speech recognition. In order to model these features and exploit the information that they contain we make use of the HMM. The HMM is used as a statistical method which characterizes the data samples of a discrete-time series [6]. It is basically a finite state automaton that models the state of a system with a random variable that changes during time on which we apply some restrictions. The HMM is called hidden because the state sequence is partially unknown, thus hidden to the observer. A probability density function attached to each state generates a sequence of acoustic feature vectors which is observed instead of the state sequence [4]. A representation of the HMM can be found below, taking into consideration the start and end states depicted as “Entry” and “Exit” and three other intermediate states:

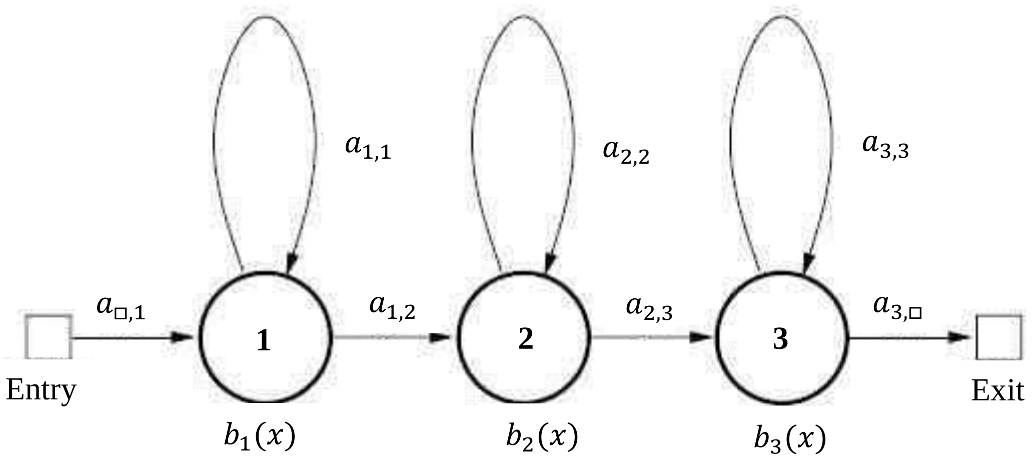


Figure 1.4 3-state left-to-right HMM diagram [12]

This HMM’s constituent parts are:

- the set of states: “Entry”, “1”, “2”, “3”, “Exit”, usually denoted $q_1,\ q_2, \dots ,\ q_N$
- the set of transition probabilities $a_{i,j} = p(q_j|q_i)$ representing the probability of transitioning to state q_j from the state q_i
- the set of observation likelihoods $b_1(x), b_2(x), b_3(x)$, where $b_i(x) = p(x|q_i)$ is the probability of an observation x being generated in the state i.

It can be observed that a particularity of the HMMs used for speech recognition is that the transitions are not arbitrary as how it is presumed. Instead, a transition can be made only to the actual state (a self-loop) or to the successive state, not allowing backward transitions or skipping transitions. This type of modeling for phones can be explained by the fact that a self-loop occurs when a phone is prolonged for more than one state, covering a larger part of the input speech or it can only advance to the next state, just as human speech behaves. The advantages for this type of modeling occur from the fact that there is no need for a memory in which to store the previous states being a first-order Markov process and from the fact that the current state contains all the information regarding the previously observed acoustic feature vectors [4].

The HMM previously presented is regarded as the best method of acoustic modeling so far, making computations faster and less complex. This is well-suited for phone modeling because it is known that a phoneme is strongly dependent on the context in which it will be found. Thus, a phoneme can be evaluated only by taking into account its neighboring phonesme. The common practice is to take a phoneme into consideration regarding its left and right neighboring phonemes, case in which we refer to it as a triphone. Another case could be when a pair of phonemes is referred to in a similar manner, regarded as a quadriphone.

Chapter 2. Language Identification – Theoretical Aspects

2.1. Introduction to Language Identification

The aim of Language Identification (LID) systems is to give a quick and accurate identification of the language that has been spoken. Over the past few decades LID systems have become increasingly important and found their way into many industrial applications. The current degree of globalization has indirectly imposed the use of many languages for a large number of applications, thus requiring these systems to give a fast and accurate response while surpassing language barriers. An example of such applications is the system used to redirect incoming calls in a telephone company to different operators depending on the language spoken by the caller [14]. That would be a time-saving action that would benefit not only the company but also the person calling as he or she wouldn't be required to go through preliminary steps in order to talk to the person in charge of solving that problem. It would be much easier to say *"I am given the number busy message whenever I call somebody"* in your mother tongue and immediately be switched to an operator that speaks your language and already knows the problem that you are confronted with. It would be less stressful and this kind of problems would be quickly solved.

Another important aspect about LID is that it is less expensive to train than people. It would take multiple days or weeks for a person to gain the capability to recognize a language that they have no knowledge about. Taking into account the fact that LID is being performed out of a number of languages it means that each person responsible for this task should be trained in the same manner for each of the languages in question. This training must be performed for each and every person that will have the task of LID. In comparison to these necessities, LID systems must be trained only once with a high enough robustness and then they can be ran on any machine simultaneously with no additional costs involved. This gives a higher degree of freedom when one resorts to such an approach.

However, the accuracy of such systems depends on the available training data as it can take wrong decisions if the system has not been properly configured. A strong advantage that humans possess in this debate is that they can make subjective assumptions about the language that they are hearing at the moment based on previous experience "*sounds like Spanish*" and with little or no expertise about what the sentence's message was [15].

Another challenging task for LID systems is that they must be reliable in the absence of prior knowledge about the speaker's identity and the utterance's message. Adding to this there is an overwhelming number of languages out of which a LID system must discriminate, with more than 6000 languages being used in the present. That is why it must easily adapt to new languages and it must also be flexible enough in order to accommodate variations of different speakers [16].

LID and ASR systems share many similarities as to what the problem formulation and system approaches are concerned. Both of them can be set up as recognizers or verifications. Regarding this problem, a LID system can be built in such a way that it is able to recognize the spoken language from a set of known languages or it can be built in such a way that it addresses the problem of accepting or rejecting the hypothesis that the given utterance was spoken in a certain language or not [14].

Human speech can be modeled in the case of the LID task into different speech features, divided mainly into two levels: spoken level and word level [17]. The first level includes acoustic, phonetic, phonotactic and prosodic information about human speech and it can be obtained from the raw speech signal. The latter refers to morphology, syntax and grammar information [14]. Its importance can be easily seen because each language contains a specific set of words, namely an own vocabulary. This is a great difference between any two languages.

LID systems are formed of two main parts: the front-end which extracts the necessary feature vectors from the input speech and the back-end which has the task of identifying the language based on certain sets of feature vectors, models and algorithms. The training of such systems must be done taking into account that the identification part must not be biased to any language, it must respond in the same manner to short or long speech inputs, it must be robust to channel and speaker variations [14] and it must be persistent to noise factors.

2.2 Basic aspects of LID systems

As stated in the previous section LID systems are composed of the front-end and the back-end, each with its specific tasks to fulfill. The front-end of the system extracts a sequence of features thus giving a characterization of the input speech's waveform. The idea here is to extract as much useful information as possible from the audio waveform and discard as much of the redundant information as possible [14]. This is done at the frame level with a single N-dimensional feature vector being extracted from each frame. N is a value much lower than the number of samples per frame, thus resulting in a reduction of the information quantity that is sent to and interpreted by the back-end. An entire speech signal is thus transformed into a sequence of vectors $\boldsymbol{X} = [\boldsymbol{x_1}, \boldsymbol{x_2}, \dots, \boldsymbol{x_k}, \dots]$, with $\boldsymbol{x_k}$ an N-dimensional vector and $\boldsymbol{k}$ the frame index [14].

The ideal case is when all redundant information, noise factors and speaker dependent features have been removed from the signal and only the characteristics of the speech waveform that are useful for discriminating between languages are left and sent to the back-end for further computations. The most common parameters that are used for LID are MFCCs, PLP, Delta, Delta-Delta and Shifted Delta Cepstra (SDC). At first, these feature vectors are used to train a model, λ_l different for each of the languages to be recognized.

In the identification phase the input signal is processed and the same set of feature vectors is extracted as in the training phase. This feature set is afterwards compared with the sets of features that the system was trained with: $\{\lambda_l | l = 1,2, \dots, L\}$, with L the number of possible languages used for identification. The back-end of the system must determine which of the L language models better fits the input signal's set of feature vectors, maximizing the a posteriori probability across the set of language models [14].

The selection goes according to the following equation:

$$\hat{l} = \arg\max_{1 \le l \le L} P(\lambda_l | X) \tag{2.1}$$

According to Bayes' Rule, Eq. (2.1) can be rewritten as:

$$\hat{l} = \arg\max_{1 \le l \le L} \frac{P(X| \lambda_l) P(\lambda_l)}{P(X)} \tag{2.2}$$

The hypothesis that the system is not biased towards any language means that all languages are assigned equal likelihoods $P(X)$, the above equation thus becomes:

$$\hat{l} = \arg\max_{1 \le l \le L} P(X | \lambda_l) \tag{2.3}$$

This leads to the fact that giving an estimate of the identified language is the same as finding the language model in which Xhas the highest probability of occurring. Putting all these parts together we get the general architecture of LID systems:

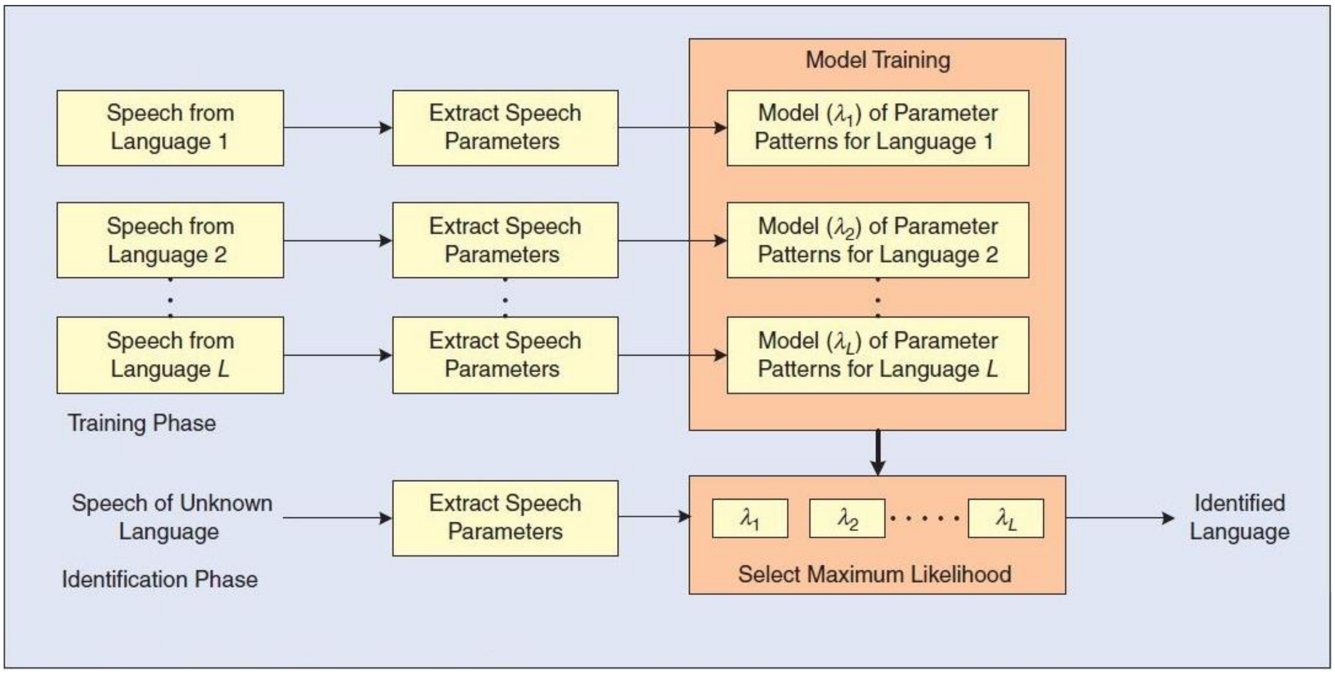

Model Training
Speech from Language 1
Extract Speech Parameters
Model (λ_1) of Parameter Patterns for Language 1
Speech from Language 2
Extract Speech Parameters
Model (λ_2) of Parameter Patterns for Language 2
Speech from Language L
Extract Speech Parameters
Model (λ_L) of Parameter Patterns for Language L
Training Phase
Speech of Unknown Language
Extract Speech Parameters
λ_1
λ_2
λ_L
Select Maximum Likelihood
Identified Language
Identification Phase

Figure 2.1 Basic block diagram of a LID system [14]

2.3 Speech Information regarding Language Identification

When using a LID system various types of information are taken into account. A certain classification of these information has been made, inspired by human's understanding mechanism. Studies have approached the methods which people use in order to discriminate between languages, whether it is in a conscious manner or not. A broad classification has split the speech features into low level and high level. At the low level, the most common are the acoustic, phonetic, phonotactic and prosodic information. At the higher level, language identification can be made based on morphology and sentence syntax [14].

In the following figure we see a gradation of these features according to the level they are assigned to:

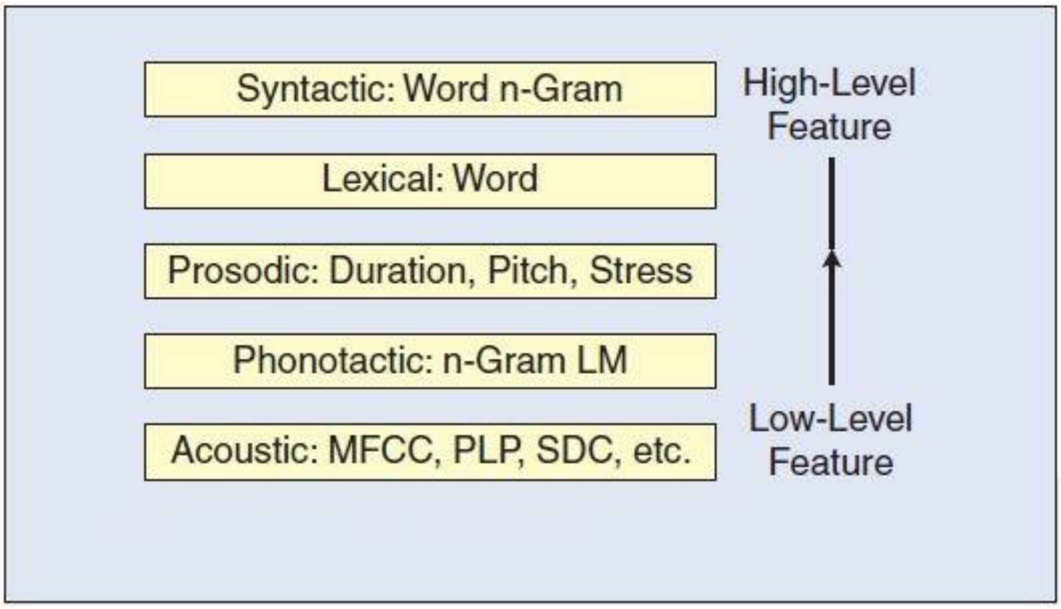


Figure 2.2 Levels of LID features [14]

The acoustic features, usually modeled by MFCCs are a compact representation of the input speech signal fulfilling a compression of the data contained in the audio waveform. The phonotactic features represent the admissible sound patterns that can be formed within a language. The N-gram language model (LM) is used to model these phonotactic features. The prosodic features make reference to duration, pitch and stress of speech and reflect elements such as the speaker's emotional state, which cannot be characterized by grammar. The lexical features address the problem of the internal structure of words. Lastly, the syntactic features are the outcome of the analysis of the way in which words are linked together in order to form phrases, clauses and sentences [14].

If we are to compare these two broad levels we can conclude that low-level features are easier to obtain but are very volatile and are easily affected by noise and speaker variations whereas high-level features contain more information regarding language discrimination. However, high-level features rely on large vocabulary recognizers, therefore on more training data. This ultimately leads to a higher complexity in obtaining these features. A brief description of each of these features is presented below.

2.3.1 Acoustic Information

Acoustic information is generally considered as the first level of speech production [19]. It is directly connected to the physical part of the speech, i.e. amplitude and frequency components of the audio signal's waves. It is the easiest to obtain form of information and it results from raw speech. Higher level features can be obtained from the acoustic information. The parameterization techniques used to model these type of information are MFCCs, Linear Prediction, PLP and Linear Prediction Cepstral Coefficients (LPCC). After basic features are obtained another intermediate step is done in which the temporal aspects of the signal are appended to each feature vector.

2.3.2 Phonotactic Information

It is known that humans can produce only a limited amount of sounds. What is of importance for LID systems is that not all of these sounds can appear in any given language. In fact, each language has its own set of sounds, out of which only a few are common to other languages as well. Phonotactics deals with the admissible phoneme combinations for each language. These phonemes' arrangement gains meaning or not depending on the given language. The phonotactics constraints are strong enough to offer a means of distinguishing a certain language. For example, Japanese does not allow two adjacent consonants but Danish and Swedish do [14]. There are many other examples based on which it has been concluded that phonotactics contain a great deal of information useful for LID.

2.3.3 Prosodic Information

Prosodic information pays attention to elements such as tone, stress, duration or rhythm. Intonation is defined as the variation of pitch during speech. The pitch, in term, is the fundamental frequency and it is used for tone representation. The intensity of one's speech is used for representing rhythm. In some Asian languages where intonation gives a certain meaning to a word [20] it is essential to explore this field. Stress can also have a strong impact on LID as some languages such as French have a word-final stress pattern as opposed to others that have a word-initial stress pattern such as Hungarian [14].

2.3.4 Morphological Information

Morphology is the field of linguistics that studies the internal structure of words [21]. Words are the fundamental units for syntax and they can be related to other words according to morphology rules. Word roots and how other words are formed are different across different languages. In a similar manner it is concluded that languages have their distinct vocabularies leading to a unique expression of these languages. Therefore, examining the characteristics of word forms can determine language discrimination.

2.3.5 Syntactic Information

As a definition, syntax is the study of the ways in which words are adjoined within a sentence. It is the study of the laws that words follow so they can form a meaningful phrase. Integrating word based grammars leads to a clear improvement of LID but it comes with a great price, as creating such a grammar is a very tedious task. It is a far more complex process than the commonly used phonetic level.

All in all, LID systems make use of the above mentioned types of information but it is not necessary that they use all of them. In fact, systems that integrate all of the above are very rare. Researches that have been made in this direction have shown that satisfactory results can be obtained by integrating only the acoustic and phonotactic information.

2.4 Acoustic Information used in the Front-End

The development of LID based on the acoustic information has been mane in a similar approach to the one used in speech recognition and speaker recognition. They are strongly related regarding the techniques used for representing the audio waveform of a signal. As it has been seen in Section 1.5.1 speech recognition relies on compressing the relevant parts of the speech into some coefficient vectors, namely MFCCs and PLP. It is the same case for LID as it aims to capture the essential differences between languages by modeling the distributions of spectral features directly [14]. In the front-end part of the system the audio signal is translated into a more compact and efficient representation which incorporates the most important aspect of speech characteristics and leaves apart the redundant information. The acoustic front-end of the system is composed of four main parts, illustrated in the figure below:

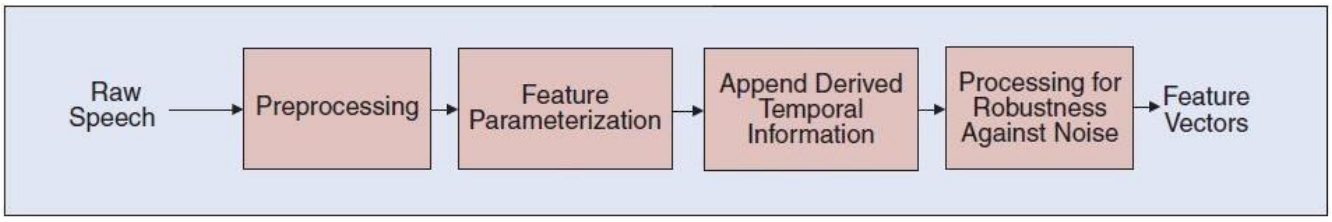

Raw Speech
Preprocessing
Feature Parameterization
Append Derived Temporal Information
Processing for Robustness Against Noise
Feature Vectors

Figure 2.3 Block Diagram of the Front-End System

These four blocks are necessary for obtaining the feature vectors from the raw speech's waveform and they work as follows:

- the preprocessing block takes the original signal as input and it includes voice activity detection, windowing and pre-emphasizing.
- feature parameterization refers to extracting only the data that is important for the back-end system in the distinguishing between languages process. The parameterization techniques will be briefly discussed in this paper as they are of equal importance to the LID and ASR systems alike.
- after the basic set of coefficients has been obtained some additional information regarding the temporal variation of the speech signal is appended. This comes in the form of delta, delta-delta cepstrum and SDC.
- the final block processes the signal such that it improves its robustness against noise and channel mismatch [14].

2.4.1 Mel Frequency Cepstral Coefficients (MFCCs)

The MFCCs are one of the most commonly used parameterization techniques used for speech and speaker recognition and LID as well. It is in fact a certain type of filter banks that best approximates the nonlinear frequency resolution of the human ear. After the magnitude-square of the Fourier Transform is calculated for the input windowed frame of speech, it is passed through a bank of triangular Mel filters and the natural logarithm of the filter bank energies is taken. The strong correlation between the log-energies imposes the need of a linear transformation such as the DCT to decorrelate the information resulting in the MFCCs [14].

2.4.2 Perceptual Linear Prediction (PLP)

The PLP coefficients come as an alternative to the MFCCs. They are useful because they incorporate some important features regarding human's subjective hearing characteristics such as critical band resolution, the equal-loudness curve and the intensity power law. The critical band is a similar approach to the MFCCs' filter bank noting that in this case we discuss about a different type of filter banks, namely the Bark filter banks. These are of trapezoidal shape, unlike MFCCs' triangular shape. The equal-loudness curve models the non-linear sensitivities of human hearing at different frequencies and the intensity power law models the non-linear relationship between the intensity of sound and the perceived loudness. After this process the auditory spectrum is estimated by an autoregressive all-pole model [14].

2.4.3 Delta and Delta-Delta Features

The two previously mentioned types of coefficients are both obtained on a short frame of the speech signal. However, the information that resides in the temporal dynamics of these features is very useful for both LID and ASR systems. The characterization of these dynamics is done in two ways:

- delta features which represent the velocity of the features, determined by its average first-order temporal derivative
- delta-delta features which represent the acceleration of the features, determined by its average second-order temporal derivative [4]

2.4.4 Shifted Delta Cepstra (SDC)

The previously described two features (delta and delta-delta) are effective when it comes to model temporal dynamics but they fail at modeling higher level temporal aspects of the speech signal because they only model the slope of the cepstra at the current point in time. This means that they are able to incorporate the temporal aspects of speech within short time windows.
LID benefits greatly from the assessment of the likelihood of one phoneme following another so in order to model the temporal aspects of a language it is necessary to take into account the transient nature of the acoustic sounds across time windows comparable to at least a phoneme's duration (which is somewhere between 50ms and 150ms) [14]. This is what the SDC has been proposed for, offering an alternative to including temporal information in the speech signal across longer time windows. They are obtained by concatenating a sampling of future delta cepstra with the current feature vector [14] and they depend on the number of basic cepstral streams to use in the calculation (the number of used MFCCs or PLP values), the number of frames from one delta calculation to the next, the total number of delta values concatenated together to form the SDC and the difference value used in the delta calculation [14].

2.5 Acoustic Information used in the Back-End

The back-end of the LID system has as main purpose the training of some form of model λ_l for each language to be recognized by the system. One very common language modeling scheme is the one that implies the modeling of the distribution of the acoustic features for each language by a separate Gaussian Mixture Model (GMM). Recently, another approach has appeared and it involves the training of a single common GMM for all the languages that are to be identified. This approach is called the Universal Background Model (UBM) and it is followed by an adaptation of a separate GMM for each language from that UBM, resulting in the GMM-UBM based LID [14].

2.5.1 Gaussian Mixture Model (GMM)

A Gaussian Mixture Model (GMM) is a parametric probability density function represented as a weighted sum of Gaussian component densities [22].

$$p(x|\lambda) = \sum_{i=1}^{M} w_i g(x|\mu_i, \textstyle\sum_i) \tag{2.4}$$

In the above equation x is a D-dimensional features vector, $w_i, i = 1,2,\dots,M$are the mixture weights and $g(x|\mu_i,\sum_i)$, $i = 1,2,\dots,M$ are the component Gaussian densities. Each component density is a D-variate Gaussian function of the form:

$$g(x|\mu_i,\textstyle\sum_i) = \frac{1}{(2\pi)^{D/2}|\sum_i|^{1/2}} \exp\{-\frac{1}{2}(x-\mu_i)^T \textstyle\sum_i^{-1}(x-\mu_i)\} \tag{2.5}$$

In Equation 2.5 μ_i represents the mean vector, $\sum_i$ the covariance matrix and it must also be taken into account that $\sum_{i=1}^{M} w_i = 1$. A complete description of the GMM can be given using only the mean vectors, covariance matrices and mixture weights, usually written as follows [22]:

$$\lambda = \{w_i, \mu_i, \textstyle\sum_{i_i}\} \quad i = 1,\dots,M \tag{2.6}$$

Training these GMMs involves forming an estimate of the probability density distribution that best characterizes the set of training data [14]. The common method to do so is using the Expectation Maximization (EM) algorithm. The use of GMMs is motivated by the fact that the individual components of the GMM can be considered to model the acoustic classes produced by that language so they can be used in training a general model of that acoustic class. A second motivation is that GMMs give a smooth approximation of varying distributions such as those produced by human speech [14] which otherwise would be very difficult to model given their strong variations.

2.5.2 GMM-UBM

GMM-UBM has been applied at first for speaker verification and then extended to LID. It has quickly become the dominant technique for acoustic based LID. Its functionality blocks are presented in Figure 3.4. In the training phase we can observe two different stages. The first one implies the training of a single GMM with the data gathered from all the languages that are desired in the testing phase. This is done by collecting a set of feature vectors from each of the languages in question and results in the UBM, which represents the characteristics common to all the different languages. In the other stage, training data from each language is used together with the UBM to train a specific GMM for each language, making use of the Maximum Aposteriori (MAP) adaptation, also known as Bayesian adaptation. The process behind this adaptation relies on the fact that the parameters for the Gaussian mixtures which bear a high probabilistic resemblance to the language specific training data will tend towards the parameters of that training data whereas the parameters of the GMMs bearing little resemblance to the language specific data will remain fairly close to their original UBM values [14]. This type of adaptation is often applied only to the means of the mixture components instead of the means, mixtures and weights. A block diagram of the working principle is illustrated in the following figure:

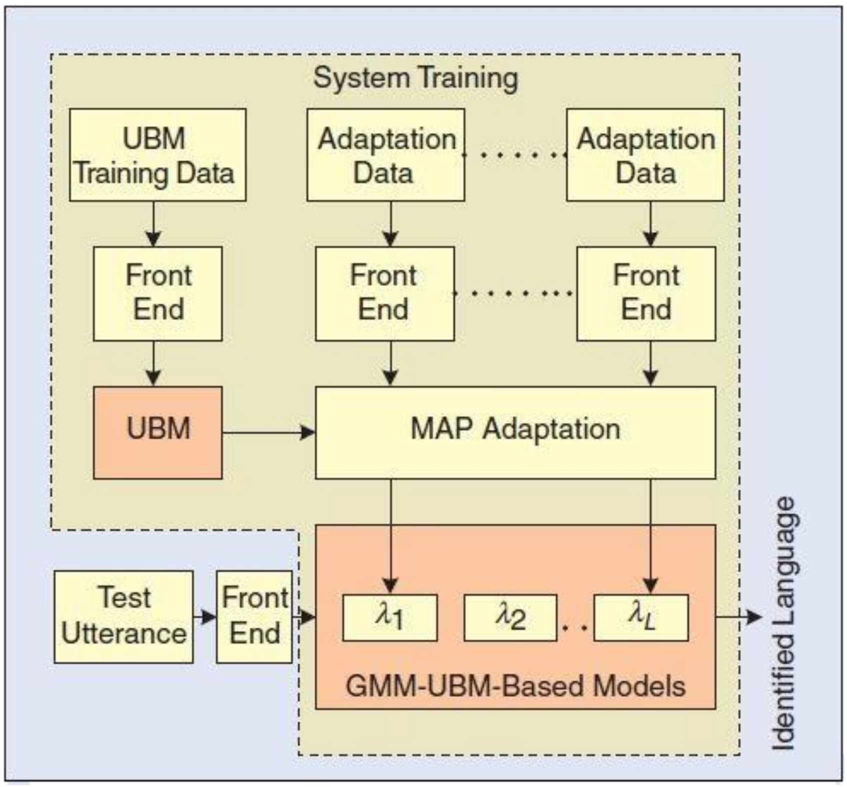


Figure 2.4 General Block Diagram of GMM-UBM Functionality [14]

2.6 Phonotactic Information in LID systems

Phonotactic information is of great importance for a LID system representing one of the main decision factors in determining the language selection. There are several techniques used at this level out of which we will discuss the two most important. For the current LID system it is worth mentioning that phoneme recognition is a key feature on which the identification mechanism heavily relies on.

2.6.1 Phoneme Recognition followed by Language Modeling (PRLM)

In the case of PRLM systems phonetic information is first extracted from the speech data using a phoneme recognizer and yields at the output a sequence of phonemes $\psi = \{\psi_1, \psi_2, \dots, \psi_P\}$. It is immediately followed by N-gram LMs which estimate the likelihood of certain phoneme sequences inside each target language. We remind that an N-gram LM gives an estimation of the probability of a certain phoneme to appear given the sequence of the previous N-1 phonemes. Afterwards, these N-gram LMs can give the LM λ that best reflects the phonotactic information about that language [14]. In order to give a comparable measure for language discrimination a likelihood score is computed for each language when an utterance is analyzed. This score is calculated with the following formula:

$$\mathcal{L}(\psi|\lambda_l) = \sum_{p=N}^{P} \log P\big(\psi_p \big| \psi_{p-1}, \psi_{p-2}, \dots, \psi_{p-(N-1)}, \lambda_l\big) \tag{2.7}$$

In Equation 2.7 λ_l represents the language model corresponding to the language l and $P\big(\psi_p \big| \psi_{p-1}, \psi_{p-2}, \dots, \psi_{p-(N-1)}, \lambda_l\big)$ represents the probability of the N-gram event $\{\psi_{p-(N-1)}, \dots, \psi_p\}$ estimated from λ_l [14]. The decision regarding the uttered language is then taken according to:

$$\hat{l} = \arg\max_{1\le l\le L} \mathcal{L}(\psi|\lambda_l) \tag{2.8}$$

In order to obtain a LID system based on such an approach it is necessary to have the phonetic transcription of the entire training corpus. This is a strong limitation as there are little available resources when it comes to phonetic transcriptions. Moreover, a phonetic recognizer must be implemented having at its output a viable phoneme sequence to work on.

2.6.2 Parallel Phone Recognition followed by Language Modeling (PPRLM)

An advance in LID has occurred when a phonotactic based LID system with a single decoder with a multilingual repertory and a variable number of phoneme units was implemented. This system uses multiple phoneme recognizers in the front-end part and it thus obtains the statistics of a language. Each of the phone recognizers will give different results according to the different characteristics that their specific language has relative to the acoustic features. Putting all these recognizers together a parallel phone recognizer is formed which is able to characterize the spoken language from a broader perspective [14].

The multiple phoneme recognizers introduce a higher robustness than in the case of a single phoneme recognizer due to a larger number of phonotactic models. However, it is a more complex task to run several recognizers at the same time and the processing speed is decreased. Thus, an improvement in accuracy comes at a higher cost. The working scheme of PPRLM is illustrated below:

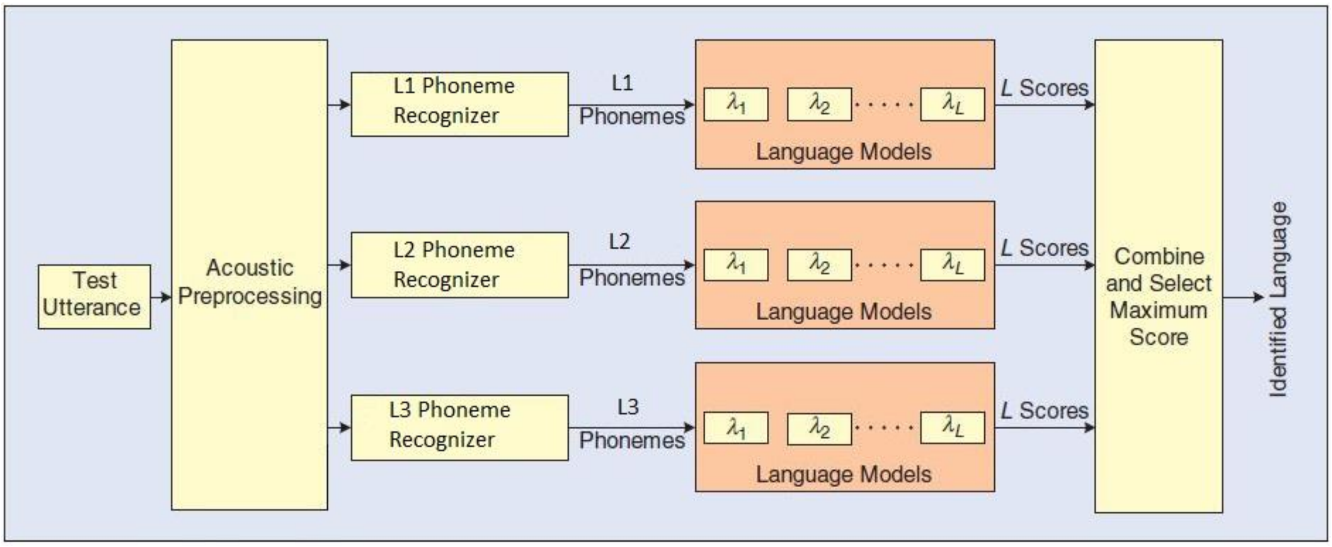


Figure 2.5 Block Diagram of PPRLM LID [14]

2.7 Prosodic, Morphological and Syntactic Information

As stated in Section 2.3.3, prosodic information stores information relative to the fundamental frequency and amplitude of the signal. Each of these may influence the LID process. Prosodic information contains duration, the pitch pattern and stress pattern in human linguistics [14]. Different combinations of the prosodic features make for different LID systems. However, the most effective approach to LID makes use of the entire available knowledge about a language's features, namely lexical and grammatical information of a language, but it comes at the price of higher complexity due to the large vocabulary needed for LID. In order to implement such a complex system which decodes a speech utterance into strings of words it is necessary to include both acoustic and phonetic features into the speech recognition process. This means that it will lead to the best accuracy as it uses most or all levels of speech information. When comparing accuracy and complexity a compromise is imminent and an acceptable one is to resort only to the first two feature levels, i.e. the acoustic and phonotactic levels in order to build a LID system.

Chapter 3. ASR Systems' Implementation and their Performances

Implementing the ASR systems before going on to the LID system was a key point in this thesis. As mentioned in the previous chapter it is vital to have a robust ASR system to build the identification mechanism on as they use similar approaches and rely on many common techniques. It is important to note that the main purpose of this thesis was to make a proof-of-concept project in which functionality overcame high performance regarding the amount of effort put in its development.

The ASRs that were developed in this project aim three different languages with the following motivations: English because it is among the most popular languages that are used in speech recognition systems and is very well documented, Romanian because it is our mother tongue and it is very interesting to see the results for it given the fact that almost anybody can assess its performance and finally, Albanian because we wanted to prove that it is possible to build such systems for a language which is not approached at all. That being said it is also worth mentioning that out of these three languages English is the only *rich-morphology language*. While Romanian has been studied by some native researchers it is fair to say that this study of Albanian by non-native speakers is amongst the first that were put to practice. This is the reason why Albanian has been given more attention than the other two, aiming to overcome resources barriers.

The ASRs proposed in this thesis address large vocabulary, speaker independent and continuous speech. In other words there are very few limitations on the way a person is supposed to make its speech for it to be analyzed by these systems. Speakers are not required to talk with a certain tonality, intensity or to train their speech before uttering it. However, some limitations appear regarding the quality of the recording, the noise level in the environment where the speech was recorded and the context of the speech. All of these will be further discussed in the following sections.

The tool that was used in order to train the models and perform the decoding process on input speech is the Carnegie Mellon University (CMU) Sphinx toolkit which is an open source project, freely available for any user. It is regarded as the best one at the moment as it offers many tutorials and there is a large community willing to aid its development but also its support.

3.1 Working with CMU Sphinx

The speech recognition process involves two phases: the training phase and the decoding phase. For the training phase it is necessary to build an acoustic model, a language model and to integrate them in the system, as stated in Chapter 1. These models consist of the following:

- a language model file
- a speech database (audio files with specific name patterns that contain speech)
- a transcription file (a text file which includes the written message found inside an audio file). These transcriptions must have certain identifiers which correlate the transcript with the audio file with the same name
- a fields file (a text file which includes the names of every audio file and transcription, of course)
- a phonetic dictionary (a text file which contains a list of all possible words that can be recognized and their phonetic transcription)

All of the above files will be discussed and analyzed in the following sections. The CMU Sphinx toolkit requires its training files to follow a certain pattern in order to work. This introduces a certain constrain on the resource gathering process and it imposes many additional processes.

The CMU Sphinx toolkit also offers language model evaluation and results evaluation tools. They also provide a Java development kit for further customization of the project which is also free to download. With the help of these tools it is possible to build an entire ASR and evaluate it as well given that one already has the necessary resources.

3.2 Resources gathering and analysis

A very important aspect to take into account when building a robust ASR is the database used for training. This can also be split into the audio database and the text corpus.

3.2.1 Audio database

The audio database consists of audio recordings of people uttering different statements. It is desirable to have a large diversity of recorded speakers, both males and females and of different ages as their voices tend to be very different. Certain health factors such as the common flu can change one's voice's features, namely its pitch making it similar to other natural nasal voices. The system best performs if it is trained with every type of voice but the number of audio samples must be weighted according to the probability of encountering a similar kind of voice in reality. For example, we cannot train a system with 90% male voices and expect it to correctly recognize female voices.

Another important aspect regarding the audio database is that all audio samples used for training must be "clean". By that we mean that there must not be any environmental noise, the speech signal must have adequate amplitude and sentences must be uttered at the same approximate speed. Other factors such as long pauses, coughs, stuttering or laughter can also appear in the audio signal so they must be treated as well. Furthermore, audio files used both for training and testing must be sampled at 16 kHz with 16 bits used to represent each sample.

Speaker independency, which is the desiderate of this thesis, is a concept very difficult to obtain because it requires a very large amount of training data, from many speakers. CMU Sphinx gives some empirical data as to what it is required to achieve satisfying performances. These numbers can be found in the table below:

ASR Task	Speaker dependent system	Speaker independent system
command and control (SV-CSR)	1 hour of recordings, 1 speaker	5 hours of recordings, 200 speakers
dictation (LV-CSR)	10 hours of recordings, 1 speaker	50 hours of recordings, 200 speakers

Table 3.1 CMU Sphinx suggested databases size [4]

The CMU Sphinx tutorial proposes these databases for several tasks. The command and control task is generally a short vocabulary pseudo-continuous speech recognition system and it is supposed to be suitable for short phrases of up to five words. As this is not a very complex application there is not a special need for numerous hours of training. For the dictation task, on the other hand, an increase of up to ten times is observed. This is due to the fact that dictation involves free speech on any topic. What the two have in common is that if a speaker independent system is desired then the number of different speakers is very large, around two hundred. The explanation for this resides in the great differences found between different speakers, as stated above.

For this thesis we have used different databases for each language as illustrated in Table 3.2. The acquisition of the Albanian speech database was the most complex part of the resources gathering because there was no free Albanian database available. Therefore we had to find a way in which we would acquire enough speech files and associate them to their respective transcription. This was resolved in a manner that did not guarantee "clean" audio clips but they were good enough to setup a decent ASR. This process took place in the following manner:

- firstly, three Albanian news websites (www.balkanweb.tv, www.vizionplus.tv, www.top-channel.tv) have been investigated and they have been discovered to contain many clips containing both audio and video information
- the second step was to iterate through all of their available pages and download them in the .html format

- a thorough search has been made on these .html files with the purpose of finding links that led to videos posted on www.youtube.com (because that is the site they would use to upload all their clips) and saved into a list
- the previous list was parsed such that only valid links would be kept for further use and the other ones would be discarded
- each of the valid clips was then downloaded from www.youtube.com using the “youtube-dl” tool under the .mp4 format. These clips would contain both video and audio information but we only needed the audio information
- the next step was to retrieve only the audio information from the .mp4 clips and to make sure that they have a good quality and storing them does not take up too much memory. This was made with the “ffmpeg” tool and the resulting clips contained only audio information, under the .wav file format, sampled at 16 kHz, represented on 16 bits. At this point we had extracted the audio information and separated them into different files for each news article.
- as we went through the audio clips we realized that many of them contained useless information such as commercials or music clips that were not correlated with the news article itself so we had to manually parse all of them so we would keep only the ones that contained useful audio information, corresponding to the news article’s information
- the audio files obtained in this way still had another problem under the form of multi-speaker audio clips. In other words, in a single audio file there were more than one speaker present and most of the times they did not have their entire speech transcribed in the news article. In order to solve this problem we had to apply a diarization process over these audio files. The diarization process would break an audio file into separate smaller audio files, each containing the amount of speech up to the next speaker change. The diarization process is illustrated in Figure 3.1.

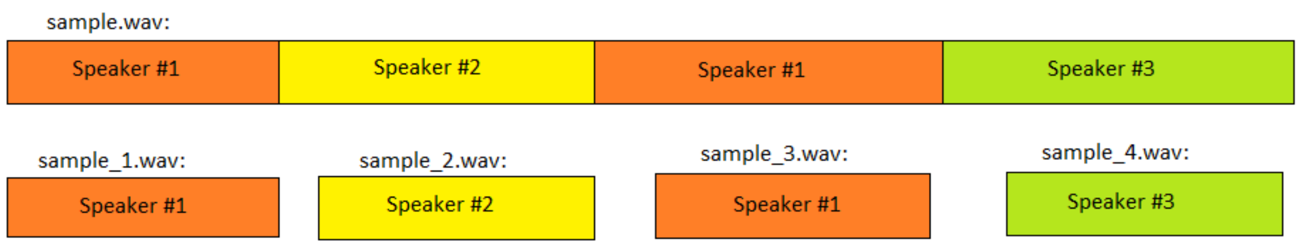


Figure 3.1 Diarization process example

- the diarization process was accompanied by a simultaneous parsing of the corresponding transcription file. In this step, a native speaker was following the target audio file and its transcription at the same time and make adjustments to the transcript where it was needed. This way we made sure that we have exact transcriptions for each part of the audio file.

Along to these audio files that we managed to extract from the Internet and bring them to the required format we also had another audio database provided by the Speech and Dialogue (SpeeD) laboratory, that was used for the MediaEval 2013 Spoken Web Search task. This database provided us with audio files that were recorded with the help of native speakers, both males and females, that were prompted a certain sentence on a screen and then asked to utter it. The entire audio database that was used for testing Albanian files is represented in the table below:

Database name	Total duration of files [h]	Type of files
SD1	2	recordings
SD2	3:20	broadcast news (web)
SD3	2	broadcast news (web)
SD4	5:40	broadcast news (web)

Table 3.2 Databases for Albanian ASR

The above databases go as follows:

- SD1 was created on the recordings used for the MediaEval 2013 Spoken Web Search task and it includes two hours of speech from twelve speakers, equally distributed between genders. These recordings were done in a controlled environment and have the best quality. It is the desirable case for any audio database.
- SD2 consists of audio files extracted from the vizionPlus news site. The number of speakers is difficult to evaluate because the audio files do not undergo a certain rule. They can have both male and female news anchors. Furthermore, they also contain interviews with different people and cannot be subject to any statistics without making an exhaustive manual search among the audio files. These files went through the diarization process and were manually transcripted by native speakers. While making tests on this database we discovered a major drawback in the form that its audio files are filtered low-pass at 5.5 kHz which comes in great contrast with what the system requires, namely files with the frequency range up to 8 kHz, sampled at 16 kHz. This cutoff frequency limits the amount of information that the system relies in the decoding process.
- SD3 is formed out of audio files selected from the topChannel news site. These files were parsed as well and a thorough transcription for each was created. They did not pose any problems, such as SD4 but the database is not as large as the others.
- SD4 was created from the topChannel news site as well. This database contains speech that does not have a perfect transcription but we noticed that many of this files corresponded to their respective transcription. Thus, we approached the problem in the following manner: from a larger list of audio files we ran the decoding process and obtained some results (hypothesis) which were aligned with the available transcriptions(references). Out of these we selected only those with higher recognition rates (number of correctly recognized words > 25%). Afterwards, we listened to each of the audio files and eliminated parts of the transcriptions that were not present in the uttered sentences. This led to a database with loose transcriptions for the audio files. Another important aspect is that these files were recorded in a noisy environment (recordings were done outdoors or in crowded rooms – interviews or political public speech).

Using these audio databases we have created the following acoustic models:

Acoustic model name	Audio database used for training the acoustic model
AM01	SD1
AM02	SD1+SD2
AM03	SD1+SD2+SD4
AM04	SD3
AM05	SD3+SD4

Table 3.3 List of Albanian acoustic models

In addition to the four audio databases mentioned above that were used for Albanian we must also specify the audio databases which were used for Romanian and English. This information can be found in the table below:

Database name	Total duration of files [h]	Type of files
TIMIT (English)	5:20-total; 3:50-training	recordings
SD5 (Romanian)	62-total;36-training	recordings

Table 3.4 Databases for English and Romanian ASR

Unlike most of the Albanian audio databases, these two databases contain only recordings, meaning clean audio files recorded in a controlled environment. The TIMIT database is available for free download on the internet and it contains utterances from 630 different speakers, both males and females. SD5 contains utterances from 18 different speakers, out of which 8 are males and 10 females. The Romanian database has been provided by the SpeeD laboratories and it was created there as well. These two databases did not require any further processing as they were already prepared to match the CMU Sphinx pattern.

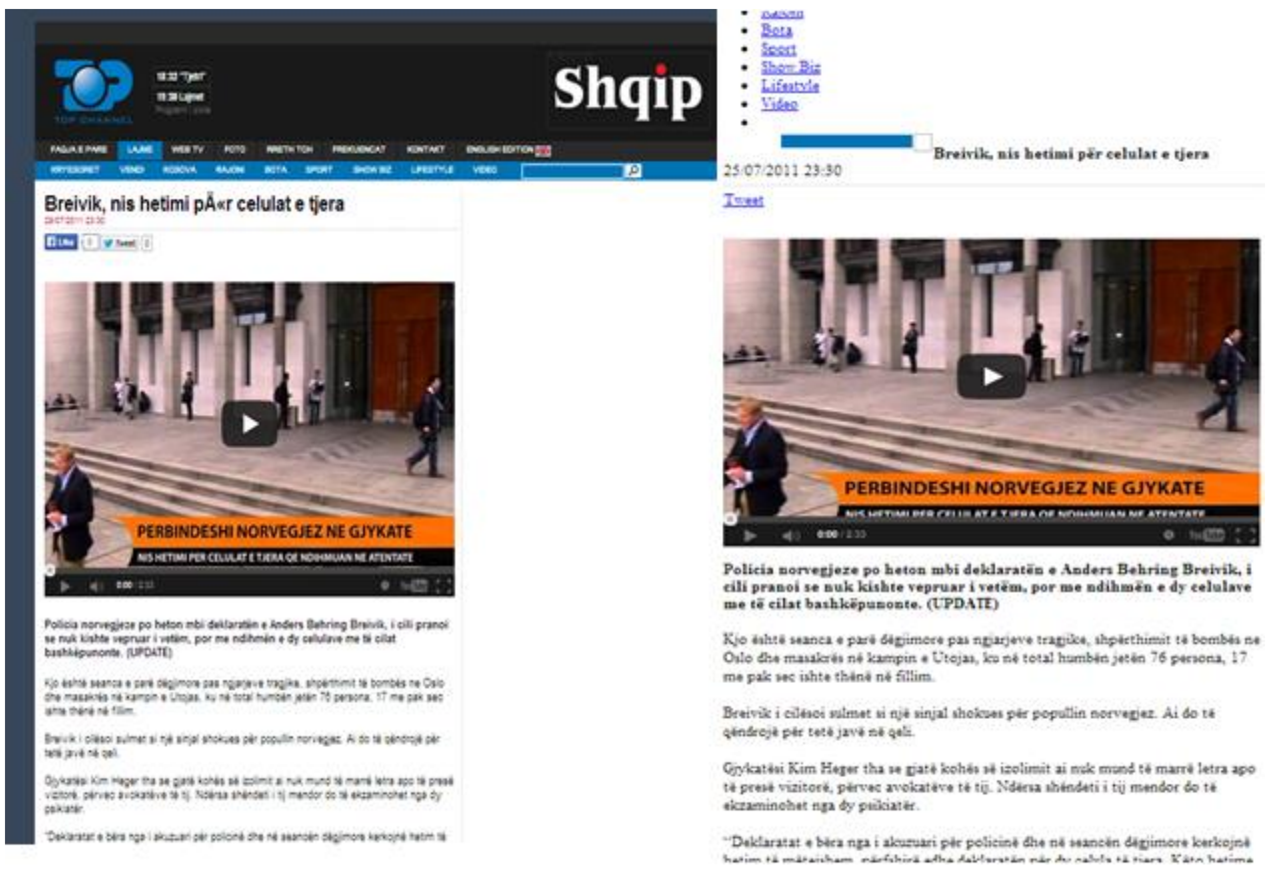
3.2.2 Text corpus

The text corpus is basically a text database, consisting of already transcribed utterances, thus forming meaningful sequences of words that can be found in different contexts. The text corpus is the starting point in any ASR system, along with the audio database, because it provides valuable information regarding many aspects. The text corpus gives relevant information about the language and it also contains the transcriptions of the audio files that are used to train the ASR system. We will approach in this section the matters of phonetic dictionary and fileids file as well, as they also contain only text.

As the audio databases for Romanian and English were already given so were their transcriptions. That left the Albanian language up for further processing so as we would get the desired pattern from the raw texts that were provided, based on the news websites' content.

3.2.2.1 Language models

The entire text databases are used not only as transcriptions of the audio files that the ASR system is trained with but also to create a phonetic dictionary, a list of phonemes and a language model, one for each of the languages in question. To start with, we will approach the extraction of text from the previously mentioned websites (www.balkanweb.tv, www.vizionplus.tv, www.top-channel.tv) and go through every step that was necessary to obtain a clean set of phrases. The idea behind text gathering was to access different news pages on the previously mentioned three news websites and to download their source pages in a more simplified format. This has been done automatically, with the help of a Java program and the results are illustrated in the figures to follow.



Breivik, nis hetimi pÃ«r celulat e tjera

Policia norvegjeze po heton mbi deklaratën e Anders Behring Breivik, i cili pranoi se nuk kishte vepruar i vetëm, por me ndihmën e dy celulave me të cilat bashkëpunonte. (UPDATE)

Kjo është seanca e parë dëgjimore pas ngjarjeve tragjike, shpërthimit të bombës ne Oslo dhe masakrës në kampin e Utojas, ku në total humbën jetën 76 persona, 17 me pak sec ishte thënë në fillim.

Breivik i cilësoi sulmet si një sinjal shokues për popullin norvegjez. Ai do të qëndrojë për tetë javë në qeli.

Gjykatësi Kim Heger tha se gjatë kohës së izolimit ai nuk mund të marrë letra apo të presë vizitorë, përvec avokatëve të tij. Ndërsa shëndeti i tij mendor do të ekzaminohet nga dy psikiatër.

"Deklaratat e bëra nga i akuzuari për policinë dhe në seancën dëgjimore kerkojnë hetim të

- Bota
- Sport
- Show Biz
- Lifestyle
- Video
-

Breivik, nis hetimi për celulat e tjera

25/07/2011 23:30

Tweet

Policia norvegjeze po heton mbi deklaratën e Anders Behring Breivik, i cili pranoi se nuk kishte vepruar i vetëm, por me ndihmën e dy celulave me të cilat bashkëpunonte. (UPDATE)

Kjo është seanca e parë dëgjimore pas ngjarjeve tragjike, shpërthimit të bombës ne Oslo dhe masakrës në kampin e Utojas, ku në total humbën jetën 76 persona, 17 me pak sec ishte thënë në fillim.

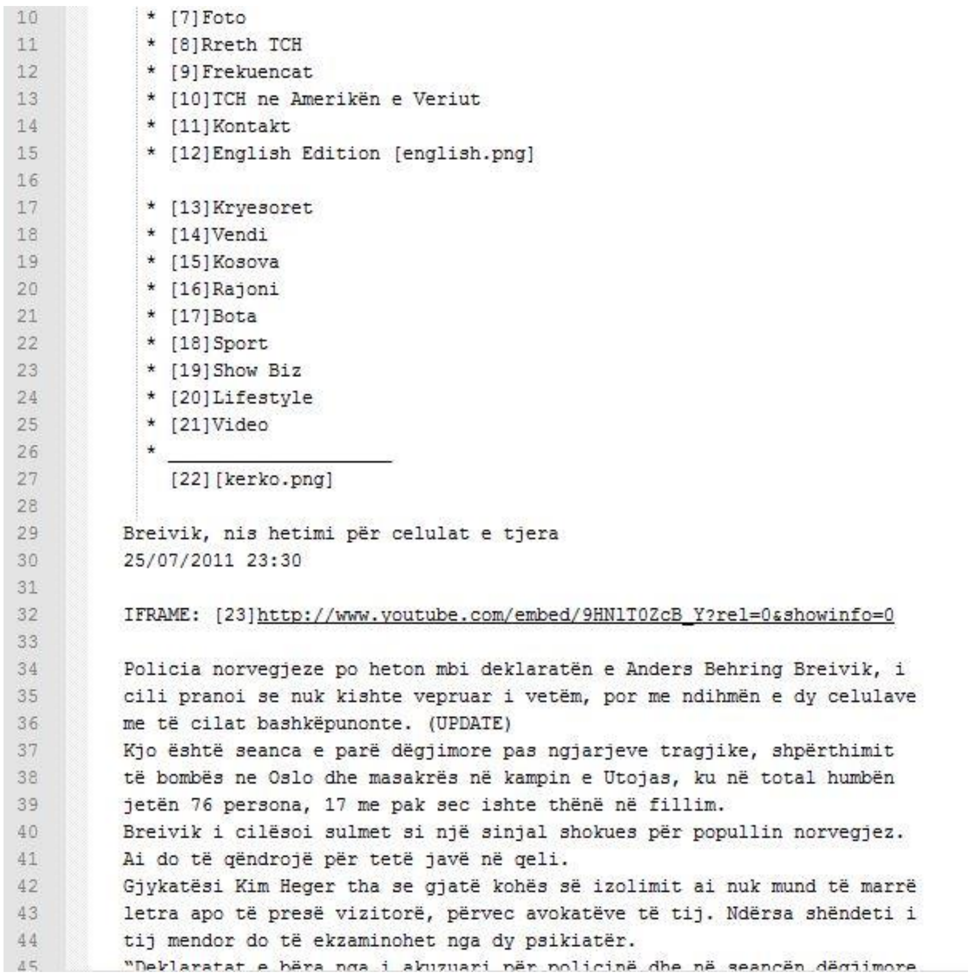
Breivik i cilësoi sulmet si një sinjal shokues për popullin norvegjez. Ai do të qëndrojë për tetë javë në qeli.

Gjykatësi Kim Heger tha se gjatë kohës së izolimit ai nuk mund të marrë letra apo të presë vizitorë, përvec avokatëve të tij. Ndërsa shëndeti i tij mendor do të ekzaminohet nga dy psikiatër.

"Deklaratat e bëra nga i akuzuari për policinë dhe në seancën dëgjimore kerkojnë

Figure 3.2 Website page in different
formats: the original page on the left-side, the downloaded format changed to .html on the right-side

Note that these pictures are valid in the case of the Top Channel website. In a similar manner, the other two websites have been approached.It can be observed that certain information, such as the background's color and numerous visual details, were removed from the original source of the web page and the useful information along with its respective video remained undamaged. We were presented with the .php version of the web page, and our task was to retrieve the news content and the audio clip associated to it. Firstly, a script that transformed the .php files into .html files by simply changing their extension was necessary in order to prepare the files for the next steps. The result is presented in the right-side, in Figure 3.2. Next, we created a script that would convert all the .html files into a more readable form, in the .txt format with the help of the "lynx" tool, available under Linux. In this way we have managed to eliminate other unimportant information and bring the file in a format that can be easily parsed and have the required information extracted from it, as it can be seen in Figure 3.3.

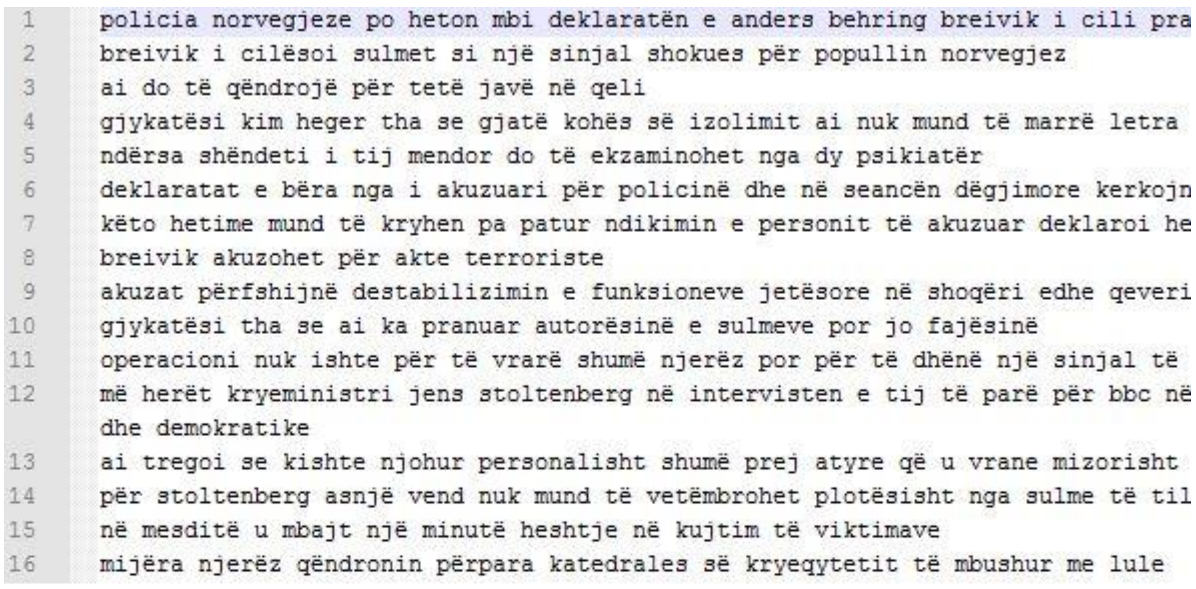


```
10       * [7]Foto
11       * [8]Rreth TCH
12       * [9]Frekuencat
13       * [10]TCH ne Amerikën e Veriut
14       * [11]Kontakt
15       * [12]English Edition [english.png]
16
17       * [13]Kryesoret
18       * [14]Vendi
19       * [15]Kosova
20       * [16]Rajoni
21       * [17]Bota
22       * [18]Sport
23       * [19]Show Biz
24       * [20]Lifestyle
25       * [21]Video
26       * ___________________
27         [22][kerko.png]
28
29     Breivik, nis hetimi për celulat e tjera
30     25/07/2011 23:30
31
32     IFRAME: [23]http://www.youtube.com/embed/9HN1T0ZcB_Y?rel=0&showinfo=0
33
34     Policia norvegjeze po heton mbi deklaratën e Anders Behring Breivik, i
35     cili pranoi se nuk kishte vepruar i vetëm, por me ndihmën e dy celulave
36     me të cilat bashkëpunonte. (UPDATE)
37     Kjo është seanca e parë dëgjimore pas ngjarjeve tragjike, shpërthimit
38     të bombës ne Oslo dhe masakrës në kampin e Utojas, ku në total humbën
39     jetën 76 persona, 17 me pak sec ishte thënë në fillim.
40     Breivik i cilësoi sulmet si një sinjal shokues për popullin norvegjez.
41     Ai do të qëndrojë për tetë javë në qeli.
42     Gjykatësi Kim Heger tha se gjatë kohës së izolimit ai nuk mund të marrë
43     letra apo të presë vizitorë, përvec avokatëve të tij. Ndërsa shëndeti i
44     tij mendor do të ekzaminohet nga dy psikiatër.
45     "Deklaratat e bëra nga i akuzuari për policinë dhe në seancën dëgjimore
```

Figure 3.3 Website page brought in the .txt format

It can be observed that the .txt format allows an easier parsing of the file as it has its information written on distinct rows and every other web page element has been replaced by a text marker. In Figure 3.3 we can see where the useful information starts (row 29, containing the title of the news) and the link to the associated video file (row 32). Every web page associated to a web site follows a certain pattern and it was up to us to discover it and use it to our advantage. In the present case it can be seen that the unwanted information for the text corpus ends on row 28, thus it can be eliminated. A similar pattern can be found at the end of the file that must also be eliminated. This header and footer patterns had to be manually identified for every news website and applied individually for the three raw text databases.

After isolating the useful information there is still need for some processing regarding punctuation and what the texts contain. Thus, all punctuation marks had to be removed, every phrase had to be written on a separate line, the entire text had to be written using only lowercase letters, some special symbols like "$" had to be replaced by their textual transcription, i.e. "dollars" and all phrases containing numbers had to be also removed because it would have been a very complex task to replace each numeral with its textual transcription. Furthermore, all tabs, trailing whitespaces and other characters or symbols that were not Albanian letters had to be erased. This posed an interesting problem because it appears that many different encodings were used when writing the news on the website and most of these special characters had to be manually identified and removed. A special program was written to perform all of the above tasks on an indefinite number of files, meaning that adding several files to the target directory would not change the way in which the program behaves and its sample output can be seen in Figure 3.4.



```
policia norvegjeze po heton mbi deklaratën e anders behring breivik i cili pra
breivik i cilësoi sulmet si një sinjal shokues për popullin norvegjez
ai do të qëndrojë për tetë javë në qeli
gjykatësi kim heger tha se gjatë kohës së izolimit ai nuk mund të marrë letra
ndërsa shëndeti i tij mendor do të ekzaminohet nga dy psikiatër
deklaratat e bëra nga i akuzuari për policinë dhe në seancën dëgjimore kerkojn
këto hetime mund të kryhen pa patur ndikimin e personit të akuzuar deklaroi he
breivik akuzohet për akte terroriste
akuzat përfshijnë destabilizimin e funksioneve jetësore në shoqëri edhe qeveri
gjykatësi tha se ai ka pranuar autorësinë e sulmeve por jo fajësinë
operacioni nuk ishte për të vrarë shumë njerëz por për të dhënë një sinjal të
më herët kryeministri jens stoltenberg në intervisten e tij të parë për bbc në
dhe demokratike
ai tregoi se kishte njohur personalisht shumë prej atyre që u vrane mizorisht
për stoltenberg asnjë vend nuk mund të vetëmbrohet plotësisht nga sulme të til
në mesditë u mbajt një minutë heshtje në kujtim të viktimave
mijëra njerëz qëndronin përpara katedrales së kryeqytetit të mbushur me lule
```

Figure 3.4 Parsed website page in its final format

The above text corpus corresponds to the news found at the address http://www.top-channel.tv/artikull.php?id=215708 (most recently accessed on 23rd June). At this moment a clean set of phrases has been obtained and another program was ran in parallel that iterated through the same files and extracted only those that contained a valid link to an online video. Based on these results we have created a list containing the names of the files in which the link was found and its respective link. The following process that led to obtaining the audio information from these links was previously presented in Section 3.2.1. At this moment we managed to retrieve the entire written information that was posted on the three websites during the past year and the audio files necessary for the audio database creation, correlated to their transcriptions.

This text corpus mining was important in the language modeling process. As stated in Chapter 2 it is necessary to have a text database as large as possible in order to obtain the best result for a language model. During the parsing process a total of 293939 files have been processed, the entire text corpus that was gathered from the websites contains 5378944 phrases adding up to a total of 53952942 words, distributed according to Table 3.4.

News website	Number of files	Number of phrases	Number of words
www.balkanweb.tv	124829	4479633	35504639
www.top-channel.tv	158408	825505	17164441
www.vizionplus.tv	10702	73806	1283862

Table 3.5 News websites parsed content

For the language model we used only 90% of the available text corpus in order to have a big enough text database (10% out of the total) on which to perform the language model evaluation. Thus, the obtained language model contains a total of 4841048 phrases, 48560551 words, out of which 377170 are distinct words.

With the language modeling process for Albanian being summed up we now refer to the Romanian and English text databases. These two languages did not require any further text processing actions as several language models were available for each of them. For the Romanian language model we have used a text corpus provided by the SpeeD laboratory, of 9794777 phrases containing 168519175 words, out of which 656647 are distinct words. For English we have trained a language model based on the transcriptions of the audio files provided in the TIMIT database, resulting in 6299 phrases, a total of 54375 words, out of which 6102 are distinct words. Several other language models have been trained with variations on the available text corpus in both the number of words to be taken into consideration and the weight associated to certain language models, all combined so that new language models with different properties would result. For all three languages we have also trained language models based only on the transcriptions of the audio files that were used in the training of the acoustic model, in order to have a reference model for further comparisons. We can find below an enumeration of the language models that were created for Albanian and the text corpuses that were used for their training.

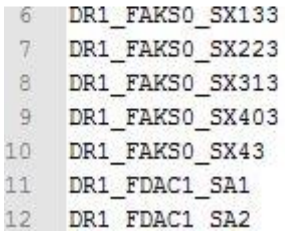
LM name	Database used for training
LM01	SD1
LM02	all news text corpus
LM03	SD1+SD2
LM04	SD1+SD2+SD4
LM05	SD3
LM06	SD4
LM07	SD3+SD4
LM08	SD3(90%)+all news text corpus(10%)
LM09	SD1+SD4(90%)+all news text corpus(10%)

Table 3.6 List of Albanian LMs

Note that on the right column, in the “SD” fields, we have denoted the audio database from whose transcriptions the language model was created and the “all news text corpus” label stands for the entire text corpus that was extracted from the news websites. The percentages found in the last two rows represent the weight associated to the respective language model when it was interpolated with the other language model written on the same row.

3.2.2.2 Fileids

The fileids files represent a listing of the audio files that we give as input for the ASR system’s training process. They are simply a list of the available audio files, from whose name the extension “.wav” was removed. An example of fileids file can be found below.

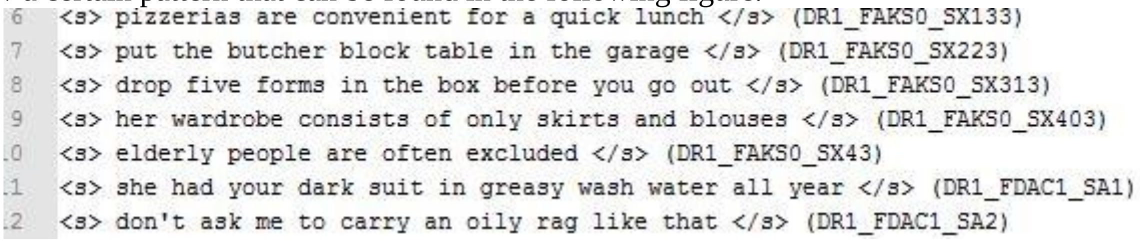


```
DR1_FAKS0_SX133
DR1_FAKS0_SX223
DR1_FAKS0_SX313
DR1_FAKS0_SX403
DR1_FAKS0_SX43
DR1_FDAC1_SA1
DR1_FDAC1_SA2
```

Figure 3.5 Fileids file example

3.2.2.3 Transcriptions

Another part of the text corpus is represented by the transcription files. These represent the written form of the speech that can be found within an utterance. These transcription files must follow a certain pattern that can be found in the following figure.



```
<s> pizzerias are convenient for a quick lunch </s> (DR1_FAKS0_SX133)
<s> put the butcher block table in the garage </s> (DR1_FAKS0_SX223)
<s> drop five forms in the box before you go out </s> (DR1_FAKS0_SX313)
<s> her wardrobe consists of only skirts and blouses </s> (DR1_FAKS0_SX403)
<s> elderly people are often excluded </s> (DR1_FAKS0_SX43)
<s> she had your dark suit in greasy wash water all year </s> (DR1_FDAC1_SA1)
<s> don't ask me to carry an oily rag like that </s> (DR1_FDAC1_SA2)
```

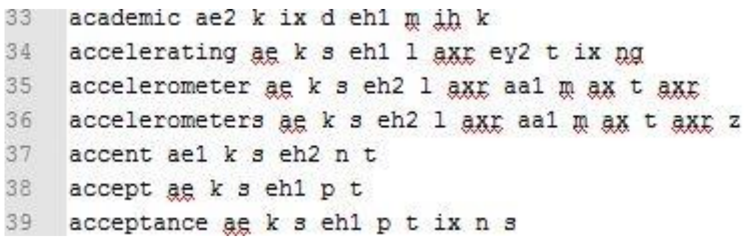
Figure 3.6 Transcription file example

The pattern used for the transcription files can be easily understood from Figure 4.5: every utterance is preceded by the “<s>” syntax and follow by the “</s>” syntax. After this we can find the audio file's name, without the ".wav" extension between round brackets. In order for the CMU Sphinx tool to work properly we need to provide it with a transcription file that corresponds precisely to the fileids file. The fileids file and the transcription file must be perfectly synchronized for the system to work and all files that are included in the fileids and transcription files must be contained in the folder containing the audio files.

3.2.2.4 Phonetic dictionary

The phonetic dictionary is a list of words from a certain language, followed by their phonetic transcription. For this, a list of phonemes is necessary to provide us with the available phonemes for that language. For the targeted three languages we used three different lists of phonemes with the following phoneme count: Albanian list of phonemes contains 36 phonemes, English contains 76 phonemes and Romanian contains 36 phonemes. These lists of phones contain a different phoneme on each row and no other text.

The phonetic dictionaries were obtained in different ways: the Romanian phonetic dictionary was automatically created based on phonetic rules and it was manually adjusted with numerous additions and corrections where it was the case, at the SpeeD laboratories. The English phonetic dictionary was provided along with the other English resources. The Albanian phonetic dictionary was created automatically based on a script in which phonetic rules have been implemented with the help of a native speaker. The phonetic dictionaries all have the same structure as presented in Figure 3.7.



```
academic ae2 k ix d eh1 m ih k
accelerating ae k s eh1 l axr ey2 t ix ng
accelerometer ae k s eh2 l axr aa1 m ax t axr
accelerometers ae k s eh2 l axr aa1 m ax t axr z
accent ae1 k s eh2 n t
accept ae k s eh1 p t
acceptance ae k s eh1 p t ix n s
```

Figure 3.7 Phonetic dictionary example

The lists of phonemes for each language will be presented in Chapter 5 as they better fit the approaches studied and explained in that chapter.

All of the above resources were either downloaded or created with the help of scripts conceived by us and all processes were brought to the form where they required only writing a single command line. All these scripts were written in such a manner that they would automatically perform all the desired actions and adding more files to the working directory would not affect the system's performances. Automatization was a must, given the large number of files that were processed, and execution time was also improved whenever it was possible as running these scripts was a time-consuming task.

The number of words in the phonetic dictionary it is also important for the outcome of the test as a large number of words can determine ambiguity for the system's decoding process whereas small number of words can be very restrictive in the context independent task that was proposed. There are 3 Albanian phonetic dictionaries containing 11k, 16k and 367k words, one English dictionary of 6k words and two Romanian ones containing 14k and 96k words. For Albanian and Romanian the dictionaries with 11k and 14k words, respectively have been used in order to have a certain balance between the languages.

3.3 Experiments and results

In the following section we will cover the experiments that were performed along with the configuration that was used at the time being, their results and an interpretation of these outcomes. All these experiments were evaluated relative to the percentage of correctly identified words within an utterance. This evaluation process was possible by aligning the decoding result (the hypothesis) with the original transcription for the target file (the reference) and interpreting the result according to the following aspects: if a word/sequence of words is found both in the hypothesis and reference files it is considered correctly recognized, if a word/sequence of words is found in the hypothesis file but not in the reference file it is considered insertion (I), if a word/sequence of words from the hypothesis file is similar to a word/sequence of words found in the reference file but not exact match then it is considered substitution (S), if a word/sequence of words from the reference file cannot be found in the hypothesis file then it is considered deletion (D). Having these in mind there are two evaluation metrics: word error rate (WER), calculated as the ratio between the sum of S, D and I and the total number of words, and accuracy, calculated as the difference between the total number of words and the sum of S, D, I, all divided by the total number of words. An interesting metric, however, is the number of correctly identified words as it gives a flavor of the ASR's performance.

In what it is to come we find the results of several experiments and their interpretations, most of them ran on the Albanian ASR as it was the most challenging of the three. On the first column we have the name of the acoustic models that were used and on the first row the names of the LMs used in the decoding process. The results are evaluated from the point of view of correctly identified number of words and WER, as accuracy is the complement of WER.

LM name / Acoustic model name	LM02	LM07
AM04	Correct=9.80% WER=90.38%	- -
AM05	Correct= 7.96% WER=92.24%	Correct= 16.02% WER= 85.54%%

Table 3.7 Albanian ASR tested on SD1

This experiment shows that running the decoding process on clean audio files when the acoustic model was trained with another audio database gives bad results, mainly because the audio files in that form the acoustic model have little to no resemblance with the audio files in the test database, mainly because of different contexts. This experiment shows the degree of context dependency of an ASR.

LM / AM	LM02	LM03	LM05	LM07
AM02	- -	Correct=46.53% WER= 54.37%%	- -	- -
AM04	Correct=8.82% WER=91.27%	- -	Correct=29.06% WER=71.48%	- -
AM05	Correct=4.35% WER=95.71%	- -	- -	Correct=10.23% WER=90.27%

Table 3.8 Albanian ASR tested on SD1+SD2

These experiments show a great difference when running the decoding process on the same database that was used for training the system and when running the decoding process on different databases. It can be seen that a system trained entirely on a database and tested on a completely different database gives the best result in the case of SD3 (WER=29%) meaning that these two databases share similar context and their speakers behave similarly, which was to be expected since the test database contains part news corpora and part clean transcriptions. This WER is better because of the news corpora part of the combined test databases.

AM \ LM	LM02	LM05	LM09
AM01	Correct=1.44% WER=98.58%	Correct=5.01% WER=95.04%	- -
AM02	Correct=26.34% WER=74.79%	Correct=68.49% WER=33.86%	- -
AM04	Correct=40.38% WER=61.62%	Correct=74.89% WER=27.61%	Correct=44.88% WER=59.13%

Table 3.9 Albanian ASR tested on SD3

From the experiments ran on SD3 audio database we can see the best results are obtained when we decode a database with the same audio files that were used for training. It is the most constrictive case but it also provides the best results. These experiments also show us that applying a different language model than the one obtained from the transcriptions used for training gives worse results as it introduces more degrees of uncertainty when it comes to selecting the words/sequences of words as the recognition outcome. These results were somewhat expected and they provide useful information mostly under constrained conditions such as command and control systems.

AM \ LM	LM06
AM04	Correct=33.55% WER=75.11%

Table 3.10 Albanian ASR tested on SD4

The last test on Albanian databases was ran mostly to verify if the manually selected audio files from SD4 matched the manually edited and parsed transcriptions that they point to. Generally, this is a bad result, but taking into account the method in which the database has been created and the factors that contributing to deter the result we can state that it is an acceptable result.

A few remarks must be made regarding the Albanian recognition task because they highly influence the results and make them look worse than they really are. For starters, Albanian includes some diacritics that are easily mistaken for simple letters, i.e. “ë” is often mistaken for “e” and “ç” for “c” mostly in the case of words that differ only with one letter, namely with the ones implied in the confusion pair stated before. One such example is the word “të” which is very often mistaken for “te” with both words existing in Albanian and a confusion between the two words is very likely since they sound very similar. Another aspect that must be taken into consideration is that some words/sequences of words are recognized as different words/sequences of words that have similar transcriptions, such as “individët” and “individë”. These kind of confusions are also often encountered and they have a negative impact on the total WER even if a person would still understand the message behind the transcription. Given the limited amount of resources available from the start (only SD4) and the fact that these methods were manually designed and implemented and are scalable we can say that the results were better than expected.

For English and Romanian the case is somewhat different because there were not any different databases available for training and testing respectively. Thus, the available database was split into test and train parts. The English database involved 3:50:00h of training and 1:30:00h of testing, while the Romanian database was trained on 36:00:00h and tested on the same amount but on different files. The English LM was created based on the transcriptions available in the TIMIT database, containing 54k words out of which 6.1k are distinct words and the Romanian transcription file was created on a text corpus of 254k words with 10k distinct words, provided by the SpeeD laboratories. The results are:

English database	Correct=42.16% WER=60.78%
Romanian database	Correct=50.32% WER=64.33%

Table 3.11 Romanian and English Results

Adding up the results for each of the three language and taking into consideration only the case of decoding a database that contains audio files which were not used for the training process we get the correctly recognized words comparison seen in Figure 3.8. For Albanian we have regarded the testing of SD1+SD2 database with both language and acoustic models built on SD3, this being the most restrictive case that can be encountered for an ASR, imposing more strict conditions than in the case of English decoding. When analyzing this result we must also note that the diacritics and the minor, but often encountered, confusions as stated above strongly influence the correctness of this decoding.

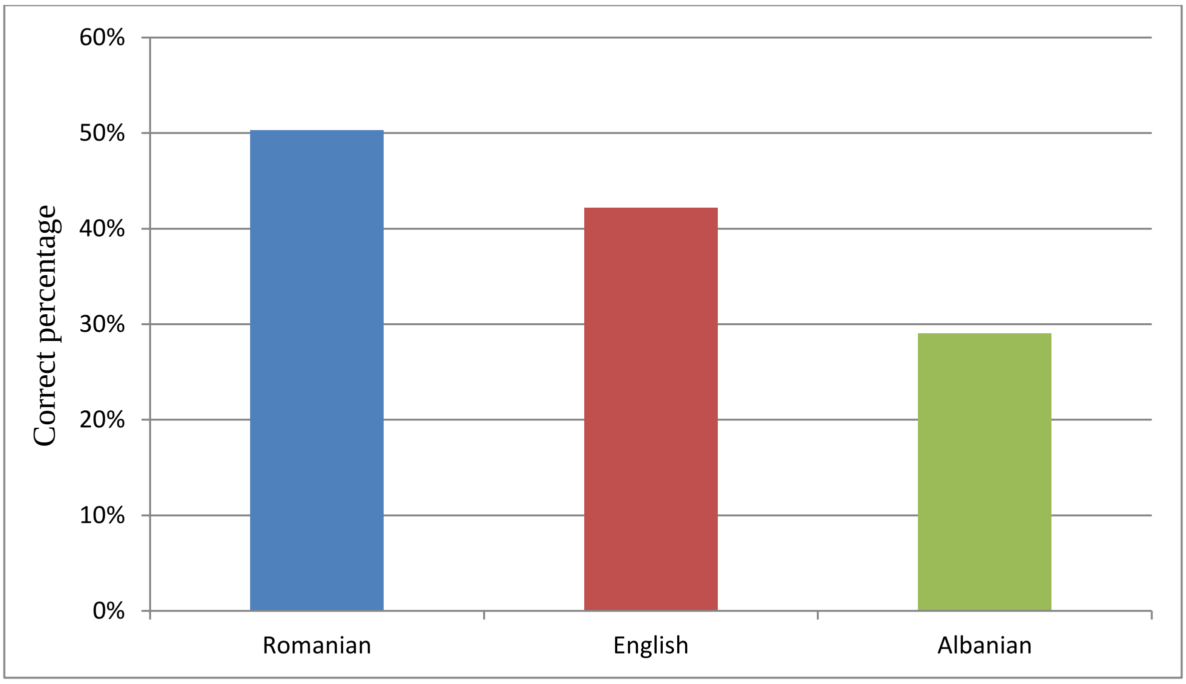


Figure 3.8 Final ASR results

Drawing a conclusion out of the experiments' results we can state that we have successfully created an ASR system for Albanian with similar performances to English and Romanian given the major drawback represented by the acoustic and text resources. This method can be subject to further optimization and improvements but it was demonstrated that is represent a strategy worth taking into consideration.

Chapter 4. LID Systems' Performance and their Implementations

As mentioned in the previous chapter, building the multilingual ASR systems was a crucial point because all of the techniques applied for the LID systems rely on the previous knowledge studied for the ASR part. It is worth mentioning that some of the programs that were used for the LID system development are open-source, thus can be modified over time by different people and that is why their stability is not guaranteed. The studied approaches came as an addition to the existing source code, bringing it a new flavor, exploiting some of the existing characteristics in a new different manner.

The aim of the this work is to make a study over some language identification methods, decide on which has the most accurate results and determine which of them is the most efficient taking into account their complexities, processing times and accuracies.

The LID systems that were created and tested in this thesis make use of the same databases as the ASR systems, some of the approaches rely heavily on the results of the ASR systems and that brings an unwanted dependency between the proposed LID and ASR systems. This dependency reflects in the fact that if the ASR system has poor performances then the LID system would also be faulty in its output, even if the language identification is done according to the technique that outperforms the others. This is why an evaluation of the LID system can be ambiguous and not give the best estimate for its accuracy. Imposing several restrictions on the developed methods gave comparable results that will be further discussed as we go through every applied method.

It is also important to note that this chapter aims only to identify the spoken language and very little importance is given to the recognized sequence of words. The thing that interests us is the language to which that sequence of words or phonemes, as we will see, belongs to.

4.1 Phoneme recognition method

The first strategy that was proposed for the LID system refers to phoneme recognition. In Chapter 4 we addressed the term of phonetic dictionary and defined it as a list of words, one on each line together with their phonetic transcription. In Chapter 2 we introduced the term "phoneme" as the fundamental unit of speech. Now we exploit these phonemes and how they help us decide over language identification. In Table 4.1 we can find a listing of all the phonemes corresponding to the three languages, written in a common encoding, making it easier to read and write them on a computing system. As it was stated in Chapter 3, we can use phonotactics to identify a language from a series of previously trained languages.

Phoneme	Word example (written form)	Word example (phonetic form)	Phoneme	Word example (written form)	Word example (phonetic form)
a	aflate	**a** f l a t e	a	afatet	**a** f a t e t
a1	află	a f l **a1**	b	baletit	**b** a l e t i t
b	aibă	a i3 **b** a1	c	ciko	**c** i k o
d	data	**d** a t a1	c1	diçka	d i **c1** k a
e	alune	a l u n **e**	d	dita	**d** i t a
e1	asemenea	a s e m e n **e1** a	dh	edhe	e **dh** e
f	asfalt	a s **f** a l t	e	ekonomia	**e** k o n o m i a
g	asigura	a s i **g** u r a	e1	dëmtuar	d **e1** m t u a r
g1	atinge	a t i n **g1** e	f	fakt	**f** a k t
g2	gheb	**g2** e b	g	figurat	f i **g** u r a t
h	hadroni	**h** a d r o n i1	gj	gjatë	**gj** a t e1
i	găsit	g a1 s **i** t	h	historia	**h** i s t o r i a
i1	găști	g a1 s1 t **i1**	i	hiçi	h **i** c1 i
i2	dâmb	d **i2** m b	j	abetarja	a b e t a r **j** a
i3	hotelului	h o t e l u l u **i3**	k	kaq	**k** a q
j	just	**j** u s t	l	klan	k **l** a n
k	cojoc	k o j o **k**	ll	vullnet	v u **ll** n e t
k1	colaci	k o l a **k1**	m	kemi	k e **m** i
k2	deschide	d e s **k2** i d e	n	kenë	k e **n** e1
l	destul	d e s t u **l**	nj	njeri	**nj** e r i
m	diametrul	d i a **m** e t r u l	o	njoftoi	nj o f t **o** i
n	din	d i **n**	p	paguar	**p** a g u a r
o	dizolv	d i z **o** l v	q	paqena	p a **q** e n a
o1	doar	d **o1** a r	r	parti	p a **r** t i
o2	maseur	m a s **o2** r	rr	rreth	**rr** e th
p	cip	k1 i **p**	s	rusisë	r u s i **s** e1
r	morav	m o **r** a v	sh	shba	**sh** b a
s	mosc	m o **s** k	t	sarajet	s a r a j e **t**
s1	mușc	m u **s1** k	th	theksoi	**th** e k s o i
t	muști	m u s1 **t** i1	u	thinjur	th i nj **u** r
t1	mutați	m u t a **t1** i	v	valbona	**v** a l b o n a
u	mărul	m a1 r **u** l	x	nxitjen	n **x** i t j e n
v	naiv	n a i **v**	xh	xhirua	**xh** i r u a
w	nouă	n o **w** a1	y	zyrtarë	z **y** r t a r e1
y	alurile	a l **y** r i l e	z	zoti	**z** o t i
z	miez	m i3 e **z**	zh	zhvillim	**zh** v i ll i m

a) b)

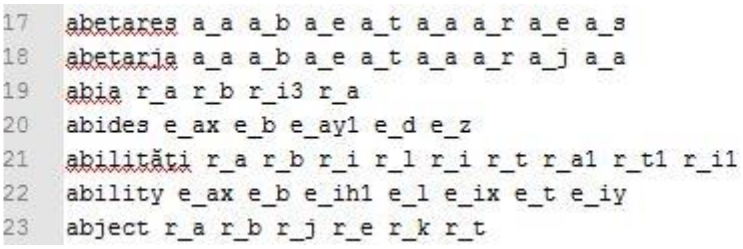
Phoneme	Word example (written form)	Word example (phonetic form)
aa	arcade	**aa** r k ey1 d
aa1	abolish	ax b **aa1** l ih sh
aa2	chaos	k ey1 **aa2** s
ae	comrades	k aa1 m r **ae** d z
ae1	crab	k r **ae1** b
ae2	diagram	d ay1 ax g r **ae2** m
ah	nobody	n ow1 b **ah** d iy
ah1	none	n **ah1** n
ah2	outcome	aw1 t k **ah2** m
ao	portray	p **ao** r t r ey1
ao1	pouring	p **ao1** r ix ng
ao2	sauce	s **ao2** s
aw	however	hh **aw** eh1 v axr
aw1	loudest	l **aw1** d ix s t
aw2	outside	**aw2** t s ay1 d
ax	parallel	p ae1 r **ax** l eh2 l
axr	parades	p **axr** ey1 d z
ay	tycoons	t **ay** k uw1 n z
ay1	type	t **ay1** p
ay2	upside	ah1 p s **ay2** d
b	variable	v ae1 r iy ax **b** el
ch	virtue	v er1 **ch** uw2
d	vivid	v ih1 v ix **d**
dh	weather	w eh1 **dh** axr
eh	aspects	ae1 s p **eh** k t s
eh1	assembled	ax s **eh1** m b el d
eh2	comment	k aa1 m **eh2** n t
el	colorful	k ah1 l er f **el**
em	column	k aa1 l **em**
en	cotton	k aa1 t **en**
er	energy	eh1 n **er** jh iy
er1	eternal	ih t **er1** n el
er2	sunburn	s ah1 n b **er2** n
ey	always	ao1 l w **ey** z
ey1	vacant	v **ey1** k ix n t
ey2	anyway	eh1 n iy w **ey2**
f	baffle	b ae1 **f** el
g	bag	b ae1 **g**
hh	forehead	f ao1 r **hh** eh2 d
ih	behavior	b **ih** hh ey1 v y axr
ih1	forbidden	f ao r b **ih1** d en
ih2	heroism	hh eh1 r ow **ih2** z
ix	hesitate	hh eh1 z **ix** t ey2 t
iy	highly	hh ay1 l **iy**
iy1	hyena	hh ay **iy1** n ax

iy2	increase	ih1 n k r **iy2** s
jh	indulge	ih n d ah1 l **jh**
k	inexact	ih2 n ih g z ae1 **k** t
l	inflated	ih n f **l** ey1 t ix d
m	informed	ih n f ao1 r **m** d
n	into	ih1 **n** t uw2
ng	ironing	ay1 axr n ix **ng**
ow	location	l **ow** k ey1 sh ix n
ow1	loads	l **ow1** d z
ow2	mango	m ae1 ng g **ow2**
oy	ellipsoids	ax l ih1 p s **oy** d
oy1	enjoy	eh n jh **oy1**
oy2	gunpoint	g ah1 n p **oy2** n t
p	happen	hh ae1 **p** ax n
r	hard	hh aa1 **r** d
s	heights	hh ay1 t **s**
sh	inertia	ih n er1 **sh** ax
t	inexact	ih2 n ih g z ae1 k **t**
th	length	l eh1 ng **th**
uh	modular	m aa1 jh **uh** l axr
uh1	poor	p **uh1** r
uh2	outputs	aw1 t p **uh2** t s
uw	ritual	r ih1 ch **uw** el
uw1	roof	r **uw1** f
uw2	statue	s t ae1 ch **uw2**
v	survey	s er1 **v** ey2
w	sweet	s **w** iy1 t
y	unit	**y** uw1 n ix t
z	used	y uw1 **z** d
zh	visual	v ih1 **zh** uw el

c)

Table 4.1 List of phonemes for: a) Romanian, b) Albanian, c) English

The most notable difference between the ASRs and the LID system is that the LID system uses all three databases, namely SD3, TIMIT for train and SD5 for train, as they were presented in Chapter 3, concatenated in order to form a single large database with audio clips and text files that cover all the targeted languages for its training process. This system uses different techniques for training and decoding but is somewhat similar to the multilingual ASR in that it is trained on the same audio databases, it uses the same transcriptions for training and it is done with the help of the CMU Sphinx tool. The phonetic dictionary used for training suffered some minor modifications, because every phoneme was added a prefix to mark the language it belongs to in the following way: every Albanian phoneme has been added the prefix “a_”, English phonemes have been added the prefix “e_” and Romanian ones the prefix “r_”.

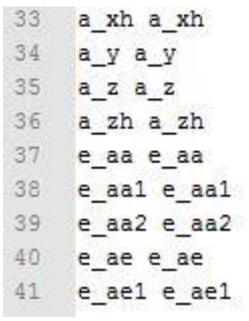


```
abetares a_a a_b a_e a_t a_a a_r a_e a_s
abetaria a_a a_b a_e a_t a_a a_r a_j a_a
abia r_a r_b r_i3 r_a
abides e_ax e_b e_ay1 e_d e_z
abilități r_a r_b r_i r_l r_i r_t r_a1 r_t1 r_i1
ability e_ax e_b e_ih1 e_l e_ix e_t e_iy
abject r_a r_b r_j r_e r_k r_t
```

Figure 4.1 Phonetic dictionary used for training

Using this type of dictionary we will have words from each of the three languages mapped to a phonetic transcription specific to that language, easily recognized by their prefix. This way, the system will know to train the acoustic model with language specific phonemes.

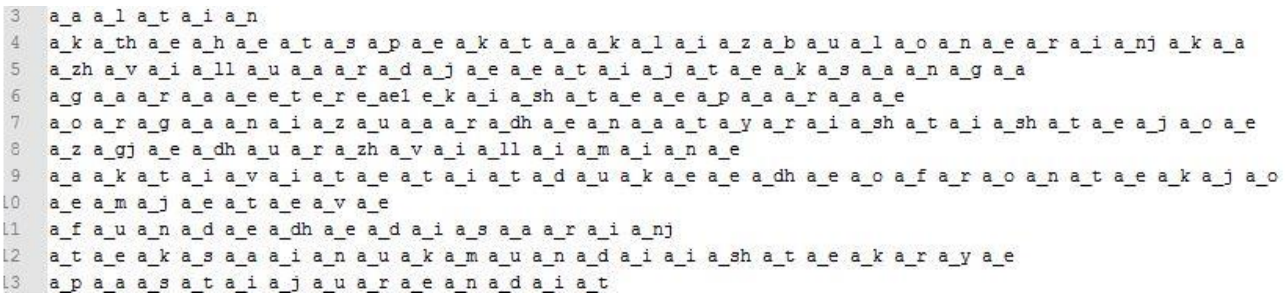
The phonetic dictionary used for decoding has been changed as well because of the fact that this system is not meant to recognize words but phonemes. That is why every phone must now be mapped to itself, instead of mapping words to their phonetic transcriptions. In the case of an ASR the system would provide a sequence of phonemes as output and according to the phonetic dictionary they would be joined in such a way to form complete words. For the phoneme recognition task we only need to recognize phonemes and mapping them to complete words is unnecessary.



```
a_xh a_xh
a_y a_y
a_z a_z
a_zh a_zh
e_aa e_aa
e_aa1 e_aa1
e_aa2 e_aa2
e_ae e_ae
e_ae1 e_ae1
```

Figure 4.2 Phonetic dictionary used for decoding

For the language model we also used a different technique because we are now interested to study not the sequences of words and based on them to build trigrams but the sequences of phonemes and build the trigrams taking sequences of phonemes as input, instead of words. For this, we had to bring the transcriptions file to a form in which it contains only phonemes and not words. Basically, we changed every word in the transcriptions file with its phonetic transcription according to the phonetic dictionary that was used in the training process by running a Java program that was specially created for this purpose. The transcriptions' form is represented in Figure 4.3.



```
a_a a_l a_t a_i a_n
a_k a_th a_e a_h a_e a_t a_s a_p a_e a_k a_t a_a a_k a_l a_i a_z a_b a_u a_l a_o a_n a_e a_r a_i a_nj a_k a_a
a_zh a_v a_i a_ll a_u a_a a_r a_d a_j a_e a_e a_t a_i a_j a_t a_e a_k a_s a_a a_n a_g a_a
a_g a_a a_r a_a a_e e_t e_r e_ae1 e_k a_i a_sh a_t a_e a_e a_p a_a a_r a_a a_e
a_o a_r a_g a_a a_n a_i a_z a_u a_a a_r a_dh a_e a_n a_a a_t a_y a_r a_i a_sh a_t a_i a_sh a_t a_e a_j a_o a_e
a_z a_gj a_e a_dh a_u a_r a_zh a_v a_i a_ll a_i a_m a_i a_n a_e
a_a a_k a_t a_i a_v a_i a_t a_e a_t a_i a_t a_d a_u a_k a_e a_e a_dh a_e a_o a_f a_r a_o a_n a_t a_e a_k a_j a_o
a_e a_m a_j a_e a_t a_e a_v a_e
a_f a_u a_n a_d a_e a_dh a_e a_d a_i a_s a_a a_r a_i a_nj
a_t a_e a_k a_s a_a a_i a_n a_u a_k a_m a_u a_n a_d a_i a_i a_sh a_t a_e a_k a_r a_y a_e
a_p a_a a_s a_t a_i a_j a_u a_r a_e a_n a_d a_i a_t
```

Figure 4.3 Phonetic transcriptions

Each sentence should contain only phonemes from a single language since we have no audio clip containing mixed languages speech. The language can be easily recognized by reading the prefix in front of the phoneme and this is the algorithm based on which the evaluation will be done as well.

Having brought the resources in the desired form, the LID system was trained and then the decoding process was ran firstly on the same audio database that the system was trained on in order to obtain a reference result and, afterwards, it was tested on different audio clips than the ones used for training. Then, we have sorted the output files according to their fileids into three categories: English, Romanian and Albanian. Afterwards, a Java program that was created for this purpose was ran and it would count all the occurrences of the prefixes "e_", "r_" and "a_" for each line in the resulted file. The maximum number determines the language that was uttered, with the restriction that if at least two of the prefixes shared the place for the maximum number of occurrences on a line or if there were not at least 6 phonemes recognized for a sentence then that sentence's language would be regarded as "unknown". The results can be found in the table below:

Tested language	Romanian phrases	English phrases	Albanian phrases	Unknown phrases	Correct phrases
Romanian	4876	0	0	124	97.52%
English	369	3883	268	99	84.06%
Albanian	0	0	561	283	66.46%

Table 4.2 Phoneme decoding results reference

Summing up all the correctly identified phrases and dividing them to the total number of test phrases we obtain a LID rate of 89.07%. Having these results as reference we performed another test on some audio clips that were not used for the system's training, namely SD1, TIMIT for test, SD5 for test. The results can be found in Table 4.3.

Tested language	Romanian phrases	English phrases	Albanian phrases	Unknown phrases	Correct phrases
Romanian	4998	0	1	110	97.82%
English	184	1332	125	39	79.28%
Albanian	80	149	733	6	75.72%

Table 4.3 Phoneme decoding results on a different audio database than the one used for training

Summing up the result it gives us a LID rate of 91.05%. It has been observed that the language model favored the languages with a greater number of words in the text corpus, because they would have more occurrences than the others resulting in greater probabilities. In order to overcome this drawback we applied a language interpolation script that combines two different language models, each of them being given a desired weight. As the script can be applied only for two language models we had to first make a language interpolation between two out of the three language models, each for a different language, and assign them equal weights (50%-50%). The resulted interpolated language model would then be interpolated with the remaining language model out of the three with the following weights: the already interpolated language model was given a weight of 66% and the other one 34%. This led to an equal distribution of weights between the three languages and the results are. The new language model was tested on the same database that was used for training to see how it would affect our reference results.

Tested language	Romanian phrases	English phrases	Albanian phrases	Unknown phrases	Correct phrases
Romanian	4857	1	16	126	97.14%
English	152	3748	599	120	81.14%
Albanian	0	0	561	283	66.46%

Table 4.4 Phoneme decoding results on trained database with interpolated LM

The total LID rate is of 87.6% in this case, slightly lower than in the reference case. We can observe a small increase in the total number of unknown phrases, due to the fact that equal weights of the three language models diminish the difference between phonemes' probability occurrence, leading to a higher number of confusions.

Adding all these various results together we have obtained the next table, for an easier understanding and reading with the following notations: "Reference" stands for the test performed on the training database, "Different database" stands for the test performed on a database containing audio files different from the ones used in the training process and "Interpolated LM" stands for the test performed on the database that was used for training with a LM that was assigned equal weights for each language. The results represent the percent of correctly identified phrases.

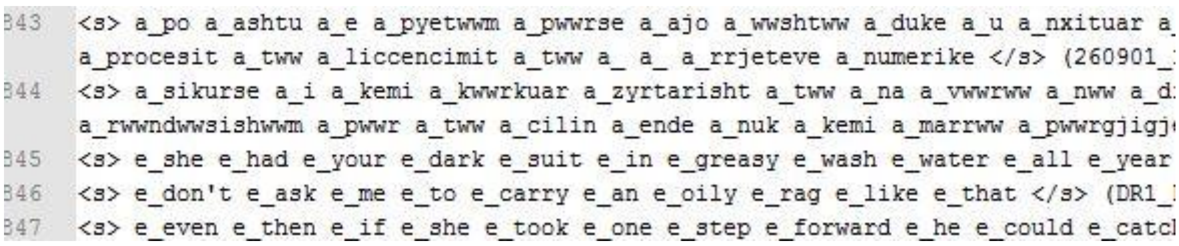
Tested language	Reference	Different database	Interpolated LM
Romanian	97.52%	97.82%	97.14%
English	84.06%	79.28%	81.14%
Albanian	66.46%	75.72%	66.46%

Table 4.5 Phoneme recognition results summary

All in all, this strategy provided a good LID rate with little additional effort. The most complex task was the creation of the new LM but given the fact that there was little additional knowledge required to put this method in practice [Annex 1] it can be considered an efficient method of LID.

4.2 Language specific word recognition method

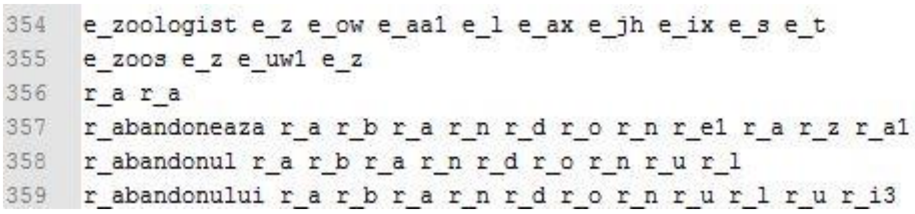
The second strategy proposed for the LID system is based on recognizing complete words having the prior knowledge regarding the language they belong to. The basic idea for this approach is to add a distinctive marker to each word so as we know what language it belongs to and after the decoding process to count the number of words specific to each language. In the end, the maximum number of words attached to a language determines the uttered language. For this method we used the same audio databases as in the previous method for training, namely SD3, TIMIT for train and SD5 for train, as they were presented in Chapter 3, concatenated in order to form a single large database with audio clips. The transcriptions have been slightly modified and so was the phonetic dictionary. Similar to the first method we added the "a_", "e_" and "r_" prefixes to each word and obtained the following transcriptions:



```
<s> a_po a_ashtu a_e a_pyetwwm a_pwwrse a_ajo a_wwshtww a_duke a_u a_nxituar a_
a_procesit a_tww a_liccencimit a_tww a_ a_ a_rrjeteve a_numerike </s> (260901_
<s> a_sikurse a_i a_kemi a_kwwrkuar a_zyrtarisht a_tww a_na a_vwwrww a_nww a_d
a_rwwndwwsishwwm a_pwwr a_tww a_cilin a_ende a_nuk a_kemi a_marrww a_pwwrgjigj
<s> e_she e_had e_your e_dark e_suit e_in e_greasy e_wash e_water e_all e_year
<s> e_don't e_ask e_me e_to e_carry e_an e_oily e_rag e_like e_that </s> (DR1_
<s> e_even e_then e_if e_she e_took e_one e_step e_forward e_he e_could e_catc
```

Figure 4.4 Transcriptions for word recognition LID example

As it can be seen, the difference between these transcriptions and the ones used for ASR lies in the additional prefix that maps every word to a certain language. The same technique has been applied for the text corpus that was used to create the LMs. The phonetic dictionary has also been modified and it was the same one used both for training and decoding having the language prefix added not only to each phoneme but also to every word as it can be seen in Figure 4.5.



```
e_zoologist e_z e_ow e_aa1 e_l e_ax e_jh e_ix e_s e_t
e_zoos e_z e_uw1 e_z
r_a r_a
r_abandoneaza r_a r_b r_a r_n r_d r_o r_n r_e1 r_a r_z r_a1
r_abandonul r_a r_b r_a r_n r_d r_o r_n r_u r_l
r_abandonului r_a r_b r_a r_n r_d r_o r_n r_u r_l r_u r_i3
```

Figure 4.5 Phonetic dictionary for word recognition LID example

What is worth mentioning about the phonetic dictionary is that a special parsing was necessary in order to remove words that belonged to several languages. These are mostly words that were borrowed from English (e.g. “show” that appears both in English and Romanian) and only one of these phonetic transcriptions was kept which in most of the cases was the English phonetic transcription. The small amount of words that posed this problem had little to no impact on the performance but having duplicated phonetic transcriptions for the same word would result in a critical error for the system, which would cause it to fail in both the training and decoding processes making this an important aspect.

A reference test has been performed on the same audio clips that were used for the system’s training. Note that the “Unknown phrases” field refers to sentences for which less than 5 words were recognized. The results are the ones in Table 4.6.

Tested language	Romanian phrases	English phrases	Albanian phrases	Unknown phrases	Correct phrases
Romanian	152	2483	1595	770	3.04%
English	0	4065	1	553	88%
Albanian	0	0	831	13	98.45%

Table 4.6 Word decoding results reference

The total LM rate was of only 48.24%, a very low value due to the poor results obtained for Romanian. The above test has been run with a LM that was obtained by interpolating the most common 6k words from each of the targeted languages. This means that the entire corpus for each language is processed and the most common 6k words from each language are selected. Afterwards, three LMs are created, one for each language and then interpolated so that they are given equal weights. The same is valid for the next test, where the decoding was performed on different audio databases, with none of the audio files being part of the training process.

It has been observed that the previously mentioned technique used for creating the LM is not perfectly realistic because LMs created on a smaller text corpus have larger probabilities for each word/sequence of words, which in term would make their presence felt more in the interpolated LM design. This explains the poor results for Romanian, where the text corpus for the LM is much larger than the ones used for English and Albanian, meaning lower probabilities are assigned to Romanian words, meaning less Romanian words would be decoded when the test is performed, this being reflected in the obtained results.

That is why another method has been proposed, namely selecting the most common 6k words from each language and create a text corpus with all the previous three text corpuses concatenated which would serve as backbone for the one single LM. This means that all the probabilities assigned to words/sequence of words would be flattened by the increased number of words in the text corpus and all the probabilities will then be calculated with the same amount of total words. This is still not the best practice, but a perfect selection of text corpuses would be too wasteful for this precise task. The test with the new LM was done on the training databases in order to make a comparison between its results and the previous ones.

Tested language	Romanian phrases	English phrases	Albanian phrases	Unknown phrases	Correct phrases
Romanian	4769	0	0	231	95.38%
English	125	2753	18	1723	59.60%
Albanian	6	0	555	283	65.75%

Table 4.7 Word decoding results reference, new LM

The total result yielded a LID rate of 77.19%. We have observed that the imposed restriction on the number of words was slightly harsh and it impacted the decision process. By lowering the threshold for the number of words per phrase that led to the decision of "Unknown phrase" by one unit, i.e. less than 4 words, we obtained the following evaluation:

Tested language	Romanian phrases	English phrases	Albanian phrases	Unknown phrases	Correct phrases
Romanian	4826	0	0	174	96.52%
English	173	3204	32	1210	69.36%
Albanian	6	0	559	279	66.23%

Table 4.8 Word decoding results reference, new LM (less restrictive)

The new total LID rate increased to 82.08%. It is now clear that tweaking the decision threshold for this method gives a visible increase of the recognition rate, but it is less reliable because it makes the LID system more dependent on the ASR system that runs behind it. A reliable ASR that introduces less confusion over the recognized words gives us the opportunity to lower the decision threshold even more. Ideally, this "Unknown phrase" threshold would be set to 0 so that we could state that a random phrase clearly belongs to a certain language. The improvement brought by the new LM is obvious and it clearly shows the importance of a more balanced LM.

The next test was performed with the second LM and on audio files that were not used during the training process.

Tested language	Romanian phrases	English phrases	Albanian phrases	Unknown phrases	Correct phrases
Romanian	4948	0	0	161	96.84%
English	96	1008	34	542	60%
Albanian	72	139	677	80	69.93%

Table 4.9 Word decoding results on different audio database (less restrictive)

This test was performed with the less restrictive threshold for the "Unknown phrases" decision, namely only less than 4 words/decoding would lead to this decision, giving a LID rate of 85.50% because of the good result that the numerous Romanian test phrases yielded. Several differences can be observed regarding the other two languages as well, most notably that the Albanian decoding performed better than in the reference case due to having clearer audio clips to run the test on than the ones used for training the system.

Adding all these various results together we have obtained the next table, for an easier understanding and reading with the following notations: "Reference" stands for the test performed on the training database, "Reference with new LM" stands for the test performed on the training database but with the LM obtained in the previously mentioned manner, "Loose reference with new LM" stands for the same test as before, but with a less restrictive threshold and "Loose different database" stands for the test performed on a different database than the one used for training with the newer LM and less restrictive threshold. The results represent the percent of correctly identified phrases.

Tested language	Reference	Reference with new LM	Loose reference with new LM	Loose different database
Romanian	3.04%	95.38%	96.52%	96.84%
English	88%	59.60%	69.36%	60%
Albanian	98.45%	65.75%	66.23%	69.93%

Table 4.10 Word recognition results summary

Another important fact is that for this strategy the LM plays an important role because when the ASR system recognizes a sequence of phonemes it will automatically try to map them into words. This decision can be somewhat forced having recognized phonemes from more than one language and being obliged to output a word from only one language based on that phoneme sequence. Overall, it is a method of low complexity as it introduces few new elements [Annex 2] with the LM being the most important aspect to take into consideration and it produces decent results.

4.3 Confidence score method

The third strategy resorts to an approach more programming oriented and it heavily relies on the Java application released by the CMU Sphinx for free usage. This application can be found under the name "sphinx4-5prealpha-src" and it is available for free downloading on the Internet, at the address http://sourceforge.net/projects/cmusphinx/files/sphinx4/5%20prealpha/ (accessed on 30.06.14). It consists of all the features that were developed so far by the Sphinx community, as it is an open-source toolkit, integrated into one dense Java program which can be modified at one's own wish. This application offers access to all the system's core functions, allows modifying the configuration files and one can also create new features based on the existing ones. Having such freedom inside the speech recognition application gives us the ability to access and model certain parameters that are otherwise not implemented in the available release.

The idea behind this strategy is to explore a certain characteristic of the recognized words/sequences of words, namely the confidence score. This is a measure of how the system interprets the correctness of the given hypothesis as outcome of a recognition system taking into account other possibilities or, to put it in other words, the probability that the resulted words are correct. The confidence score feature resides in the program's core source code but it has not been approached in any release and is not yet supported by the available demos, thus we had to investigate the source code, isolate and understand how the confidence scorer works and, lastly, implement it in a brand new demo application. Afterwards, several tests have been performed on different audio files and the results have been carefully parsed as to keep only relevant situations from which the required information has been extracted. These steps have been applied three times as a consequence to building three different recognizers, one for each language and finally summing up the results.

Firstly, a brief description of how the confidence score is obtained is necessary to allow a better understanding of it and of its impact on the analyzed data as well. It is important to note that the recognized sentence that is presented to us by the decoding program is the outcome of the last stage through which the uttered sentence passes. After the actual decoding of the audio signal into a sequence of words, under the form of text, a very rigorous selection process takes place in order to allow only the best result to pass. In the first phase of this selection a lattice formed of all the theories considered by the recognizer that have not been pruned out is created [23]. This lattice is, in fact, a directed graph with nodes and edges. The nodes represent the theory that a word has been uttered over a certain time period while the edges represent the score given to a word following another. A lattice example can be found in Figure 4.6. Initially, a lattice can have redundant nodes, i.e., nodes referring to the same word and that originate from the same parent node. These nodes can be collapsed and they result in a cleaner, more readable lattice [23]. The lattice is useful to analyze alternative results meaning alternative recognized words on the same position, forming the so called “confusion sets”. Note that <s> means the start of the utterance, </s> its end and <sil> stands for a short period of silence.

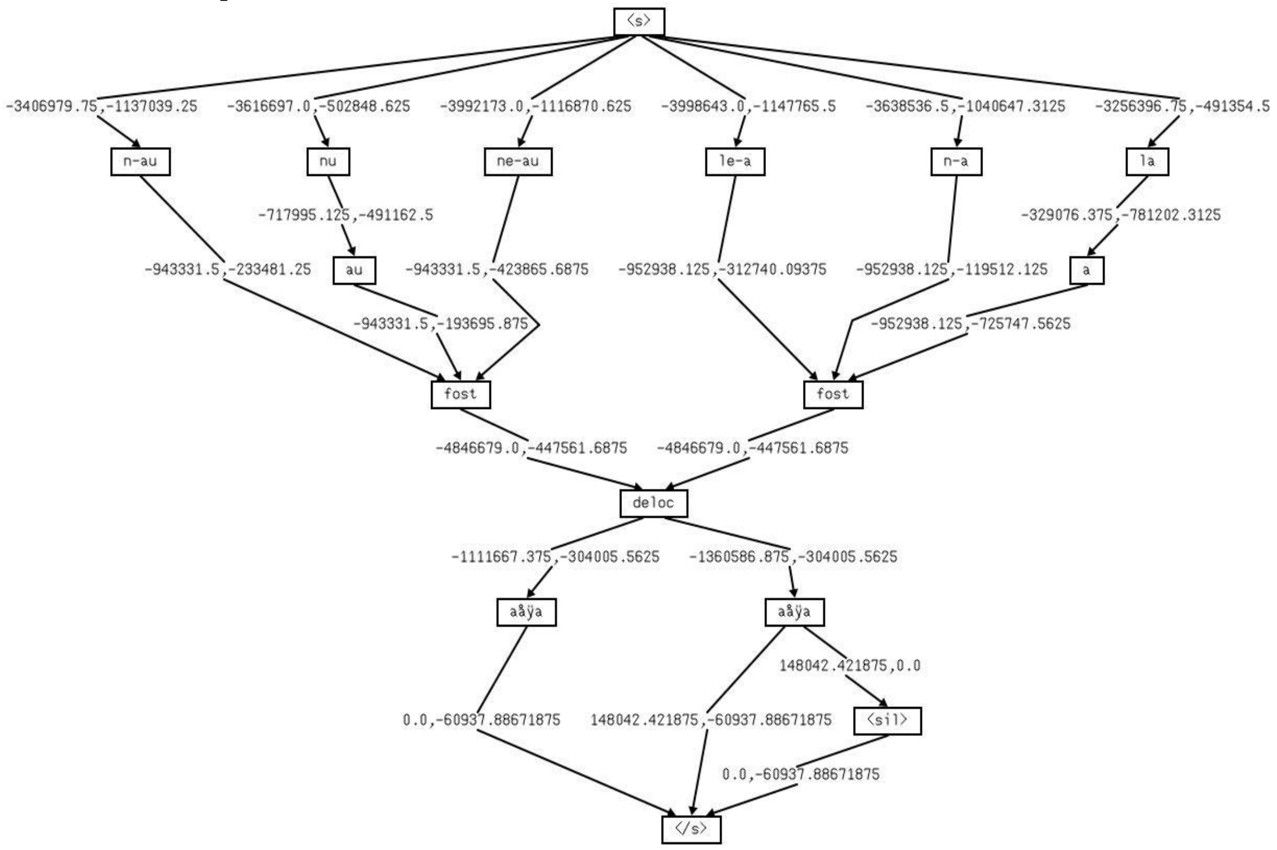


Figure 4.6 Lattice example

Figure 4.6 represents the lattice example for a Romanian sentence recognized using Romanian language and acoustic models. The next step of the selection process and obtaining the confidence score is to create a sequence of confusion sets, one for each position in an utterance. This process is called “sausage making” and as awkward as it may sound it is very representative for its task. A graphical representation of this step can be seen in Figure 4.7.

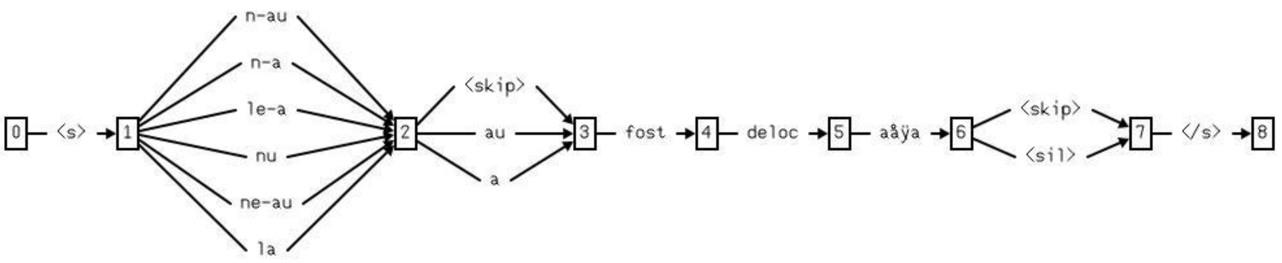


Figure 4.7 Sausage example

As it can be observed, the lattice has suffered some modifications and the sausage is a graph as well, but this time the nodes and edges are arranged in a different manner and some decisions will be taken on each node of the sausage. The scores have been removed from the graphical representation for an easy reading and understanding, but the important aspect about the confidence score is that it gets higher when a word appears more often in one of the nodes described above. In the cases when there is no alternative at the node level (e.g. between boxes 4 and 5 in Figure 4.7) a maximum confidence score is assigned to the selected word. Thus, each word in the recognized utterance has its own confidence score and the fewer the alternatives for that word are the better the confidence result will be. These alternatives appear when the decoder has doubts regarding what it processed but maximum confidence score means that there is no room for doubt regarding that certain word. The confidence score is expressed as a non-positive number with 0 representing the best confidence assigned and no determined lower limit because it represents the logarithmic value of a probability.

In this context we have designed three distinct ASRs, one for each language and started testing audio clips from all languages with every decoder. Firstly, each decoder had the task to recognize audio files corresponding to the language they were built on, even part of the training database, and by comparing the hypothesis' and references some restrictions have been imposed in order to have a common metric used for comparisons between languages. The purpose was to extract enough data based on how each recognizer acts for both native audio files and foreign ones as well and observe how the confidence score is influenced.

It has been observed that the three ASRs behaved somewhat differently in that they assigned different confidence scores to correctly identified words mainly because of the acoustic and language model that did not possess the same characteristics above all languages. When analyzing the results we have seen that the correctly recognized words of less than 4 characters in length gave very bad confidence results. It has also been seen that there were very few correctly recognized words with a confidence score of less than -2000. Thus, we have decided to impose the threshold of a word to have at least 4 characters in size and have a confidence score of at least -2000 in order to be evaluated as a correctly recognized word. Moreover, it has been seen that the correctly recognized phrases gave at the output at least 5 words that met the above restrictions. Thus, a counter for the words that met the imposed requirements has been implemented for each decoder to keep a statistic of the "good words" that have been recognized.

The next step was to find a common metric for each sentence that could be used for comparisons. The following metric has been determined by trial and error while observing its impact on the identified words and it has been applied for all the "good words".

$$mean = \frac{\sum_{i=1}^{n} \frac{\lambda}{\mu}}{n} \tag{4.1}$$

In Equation 4.1 n represents the total number of words in the sentence, μ represents the word size (the number of characters of that word) and λ represents the word's confidence score, thus taking into account the number of words that were recognized, the number of characters in the recognized word and the decoder's uncertainty regarding the hypothetical results. In this way, for a sentence that was used as input for each of the three transcribers the mean would be evaluated and the highest mean out of the three would point towards the presumably correct language.

Unfortunately, the designed demo [Annex 3] would require a great deal of computational resources mainly because it was built on an untested, not optimized subversion of the core source code which did not provide the appropriate support for the targeted task. That resulted in very high processing times, up to 20 times the decoded file's time. However, a total of 275 files, equally balanced between the three languages, were still analyzed and the results can be found below. The "Can't decide" field refers to sentences that were decoded and did not have more than 5 "good words".

Transcriber's main language	Romanian phrases	English phrases	Albanian phrases	Can't decide	Percent correct
Romanian	50	18	6	17	54.94%
English	5	78	4	13	78%
Albanian	4	7	69	4	82.14%

Table 4.11 Confidence score based experiment's results

This experiment was performed on a database with audio files that were not used during the training process and yielded a total LID rate of 71.63% which is satisfying taking into consideration the imposed restrictions the small amount of tested data and, most of all, the dependency on the previously built ASRs. Another notable aspect about this strategy is that it somewhat forces the system into decoding a certain language. By that we mean that if, for instance, the English transcriber is passed an Albanian sentence for decoding then it would automatically try decoding it by the English rules that were implemented in the acoustic and language model. In this way the system gives an opinion about what written sequence of words from English best fits the uttered sentence from Albanian, instead of making a clear-cut distinction between languages. It still is a solution for LID but it is not as reliable as the first two that were proposed.

Another strategy has been proposed, closely related to the third method, which involved building two cumulative distribution functions for each language, one composed of the low confidence scores and one of the high confidence scores. These were supposed to represent the wrongly and correctly identified words, respectively, but after observing how some correctly identified words are given very low confidence scores and vice versa it has been concluded that there is not enough available data to build such distributions and extract relevant data out of it. Therefore, this strategy has been aborted but it may still be viable on a larger data set.

The following chart brings together the results for each of the three methods in the case of decoding a database that was not used for training and with the characteristics that maximized the performance in each case.

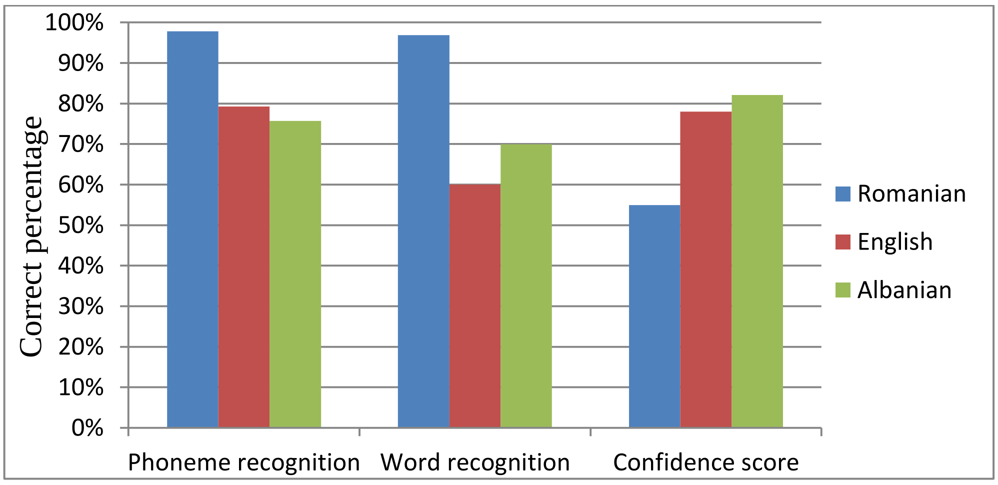


Figure 4.8 Results of the three methods

Taking into account the total number of files that were decoded and expressing the results relative to the entire database that the test was performed on (i.e. taking into account all three languages for each test) we get the following chart that summarizes the total results for each strategy.

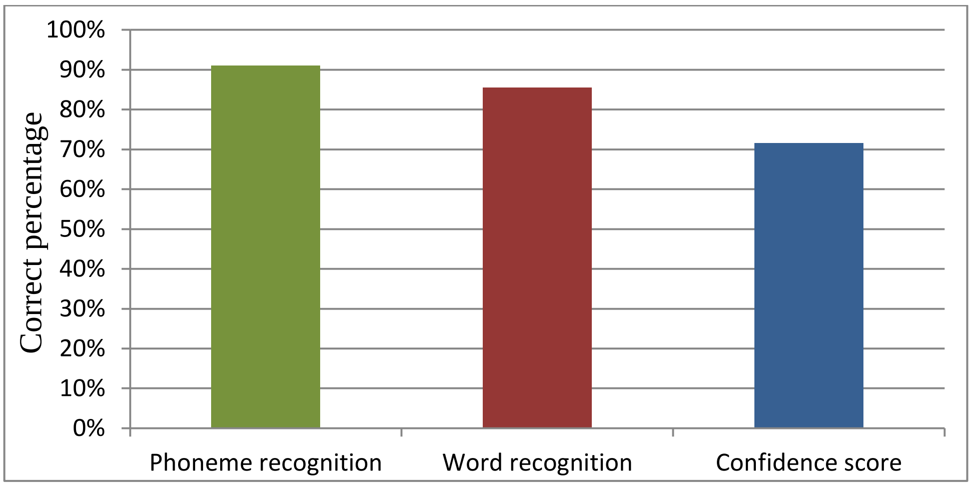


Figure 4.9 Total results of the three methods

Conclusions

The present paper has shown that in order to elaborate a study over language identification methods one needs both an ASR and a LID system as the latter depends in a great extent on the first one. Both systems have been presented along with their characteristics and, most importantly, how modifying these characteristics altered the final results. Emphasizing how each aspect, such as acoustic model, audio files clarity, language model or recognition pattern, influences the outcome of the experiments we can outline the importance of trial and error when it comes to this kind of approaches.

The major drawback when designing both systems was undoubtedly the lack of resources that were available for the ASR task, which, in term, impacted the performances of the LID strategies. However, this problem has been partially solved by the very interesting approach of exploiting a free to use resource, namely the Internet. Even if the Internet does not always provide perfectly accurate information or the desired pattern it is still a great source which should be given the appropriate attention.

This thesis presented an out-of-the-box idea that was applied with a considerable amount of efforts but the results that it provided made the entire process worth-while. A major aspect of the way in which Albanian resources were gathered is that it is a scalable process, giving us the possibility to extend it to any other language. This opens up numerous opportunities either for developing other similar systems or improving this one. Being amongst the few ASR systems for Albanian ever developed it made it even harder to overcome the problems that emerged at every step. This made the result even more satisfying, knowing what the start point was and what the end point came to be, thus not only creating an acceptable ASR but also proving that this idea yielded good results. The Romanian and English ASRs behaved both as a reference point for the Albanian one and part of a multilingual ASR that was very interesting to develop. Better results would have been obtained with a larger resource database, both for audio and text files, preferably in the desired format. Parsing the entire databases and creating each necessary file were very time consuming and challenging as they required a lot of imagination and programming skills, represented by numerous scripts [Annex 4] that were ran under Linux and Java programs that, in the end, came to work as a whole.

Regarding the LID system the task was to evaluate a number of methods and given the ASR background it was easier to design them and work with them. However, differences in the LM or the acoustic model had a greater impact in this case making the LID system more sensitive to this sort of changes. The processing time was a drawback in the case of the confidence scores method but its results proved that it is a method worth not only approaching but also optimizing. The other two strategies that were designed had the advantage that they did not require perfect speech recognition and even a bad recognition from the context meaning point of view would benefit the system in its task. It is important to note that the LID system relied on the language of what it recognized and not on the meaning of what it recognized. This offered a greater degree of freedom when discriminating between languages than when we were to discriminate between an accurate recognition and a bad one, as it was the case of the ASRs.

All in all, the approaches that were studied in this thesis were new to this field and they would have brought even greater contributions if the available databases were denser and with a better quality. The study over LID systems gave comparable results which were not ambiguous and most importantly, could be extended to match numerous other applications' requirements. In a world dominated by technological progress and highly reliant on verbal communication these methods provided an interesting insight and an additional aid to obtaining better performances.

References

[1] http://en.wikipedia.org/wiki/Telephone accessed on 27.05.14

[2] https://www.princeton.edu/~achaney/tmve/wiki100k/docs/Agglutinative_language.html accessed on 27.05.14

[3] http://privatewww.essex.ac.uk/~nckula/LG105_Class10_MorphologyAcrossLanguages.pdf accessed on 27.05.14

[4] H. Cucu, "Towards a speaker-independent, large-vocabulary continuous speech recognition system for Romanian", București, 2011

[5] http://www-i6.informatik.rwth-aachen.de/web/Research/speech_recog.html accessed on 28.05.14

[6] Xuedong Huang, Alex Acero, Hsiao-Wuen Hon, "Spoken language processing: A Guide to Theory, Algorithm and System Development" , Redmond WA, October 2000

[7] http://research.microsoft.com/en-us/projects/language-modeling/ accesed on 30.05.14

[8] Philipp Koehn, "Statistical Machine Translation", 2009, Cambridge University Press

[9] http://www.stanford.edu/class/cs124/lec/languagemodeling.pdf accesed on 30.05.14

[10] András Zolnay Thesis, "Acoustic Feature Combination for Speech Recognition", 14 August 2006, Fakulät für Mathematik, Informatik und Naturwissenschaften, Aachen

[11] http://svr-www.eng.cam.ac.uk/~ajr/SA95/node55.html accesed on 01.06.14

[12] http://masters.donntu.edu.ua/2008/fvti/verenich/library/th_eng.htm accesed on 01.06.14

[13] Stanley Chen, Douglas Beeferman, Ronald Rosenfeld, "Evaluation Metric for Language Models", School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213

[14] Eliathamby Ambikairajah, Haizhou Li, Liang Wang, Bo Yin and Vidhyasaharan Sethu, "Language Identification: A Tutorial", Circuits and Systems Magazine, IEEE (Volume:11 , Issue: 2), p. 82-108, 27 May, 2011

[15] D. Jurafsky and J. Martin, "Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistic and Speech Recognition", 2nd ed. New Jersey: Prentice Hall, 2008

[16] E. Wong, "Automatic spoken language identification utilizing acoustic and phonetic speech information", Ph.D. dissertation, Speech and Audio Research Laboratory, Queensland Univ. Technol., 2004

[17] T. Rong, M. Bin, Z. Donglai, L. Haizhou and C. Eng Siong, "Integrating acoustic, prosodic and phonotactic features for spoken language identification", in Proc. 2006 IEEE Int. Conf. Acoustics, Speech and Signal Processing 2006 (ICASSP 2006), pp. I-I.

[18] T. Schultz, I. Rogina, and A. Waibel, "LVCSR-based language identification," in Proc. 1996 IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP-96), 1996, vol. 2, pp. 781–784.

[19] J. Laver, „Principles of Phonetics”, Cambridge, U.K.: Cambridge Univ. Press, 1994

[20] M. Yip, „Tone” Cambridge, U.K.: Cambridge Univ. Press, 2002.

[21] L. Bauer, „Introducing Linguistic Morphology” Georgetown Univ. Press, 2003.

[22] D. Reynolds, „Gaussian Mixture Models”, MIT Lincoln Laboratory, 244 Wood St., Lexington, MA 02140, USA

[23] http://cmusphinx.sourceforge.net/doc/sphinx4/edu/cmu/sphinx/result/Lattice.html accesed on 29.07.14

Annex 1

```
String phrase;
Configuration configuration = new Configuration();

configuration.setAcousticModelPath("resource:/edu/cmu/sphinx/models/acoustic/LID_3.cd_cont_1000");
configuration.setDictionaryPath("resource:/edu/cmu/sphinx/models/acoustic/LID_3.cd_cont_1000/LID_3_d
ecoding.dic");
configuration.setLanguageModelPath("resource:/edu/cmu/sphinx/models/language/LID_3.1_OnlyFromPhones.
3GramLM.sorted.dmp");
//Iterate through the entire wavs directory:
File targetInput = new File("C:\\Users\\Mihai\\Desktop\\smallTestFolder");
       for (File rawInput : targetInput.listFiles()) {
            try {
                int alWordCount = 0;
                int roWordCount = 0;
                int enWordCount = 0;

                StreamSpeechRecognizer recognizer = new StreamSpeechRecognizer(configuration);
                InputStream stream = new FileInputStream(rawInput);
                recognizer.startRecognition(stream);
                SpeechResult result;

                while ((result = recognizer.getResult()) != null) {
                    phrase = ("" + result.getHypothesis());
                    System.out.println("\n\n");
                    System.out.println("Phrase:" + phrase + " " + rawInput.getName());
                    String[] wordsInRecognizedPhrase = phrase.split(" ");

                    for (String word : wordsInRecognizedPhrase) {
                        if (word.contains("a_")) {
                            alWordCount++;
                        } else if (word.contains("e_")) {
                            enWordCount++;
                        } else if (word.contains("r_")) {
                            roWordCount++;
                        }
                    }
                    int max = Math.max(alWordCount, Math.max(enWordCount, roWordCount));
                    if (max == alWordCount) {
                        albanianLID_3Phrase.append(rawInput.getName() + " transcription: " + phrase)
                              .append("\n" + rawInput.getName() + " is Albanian!");
                    } else if (max == enWordCount) {
                        englishLID_3Phrase.append(rawInput.getName() + " transcription: " + phrase)
                              .append("\n" + rawInput.getName() + " is English!");
                    } else {
                        romanianLID_3Phrase.append(rawInput.getName() + " transcription: " + phrase)
                              .append("\n" + rawInput.getName() + " is Romanian!");
                    }

                }
                recognizer.stopRecognition();
            }
            catch (IOException ex) {
                Logger.getLogger(DemoGUI.class.getName()).log(Level.SEVERE, null, ex);
            }
        }
```

Annex 2

```
String phrase;
Configuration configuration2 = new Configuration();

configuration2.setAcousticModelPath("resource:/edu/cmu/sphinx/models/acoustic/LID_4.cd_cont_1000");
configuration2.setDictionaryPath("resource:/edu/cmu/sphinx/models/acoustic/LID_4.cd_cont_1000/LID_4.
dic");
configuration2.setLanguageModelPath("resource:/edu/cmu/sphinx/models/language/all_languages.6k_each.
3GramLM.sorted.dmp");
        //Iterate through the entire wavs directory:
        for (File rawInput : targetInput.listFiles()) {
            try {
                int alWordCount = 0;
                int roWordCount = 0;
                int enWordCount = 0;

                StreamSpeechRecognizer recognizer2 = new StreamSpeechRecognizer(configuration2);
                InputStream stream2 = new FileInputStream(rawInput);
                recognizer2.startRecognition(stream2);
                SpeechResult result2;

                while ((result2 = recognizer2.getResult()) != null) {
                    phrase = ("" + result2.getHypothesis());
                    System.out.println("\n\n");
                    System.out.println("Phrase:" + phrase + " " + rawInput.getName());
                    String[] wordsInRecognizedPhrase = phrase.split(" ");

                    StringBuilder wordsOnLine = new StringBuilder();
                    for (String word : wordsInRecognizedPhrase) {
                        wordsOnLine.append(word).append(" ");
                        if (word.contains("a_")) {
                            alWordCount++;
                        } else if (word.contains("e_")) {
                            enWordCount++;
                        } else if (word.contains("r_")) {
                            roWordCount++;
                        }
                    }
                    int max = Math.max(alWordCount, Math.max(enWordCount, roWordCount));
                    if (max == alWordCount) {
                        albanianLID_4Phrase.append(rawInput.getName() + " transcription: " + phrase)
                              .append("\n" + rawInput.getName() + " is Albanian!");
                    } else if (max == enWordCount) {
                        englishLID_4Phrase.append(rawInput.getName() + " transcription: " + phrase)
                              .append("\n" + rawInput.getName() + " is English!");
                    } else {
                        romanianLID_4Phrase.append(rawInput.getName() + " transcription: " + phrase)
                              .append("\n" + rawInput.getName() + " is Romanian!");
                    }
                }
            } catch (IOException ex) {
                Logger.getLogger(DemoGUI.class.getName()).log(Level.SEVERE, null, ex);
            }
        }
```

Annex 3

```
public class AlbanianTranscriber {

    public static void main(String[] args) throws Exception {
        System.out.println("Loading models...");
        String phrase = new String();
        Configuration configuration = new Configuration();
        configuration.setAcousticModelPath("resource:/edu/cmu/sphinx/models/acoustic/
albanianChunks7-10.cd_cont_1000");
        configuration.setDictionaryPath("resource:/edu/cmu/sphinx/models/acoustic/
albanianChunks7-10.cd_cont_1000/albanianTrain.dic");
        configuration.setLanguageModelPath("resource:/edu/cmu/sphinx/models/language/
albanianChunks7-10.3GramLM.sorted.dmp");

        //Iterate through the entire wavs directory:
        File targetInput = new File("C:\\Users\\Mihai\\Desktop\\smallTestFolder");
        int totalWordCount = 0;
        float totalConfidencePerFile = 0;
        float totalGoodConfidencePerFile = 0;
        int goodWordsPerFile = 0;

        for (File rawInput : targetInput.listFiles()) {
            String outputID = "(" + rawInput.getName().replace(".wav", "") + ")";
            File textResultsOutput = new
File("C:\\Users\\Mihai\\Desktop\\LID_1_results\\albanianTranscriber\\", outputID.replace("(",
"").replace(")", "").concat(".result.txt"));
            File confidenceResultsOutput = new
File("C:\\Users\\Mihai\\Desktop\\LID_1_results\\albanianTranscriber\\", outputID.replace("(",
"").replace(")", "").concat(".confidence.txt"));

            BufferedWriter bufferedTextWriter = new BufferedWriter(new
FileWriter(textResultsOutput));
            BufferedWriter bufferedConfidenceWriter = new BufferedWriter(new
FileWriter(confidenceResultsOutput));
            int k = 0;
            int contor = 0;
            int goodWordsPerPhrase = 0;
            float totalConfidencePerLine = 0;

            StreamSpeechRecognizer recognizer
                    = new StreamSpeechRecognizer(configuration);
            InputStream stream = new FileInputStream(rawInput);
            recognizer.startRecognition(stream);

            SpeechResult result;

            while ((result = recognizer.getResult()) != null) {
                phrase = ("" + result.getHypothesis());
                System.out.println("\n\n");
                System.out.println("Phrase:" + phrase + " " + rawInput.getName());
                String[] wordsInRecognizedPhrase = phrase.split(" ");

                Lattice resultedLattice = result.getLattice();
                LatticeOptimizer optimizer = new LatticeOptimizer(resultedLattice);
                optimizer.optimize();
                resultedLattice.computeNodePosteriors(1);
                SausageMaker sausageMaker = new SausageMaker(resultedLattice);
                Sausage sausage = sausageMaker.makeSausage();
                sausage.removeFillers();
                ArrayList<String> wordsInRecognizedPhraseFromSet = new ArrayList();
                ArrayList<Float> confidenceArrayFromSet = new ArrayList();

                StringBuilder wordsOnLine = new StringBuilder();
                StringBuilder confidenceOnLine = new StringBuilder();
                for (ConfusionSet confusionSet : sausage) {
                    for (Set<WordResult> wordResultSet : confusionSet.values()) {
                        for (WordResult wordResult : wordResultSet) {
                             wordsInRecognizedPhraseFromSet.add(wordResult.
                             getPronunciation().getWord().getSpelling());
                             confidenceArrayFromSet.add((float) wordResult.getConfidence());
                        }
                    }
                }
                for (int i = 0; i < wordsInRecognizedPhrase.length; i++) {
                    for (int j = k; j < wordsInRecognizedPhraseFromSet.size(); j++) {
                        if (wordsInRecognizedPhrase[i].
                             equals(wordsInRecognizedPhraseFromSet.get(j))) {
                            wordsOnLine.append(wordsInRecognizedPhraseFromSet.get(j)).append(" ");
                            confidenceOnLine.append(confidenceArrayFromSet.get(j)).append(" ");
                            if ((wordsInRecognizedPhraseFromSet.get(j).length() >= 3) &&
(confidenceArrayFromSet.get(j) > -2000)) {
                                totalConfidencePerLine = totalConfidencePerLine
                                        + confidenceArrayFromSet.get(j) /
wordsInRecognizedPhraseFromSet.get(j).length();
                                goodWordsPerPhrase++;
                                goodWordsPerFile++;
                            }
                            contor = j;
                            break;
                        }
                    }
                    k = contor + 1;
                }
                wordsOnLine.append(outputID);
                if (goodWordsPerPhrase <= 5) {
                    confidenceOnLine.append(outputID).
                            append(" not enough (good) words in phrase");
                } else {
                    confidenceOnLine.append(outputID).
                            append(" word count: ").
                            append(wordsInRecognizedPhrase.length).
                            append(" sum of confidence per line: ").
                            append(totalConfidencePerLine).
                            append(" mean: ").
                            append(totalConfidencePerLine / wordsInRecognizedPhrase.length);
                    totalGoodConfidencePerFile = totalGoodConfidencePerFile +
totalConfidencePerLine;
                }
                totalWordCount = totalWordCount + wordsInRecognizedPhrase.length;
                totalConfidencePerFile = totalConfidencePerFile + totalConfidencePerLine;

                bufferedTextWriter.write(wordsOnLine.toString());
                bufferedTextWriter.newLine();
                bufferedConfidenceWriter.write(confidenceOnLine.toString());
                bufferedConfidenceWriter.newLine();
            }
            recognizer.stopRecognition();
            bufferedConfidenceWriter.newLine();
            bufferedConfidenceWriter.write("Total number of recognized words: " + totalWordCount);
            bufferedConfidenceWriter.newLine();
            bufferedConfidenceWriter.write("Total number of good words: " + goodWordsPerFile);
            bufferedConfidenceWriter.newLine();
            bufferedConfidenceWriter.write("Total sum of confidences: " + totalConfidencePerFile);
            bufferedConfidenceWriter.newLine();
            bufferedConfidenceWriter.write("Total sum of good confidences: " +
totalGoodConfidencePerFile);
            bufferedConfidenceWriter.newLine();
            bufferedConfidenceWriter.write("Total mean confidence of good words: " +
totalGoodConfidencePerFile / goodWordsPerFile);
            bufferedConfidenceWriter.newLine();
            bufferedTextWriter.close();
            bufferedConfidenceWriter.close();
        }
    }
}
```

Annex 4

```
// .php to.html conversion script:
for file in /home/dogariu/studentsShare/multilingualASRTask/albanian/text/raw/topchannel2/*.php; do
       cp "$file" /home/dogariu/nasHome/albanian/Task1/topchannel2/"`basename $file .php`.html"
done
----------------------------------------------------------------------------------------------------
// .html to .txt conversion script:
#!/bin/bash
for i in `ls /home/dogariu/nasHome/albanian/Task1/topchannel2/`;do
       lynx -dump /home/dogariu/nasHome/albanian/Task1/topchannel2/$i >
       /home/dogariu/nasHome/albanian/Task1/texte_formatate/topchannelTemp/"`basename $i .html`.txt"
done
----------------------------------------------------------------------------------------------------
// conversion to UTF-8 encoding script:
TARGET_FOLDER=/home/dogariu/nasHome/albanian/Task1/texte_formatate/topchannelTemp
DEST_FOLDER=/home/dogariu/nasHome/albanian/Task1/texte_formatate/topchannel2
for i in `ls /home/dogariu/nasHome/albanian/Task1/texte_formatate/topchannelTemp/`;do
       iconv -f UCS-2le -t UTF-8 $TARGET_FOLDER/$i > $DEST_FOLDER/$i
done;
----------------------------------------------------------------------------------------------------
// concatenating large numbers of files:
ls /home/dogariu/nasHome/albanian/Task1/texte_finale/topchannel2/ >
/home/dogariu/nasHome/albanian/Task1/texteFinaleOneFile/texteFinaleOneFile.txt
sort -n /home/dogariu/nasHome/albanian/Task1/texteFinaleOneFile/texteFinaleOneFile.txt >
/home/dogariu/nasHome/albanian/Task1/texteFinaleOneFile/texteFinaleOneFileSorted.txt
while read LINE
do
       echo $LINE
       cat /home/dogariu/nasHome/albanian/Task1/texte_finale/topchannel2/$LINE >>
/home/dogariu/nasHome/albanian/Task1/texteFinaleOneFile/topchannel2.txt
done < /home/dogariu/nasHome/albanian/Task1/texteFinaleOneFile/texteFinaleOneFileSorted.txt
----------------------------------------------------------------------------------------------------
// selecting 90% of the text corpus for training and leaving the other 10% for testing:
shuf topchannel2.txt > topchannel2Shuffled.txt
head -82550 topchannel2Shuffled.txt > firstTopchannel2.txt
diff firstTopchannel2.txt topchannel2Shuffled.txt | grep '>' | sed 's/> //' > lastTopchannel2.txt
cat firstTopchannel2.txt >> toTest.txt
cat lastTopchannel2.txt >> toTrain.txt
----------------------------------------------------------------------------------------------------
// copying .wav files corresponding to the files in the fileids list of files:
#!/bin/sh
FROM_WAV_FOLDER=/home/dogariu/studentsShare/old140402/resources/speech/database4/wav/*
TO_WAV_FOLDER=/home/dogariu/romanianTrainDatabase
INPUT_FILEIDS=/home/dogariu/licentaMihai/romanian/etc/01-20_00-04_train.fileids
while read line
do
       cp -r $FROM_WAV_FOLDER/$line.wav $TO_WAV_FOLDER/$line.wav
done < $INPUT_FILEIDS
----------------------------------------------------------------------------------------------------
// creating fileids file based on the list of available audio files:
for file in /home/dogariu/licentaMihai/albanian_task4_annotated/wav/*.wav;
       do
       echo "`basename $file .wav`" >>
/home/dogariu/licentaMihai/albanian_task4_annotated/etc/fileidsFromWavs
done
----------------------------------------------------------------------------------------------------
//creating fileids file based on the available transcriptions:
TARGET_FOLDER=/home/dogariu/licentaMihai/albanian_task4_annotated/etc
TARGET_FILE=/home/dogariu/licentaMihai/albanian_task4_annotated/etc/albanian.fileids
# reverse file such that last column on each row becomes first column on each row
rev $TARGET_FOLDER/albanian.all.transcription > temp
awk -F " " '{print $1}' temp > temp2
rev temp2 > temp
sed 's/(//g; s/)//g' temp > $TARGET_FILE
rm temp
rm temp2
----------------------------------------------------------------------------------------------------
// copying .txt files corresponding only to the available audio files:
for file in /home/dogariu/wav/*.wav;do
       cp /home/dogariu/nasHome/albanian/Task1/texte_finale/vizionPlus/"`basename $file .wav`.txt"
/home/dogariu/wav_transcripts/
done
----------------------------------------------------------------------------------------------------
```

```
// bringing the text corpus in the desired format for LM creation:
TARGET_FOLDER=/home/dogariu/licentaMihai/albanianChunks7-10/etc
sed 's/<s>//g;s/<\/s>//g;s/[ \t]*$//g' $TARGET_FOLDER/albanianChunks7-10.transcription >
$TARGET_FOLDER/transcription_for_lm
rev $TARGET_FOLDER/transcription_for_lm > $TARGET_FOLDER/temp
cut -d " " -f 2- $TARGET_FOLDER/temp > $TARGET_FOLDER/temp2
rev $TARGET_FOLDER/temp2 > $TARGET_FOLDER/temp
sed 's/[ \t]*$//g' $TARGET_FOLDER/temp > $TARGET_FOLDER/transcription_for_lm
rm $TARGET_FOLDER/temp
rm $TARGET_FOLDER/temp2
----------------------------------------------------------------------------------------------------
// creating LM script:
ENGLISH_LM_FOLDER=/home/dogariu/licentaMihai
TARGET_FOLDER=/home/dogariu/studentsShare/old140402/resources/text/europarl9amHotnews
#create the counts file (english.counts) and the vocabulary file (english.vocab) for english corpus
#english-one-phrase-per-line is the file that contains one phrase per line
ngram-count -order 3 -write-vocab $ENGLISH_LM_FOLDER/ro.vocab  -text
$TARGET_FOLDER/europarl9amHotnews.rsDiacriticsGre -write $ENGLISH_LM_FOLDER/ro.counts
#create the language model (english.3GramLM)
ngram-count -sort -order 3 -read $ENGLISH_LM_FOLDER/ro.counts -lm $ENGLISH_LM_FOLDER/ro.3GramLM
#sort the language model and create sphinx format language model (english.3GramLM.sorted.dmp)
sphinx_lm_sort $ENGLISH_LM_FOLDER/LID_4_english.3GramLM >
$ENGLISH_LM_FOLDER/LID_4_english.3GramLM.sorted
sphinx_lm_convert -i $ENGLISH_LM_FOLDER/LID_4_english.3GramLM.sorted -o
$ENGLISH_LM_FOLDER/LID_4_english.3GramLM.sorted.dmp
----------------------------------------------------------------------------------------------------
// downloading content of URLs under a specified format:
i=0;
j=0;
ID_FILE=/home/dogariu/nasHome/albanian/Task2/OutputTask2_vizionPlus/ID_vizionPlus.txt
URL_FILE=/home/dogariu/nasHome/albanian/Task2/OutputTask2_vizionPlus/URL_vizionPlus.txt
OUTPUT_WAV_FOLDER=/home/dogariu/vizionPlusWav
while read p;do
        a[i]=$p;
        c[i]=$i;
        #echo ${a[i]}
        i=`expr $i + 1`;
done < $ID_FILE
echo $i
echo ${c[*]}
while read r;do
        b[j]=$r;
        #echo ${b[j]}
        j=`expr $j + 1`;
done < $URL_FILE
#echo $j
for k in ${c[*]}
do
        #echo $k
        #echo ${b[k]}
        youtube-dl -o temp.mp4 -f 17 "${b[k]}"
        ffmpeg -i temp.mp4 -f wav -ar 16000 $OUTPUT_WAV_FOLDER/${a[k]}.wav
        rm -r temp.mp4
done
----------------------------------------------------------------------------------------------------
// LM interpolation script:
ngram -lm <LM#1> -lambda <weight> -mix-lm <LM#2> -write-lm <interpolatedLM>
<LM#1> path to first LM
<weight> a number between 0 and 1, representing LM1’s weight
<LM#2> path to second LM
<interpolatedLM> path to resulting LM

ngram -lm /home/dogariu/licentaMihai/creatingLMTopChannel+bd+bigLM/topChannel2+bd.3GramLM.sorted -
lambda 0.9 -mix-lm
/home/dogariu/nasHome/albanian/Task1/albanianLMfolder/albanian.task1.3GramLM.sorted -write-lm
/home/dogariu/licentaMihai/AdaptiveTraining2/models/language/albanianInterpolated90topChannel2+bdWit
h10BigLM
----------------------------------------------------------------------------------------------------
```