University of Tunis
Institut Supérieur de Gestion de Tunis
LARODEC Laboratory

Belief Hidden Markov Model for Speech Recognition

Elaborated by:

Siwar Jendoubi

Supervised by:

Dr. Boutheina Ben Yaghlane Ben Slimen (IHEC, Université de Carthage)
Dr. Arnaud Martin (IUT de Lannion, Université de Rennes 1)

April 2012

Contents

List of Tables

List of Figures

Introduction

Context

Nowadays, belief functions are widely used in several domains of research where incertitude and imprecision dominates. They provide many tools for managing and processing the existing pieces of evidence in order to extract knowledge and make better decision. They allow experts to have a more clear vision about their problems, which is helpful for finding better solutions. Belief functions theory presents a more flexible way to model uncertainty and imprecise data than probability theory. In addition, it allows expert to model conflict and ignorance. Also, it offers many tools with a higher ability to combine a great number of pieces of evidence. Finally, the theory of belief functions is more general than probability and possibility theory.

Recently, (Ramasso and al, 2007; Ramasso, 2009) present a new way for application of belief functions which is the Belief Hidden Markov Model. The belief HMM extends the probabilistic HMM to belief function theory.

In this research, we seek to use belief functions theory and the belief HMM in the speech processing especially in the speech segmentation process. Speech segmentation involves two classes of methods which are: speech segmentation with linguistic constraint (often called speech segmentation) and speech segmentation without linguistic constraint (called speech recognition). The first one is supervised; in fact, it has as inputs the speech signal and its corresponding text (the phonetic transcription). The goal is to find boundary of each acoustic unit (phoneme) in the speech signal. Speech segments are widely used especially in the speech synthesis by concatenation that concatenates these segments in order to generate the speech realization of the input text. The second one is unsupervised; it has the speech signal as input and it searches the corresponding text as output.

Speech recognition merges many disciplines and technologies (Rabiner and Juang, 1993) which include the following:

- ***Signal processing***: Allows us to study and analyze the speech signal characteristics in order to extract useful information and interesting properties.
- ***Physics (acoustics)***: Allows us to understand the relationship between the physical speech signal and its production and perception mechanisms which are called physiological mechanisms.
- ***Pattern recognition***: In this discipline, a set of algorithms are used to create a prototypical pattern (a model) that better describes the data (speech signal), this pattern is then used to recognize new speech signal. This research can be classified under the pattern recognition discipline.
- ***Communication and information theory***: In this discipline we found procedures of parameter estimations, methods that recognise a particular speech pattern and algorithms of coding and decoding which are used to recognize the best sequence of words that corresponds to the speech signal.
- ***Linguistics***: Includes the syntactic, semantic and pragmatic levels.
- ***Computer science***: Algorithms implementation (software or hardware) and practical methods used in speech recognition system.

Contribution

HMMs are widely used in the speech segmentation and recognition processes. HMM based speech recognizer is an efficient method that allows us to recognize about 80% of a given speech signal. But this recognition rate still not yet satisfying. Recently, (Ramasso and al, 2007; Ramasso, 2009) presented a new version of HMM based on belief functions. Belief HMM gives a better classification rate than the ordinary HMM, when it is applied in a classification problem. Consequently, we propose the new Belief HMM isolated word recognizer, which is a speech recognizer based on belief HMM and used for recognizing isolated words.

Document organization

This document is composed by four chapters:

- Chapter 1: an overview of the Transferable Belief Model will be presented. Then we will talk about belief functions, the principle of minimal commitment, combination rules, frame of discernment operations, the generalized Bayesian theorem, the deconditionalization rule and some transformations used for making decision.
- Chapter 2: we will present about speech production and perception mechanisms, some methods used for extracting features from the speech signal. Then, we will present the

phonetic and the phonology. Finally, a literature review of some speech processing methods will be presented.

- Chapter 3: in which we will present the probabilistic HMM, the belief HMM and finally, we will introduce our belief HMM based speech recognizer.
- Chapter 4: will be dedicated to experiments and results.

1

Transferable Belief Model

1.1 Introduction

The Transferable Belief Model (TBM) (Smets and Kennes, 1994) is a variant of belief functions theories. It is a more general system than the Bayesian model. "It is a model for representing the quantified beliefs held by an agent at a given time on a given frame of discernment" (Smets and Kennes, 1994). As shown in Figure 1.1, the TBM is a two-levels model which are:

- ***Credal level***: This level models our belief and quantifies it by belief functions (static part). Also, it allows many processing to update our model including combination rules (dynamic part).
- ***Pignistic level***: Which is used for making decision, it is preceeded by the credal level. To make decision, we proceed in this level with the conversion of the existing belief functions to probability functions using the pignistic transformation.

According to this process, we will present TBM tools all over this chapter. We will talk about the basic functions which are used in the static part of the credal level and allow as modeling our belief in many forms. Then we will present the principle of minimal commitment, some combination rules, operations that can be made on frames of discernment and the generalized Bayesian theorem which generalizes the Bayes' theorem to belief functions. All these tools are used in the dynamic part of the credal level. Finally, we will present some operation for the pignistic level in the section called making decision.

1.2 Basic functions

1.2.1 Basic Belief Assignment

Let $\Omega = \{d_1, d_2, ..., d_n\}$ called ***frame of discernment***, is a set of all possible decisions that can be made in a particular problem, for example, in a classification problem, Ω contains all possible classes that an object can have, in a similar case, we search to predict the real class d_0 of a given object. Decisions d_i have to be mutually exclusive but not necessary exhaustive. In

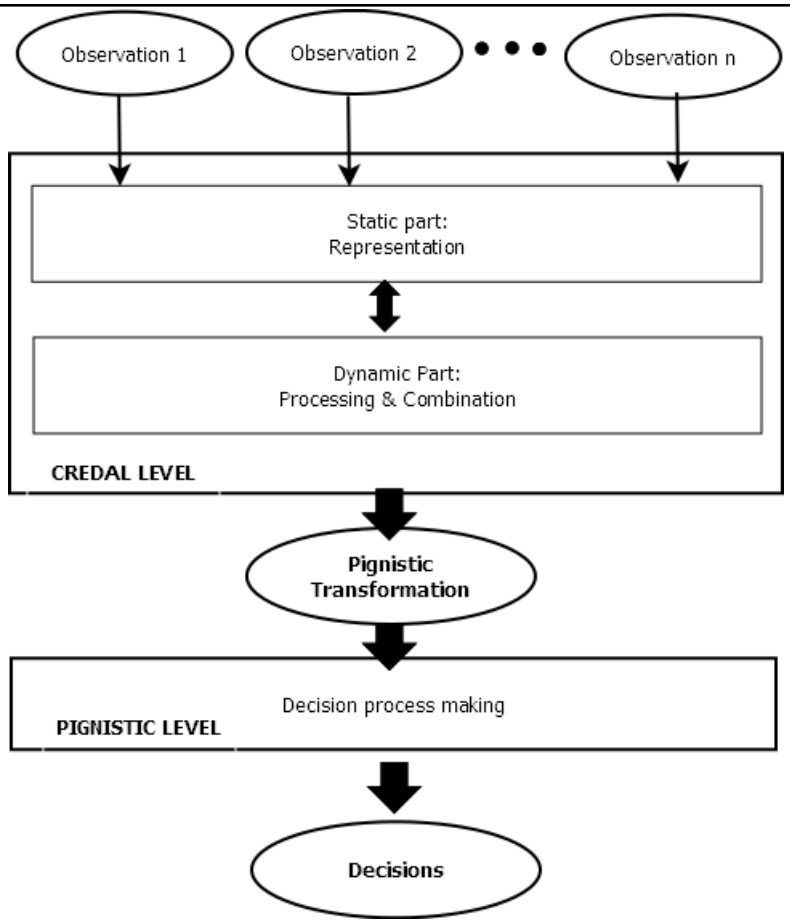

Figure 1.1: Transferable belief model mechanism

the case where Ω is exhaustive, i.e. Ω contains all possible decision, then we say that we work under a ***closed world assumption*** (Shafer, 1976). Otherwise, i.e Ω is not exhaustive, we say that we work under an ***open world assumption*** (Smets, 1990).

The agent belief on Ω is represented by the basic belief assignment[1] (BBA) m^{Ω}, defined as:

$$\begin{aligned} 2^{\Omega} &\rightarrow [0,1] \\ A &\mapsto m^{\Omega}(A) \end{aligned} \tag{1.1}$$

where $2^{\Omega} = \{\emptyset, \{d_1\}, \{d_2\}, \{d_1, d_2\}, ..., \{d_1, d_2, ..., d_n\}\}$is the set of all subsets of Ω, it is called ***power set***. $m^{\Omega}(A)$ is the mass value assigned to the proposition $A \subseteq \Omega$ and it must respect:

$$\sum_{A \subseteq \Omega} m^{\Omega}(A) = 1 \tag{1.2}$$

If we have $m^{\Omega}(A) > 0$ then A is called focal set of m^{Ω}.

Example 1.1 We consider the example of the murder of Mr. Jones presented in (Smets and Kennes, 1994). Our goal is to help the judge to know who the killer was. The judge knows that:

- Mr. Jones has been killed by one member of the team of Big Boss.
- The team of Big Boss includes three people: Peter, Paul and Mary.
- Big Boss selected the killer by a throw of a dice: if he has got an even number so the killer will be female, else, the killer will be male.

There is no information about the choice between Peter and Paul in the case where the killer is male.

First of all, we define our frame of discernment $\Omega = \{Peter\ (P), Paul\ (Pa), Mary\ (M)\}$, then our power set will be $2^{\Omega} = \{\emptyset, \{P\}, \{Pa\}, \{P, Pa\}, \{M\}, \{P, M\}, \{Pa, M\}, \{P, Pa, M\}\}$, finally, our BBA:

$m^{\Omega}(\{P, Pa\}) = 0.5$

$m^{\Omega}(\{M\}) = 0.5$

■

1.2.1.1 Particular BBA

There are some particular BBA from which we note the following (Denoeux, 2007):

- ***Normal BBA***: a BBA is called normal when we have $m^{\Omega}(\emptyset) = 0$. If we have a BBA with $m^{\Omega}(\emptyset) \neq 0$, named *subnormal BBA*, it can be normalized by applying these formulas:

$$\begin{aligned} m_*^{\Omega}(A) &= \frac{m^{\Omega}(A)}{1 - m^{\Omega}(\emptyset)}\ \forall A \subseteq \Omega;\ A \neq \emptyset \\ m_*^{\Omega}(\emptyset) &= 0 \end{aligned} \tag{1.3}$$

[1] Notation: we will use the following notation throughout this document for all belief functions: $m^{domain}(subset)$, where domain represents our frame of discernment Ω and subset is any subset of Ω.

Table 1.1: BBA examples

$A \subseteq \Omega$	**Nor**	**Subn**	**Dogm**	**Vac**	**Sple**	**Categ**	**Bay**	**Con**
$\emptyset$	0	0.3	0	0	0	0	0	0
$\{d_1\}$	0.2	0	0.4	0	0	0	0.2	0
$\{d_2\}$	0	0	0.2	0	0	0	0.4	0.2
$\{d_1, d_2\}$	0.1	0.5	0.1	0	0.4	0	0	0.4
$\{d_3\}$	0	0	0.1	0	0	0	0.4	0
$\{d_1, d_3\}$	0	0.1	0	0	0	1	0	0
$\{d_2, d_3\}$	0.4	0.1	0.2	0	0	0	0	0
$\{d_1, d_2, d_3\}$	0.3	0	0	1	0.6	0	0	0.4

Table 1.1 shows an example of a normal BBA (column Nor) and a subnormal BBA (column Subn).

- ***Dogmatic BBA***: in this BBA, the proposition Ω is not a focal set, i.e. $m^{\Omega}(\Omega) = 0$. As an example, see the column titled Dogm in the Table 1.1.

- ***Vacuous BBA***: we have only one focal set which is Ω, i.e. $m^{\Omega}(\Omega) = 1$. Such a BBA is used in the case of total ignorance, i.e. we have no information about the actual state of our system. Example is given in the Table 1.1 in column five.

- ***Simple BBA***: it has at most two focal sets (see example in column six of the Table 1.1). When it has two, Ω is one of those, i.e. for $\alpha \in [0..1]$ we have:

$$\begin{cases} m^{\Omega}(A) & = 1 - \alpha,\ A \subseteq \Omega \\ m^{\Omega}(\Omega) & = \alpha \end{cases} \tag{1.4}$$

- ***Categorical BBA***: it has only one focal set, i.e.

$$\begin{cases} m^{\Omega}(A) = 1, & A \subseteq \Omega \\ m^{\Omega}(B) = 0, & \forall B \subseteq \Omega \, and \, B \neq A \end{cases} \tag{1.5}$$

  An example of a categorical BBA is given in column Categ of the Table 1.1.

- ***Bayesian BBA***: in the case of a Bayesian BBA, focal sets are singletons, i.e. $m^{\Omega}(\{d_i\}) \geq 0$, $d_i \in \Omega$ for example see the column titled Bay of the Table 1.1.

- ***Consonant BBA***: a BBA is called consonant if its focal sets are nested, i.e. $A \subseteq B \subseteq C...$ Consequently, it has the same characteristics as a possibility distribution, for example see the last column of the Table 1.1 .

1.2.1.2 BBA discounting

Discounting is used in the case where we have a doubt about the pieces of evidence that we have got. Then we can take into account the reliability of our sources of information. In many applications, it is possible to quantify our ***reliability*** by a coefficient $\alpha \in [0,1]$, then the ***discounting rate*** is $1-\alpha$. Our discounted BBA is given by the use of this formula:

$$\begin{cases} m_{\alpha}^{\Omega}(A) = \alpha m^{\Omega}(A), & \forall A \subset \Omega \\ m_{\alpha}^{\Omega}(\Omega) = (1-\alpha) + \alpha m^{\Omega}(\Omega) \end{cases} \tag{1.6}$$

1.2.1.3 Canonical decomposition

Canonical conjunctive decomposition: Before defining the canonical conjunctive decomposition of a non dogmatic BBA, it is necessary to introduce the concept of ***separability***. This concept was firstly introduced by Shafer as: "a separable support function is appropriate whenever the evidence can be decomposed into components that are homogeneous with respect to one's frame of discernment" (Shafer, 1976). Then a BBA function is said separable if and only if it can be decomposed into simple BBAs. If our function is a non dogmatic BBA then the decomposition will be unique and it called the canonical conjunctive decomposition.

We define the canonical conjunctive decomposition of a BBA m (Smets, 1995; Denoeux, 2007) as:

$$m^{\Omega} = \bigcap_{A \subset \Omega} A^{\omega(A)} \tag{1.7}$$

where:

- $A^{\omega(A)}$ is a simple BBA defined by:

$$\begin{cases} m_{A}^{\Omega}(A) = 1 - \omega(A), & if\ A \neq \Omega \\ m_{A}^{\Omega}(\Omega) = \omega(A) \end{cases} \tag{1.8}$$

- $\omega(A)$ is a value in [0..1] which represents the weights of the canonical conjunctive decomposition (WCD) and it can be obtained via this formula:

$$\omega(A) = \prod_{B \subseteq A} q(B)^{(-1)^{|B|-|A|+1}} \tag{1.9}$$

Table 1.2 shows an example of the canonical conjunctive decomposition calculation.

Canonical disjunctive decomposition: It is used to decompose subnormal BBAs (Denoeux, 2007). Let m^{Ω} be a subnormal BBA, then $\overline{m}^{\Omega}$ (its complement) is non dogmatic BBA and it is decomposed as described in subsubsection 1.2.1.3, we will have $\overline{m}^{\Omega} = \bigcap_{A \subset \Omega} A^{\overline{\omega}(A)}$. Assume that $\nu\left(\overline{A}\right) = \overline{\omega}(A)$, the canonical disjunctive decomposition will be defined as:

$$m^{\Omega} = \bigcup_{A \neq \emptyset} A_{\nu(A)} \tag{1.10}$$

Table 1.2: Canonical conjunctive decomposition example

$A \subseteq \Omega$	$\emptyset$	$\{d_1\}$	$\{d_2\}$	$\{d_1, d_2\}$	$\{d_3\}$	$\{d_1, d_3\}$	$\{d_2, d_3\}$	$\{d_1, d_2, d_3\}$
$m(A)$	0	0.2	0.15	0	0.1	0.05	0.2	0.2
$\omega(A)$	1.5125	0.5556	0.7273	1	0.9091	0.8	0.5	1

Table 1.3: Canonical disjunctive decomposition example

$A \subseteq \Omega$	$\emptyset$	$\{d_1\}$	$\{d_2\}$	$\{d_1, d_2\}$	$\{d_3\}$	$\{d_1, d_3\}$	$\{d_2, d_3\}$	$\{d_1, d_2, d_3\}$
$m(A)$	0.1	0	0	0.3	0	0	0.6	0
$\overline{m}(A)$	0	0.6	0	0	0.3	0	0	0.1
$\overline{\omega}(A)$	2.8	0.1429	1	1	0.25	1	1	1
$\nu(A)$	1	1	1	0.25	1	1	0.1429	2.8

For more detail, the reader can see (Denoeux, 2007). Table 1.3 presents an example of the canonical disjunctive decomposition calculation.

1.2.2 BBA conversions

Basic belief assignment can be converted into other functions. They represent the same information under other forms. What's more, they are in one to one correspondence and they are defined from 2^Ω to $[0, 1]$. We will use functions described above:

- ***Belief (bel)***: $bel^\Omega(A)$ is the degree of belief of A. To obtain $bel^\Omega(A)$ we sum all BBAs given to subsets of A, such that $m^\Omega(\emptyset)$ should not be included in $bel^\Omega(A)$. This function quantifies the total belief that the actual state d_0 belongs to A (Shafer, 1976).

$$bel^\Omega(\emptyset) = 0 \, and \, bel^\Omega(A) = \sum_{\emptyset \neq B \subseteq A} m^\Omega(B), \forall A \subseteq \emptyset, A \neq \emptyset \tag{1.11}$$

$$m^\Omega(A) = \sum_{B \subseteq A} (-1)^{|A|-|B|} bel^\Omega(B), \forall A \subseteq \Omega \tag{1.12}$$

- ***Plausibility (pl)***: it is the dual of the belief function. $pl(A)$ measures the maximum belief that could given to the fact that the actual state d_0 belongs to A (Smets, 2000).

$$pl^\Omega(A) = bel^\Omega(\Omega) - bel^\Omega\left(\bar{A}\right), \forall A \subseteq \Omega \tag{1.13}$$

$$pl^\Omega(A) = \sum_{B \cap A = \emptyset} m^\Omega(B), \forall A \subseteq \Omega \tag{1.14}$$

$$m^\Omega(A) = \sum_{B \subseteq A} (-1)^{|A|-|B|-1} pl^\Omega\left(\bar{B}\right), \forall A \subseteq \Omega \tag{1.15}$$

Table 1.4: BBA conversions example

$A \subseteq \Omega$	m	pl	bel	b	q
$\emptyset$	0.15	0	0	0.15	1
$\{d_1\}$	0.1	0.44	0.1176	0.25	0.44
$\{d_2\}$	0.05	0.59	0.0588	0.2	0.59
$\{d_1, d_2\}$	0.2	0.73	0.4118	0.5	0.3
$\{d_3\}$	0.12	0.5	0.1412	0.27	0.5
$\{d_1, d_3\}$	0.04	0.8	0.3059	0.41	0.14
$\{d_2, d_3\}$	0.24	0.75	0.4824	0.56	0.34
$\{d_1, d_2, d_3\}$	0.1	0.85	1	1	0.1

- ***Commonality (q)***: $q^{\Omega}(A)$ is the sum of BBAs allocated to under-sets of A.

$$q^{\Omega}(A) = \sum_{B \supseteq A} m^{\Omega}(B), \forall A \subseteq \Omega \tag{1.16}$$

$$m^{\Omega}(A) = \sum_{A \subseteq B} (-1)^{|B|-|A|} q^{\Omega}(B), \forall A \subseteq \Omega \tag{1.17}$$

- ***Implicability (b)***: $b^{\Omega}(A)$ is the sum BBAs given to sub-sets of A.

$$b^{\Omega}(A) = \sum_{B \subseteq A} m^{\Omega}(B), \forall A \subseteq \Omega \tag{1.18}$$

$$b^{\Omega}(A) = bel^{\Omega}(A) + bel^{\Omega}(\emptyset), \forall A \subseteq \Omega \tag{1.19}$$

$$m^{\Omega}(A) = \sum_{B \subseteq A} (-1)^{|A|-|B|} b^{\Omega}(B), \forall A \subseteq \Omega \tag{1.20}$$

Table 1.4 shows an example of BBA conversions.

1.3 Principle of minimal commitment

The ***Principle of minimal commitment (PMC)*** is used when we have to choose a BBA distribution from a set of all possible BBAs. This principle "is really at the core of the TBM, where degrees of belief are degrees of justified supports" (Smets, 2000). As an example, imagine that we have to choose a belief function over $\Omega = \{a, b, c\}$, suppose we know that $bel(\{a\}) = 0.2$ and $bel(\{b, c\}) = 0.5$, suppose also, we have no other information on our frame of discernment Ω. How can we adopt a BBA distribution given these partial constraints? There are many BBAs that can satisfy them. To resolve a similar problem, we can use the PMC by choosing the least committed BBA. It "formalizes the idea that one should never give more support than justified to any subset of Ω" (Smets, 2000). The PMC is a variant of the principle of minimum specificity introduced by Dubois and Prade in (Dubois and Prade, 1986), it is presented in detail by Hsia in (Hsia, 1991).

To choose the least committed BBA we can use the specificity measure proposed by (Dubois and Prade, 1986).

$$S\left(m^{\Omega}\right) = \sum_{\emptyset \neq A \subseteq \Omega} m^{\Omega}\left(A\right).\log_2\left(|A|\right) \tag{1.21}$$

This measure gives values in $[0, \log_2(|\Omega|)]$. Then the least committed BBA is the BBA that maximizes the specificity measure. Also, we can choose a BBA by the mean of a comparison between plausibility functions or belief functions (Smets, 2000). Then consider the case in which we have two BBAs m_1 and m_2, and must choose the least committed one. First of all, we transform BBAs into plausibility or belief functions, then we say that m_2 is not more committed than m_1 if we have $pl_1(A) \leq pl_2(A)$, $\forall A \subseteq \Omega$ or $bel_1(A) + m_1(\emptyset) \geq bel_1(A) + m_1(\emptyset)$, $\forall A \subseteq \Omega$. m_2 is less committed than m_1 if there is one strict inequality.

1.4 Combination rules

To combine bodies of evidence, first of all, we have to know if their sources are distinct or not. Then we can use the appropriate combination rule. In this section, we present the most known rules of combination of distinct body of evidence. We also, introduce some rules used to combine those obtained from non distinct sources.

1.4.1 Distinct bodies of evidence

In this section, we introduce conjunctive and disjunctive combination rules that are useful to combine distinct pieces of evidence.

1.4.1.1 Conjunctive combination rule

TBM conjunctive rule Consider two distinct BBA m_1^{Ω} and m_2^{Ω} defined on Ω, we can obtain $m_{1\cap 2}^{\Omega}$ through the TBM conjunctive rule (also called conjunctive rule of combination CRC) (Smets, 1993) as follows:

$$m_{1\cap 2}^{\Omega}\left(A\right) = \sum_{B\cap C = A} m_1^{\Omega}\left(B\right) m_2^{\Omega}\left(C\right), \forall A \subseteq \Omega \tag{1.22}$$

Equivalently, we can calculate the CRC via a more simple expression defined with the commonality function as:

$$q_{1\cap 2}^{\Omega}\left(A\right) = q_1^{\Omega}\left(A\right) q_2^{\Omega}\left(A\right), \forall A \subseteq \Omega \tag{1.23}$$

We can combine m_1 with m_2 and obtain the same result of the CRC, via their weight functions ω_1 and ω_2 respectively (Denoeux, 2007). Then

$$m_{1\cap 2}^{\Omega} = \bigcap_{A \subset \Omega} A^{\omega_1(A)\omega_2(A)} \tag{1.24}$$

Table 1.5: Combination rules example

$A \subseteq \Omega$	m_1^Ω	m_2^Ω	$m_{1\cap 2}^\Omega$	$m_{1\oplus 2}^\Omega$	$m_{1\cup 2}^\Omega$
$\emptyset$	0	0	0.355	0	0
$\{d_1\}$	0.1	0.1	0.09	0.1395	0.01
$\{d_2\}$	0.25	0.15	0.2175	0.3372	0.0375
$\{d_1, d_2\}$	0.2	0.1	0.045	0.0698	0.145
$\{d_3\}$	0.2	0.35	0.2375	0.3682	0.07
$\{d_1, d_3\}$	0.1	0.05	0.0175	0.0271	0.12
$\{d_2, d_3\}$	0.1	0.15	0.0325	0.0504	0.25
$\{d_1, d_2, d_3\}$	0.05	0.1	0.005	0.0078	0.3675

Properties

- ***Commutativity***: $m_1^\Omega \cap m_2^\Omega = m_2^\Omega \cap m_1^\Omega$
- ***Associativity***: $m_1^\Omega \cap \left(m_2^\Omega \cap m_3^\Omega\right) = \left(m_1^\Omega \cap m_2^\Omega\right) \cap m_3^\Omega$
- ***Neutral element***: the vacuous BBA (see subsection 1.2.1.1) is the neutral element of the CRC.

Dempster's rule It was firstly defined in (Dempster, 1967), it allows us to combine two distinct BBAs and obtain a normalized one. Consider two distinct BBA m_1^Ω and m_2^Ω defined on Ω, Dempster's rule of combination is then:

$$m_{1\oplus 2}^\Omega(A) = \begin{cases} \frac{\sum_{B\cap C=A} m_1^\Omega(B) m_2^\Omega(C)}{1-\sum_{B\cap C=\emptyset} m_1^\Omega(B) m_2^\Omega(C)}, & \forall A \subseteq \Omega,\ A \neq \emptyset \\ 0 & if\ A = \emptyset \end{cases} \tag{1.25}$$

As we can see, if we apply the normalization rule to the result of the CRC we can obtain the same result of the Dempster's rule.

Example 1.2 Let m_1^Ω and m_2^Ω be two BBA functions defined on Ω. Table 1.5 gives a calculation example of the CRC, the Dempster's rule of combination and the DRC.

■

1.4.1.2 Disjunctive combination rule

Consider two distinct BBAs m_1^Ω and m_2^Ω defined on Ω, we can obtain $m_{1\cup 2}^\Omega$ by the use of the disjunctive rule of combination DRC (Smets, 1993; Denoeux, 2007) as follows:

$$m_{1\cup 2}^\Omega(A) = \sum_{B\cup C=A} m_1^\Omega(B)\, m_2^\Omega(C),\ \forall A \subseteq \Omega \tag{1.26}$$

The DRC has also, a simple expression defined by the implicability function as:

$$b_{1\cup 2}^\Omega(A) = b_1^\Omega(A)\, b_2^\Omega(A),\ \forall A \subseteq \Omega \tag{1.27}$$

1.4.2 Non distinct body of evidence

Rules defined above are used in the case where all sources are distinct and there is no relation between them, i.e. every source defines his degree of belief on Ω independently of others. In the case where there is some dependency between sources, combination rules defined here after cannot be used. Then, there is a necessity to use appropriate rules. Denoeux introduces some rules in (Denoeux, 2007) which are used to combine beliefs obtained from dependent sources.

1.4.2.1 Cautious conjunctive rule

(Denoeux, 2007) presents another rule for conjunctive combination called the ***cautious conjunctive rule (CCRC)*** used to combine non distinct bodies of evidence. Let m_1^Ω and m_2^Ω be two non dogmatic BBAs defined on Ω, then the resulting BBA $m_{1\wedge 2}^\Omega$ obtained after the use of CCRC will be defined by its corresponding weight function as:

$$\omega_{1\wedge 2}(A) = \min\left(\omega_1(A), \omega_2(A)\right), \forall A \subset \Omega \tag{1.28}$$

$$m_{1\wedge 2}^\Omega(A) = \bigcap_{A\subset\Omega} A^{\omega_{1\wedge 2}} \tag{1.29}$$

As in the case of distinct functions, there exists a normalized version of the CCRC called ***normalized cautious rule*** (Denoeux, 2007). Furthermore, we obtain a normal BBA by the use of this formula:

$$\begin{cases} m_{1\wedge *2}^\Omega(A) = k * m_{1\wedge 2}^\Omega(A), & \forall A \subset \Omega,\ A \neq \emptyset \\ m_{1\wedge *2}^\Omega(\emptyset) = 0 \\ k = \left(1 - m_{1\wedge 2}^\Omega(\emptyset)\right)^{-1} \end{cases} \tag{1.30}$$

Properties

- ***Commutativity***: $m_1^\Omega \wedge m_2^\Omega = m_2^\Omega \wedge m_1^\Omega$
- ***Associativity***: $m_1^\Omega \wedge \left(m_2^\Omega \wedge m_3^\Omega\right) = \left(m_1^\Omega \wedge m_2^\Omega\right) \wedge m_3^\Omega$
- ***Idempotence***: $m^\Omega \wedge m^\Omega = m^\Omega$
- ***Distributivity of $\bigcap$ with respect to $\bigwedge$***: $m_1^\Omega \cap \left(m_2^\Omega \wedge m_3^\Omega\right) = \left(m_1^\Omega \cap m_2^\Omega\right) \wedge \left(m_1^\Omega \cap m_3^\Omega\right)$

(Denoeux, 2007), also, generalized these rules by using positive t-norms/t-conorms. Then we can replace the minimum operator by a positive t-norm/t-conorm. A positive triangular norm or conorms (t-norm or t-conorm) is a binary operator (Klement and al, 2000). This offers the possibility to use many t-norm ($\top$) and t-conorm ($\perp$) operators. We give some examples of these operators in the Table 1.6.

Example 1.3 Let m_1^Ω and m_2^Ω be two BBA functions defined on Ω. Table 1.7 gives a calculation example of the CCRC.

■

Table 1.6: Triangular norm/conorm (Klement and al, 2000)

t-norms	
Minimum t-norm	$x \top_{Min} y = \min(x, y)$
Lukasiewicz t-norm	$x \top_{L} y = \max(x + y - 1,\ 0)$
Drastic product	$x \top_{D} y = \begin{cases} 0 & if\ (x,y) \in [0,1]^2 \\ \min(x,y) & otherwise \end{cases}$
Frank t-norm	$x \top_{s} y = \begin{cases} \min(x,\ y) & if\ s = 0 \\ x * y & if\ s = 1 \\ \log_s\left(1 + \frac{(s^x-1)*(s^y-1)}{(s-1)}\right) & otherwise \end{cases}$
t-conorms	
Maximum t-conorm	$x \perp_{Max} y = \max(x, y)$
Lukasiewicz t-conorm	$x \perp_{L} y = \max(x + y,\ 1)$
Drastic sum	$x \perp_{D} y = \begin{cases} 1 & if\ (x,y) \in\ ]0,1]^2 \\ \max(x,y) & otherwise \end{cases}$
Probabilistic sum	$x \perp_{P} y = x + y - x.y$

Table 1.7: CCRC example

$A \subseteq \Omega$	m_1^Ω	m_2^Ω	$m_{1\wedge 2}^\Omega$
$\emptyset$	0	0	0.5478
$\{d_1\}$	0.1	0.1	0.1246
$\{d_2\}$	0.25	0.15	0.0949
$\{d_1, d_2\}$	0.2	0.1	0.0475
$\{d_3\}$	0.2	0.35	0.1258
$\{d_1, d_3\}$	0.1	0.05	0.0237
$\{d_2, d_3\}$	0.1	0.15	0.0237
$\{d_1, d_2, d_3\}$	0.05	0.1	0.012

1.4.2.2 Bold disjunctive rule

Let m_1^Ω and m_2^Ω be two subnormal BBAs (Denoeux, 2007), then the resulting BBA $m_{1\vee 2}^\Omega$ obtained after the use of the ***bold disjunctive rule of combination (BDRC)*** will be defined by its disjunctive weight function (see paragraph 1.2.1.3) as:

$$\nu_{1\vee 2}(A) = \min\left(\nu_1(A), \nu_2(A)\right) \forall A \subseteq \Omega,\ A \neq \emptyset \tag{1.31}$$

$$m_{1\vee 2}^\Omega(A) = \bigcup_{A\neq\emptyset} A_{\nu_{1\vee 2}(A)} \tag{1.32}$$

Properties

- ***Commutativity***: $m_1^\Omega \vee m_2^\Omega = m_2^\Omega \vee m_1^\Omega$
- ***Associativity***: $m_1^\Omega \vee \left(m_2^\Omega \vee m_3^\Omega\right) = \left(m_1^\Omega \vee m_2^\Omega\right) \vee m_3^\Omega$
- ***Idempotence***: $m^\Omega \vee m^\Omega = m^\Omega$
- ***Distributivity of $\bigcup$ with respect to $\bigvee$***: $m_1^\Omega \cup \left(m_2^\Omega \vee m_3^\Omega\right) = \left(m_1^\Omega \cup m_2^\Omega\right) \vee \left(m_1^\Omega \cup m_3^\Omega\right)$

1.4.3 Conditioning rule

Suppose we learn that $A \subseteq \Omega$ is true, then we should update our system through the conditioning rule. It is a particular case of the conjunctive rule of combination, using the CRC between the given BBA m_1^Ω defined on Ω, and a second BBA m_2^Ω defined as:

$$m_2^\Omega(B) = \begin{cases} 1 & \text{if } B = A \\ 0 & \text{otherwise} \end{cases} \tag{1.33}$$

Hence, we obtain the conditional BBA through the Dempster's conditioning rule

$$m^\Omega[B](A) = \sum_{C\subseteq\overline{B}} m_1^\Omega(A \cup C),\ \forall A \subseteq B \tag{1.34}$$

$bel^\Omega[A](B),\ \forall B \subseteq \Omega$ which is the belief of B given A (Smets, 1993), as follows:

$$bel^\Omega[A](B) = bel^\Omega\left(B \cup \overline{A}\right) - bel^\Omega\left(\overline{A}\right),\ \forall B \subseteq \Omega \tag{1.35}$$

1.5 Frame of discernment operations

In real word applications, we have pieces of evidence defined on many frame of discernment. To work flexibly under this case, TBM provides many tools that allow us to redefine these pieces under a common space. In this section we present the well known operations which are the refinement, the vacuous extension, the coarsening and the marginalization.

1.5.1 Refinement, coarsening and vacuous extension

Let $\Omega = \{d_1, d_2, ..., d_n\}$ and $\Theta = \{o_1, o_2, \ldots, o_m\}$ be two distinct frames of discernment, we said that there is a ***refinement*** between Ω and Θ if there exists an application $\Re$ defined as follows:

$$\begin{aligned} \Re : 2^{\Omega} &\rightarrow 2^{\Theta} \\ A &\mapsto \Re(A) \end{aligned} \tag{1.36}$$

Hence, the refinement is a mapping between two frames of discernment (Shafer, 1976; Smets, 1993; Denoeux and Ben Yaghlane 2002), i.e. in our case from Ω to Θ, so every element A of Ω has an image $\Re(A)$ on Θ, this image can be a subset of Θ. The refinement is used to refine hypothesis (decisions) included in Ω by those in Θ.

The ***coarsening*** is the dual operation of the refinement (Denoeux and Ben Yaghlane 2002), then we said that Θ is a coarsening of Ω if there exists a refinement that maps Ω to Θ.

In addition, if we have a BBA m_1^{Ω} defined on Ω we can obtain a BBA m_2^{Θ}defined on Θ by the mean of an operation called ***vacuous extension*** (Shafer, 1976; Smets, 1993), noted $\uparrow$, as:

$$\begin{aligned} 2^{\Omega} &\rightarrow 2^{\Theta} \\ m_1^{\Omega}(A) &\mapsto m_2^{\Theta}(\Re(A)) = m_1^{\Omega\uparrow\Theta}(\Re(A)) \end{aligned} \tag{1.37}$$

Example 1.4 Consider the example 1.1, we have $\Omega = \{P, Pa, M\}$ and the killer was a male or a female, then we can define a second frame of discernment which is $\Theta = \{male, female\}$. Then Θ is a caorsening of Ω, its corresponding refinement $\Re$ is defined from 2^{Θ} to 2^{Ω} as:

- $\Re(\{male\}) = \{P, Pa\}$
- $\Re(\{female\}) = \{M\}$

We define also a BBA function on Θ by the mean of the vacuous extension as:

- $m^{\Theta}(\{male\}) = m^{\Omega}(\{P, Pa\}) = 0.5$
- $m^{\Theta}(\{female\}) = m^{\Omega}(\{M\}) = 0.5$

■

Extension on the product space: First, we define the ***product space*** $\Omega \times \Theta$ (Smets, 1993), which is the set of couples defined by $\{(d_i, o_j), \forall d_i \in \Omega, \forall o_j \in \Theta\}$. Then, if we would like to combine the two BBAs m_1^{Ω} and m_2^{Θ}, we should define them onto the same space which is $\Omega \times \Theta$. This can be made by the use of the vacuous extension operator, hence, to obtain $m_1^{\Omega\uparrow\Omega\times\Theta}$, mass value initially allocated to $A \subseteq \Omega$ will be reallocated to $A \times \Theta$, i.e. to the set $\{(d_i, o_j), \forall d_i \in A, \forall o_j \in \Theta\}$, so:

$$m_1^{\Omega\uparrow\Omega\times\Theta}(B) = \begin{cases} m_1^{\Omega}(A) & if\ B = A \times \Theta,\ A \subseteq \Omega \\ 0 & otherwise \end{cases} \tag{1.38}$$

By a same manner, we obtain $m_2^{\Theta\uparrow\Omega\times\Theta}$. Then we can combine the BBAs obtained by the mean of an appropriate combination rule as the CRC (see paragraph 1.4.1.1) if our BBAs are distinct. The resultant BBA is given by

$$m_{1\cap 2}^{\Omega\times\Theta}(C) = \left(m_1^{\Omega\uparrow\Omega\times\Theta} \cap m_2^{\Theta\uparrow\Omega\times\Theta}\right)(C) \tag{1.39}$$

$$= \begin{cases} m_1^{\Omega}(A).m_2^{\Theta}(B) & if\ C = A\times B,\ A\subseteq\Omega,\ B\subseteq\Theta \\ 0 & \text{otherwise} \end{cases} \tag{1.40}$$

Example 1.5 Suppose that we have two frames of discernment $\Omega = \{E, F\}$ and $\Theta = \{A, B, C\}$, and we have two BBAs m^{Ω} and m^{Θ} defined respectively on Ω and Θ:

- $m^{\Omega}(\{E\}) = 0.2$, $m^{\Omega}(\{F\}) = 0.5$ and $m^{\Omega}(\{E, F\}) = 0.3$
- $m^{\Theta}(\{A\}) = 0.8$, $m^{\Theta}(\{B, C\}) = 0.1$ and $m^{\Theta}(\{A, B, C\}) = 0.1$

We want to combine these two BBAs. We first define the product space $\Omega\times\Theta = \{(E,A),(E,B),(E,C),(F,A),(F,B),(F,C)\}$, then we obtain $m_{1\cap 2}^{\Omega\times\Theta}$ by applying the CRC:

$$\begin{aligned}
m_{1\cap 2}^{\Omega\times\Theta}(\{(E,A)\}) &= 0.16 \\
m_{1\cap 2}^{\Omega\times\Theta}(\{(F,A)\}) &= 0.40 \\
m_{1\cap 2}^{\Omega\times\Theta}(\{(E,A),(F,A)\}) &= 0.24 \\
m_{1\cap 2}^{\Omega\times\Theta}(\{(E,B),(E,C)\}) &= 0.02 \\
m_{1\cap 2}^{\Omega\times\Theta}(\{(E,A),(E,B),(E,C)\}) &= 0.02 \\
m_{1\cap 2}^{\Omega\times\Theta}(\{(F,B)(F,C)\}) &= 0.05 \\
m_{1\cap 2}^{\Omega\times\Theta}(\{(F,A),(F,B),(F,C)\}) &= 0.05 \\
m_{1\cap 2}^{\Omega\times\Theta}(\{(E,B),(E,C),(F,B),(F,C)\}) &= 0.03 \\
m_{1\cap 2}^{\Omega\times\Theta}(\{(E,A),(E,B),(E,C),(F,A),(F,B),(F,C)\}) &= 0.03
\end{aligned} \tag{1.41}$$

■

1.5.2 Marginalization

The marginalization, noted $\downarrow$, is a particular case of the coarsening (Smets, 1993). This operator is used when we have a BBA function defined on the product space $\Omega\times\Theta$ and we want to redefine it on Ω as:

$$m^{\Omega\times\Theta\downarrow\Omega}(A) = \sum_{B\subseteq\Omega\times\Theta,\ B\downarrow\Omega=A} m^{\Omega\times\Theta}(C),\ \forall A\subseteq\Omega \tag{1.42}$$

such that $B\downarrow\Omega$ is the projection of B on Ω.

1.6 Generalized Bayesian theorem

Generalized Bayesian Theorem (GBT), proposed by (Smets, 1993), is a generalization of the Bayes' theorem to belief functions under the TBM. GBT uses belief functions instead of probability functions. Hence, it performs a more flexible modeling of uncertainty, imprecise and conflictual pieces of information.

In this section, we present the generalized likelihood principle which is in the heart of the GBT. After that, we talk about the GBT and its duality with the DRC (see subsection 1.4.1.2).

1.6.1 Generalized likelihood principle

Smets generalizes the likelihood principle, defined under the probability theory, in order to derive the GBT and the DRC. The new principle is called ***generalized likelihood principle***. "It simply postulates that the belief function induced by the disjunction of two pieces of evidence is only a function of the belief function induced by each piece of evidence" (Smets, 1993). It is applied to plausibility function as:

$$\forall A \subseteq \Theta,\ \forall d \subseteq \Omega,\ pl^{\Omega}[A](d) \text{ depends only on } \left\{pl^{\Omega}[a_i](d),\ pl^{\Omega}[a_i]\left(\bar{d}\right) :\ a_i \in A\right\} \tag{1.43}$$

Hence, if we have a set of plausibility distributions conditioned to singletons w can obtain the plausibility distribution conditioned to the union of singletons by the mean of the DRC as:

$$pl^{\Omega}[A](d) = 1 - \prod_{a_i \in A} \left(1 - pl^{\Omega}[a_i](d)\right) \tag{1.44}$$

1.6.2 Generalized Bayesian theorem and disjunctive rule of combination

The GBT has the advantage of taking many forms thanks to belief functions. What's more, Smets presents several expressions (Smets, 1993) used to generate many forms of distributions of belief functions by the mean of the GBT. In this document, we present those based on the conditional plausibility. Let Ω and Θ be two frames of discernment, $\omega \in \Omega$ and $\theta \in \Theta$:

$$pl^{\Theta}[\omega](\theta) = 1 - \prod_{\theta_i \in \theta} \left(1 - pl^{\Omega}[\theta_i](\omega)\right) \tag{1.45}$$

$$q^{\Theta}[\omega](\theta) = \prod_{\theta_i \in \theta} pl^{\Omega}[\theta_i](\omega) \tag{1.46}$$

$$m^{\Theta}[\omega](\theta) = \prod_{\theta_i \in \theta} pl^{\Omega}[\theta_i](\omega) \cdot \prod_{\theta_i \in \bar{\theta}} \left(1 - pl^{\Omega}[\theta_i](\omega)\right) \tag{1.47}$$

Duality between the GBT and the DRC: This duality comes from the fact that $pl^{\Omega}[\theta](\omega) = pl^{\Omega \times \Theta}(\omega \times \theta)$, hence we can write:

$$pl^{\Omega}[\theta](\omega) = pl^{\Theta}[\omega](\theta) \tag{1.48}$$

Then, functions 1.44 and 1.48 allow as to derive function 1.45.

1.7 Deconditionalization

Deconditionalization (Smets, 1993) is the inverse operation of conditioning. It aims to transform the conditional belief distribution in order to return to the original distribution, i.e. before conditioning. But, in the majority of cases we cannot obtain the original distribution. That's why Smets (Smets, 1993) proposed to choose the least committed distribution given via this function:

$$m^{\Omega\Uparrow}\left(A \cup \bar{B}\right) = m^{\Omega}\left[B\right](A), \forall C \subseteq B \tag{1.49}$$

This operation can be used to merge a set of BBA $m^{\Omega}\left[\theta_i\right]$ defined on Ω conditionally to singletons $\theta_i \subseteq \Theta$. In this case the resultant BBA is defined on the product space.

$$m^{\Omega\times\Theta}(S) = \prod_{\theta_i \in \Theta} m^{\Omega}\left[\theta_i\right](\upsilon), \forall S \subseteq \Omega \times \Theta \tag{1.50}$$

$$q^{\Omega\times\Theta}(S) = \prod_{\theta_i \in \Theta} q^{\Omega}\left[\theta_i\right](\upsilon), \forall S \subseteq \Omega \times \Theta \tag{1.51}$$

where $\upsilon = ((\theta_i \times \Omega) \cap S)^{\downarrow\Omega}$, this operation is derived by Smets from the GBT and the deconditionalization operation.

1.8 Making decision

The main purpose of the TBM is to make the optimal decision in a world dominated by uncertainty, imprecision and conflict. Making a decision is to choose one hypothesis among those belonging to our frame of discernment and it can be made automatically or by an expert of the domain where TBM is used. As mentioned in (Smets and Kennes, 1994), we can make a coherent decision if our model (system) can be described by a probability distribution defined on our power set $\left(2^{\Omega}\right)$. As shown in Figure 1.1, pignistic level is used to transform belief functions, resulting from the credal level, to a probability distribution via the pignistic transformation.

Pignistic probability Pignistic probability can be used to make decision, it transforms belief function to a classical probability function defined on same frame of discernment Ω. This transformation is called the ***pignistic transformation*** (Smets, 2005).

Let m^{Ω} be a BBA function defined on Ω, m^{Ω} can be transformed into a probability distribution via the pignistic transformation (Smets, 2005) using this formula:

$$\mathrm{BetP}\left\{m^{\Omega}\right\}(d_i) = \frac{1}{\left(1 - m^{\Omega}(\emptyset)\right)} \sum_{A \subset \Omega, d_i \in A} \frac{m^{\Omega}(A)}{|A|} \tag{1.52}$$

where $|A|$ is the number of elements belonging to A. We choose generally the hypothesis that

maximizes the pignistic probability:

$$d_0 = \arg\max_{d_i \in \Omega} \text{BetP}\left\{m^{\Omega}\right\}(d_i) \tag{1.53}$$

Plausibility criteria We can use other criteria to make a decision such as the plausibility criteria. Therefore, if we consider the plausibility as a criteria for making decision, we choose the decision d_0 that maximizes the plausibility distribution.

$$d_0 = \arg\max_{d_i \in \Omega} pl(d_i) \tag{1.54}$$

1.9 Conclusion

The Transferable Belief Model (TBM) allows us to represent our belief under many formats via belief functions. Also, it offers many tools which are use to extract new information. From these tools we mention the combination rules that enable the fusion of several pieces of evidence which can be defined on the same frame of discernment or on different frames. We mention also the generalized Bayesian theorem that performs many tools used in inference processes.

These tools will be used after in this document to present and explain the belief hidden Markov model.

2

Speech processing

2.1 Introduction

We use speech very often in our everyday life; it is our common form of communication. However, we hardly ever ask about speech production and perception mechanisms. Then to give to the computer the ability to understand and produce speech sounds, we must begin by understanding the natural production and perception mechanisms.

Speech processing is widely used and it covers many categories, as:

- ***Speech synthesis***: its goal is to produce an artificial speech. The input of speech synthesis system is the text that we want it to be read, and its output is the speech signal that corresponds to our text.
- ***Speech segmentation***: is the process of identifying boundaries between acoustic units (phonemes, diphones, triphones, syllables, words,...) in the spoken speech signal. It can be divided into two subcategories:
  - *With linguistic constraint*: The output of this discipline will be used in the process of speech synthesis. Its entry will be a text with his corresponding speech signal and its output will be signal by acoustic unit.
  - *Without linguistic constraint*: named speech recognition, it aims to identify the spoken text; its input will be the speech signal, and its output will be spoken text.
- ***Speaker recognition***: its purpose is to recognize the speaker. A speaker recognizer can either identify the speaker who produced the speech signal, or it can be used in case when we have doubt in a person and we would like to check if that person has produce the speech signal or not.

In this chapter, we will talk about the speech signal characteristics, its production and perception processes, and its coding methods. We will introduce an overview of phonetics and phonology. Finally, a literature review of speech synthesis and segmentation methods will be presented.

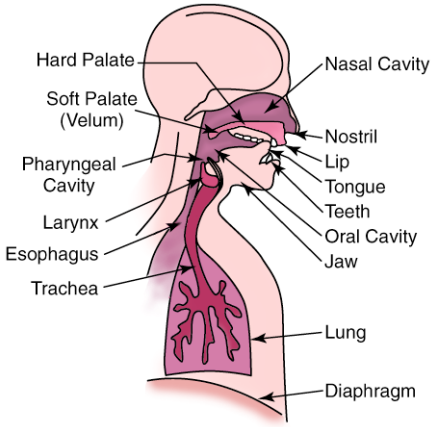

Figure 2.1: Phonetic apparatus (Bouman, 2009)

2.2 Speech signal characteristics

2.2.1 Speech production

Human speech is distinguishable from other sounds by its characteristics and its production mechanism. The phonetic apparatus (Figure 2.1) is the responsible system of the production of the speech. There are many elements that contribute in the process of generation of speech signal, the most important are the following:

- The nervous system,
- The air generated by the respiratory system,
- Vocal cords which are located in the larynx,
- Tongue and lips,
- Oral and nasal cavities.

The nervous system is the first responsible of the generation of the speech signal. First of all, we choose words to be said by the mean of our brain. Then, our choice is converted to orders given by the nervous system to the phonetic apparatus. This last transforms these orders into speech signal.

The respiratory system is the source of the energy required to produce sounds. The air comes from the two lungs through the trachea. Then comes the role of the larynx (Figure[1] 2.2 shows a top view of the larynx). As shown in Figure 2.1, the larynx is located at the top of the trachea and it is responsible of the phonation.

[1] Available on: http://ent4students.blogspot.com/2008/05/larynx-examination.html

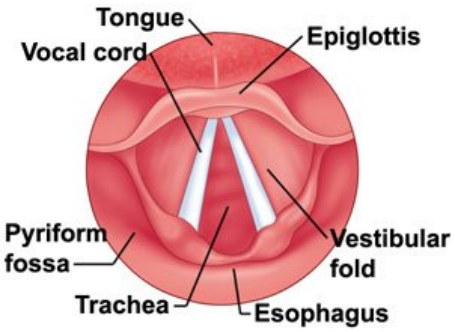

Figure 2.2: Larynx top view

Phonation is made by the rapid opening and closing of the vocal cords, this process is called vibration. Figure 2.3 shows the vocal cord vibration cycle (The Voice Problem Website, 2003), this cycle occurs repeatedly and many times every second. One vibratory cycle is as follows:

- Schema 1: Air is moved out of the lungs and towards the closed vocal cords.
- Schema 2 and 3: Air pressure develops below vocal cords and starts opening them.
- Schema 4 and 5: Brief opening of the vocal cords with the release of air.
- Schema 6, 7, 8 and 9: The release of air causes a low pressure. Then vocal cords re-approximate.
- Schema 10: Vocal folds are closed again, the air cuts off and a pulse of air is released.

This process is the cause of the production of ***voiced sounds***. Voiced sounds, vowels for example, are characterized by the vibration of the vocal folds. The vibration rate is called ***fundamental frequency*** and noted (F_0). In contrast, ***unvoiced sounds***, like f and h, are characterized by the passage of air without vibration of the vocal cords.

After this passage of air (which produces a sound), the sound passes through the vocal tract towards the mouth and the nose. Many speech sounds are then produced depending on the geometric configuration of articulators (tongue, lips, teeth, etc).

2.2.2 Speech perception

In this section, we explain the natural process by which the sounds of language are heard, interpreted and understood, this process is called speech perception. Human auditory system (see Figure[2] 2.4) is the responsible of the perception of sounds, it includes external, middle and inner ear.

[2]Available on: http://www.stanford.edu/class/me220/data/lectures/lect01/auditory.html

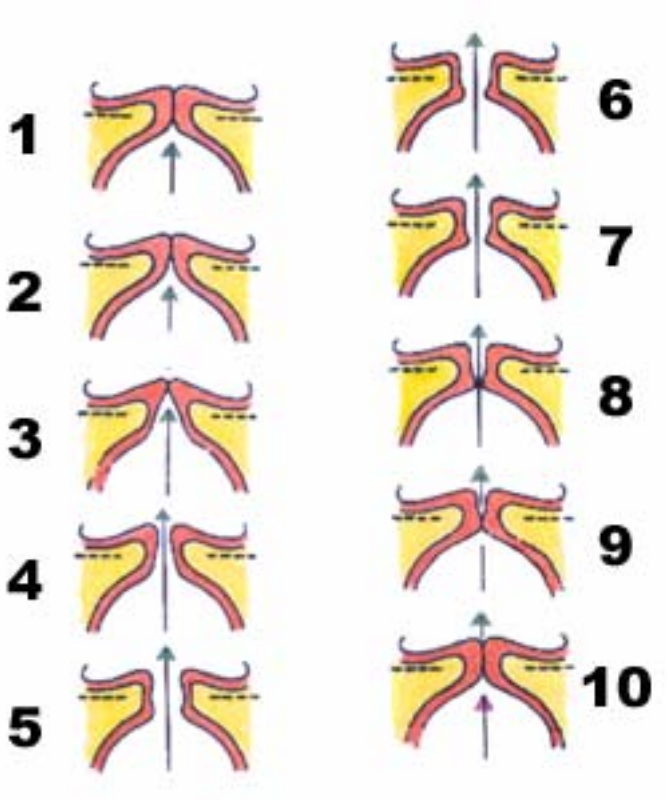

Figure 2.3: The vocal cord vibration cycle (The Voice Problem Website, 2003)

- The outer ear is the responsible of the detection of the sound waves by the pinna of the ear, then waves enter the auditory canal until reaching the ear drum.
- The middle ear contains the ear drum and three bones: malleus, incus and stapes. They are connected by joints and ligaments and they form a pathway that transports vibrations from the eardrum to the inner ear.
- The inner ear encodes vibrations and transmits them to neurons. In fact, the cochlea converts sounds from the outer ear into electrical impulses that can be transmitted to the brain via the auditory nerve.

After this process, the sound waves are converted into electrical impulses and transmitted to the brain, exactly to the auditory cortex. The auditory cortex is the part of the brain that is responsible of decoding and comprehension of speech.

2.2.3 Signal representation

Sound waves should be transformed into a variety of representations in order to be understood. In this document, we talk about graphic representations and some signal characteristics.

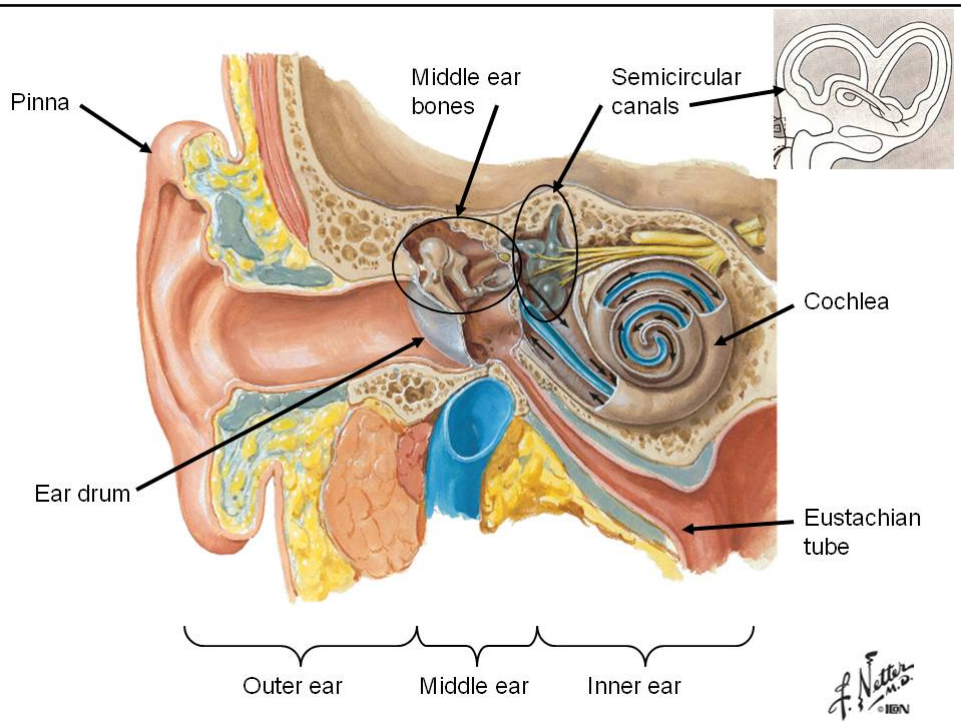

Figure 2.4: Auditory system

2.2.3.1 Signal characteristics

Speech signal is an acoustic phenomenon. It appears as an air pressure caused by the phonetic apparatus. It has many characteristics, from which we introduce the following:

Fundamental frequency (F_0) also called pitch, it measures the vibration rate of vocal cords and it is expressed on Hertz (number of cycles per second). The fundamental frequency is a function of the fundamental period (T) which is the duration of an oscillation, then $F_0 = \frac{1}{T}$. Fundamental frequency varies approximately from 70 to 250 Hz for men, from 150 to 400 Hz for women and from 200 to 600 Hz for children.

Energy is related to the air pressure moved out of the lungs and towards the larynx. It is produced by larynx vibrations and it characterizes the sound intensity. It is measured on decibel (dB).

Signal spectrum is a representation of the signal in the frequency domain, i.e. in terms of the vibration rate at each individual frequency. It is measured on dB/Hz.

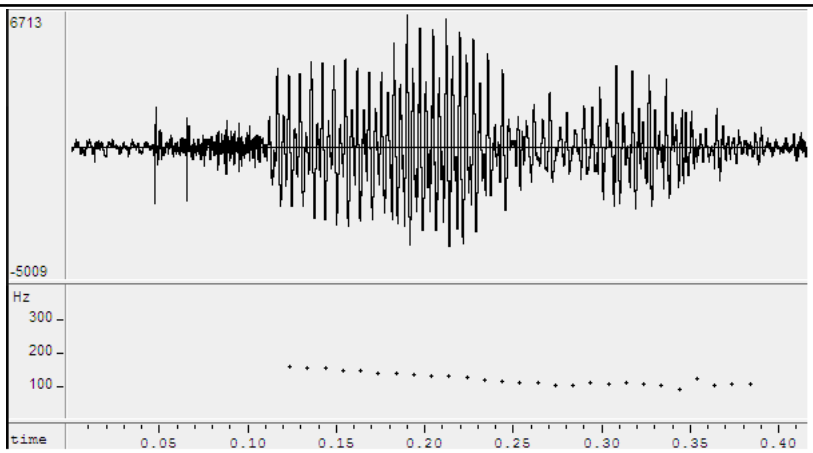

Figure 2.5: Time-frequency representation

2.2.3.2 Graphic representations

Time-frequency representation is a two-dimensional representation with time along the first axis and frequency along the second (Riley, 1987). Figure[3] 2.5 shows an example of time-frequency representation (the schema at the bottom) of the signal corresponding to the word "kiwi". The top schema of this figure shows the time evolution of the amplitude of the signal.

Spectrogram is a three-dimensional representation, generally, it is presented as a graph with two geometric dimensions which are the time in the first axis and the frequency in the second. The third dimension indicates the amplitude of the signal (of a particular frequency at a particular time) and it is represented by the intensity of the gray color in the schema. An example is shown in Figure 2.6.

2.2.4 Feature extraction

2.2.4.1 Linear predictive coding

Linear predictive coding (LPC) is a digital method for encoding the speech signal. It predicts the current speech sample $s(n)$ by a linear function of the past p speech samples (Rabiner and Juang, 1993). It is generally calculated as a weighted sum of the previous speech samples.

$$s(n) = \sum_{i=1}^{p} a_i s(n-1) + Gu(n) \tag{2.1}$$

Such that $a_1, a_2, \ldots a_p$ are the linear prediction coefficients and they are supposed to be constant, $u(n)$ is a normalized excitation and G is the gain of excitation. The transformation function is

[3]The figure is a screen shot of the open-source program WaveSurfer.

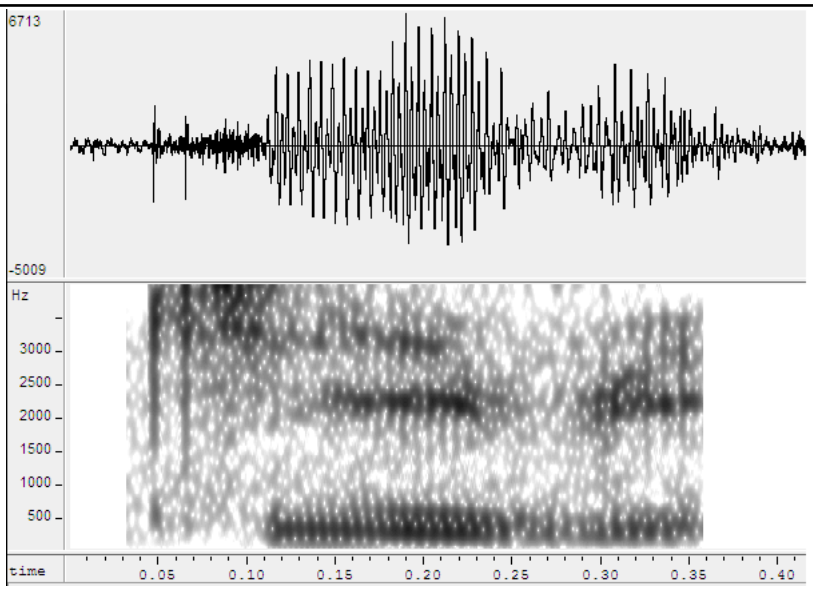

Figure 2.6: Spectrogram of the word "kiwi"

then given by:

$$H(z) = \frac{1}{A(z)} = \frac{1}{1 - \sum_{i=1}^{p} a_i z^{-i}} \tag{2.2}$$

Estimation of this model consists on estimating the linear prediction coefficients of the digital filter $H(z)$ such that we know its output signal which is $s(n)$. To solve this problem, many methods can be used like the autocorrelation method and covariance method. More detail and examples can be found in (Rabiner and Juang, 1993).

2.2.4.2 Mel-frequency cepstral coefficients

Mel-frequency cepstral coefficients (MFCC) are commonly used in the speech recognition process because they take human perception sensitivity with respect to frequencies into account.

The computational steps (Nefti, 2004) are the following:

1. ***Pre-emphasis***: In this step the speech signal $s(n)$ passes through a filter which emphasizes higher frequencies.

$$s_2(n) = s(n) - a.s(n-1) \tag{2.3}$$

   where $s_2(n)$ is the output signal and a coefficient is a value between 0.9 and 0.1. The goal of this step is to maintain all frequencies to be above the perceptual hearing threshold.

2. ***Farme blocking***: The speech signal is segmented into N frames with the length within the range of 20 to 40 ms. A frame is taken every 5 ms.

3. ***Hamming windowing***: Each frame is multiplied with a hamming window of the form:

$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right), 0 \leq n \leq N-1 \tag{2.4}$$

Then if our input frame is $s(n)$ our output frame will be $s(n)\,w(n)$.

4. ***Fast Fourier Transform or FFT***: This step is used to convert each frame from time domain into frequency domain. Then we will obtain the magnitude frequency of every frame.

5. ***Mel Filter Bank Processing***: In this step, we multiply the magnitude frequency obtained by a set of 24 triangular bandpass filters equally spaced along the Mel frequency scale. We get the log energy of each triangular bandpass filter. Then each magnitude frequency is transformed into a vector of 24 log energy E_j.

6. ***Discrete cosine transform or DCT***: We apply discrete cosine transform on the 24 log energy E_j to have 12 Mel-scale cepstral coefficients. DCT is performed using this formula:

$$C_i = \sum_{j=1}^{N} E_j . \cos\left(\frac{\pi . i}{N}\,(j-0.5)\right), 1 \leq i \leq L \tag{2.5}$$

where N is the number of the log energies and L is the number of coefficients. Then we obtain 12 Mel-frequency cepstral coefficients MFCC that can be used alone. To improve performance, we can add other features which are the frame energy and delta cepstrum.

7. ***Log Energy:*** The frame energy is also important and can be added as a 13th features. The log energy is obtained by applying this formula:

$$E = \log\left(\sum_{n=1}^{N} s(n)^2\right) \tag{2.6}$$

where $s(n)$ is the speech frame.

8. ***Delta Cepstrum***: are the time derivatives of the MFCCs. Using them as features can ameliorate the performance of our system.

2.3 Phonetic and phonology

The phonetic and the phonology are two linguistic disciplines (Pierrehumbert, 1990), they cover the field of sentence utterance. Phonetics looks how sounds are produced, transmitted and perceived. Whereas, phonology is concerned with how sounds function are related to each other in a language. In other words, phonetics study sounds (phones) of the language and phonology study relations and correlations between these sounds. Phonetics provide the descriptive tools that are used in the study of the phonological aspects of a language.

The phoneme is the basic unit of a given language; words are analyzed as a sequence of phonemes. They are used in the phonology as a linguistic coding unit. Whereas phonetic study the production and perception mechanisms and physical properties of these phonemes. Hence we distinguish between the articulatory phonetic, the perceptive phonetic and the acoustic phonetic (Jarifi, 2007).

- ***Articulatory phonetic***: studies how humans produce speech sounds. It studies the roles of different organs of the phonetic apparatus and their configuration during the production of speech sounds.
- ***Perceptive phonetic***: studies the human auditory system and the mechanisms of perception of speech sounds.
- ***Acoustic phonetic***: studies the aspects of the speech sound, it is interested by features and properties of the speech signal.

Speech sounds can be classified according to their articulatory characteristics. The two major classes that can be distinguished in any language are: consonants and vowels.

Vowels are spoken sounds pronounced with an open vocal tract (Wikipedia, 2012). Vowels can be classified according to the position of the articulatory features while they are produced. As shown in Figure[4] 2.7, in English we can distinguish between:

- High and low vowel: according to the vertical position of the tongue. Then it is positioned high for high vowel and low for low vowel. For example $[i]$ and $[u]$ are high vowels, $[a]$ is a low vowel.
- Front and back vowel: according to the position of the tongue in the mouth. We have a front vowel when the tongue is positioned forward in the mouth such as $[i]$ and $[e]$. We have a back vowel when the tongue is positioned back in the month such as $[u]$ and $[o]$.
- Rounded and unrounded vowels: according to the form of the lips. When the lips are rounded, so we have rounded vowel such as $[u]$ and $[o]$. Otherwise we have unrounded vowel like $[i]$.

Some vowel examples are given in Figure 2.8.

Consonants are spoken sounds produced with complete or partial closure of the vocal tract. There exist many classifications of consonants. According to (Peccei, 2006), consonant can be classified by: voice, place of articulation or manner of articulation.

- We distinguish between voiced and unvoiced consonant. Voiced consonant are characterized by the vibration of vocal cords such as $[b]$, whereas, unvoiced consonant are not like $[p]$.

[4]Available on: http://www.utexas.edu/courses/linguistics/resources/phonetics/vowelmap/index.html

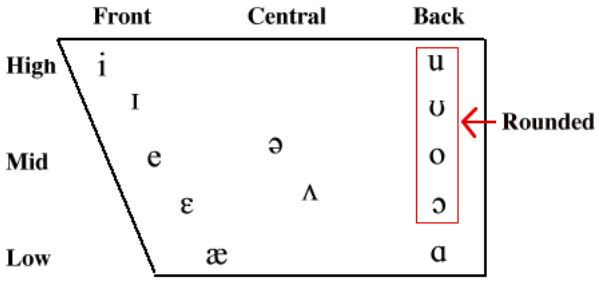

Figure 2.7: English vowel

Vowels

ɪ	pit	ɔ:	born
e	pet	u:	boon
æ	pat	aɪ	bite
ɒ	pot	eɪ	bait
ʌ	but	ɔɪ	boy
ʊ	book	əʊ	toe
ə	mother	aʊ	house
i:	bean	ʊə	poor
ɜ:	burn	ɪə	ear
ɑ:	barn	eə	air

Figure 2.8: Vowel examples (Peccei, 2006)

- The main place of articulation of the consonant can be used to classify them, for example we distinguish:
  - Labial: the air flow is obstructed by lips, for example $[p]$.
  - Labio-dental: the air flow is obstructed by lips and teeth, for example $[f]$.
  - dental: the air flow is obstructed by placing the tongue between the teeth, like $[t]$.
- The main manner of articulation can also be used to distinguish between consonant as:
  - Plosive: also called occlusive or oral stop. It is produced when the vocal tract is blocked so that all airflow ceases, hence it is a stop consonant. It can be done by the tongue as $[t]$, $[k]$ or by the lips as $[b]$, $[p]$.
  - Fricative: When these sounds are produced, a turbulence is caused by forcing the air trough a smaller opening, such as $[s]$, $[z]$.
  - Nasal: produced when the airflow is directed through the nose like $[m]$, $[n]$.

Some consonant examples are given in Figure 2.9.

Consonants

p	pip	ʒ	measure
b	bib	h	hen
t	ten	tʃ	church
d	den	dʒ	judge
k	cat	m	man
g	get	n	now
f	fish	ŋ	sing
θ	thigh	l	let
ð	this	r	ride
s	set	w	wet
z	zoo	j	yet
ʃ	ship		

Figure 2.9: Consonant examples (Peccei, 2006)

2.4 Speech synthesis

The main purpose of the speech synthesis is to read any text. It is defined by (Dutoit, 1996) as "*the automatic production of speech, through a grapheme-to-phoneme transcription of the sentences to utter*". According to this definition, a Test-To-Speech synthesizer includes two important processes:

- Natural Language Processing (NLP): transforms the text into a phoneme sequence with its desired prosody[5].
- Digital Signal Processing (DSP): generates the speech signal using received information (phoneme sequence and prosody).

Speech synthesis can be classified into three major classes of methods: synthesis-by-rule, articulatory synthesis and synthesis-by-concatenation.

2.4.1 Synthesis-by-rule

Rule based speech synthesis systems study the speech signal in order to describe it by the evolution of a set of parameters. These parameters are then used to deduce a set of rules that models phonemes representation, transition between phonemes, etc. Synthesis system uses these rules to create an artificial spectrogram that corresponds to the input text. Electrical generators and resonators are then used in order to generate the speech signal (Dutoit and al, 2002). Formant synthesis is a variant of the synthesis by rule (Klatt, 1980), it models speech waveform by a set of rules in the acoustic domain. This last is the most used in synthesis by rule systems.

[5]Prosody is the music of a spoken language, it includes intonation and rhythm.

2.4.2 Articulatory synthesis

Articulatory synthesis reproduces the functioning of the human vocal apparatus. It models the vocal tract, larynx, lips, nasal cavity and the different articulators. It analyzes their positions during the production of different speech sounds (Maeda, 1990). These information are then used to generate the artificial speech.

2.4.3 Synthesis-by-concatenation

Concatenative synthesizers are based on the use of a speech database in which we found acoustic units and their corresponding speech sounds. Before starting, the synthesis process speech database should be created. First of all, we choose the speech units (acoustic units) that can be the phone (realization of one phoneme), the diphone (unit that begin in the middle of a phone and end in the middle of the following one), the triphone (like diphone except that they include a complete central phone). The second step is to create an exhaustive list of chosen acoustic units. A speech corpus is then digitally recorded ensuring the appearance of all units. A segmentation step is then needed, it can be done manually by expert or by the mean of a segmentation algorithm (see subsection 2.5.1). Finally, the speech database is created (Dutoit, 1996). After the creation of the speech database, the speech synthesis can begin. The first step consists on transforming the sequence of phonemes, performed to the synthesizer, into acoustic units sequence. Then, the speech sound is produced by concatenating speech units pre-prepared.

Unit selection concatenation is a variant of speech synthesis by concatenation (Hunt and Black, 1996; Dutoit and al, 2002). It uses a large database that contains several exemplary of the same acoustic unit instead of only one. The advantage of having many exemplary is that we can take into account the phonetic context of the unit, in fact varying the context may vary the pronunciation. Also, we can have acoustic units of variable size, and then the database can contains phones, diphones, subwords, words, etc. Through the synthesis process, a unit selection step should be done. In fact, we must choose the appropriate units taking into account the context prosodic. Unit selection concatenation gives the most natural speech signal.

2.5 Speech segmentation

Speech segmentation can be classified into two classes of methods, the first one is called speech segmentation with linguistic constraint, and the second is called speech segmentation without linguistic constraint. Speech segmentation can be done manually, however, it requires an expert and needs a lot of time also it is very expensive. Hence automatic segmentation will be easier, faster and cheaper.

2.5.1 Speech segmentation with linguistic constraint

This category of segmentation takes two inputs which are the speech signal and its corresponding text (phone sequence). It is used to detect boundaries between acoustic units (generally, between phones). Results of this category of segmentation are used to create the speech data base that is used for the speech synthesis by concatenation of acoustic units. It is also called ***speech segmentation***.

2.5.1.1 Dynamic time warping segmentation

Segmentation by ***Dynamic time warping (DTW)*** is also called ***speech synthesis-based phonetic alignment*** because it is based on the use of a speech synthesis system (Malfère and al, 2003). The speech synthesis system is used to create a reference speech pattern such that its phonetic segmentation is known. These patterns are then used by the DTW algorithm to detect boundaries between acoustic units by comparing the natural speech signal and the reference speech pattern. Therefore, the segmentation process is based on the minimization of an accumulated distance calculated between the two speech signals. Finally, the DTW algorithm chooses the alignment that minimizes the spectral distortion between the two acoustic frames.

The advantage of this segmentation method is the absence of the learning phase, and its disadvantage is that it must use a synthesis system (specific to the language) and natural speech (that will be segmented) should be obtained from the same speaker who had generated the reference speech signal (Nefti, 2004).

2.5.1.2 Neural network based segmentation

This method is based on the use of a state-transition model associated to every acoustic unit. Each acoustic unit is transformed into a sequence of phone; the length of this sequence is used to determine the number of state. Parameters of these models (transition probabilities) are then estimated (Vorstermans and al, 1996). These models are then used to align the speech signal to its corresponding phonetic transcription. Next, a first segmentation is made and the possible boundaries are generated. After, two neural networks are used. Before their use, they required a training phase using a corpus of 10 minutes. The neural networks are used to estimate the posterior phonetic segment and the phonetic class probabilities of the language. These probabilities are then used to adjust the segmentation and to eliminate the least probable boundary. The reader can find more detail in (Vorstermans and al, 1996).

2.5.1.3 Fusion approach for speech segmentation

It is known that segmentation algorithms do not provide the same segmentation results, in fact, some algorithms are more preferment in the detection of some types of phoneme transitions (segmentation marks) than others. Hence, the fusion method is proposed by (Jarifi and al, 2008). It allows us to use many segmentation algorithms that they are complementary in terms

of transition mark detection. Furthermore, a score is attributed to every algorithm and type of transition in order to select the appropriate mark of transition or to combine many selected marks.

2.5.2 Speech segmentation without linguistic constraint

Speech segmentation without linguistic constraint aims to label the speech signal. In other words, it searches to predict the spoken words automatically. It is also named ***speech recognition***. There exist many methods that can be classified according to their recognition performance to detect some classes of acoustic-phonetic units than other. Some of these methods are described in this section.

2.5.2.1 Detection of breaks of signal stationarity

These methods suppose that the speech is a sequence of stationary segments, hence, statistical models are used in order to detect breaks of stationarities in the speech signal. Finding breaks of stationarity is equivalent to to detect changes in the parameters of the models. Among these methods we present the following.

Brandt's GLR method supposes that the speech signal is represented by a window of observation w_0 of length n and it is characterized by a vector of parameters θ_0. The goal of the algorithm is to find the change instant r in w_0 that corresponds to the change detected in θ_0 (André-Obrecht, 1988; Jarifi and al, 2005). In the first step, the algorithm suppose that there exist two hypothesis H_0 and H_1: H_0 supposes that there is no change instance and θ_0 corresponds to one segment, H_1 considers that there exists a change instant r and the parameter vector is divided into two vectors θ_1 and θ_2 that correspond to w_1 of length r and w_2 of length $(n-r)$, respectively. Brandt's Generalized Likelihood Ratio (GLR) decides between the two hypothesis by calculating the likelihood ratio D and comparing it to a threshold λ for all possible r, then it chooses the instant r that maximizes this test.

Divergence method is based on the use of the divergence test which is a statistical test proposed by (Basseville and Benveniste, 1983). This test is used for making a decision between segmenting the speech signal in a given instant or not. Divergence test is a distance measure obtained by the cumulative sum of the mutual entropies between two statistical models.

So in the divergence method, we define two models, the first one is estimated in a fixed window (signal sample), whereas, the second is adjusted to moving window with the same size. We fix the first model and at each iteration of the algorithm we move our second window, we calculate a model and the divergence test. When our models are too much different from each other, a segmentation mark is done and the current second model becomes a reference for the next pass of the algorithm.

The problem with this method is its omission of some transitions between voiced segments. This omission is caused by the asymmetric criteria of the divergence test, that is why the forward-backward divergence method is introduced.

Forward-backward divergence method is used with voiced segment and it is introduced by (André-Obrecht, 1988) in order to avoid omissions caused by the asymmetric criteria of the divergence test. This method fixes the minimum duration for two successive voiced segments noted L_{min}. The forward procedure is used to segment the speech signal. If the length of the current segment is greater than L_{min}, then we use the backward procedure. If this last detects a break then the signal is re-segmented and the forward procedure will begin its next iteration from the new segmentation mark. Forward and backward procedures use the same principle of the divergence method, except that the moving window in the backward procedure is moved in the reverse direction.

2.5.2.2 Spectral variation detection

Spectral variation detection is a speech recognition technique based on the use of the ***Spectral Variation Function (SVF)***. "SVF is defined as a correlation measure between successive windows of acoustic observation vectors" (Petek and al,1996). The definition of this measure comes from the property that the signal characteristics change rapidly between two successive speech segments. SVF detects this rapid change by its local maxima (Brugnara and al, 1993).

2.5.2.3 Voicing detection based method

Voicing detection is used in many applications of speech processing from which the speech recognition. The goal of the voicing detection based methods is to estimate the fundamental frequency F_0 in speech samples, then these frequencies are used as a feature in the recognition process as the case of (Martin and Mauuary, 2003). Voicing detection can be performed by several techniques like RAPT (Talkin, 1995), YIN (De Cheveigne and Kawahara, 2002) and voicing detection based on residual harmonic (Drugman and Alwan, 2011).

2.5.2.4 HMM segmentation

Hidden Markov Models (HMM) have shown the capacity and performance to treat large speech corpus for several years (Rabiner, 1989; Rabiner and Juang, 1993). It is a statistical approach that can be used for speech segmentation with and without linguistic constraint. It is performed in two steps. The first is the learning step in which model parameters are learned using a pre-segmented speech signal. The second is the decoding or alignment step. In the case of the speech segmentation with linguistic constraint, HMMs are used in order to detect boundaries of each acoustic unit. Whereas, in the case of the speech segmentation without linguistic constraint, HMMs are used to find the sequence of the acoustic units in the signal and their boundaries. More details and literature review of this method are given in the next chapter.

2.6 Conclusion

In this chapter, we presented some linguistic disciplines which are the phonetic and the phonology, they are used by most speech processing disciplines and methods. Speech synthesis and its most known techniques are introduced. Finally, Speech segmentation with and without linguistic constraint is presented.

In the following chapter, we will present our new approach which is the belief HMM recognizer.

3

Belief HMM for speech recognition

3.1 Introduction

Our goal is to develop a speech recognizer system based on belief HMMs instead of HMMs. In this chapter, we will present the hidden Markov model and algorithms used for training and for inference, also, we will introduce the HMM based recognizer. Then belief HMM will be presented and finally we will talk about our belief HMM based recognizer.

3.2 Probabilistic HMM

3.2.1 HMM definition

A Hidden Markov Model is a combination of two stochastic processes; the first one is a Markov chain that is characterized by a finite set of non observable states (hidden) and the transition probabilities between them. The second stochastic process produces the sequence of observations which depends on a state-dependent probability distribution. To formally define an HMM[1], we should specify five characteristics (Rabiner, 1989), as follows:

1. $\Omega_t = \{s_1^t, s_2^t, \ldots, s_N^t\}$ the set of N states of the model[2].
2. $V = \{v_1, v_2, \ldots, v_M\}$ the set of M possible observations that can be generated by our model. We note $O = O_1 O_2 \ldots O_T$ the sequence of observations, such that $O_t \in V$, $1 \leq t \leq T$.
3. $A = \{a_{ij}\}$ the set of N transition probability distributions, where $a_{ij} = \mathrm{P}\left(s_j^{t+1} \mid s_i^t\right)$, $1 \leq i, j \leq N$.

[1] Conventionally, the compact notation is used: $\lambda(A, B, \Pi)$

[2] We note the currant instant t in exponent of states for simplicity.

4. $B = \{b_j(O_t)\}$ the observation symbol probability distributions defined conditionally to every state j, where $b_j(O_t) = \mathrm{P}\left(O_t \mid s_j^t\right)$, $1 \leq j \leq N$, $1 \leq k \leq M$ and $\sum_{k=1}^{N} b_j(k) = 1$.
5. $\Pi = \{\pi_i\}$ the initial state distribution, where $\pi_i = \mathrm{P}\left(s_i^1\right)$, $1 \leq i \leq N$ and $\sum_{i=1}^{N} \pi_i = 1$.

3.2.2 Three basic problems of HMMs

There exist three basic problems of HMMs that must be solved in order to be able to use these models in real world applications. In this section, we will present these problems and their solutions.

3.2.2.1 Evaluation problem

The first problem is named the evaluation problem, it searches to compute the probability that the observation sequence O was generated by the model λ. Solutions of this problem can be used in classification, where we have a set of models, i.e. each model corresponds to a different class, and we have an observation sequence; the goal is to predict its class. The probability $P(O/\lambda)$ can be used as a classification criteria to choose the best model that matches the observation sequence.Then, how do we compute the probability of the observation sequence $P(O/\lambda)$ given the model $\lambda(A, B, \Pi)$?

Forward and Backward propagation can solve this problem (Rabiner, 1989). Their inference mechanisms reduce the calculation complexity by avoiding the summing of the joint probability over all possible state sequences.

Forward propagation It is an inference algorithm which allows as estimating the likelihood of all hidden states at every time instant t. It goes forward starting with the first observation going to the last, and at every time instance it calculates the forward variable. This estimation is called on line estimation.

Let $\alpha_t(i)$ be the forward variable such as: $\alpha_t(i) = \mathrm{P}(O_1 O_2 \ldots O_t, q_t = s_i \mid \lambda)$, this variable represents the probability of the partial observation sequence $(O_1 O_2 \ldots O_t)$ and state s_i at time t, given the model λ. Recursively, we can calculate $\alpha_t(i)$ from $t = 1$ until $t = T$ as shown below:

1. Initialization
$$\alpha_1(i) = \pi_i b_i(O_1), \; 1 \leq i \leq N \tag{3.1}$$
2. Induction
$$\alpha_{t+1}(j) = \left(\sum_{i=1}^{N} \alpha_t(i)\, a_{ij}\right) b_j(O_{t+1}), \; 1 \leq j \leq N, \; 1 \leq t \leq T \tag{3.2}$$
3. Termination
$$\mathrm{P}(O \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i) \tag{3.3}$$

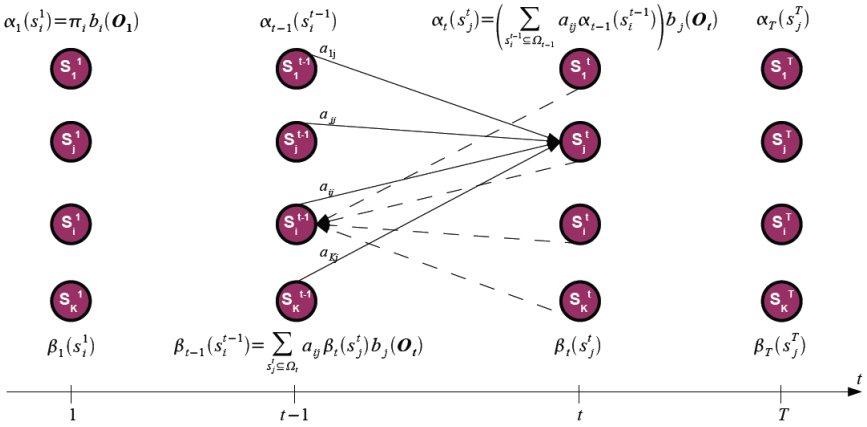

Figure 3.1: Forward and backward propagation (Ramasso, 2007)

Then, the probability that the observation sequence O was generated by the model λ is obtained by summing the terminal forward variables. Figure 3.1 (taken from (Ramasso, 2007)) explains the principle of the calculation of the forward variable.

Backward propagation Like forward, the backward is an inference algorithm that estimates the likelihood of all hidden states at every time instant t. It goes backward from the last observation returning back to the first, and at every time instance it calculates the backward variable. This estimation is called off line estimation. Figure 3.1 shows the backward pass (discontinuous lines) and explain the calculation principle of the backward variable.

In a similar manner, let $\beta_t(i)$ be the backward variable which defined as:
$\beta_t(i) = \mathrm{P}\left(O_{t+1}O_{t+1}\ldots O_T \mid q_t = s_i, \lambda\right)$, this variable represents the probability of the partial observation sequence from $t+1$ to the end, given state s_i at time t and the model λ. $\beta_t(i)$ can be calculated by using a recursive as shown in the following steps.

1. Initialization

$$\beta_T\left(i\right) = 1 \tag{3.4}$$

2. Induction

$$\beta_t\left(i\right) = \sum_{j=1}^{N} a_{ij} b_j\left(O_{t+1}\right)\beta_{t+1}\left(i\right),\ 1 \leq i \leq N,\ t = T-1, T-2, \ldots, 1 \tag{3.5}$$

Then we can calculate $P(O \mid \lambda)$ as:

$$\mathrm{P}\left(O \mid \lambda\right) = \sum_{i=1}^{N} \alpha_t(i)\beta_t(i) \tag{3.6}$$

3.2.2.2 Decoding problem

Given the model λ and the observation sequence $O = O_1 O_2 \dots O_T$, how do we choose the most likely state sequence that produced O?

It should be clear that we cannot find the best state sequence that generates O, and there is no correct one. In order to solve this problem, the Viterbi algorithm is commonly used. This method allows as finding the single best state sequence for the given observation sequence (Rabiner, 1989). To reach this result, we need to calculate a score along every single path, eventually $\delta_t(i) = \max_{q_1,q_2,\dots,q_{t-1}} \mathrm{P}(q_1, q_2, \dots q_{t-1}, q_t = i, O_1 O_2 \dots O_{t-1} \mid \lambda)$. This score can be calculated recursively.

Viterbi starts from the first instant, $t = 1$, for each moment t, it calculates $\delta_t(i)$ for every state i, then it keeps the state which has the maximum δ_t. When, the algorithm reaches the last instance $t = T$, it keeps the state which maximizes δ_T. Finally, Viterbi algorithm back-track the sequence of states as the pointer in each moment t indicates. Steps below explain Viterbi algorithm:

1. Initialization

$$\delta_1(i) = \pi_i b_i(O_1), \ 1 \le i \le N \tag{3.7}$$

$$\psi_1(i) = 0 \tag{3.8}$$

   ψ variable keep track of the argument which maximized

2. Recursion

$$\delta_t(j) = \max_{1 \le i \le N} (\delta_{t-1}(i)\, a_{ij})\, b_j(O_t), \ 1 \le j \le N, \ 2 \le t \le T \tag{3.9}$$

$$\psi_t(j) = \arg \max_{1 \le i \le N} (\delta_{t-1}(i)\, a_{ij}) \tag{3.10}$$

3. Termination

$$P^* = \max_{1 \le i \le N} (\delta_T(i)) \tag{3.11}$$

$$q_T^* = \arg \max_{1 \le i \le N} (\delta_T(i)) \tag{3.12}$$

4. Path backtracking

$$q_t^* = \psi_{t+1}\left(q_{t+1}^*\right), \ t = T-1, T-2, \dots, 1 \tag{3.13}$$

3.2.2.3 Learning problem

How do we adjust the HMM parameters $\lambda = (A, B, \Pi)$ in order to maximize $\mathrm{P}(O \mid \lambda)$?
Existing methods cannot find the optimal λ of the given observation sequence and the given model. However, they can find parameters which locally maximize $\mathrm{P}(O \mid \lambda)$. Among these methods, Baum-Welch (Rabiner, 1989) method is widely used. This algorithm uses the forward

and backward variables to re-estimate the model parameters. Before describing the whole method, we need to define two more variables, which are $\gamma_t(i)$ and $\xi_t(i,j)$ as follows:

$$\gamma_t(i) = \mathrm{P}(q_t = s_i \mid O, \lambda) \tag{3.14}$$
$$\gamma_t(i) = \frac{\alpha_t(i)\,\beta_t(i)}{\mathrm{P}(O \mid \lambda)} \tag{3.15}$$
$$\gamma_t(i) = \frac{\alpha_t(i)\,\beta_t(i)}{\sum_{t=1}^{N} \alpha_t(i)\,\beta_t(i)} \tag{3.16}$$
$$\xi_t(i,j) = \mathrm{P}(q_t = s_i \mid O, \lambda) \tag{3.17}$$
$$\xi_t(i,j) = \frac{\alpha_t(i)\,a_{ij}b_j(O_{t+1})\,\beta_{t+1}(j)}{\mathrm{P}(O \mid \lambda)} \tag{3.18}$$
$$\xi_t(i,j) = \frac{\alpha_t(i)\,a_{ij}b_j(O_{t+1})\,\beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{i=1}^{N} \alpha_t(i)\,a_{ij}b_j(O_{t+1})\,\beta_{t+1}(j)} \tag{3.19}$$

Also, we can relate $\gamma_t(i)$ to $\xi_t(i,j)$ via this formula:

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j) \tag{3.20}$$

Then, we can use these variables to re-estimate the model parameters via a set of re-estimation formulas which are:

$$\overline{\pi}_i = \gamma_1(i) \tag{3.21}$$
$$\overline{a}_{ij} = \frac{\sum_{t=1}^{T} \xi_t(i,j)}{\sum_{t=1}^{T} \gamma_t(i)} \tag{3.22}$$
$$\overline{b}_j(k) = \frac{\sum_{t=1\ s.t.\ O_t=v_k}^{T} \gamma_t(i)}{\sum_{t=1}^{T} \gamma_t(i)} \tag{3.23}$$

Thereby, we define the re-estimated model as $\overline{\lambda} = \left(\overline{A}, \overline{B}, \overline{\Pi}\right)$. Now, we can describe the above Baum-Welch procedure as shown in these steps:

1. Maximization:
   - Calculate forward and backward variables
   - Calculate $\gamma_t(i)$ and $\xi_t(i,j)$
   - Use 3.22, 3.23 and 3.21 formulas to re-estimate $\bar{A}$, $\overline{B}$ and $\overline{\Pi}$.
2. Expectation:
   - $Q\left(\lambda, \overline{\lambda}\right) = \sum_{S \subseteq \Omega_t} p(S \mid O, \lambda) \log\left(P\left(O, S \mid \overline{\lambda}\right)\right)$, $\overline{\lambda}$ is the re-estimated model.

Then, we choose the model λ_* that maximizes $Q\left(\lambda, \overline{\lambda}\right)$ as:

$$\lambda_* = \arg\max_{\overline{\lambda}} Q\left(\lambda, \overline{\lambda}\right) \tag{3.24}$$

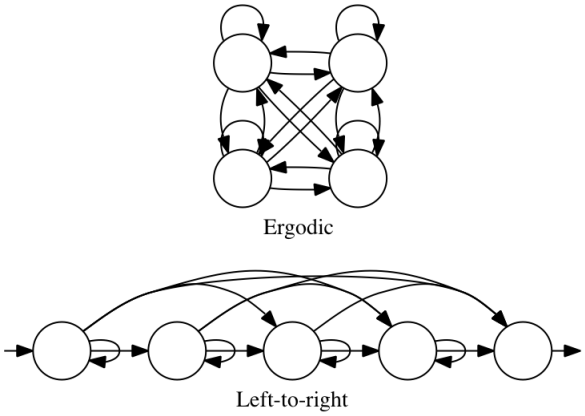

Figure 3.2: Types of models

3.2.3 Types of models

Types of models of the HMM are given by their transition probability matrix, we present some types:

- ***Ergodic model***: has the property that it is possible to reach every state from every other state (in one move or more).
- ***Left-right model***: (also called Bakis model) has the property that, it is not possible to transit to states whose indices are lower than the current state (Rabiner, 1989).

Figure[3] 3.2 gives an example of these models.

3.2.4 Types of HMM

3.2.4.1 Discrete HMM

Above this section, we considered the case of discrete HMM. We should note that, in real word problems observations are, generally, continuous. Then, we must process as follows in order to use discrete HMM (Kouemou, 2011):

1. We must reduce a set of real valued d-dimensional vectors in k d-dimensional vectors, by using a clustering algorithm like k-means.
2. Classify each feature vector with the appropriate codebook vector (nearest).
3. Index of codebook vector will then be used to generate the set of observation symbols.

[3]Taken from http://intoverflow.wordpress.com/2008/05/27/what-is-a-hidden-markov-model/

3.2.4.2 Continuous HMM

It is more profitable to work with the continuous observation densities. In fact, conversion discussed above can influence results. There are some restrictions on the form of the probability density function in order to use continuous HMM. In fact, we cannot use the observation symbol probability distributions because they are continuous, i.e. each observation is a vector of values instead of symbols. Then, we search to use the probability density function (pdf) that better represents our observation vectors, this pdf can be estimated using a mixture of continuous probability density as:

$$b_j(O) = \sum_{g=1}^{G} c_{jg} \Im(O, \mu_{jg}, \Sigma_{jg}), \ 1 \leq j \leq N \tag{3.25}$$

Where O is the observation vector, c_{jg} is the mixture coefficient $\left(\sum_{g=1}^{G} c_{jg} = 1\right)$ and $c_{jg} \geq 0$, $\Im$ is generally a Gaussian density with mean vector μ_{jg} and covariance matrix Σ_{jg} (Rabiner, 1989). Parameters of the pdf can be re estimated using these formulas:

$$\overline{c}_{jg} = \frac{\sum_{t=1}^{T} \gamma_t(j,g)}{\sum_{t=1}^{T} \sum_{g=1}^{G} \gamma_t(j,g)} \tag{3.26}$$

$$\overline{\mu}_{jg} = \frac{\sum_{t=1}^{T} \gamma_t(j,g) . O_t}{\sum_{t=1}^{T} \gamma_t(j,g)} \tag{3.27}$$

$$\overline{\Sigma}_{jg} = \frac{\sum_{t=1}^{T} \gamma_t(j,g) . (O_t - \mu_{jg})(O_t - \mu_{jg})'}{\sum_{t=1}^{T} \gamma_t(j,g)} \tag{3.28}$$

$$\gamma_t(j,g) = \left[\frac{\alpha_t(j)\beta_t(j)}{\sum_{i=1}^{N} \alpha_t(j)\beta_t(j)}\right] \left[\frac{c_{jg}\Im(O, \mu_{jg}, \Sigma_{jg})}{\sum_{g=1}^{G} c_{jg}\Im(O, \mu_{jg}, \Sigma_{jg})}\right] \tag{3.29}$$

3.2.5 Training with multiple observation sequences

The training algorithms discussed above are used for training parameters on one observation sequence. However, in real word applications, it is not practical to use only one observation for training. In fact, HMM models are generally used for classification issues, then a new observation sequence (to be classified) that presents some statistical variations might not be correctly classified because its values are different from those used for training. Then, using many sequences of observations for training will encode many statistical variations of the same class, hence we obtain a more reliable model.

A modification of the re-estimation procedure must be done to take into account the use of multiple observation sequences (Rabiner, 1989). This procedure is given for training of left-right models. Let K be the number of the observation sequences, then our training set will be:

$$O = \left[O^1, O^2, \ldots, O^K\right] \tag{3.30}$$

where $O^k = \left(O_1^k, O_2^k, \ldots, O_{T_k}^k\right)$ is the k^{th} observation sequence of length T_k. These observations are supposed to be independent to each other. Our goal is to have a model that better describes

O, then the joint probability of the observation sequences such the model must be maximized:

$$\mathrm{P}\left(O|\lambda\right) = \prod_{k=1}^{K} \mathrm{P}\left(O^k|\lambda\right) \tag{3.31}$$

$$= \prod_{k=1}^{K} \mathrm{P}_k \tag{3.32}$$

The new re-estimation formulas are given by modifying the re-estimation formulas given above to take into account P_k which is used as a scaling factor:

$$\overline{a}_{ij} = \frac{\sum_{k=1}^{K} \frac{1}{\mathrm{P}_k} \sum_{t=1}^{T_k-1} \alpha_t^k\left(i\right) a_{ij} b_j\left(O_{t+1}^k\right) \beta_{t+1}^k\left(j\right)}{\sum_{k=1}^{K} \frac{1}{\mathrm{P}_k} \sum_{t=1}^{T_k-1} \alpha_t^k\left(i\right) \beta_{t+1}^k\left(j\right)} \tag{3.33}$$

$$\overline{b}_j\left(l\right) = \frac{\sum_{k=1}^{K} \frac{1}{\mathrm{P}_k} \sum_{t=1 \, s.t. \, O_t=v_l}^{T_k-1} \alpha_t^k\left(i\right) \beta_{t+1}^k\left(j\right)}{\sum_{k=1}^{K} \frac{1}{\mathrm{P}_k} \sum_{t=1}^{T_k-1} \alpha_t^k\left(i\right) \beta_{t+1}^k\left(j\right)} \tag{3.34}$$

3.3 HMM recognizer

3.3.1 Acoustic model

The acoustic model attempts to mimic the human auditory system. It is the model used by the HMM-based speech recognizer in order to transform the speech signal into a sequence of acoustic units, this last will be transformed into phoneme sequence and finally the desired text is generated by converting the phoneme sequence into text. Acoustic models are used by speech segmentation and speech recognition systems. We will explain the creation of the acoustic model and how it can be used for speech recognition and segmentation.

3.3.1.1 Structure of the acoustic model

The acoustic model is composed of a set of HMMs (Rabiner, 1989), each HMM corresponds to an acoustic unit. To have a good acoustic model some choices have to be done:

The acoustic unit the choice of the acoustic unit is very important, in fact, the number of them will influence the complexity of the model (more large the number, more complex the model). If we choose a small unit like the phone, we will have an HMM for every possible phone in the language (this number is small), the problem with this choice is that the phone do not model its context (we mean by the context, the previous and the next phones influence the pronunciation of the current one). Such a model is called ***context independent model***. These models are generally used for speech segmentation systems.

Other units that take the context into account can be used as acoustic unit as the diphone which model the transition between two phones, the triphone which model the transition between three phones, subwords, words. These models are called ***context dependent models***.

According to (Rabiner and Juang, 1993), when the context is greater, the recognition performance improve. However, the number of units increases when we use a greater context. Hence we will need a larger number of HMM models and a greater speech corpus. Then a compromise between performance and model complexity should be done. For speech recognition systems, it is important to model the transition between phones which explains the use of the diphone or the triphone in most of these systems.

The model for each acoustic unit, we associate an HMM, then types of HMM model and the probability density function of the observation must be chosen. Generally, left-right models are used for speech recognition and speech synthesis systems (Rabiner, 1989). In fact, speech signal has the property that it changes over time, then the choice of the left-right model is justified by the fact that there is no back transitions and all transitions go forward.

The number of states is fixed in advance or chosen experimentally. (Carvalho and al, 1998; Cox and al, 1998) fixed the number of state to three. This choice is justified by the fact that most phoneme acoustic realization is characterized by three sub-segments, hence we have a state for each sub-segment. (Brugnara and al, 1993; Toledano and al, 2003) used an HMM of six states. Finally, we choose the probability density function of the observation. They are represented by a mixture of Gaussian pdf (formula 3.25), the number of mixtures is generally chosen experimentally.

3.3.1.2 Learning parameters

Learning parameters is based on the use of a training corpus which is a pre-segmented corpus, then we should know its contents (text and acoustic units) and the segmentation marks between acoustic units[4]. As shown in Figure 3.3, the first step is extraction of features. Speech segments are transformed into sequence of acoustic vectors (see subsection 2.2.4), these acoustic vectors are our sequence of observations. We should note that we can have many observation sequences of the same acoustic unit. Then every acoustic unit have its corresponding observation sequences that will be used for learning parameters of its HMM. Every HMM model is learned independently of the other, using the Baum-Welch algorithm (see section 3.2.2.3 and 3.2.5 for more details).

3.3.1.3 Creation of the acoustic model

Learned models are used to create the acoustic model. For the speech segmentation problem, we know the phonetic transcription of the signal to be segmented, this information will be used to create the acoustic model. Then HMMs will be linearly concatenated to each other by following the acoustic units sequence. Transition marks will be added between HMMs, achieve

[4] Generally speech corpus are segmented into phones, in fact, other acoustic units can be obtained by concatenation of phones.

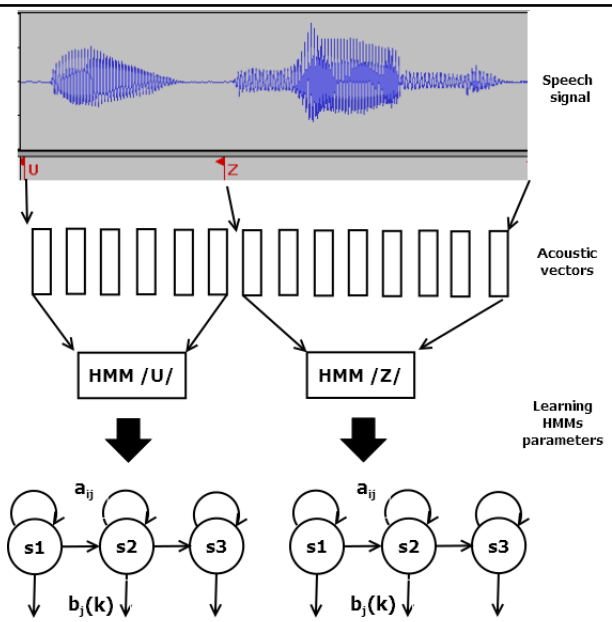

Figure 3.3: Learning HMMs parameters

the transition mark means that the segmentation mark is found.

For speech recognition problem, we do not know the phonetic transcription. Then we will create a model that takes into account all possible acoustic units sequences. In the case where we have a small set of acoustic units, enumerating all possible sequences is not hard and then it is easy to create such acoustic model. However, when we have a large set of acoustic units it will be impossible to doing so. Hence, a language model should be created (Rabiner and Juang, 1993). The language model is a statistical model that assigns a probability $\mathrm{P}(W)$ to each word sequence W. This probability is generally estimated from the text corpus. Hence, for a sentence with N words, this probability is given by:

$$\mathrm{P}(W) = \prod_{i=1}^{N} \mathrm{P}(w_i | w_0, w_1, \ldots, w_{i-1}) \tag{3.35}$$

Generally, we have $N = 2$ or $N = 3$ and our language model is called respectively ***bigram*** or ***trigram***.

Figure 3.4 shows an example of isolated word recognizer, the system is designed to recognize three words which are: one, two and three, and we suppose that there exists a silence in the beginning and the end of each word. The recognizer contains three levels; the first one is the ***syntactic level***, it represents all possible word sequences that can be recognized by our model. The second one is the ***lexical level***, it represents the phonetic transcription (the phoneme sequence) of each word. Finally, the third one is the ***acoustic level***, it models the realization

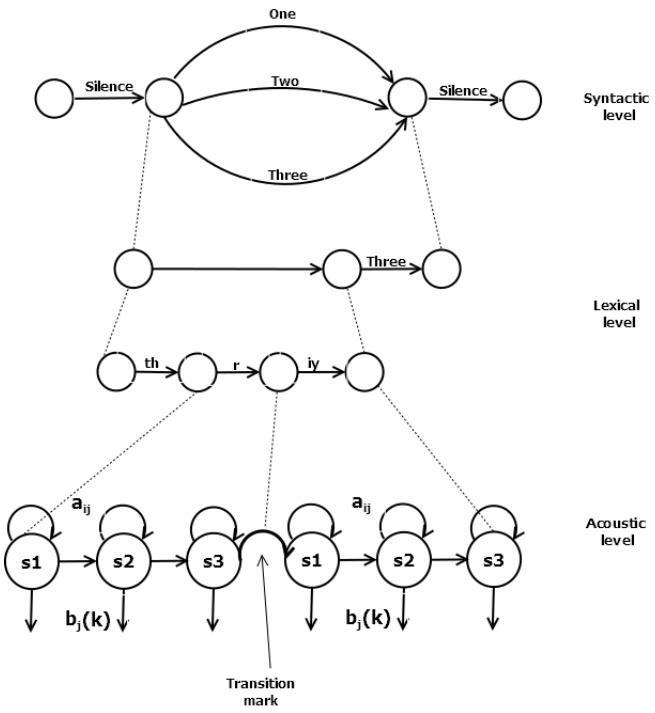

Figure 3.4: Speech recognizer model

of each acoustic unit (in this case the phone). We should note that transition marks are added between successive HMMs, when this mark is achieved, then the next phone begins.

3.3.2 Speech recognition process

The model described above is used for the speech recognition process. To explain this process, we will follow the example of the Figure 3.4 and we suppose that we have four HMMs: one, two, three and silence. Let S be our speech signal to be recognized. Recognizing S consists on finding the most likely path in the syntactic network.

The first step is to transform S into a sequence of acoustic vectors using the same feature extraction method used for training, then we obtain our sequence of observation O. The most likely path is the path that maximizes the probability of observing O such the model $\mathrm{P}(O|\lambda)$. This probability can be done either by using the forward algorithm or the Viterbi algorithm.

The second step consists on turning our HMMs from the beginning of the network until reaching the end. At the first instant (the first acoustic vector) we run the silence HMM until achieving the transition mark. The transition instant is saved because it will be the first instant for the next HMM. According to our network, we have three possible paths and we have to

choose one. Then we run tour HMMs: one , two and three in order to calculate the probability of observing O (sub-sequence of O) such each model. We choose the path that maximizes this probability and the transition mark of its corresponding model is also chosen. The last HMM is run from this last transition mark until reaching the end of the sequence. Finally, the most likely path is found and sequence S is recognized.

3.4 Belief HMM

Belief HMM is an extension of the probabilistic HMM to belief functions (Ramasso, 2007; Ramasso and al, 2007; Ramasso, 2009; Serir and al, 2011). Analogically to HMM, we will introduce the belief HMM. Furthermore, we will present the three problems and their belief solutions.

3.4.1 Definition

Like probabilistic HMM, the belief HMM is a combination of two stochastic processes where the first one is hidden and the second is observable. Therefore, we will start with the definition of the five characteristics of HMM as follows:

1. $\Omega_t = \{s_1^t, s_2^t, s_3^t, \ldots, s_N^t\}$ the set of all possible states. Note that, at time t we can have a set S_i^t of possible sates ($|S_i^t| \geq 1$, and $S_i^t \subseteq \Omega_t$).
2. $V = \{v_1, v_2, \ldots, v_M\}$ the set of M possible observation that can be generated by our model, we note $O = O_1 O_2 \ldots O_T$ the sequence of observations, such that $O_t \in V$, $1 \leq t \leq T$.
3. $m_a^{\Omega_t}\left[S_i^{t-1}\right]\left(S_j^t\right)$ a set of BBA functions defined conditionally to all possible subsets of states S_i^{t-1}, then we have $\sum_{j=1}^{2^\Omega} m_a^{\Omega_t}\left[S_i^{t-1}\right]\left(S_j^t\right) = 1$
4. $m_b^{\Omega_t}\left[O_t\right]\left(S_j^t\right)$ a set of BBA functions defined conditionally to the set of possible observation O_t.
5. $m_\pi^{\Omega_1}\left(S_i^{\Omega_1}\right)$ the initial state distribution, generally it is defined as vacuous.

Table 3.1 presents the analogy between the probabilistic HMM and the belief HMM variables.

We notice that all HMM probabilities are replaced by BBA functions when we talk about belief HMM. The advantages of this fact is that BBAs are convertible to many other formats (*bel*, *pl*, *q*, etc).

3.4.2 Three basic problems of belief HMM

The three basic problems of HMM and their solutions are extended to belief functions in (Ramasso, 2007; Ramasso and al, 2007). Like probabilistic HMM, in this section, we will define the three solutions of the three most known problems of HMMs.

Table 3.1: Analogy between HMM and belief HMM variables

	HMM	**Belief HMM**
Set of states	Ω_t	Ω_t
Set of observations	V	V
Transition matrix	A	m_a
Observations	B	m_b
A priori	Π	m_π
Forward variable	α	m_α
Backward variable	β	m_β
γ variable	γ	m_γ
ξ variable	ξ	m_ξ
Viterbi score	δ	m_δ

3.4.2.1 Evaluation problem

As we know the forward algorithm resolves the evaluation problem in the probabilistic case. Ramasso introduced the ***credal forward algorithm*** in order to resolve this problem in the evidential case. Furthermore, he presents two versions, the first one (Ramasso and al, 2007) uses the CRC combination rule (see section 1.4 for more detail), then, it is used for distinct beliefs. The second solution (Ramasso, 2009) is similar to the first one except that it uses the CCRC combination rule, hence, it generalizes the first one and it is used for non distinct beliefs. The credal forward needs as inputs $m_a^{\Omega_t}\left[S_i^{t-1}\right]\left(S_j^t\right)$ and $m_b^{\Omega_t}\left[O_t\right]\left(S_j^t\right)$ which are transformed into commonalities by the mean of the function 1.16. It calculates recursively the forward BBA as:

1. Initialization: we initialize the forward BBA at the first time instant to vacuous.

$$q_\alpha^{\Omega_1}\left(S_i^1\right) = 1 \tag{3.36}$$

2. Induction

$$q_\alpha^{\Omega_{t+1}}\left(S_j^{t+1}\right) = \left(\sum_{S_i^t \subseteq \Omega_t} m_\alpha^{\Omega_t}\left(S_i^t\right).q_a^{\Omega_{t+1}}[S_i^t]\left(S_j^{t+1}\right)\right) \circledast q_b^{\Omega_{t+1}}\left[O_t\right]\left(S_j^t + 1\right) \tag{3.37}$$

   We note the combination rule as $\circledast$, it can be replaced by the CRC or the CCRC. After calculation of $q_\alpha^{\Omega_{t+1}}$, it is transformed into a BBA using the formula 1.17, we save $m_\alpha^{\Omega_{t+1}}(\emptyset)$ then we normalize our BBA to be used in the next iteration.

3. Termination: (Ramasso, 2009) exploits the conflict of the forward BBA to define an evaluation metric that can be used for classification to choose the model that best fits the observation sequence or it can also be used to evaluate the model. He justified his metric

by the fact that "*the lower is the conflict throughout the whole observation sequence, the better is the model λ for explaining these observations*". Then, given a model λ and an observation sequence of length T, the conflict metric is defined by:

$$L_c(\lambda) = \frac{1}{T}\sum_{t=1}^{T} \log\left(1 - m_{\alpha}^{\Omega_{t+1}}[\lambda](\emptyset)\right) \tag{3.38}$$

$$\lambda_* = \arg\max_{\lambda} L_c(\lambda) \tag{3.39}$$

Also the backward algorithm has been extended to belief function. As the case of the credal forward, Ramasso introduces also, two versions of the ***credal backward***. The first one is used for the distinct body of evidence (Ramasso and al, 2007). The second one generalizes the first one and is used for the non distinct body of evidence (Ramasso, 2009). As inputs, the credal backward algorithm requires $m_a^{\Omega_t}\left[S_i^{t-1}\right]\left(S_j^t\right)$ and $m_b^{\Omega_t}[O_t]\left(S_j^t\right)$. The backward BBA is then calculated recursively as:

1. Initialization
$$q_{\beta}^{\Omega_T}\left(S_i^T\right) = 1 \tag{3.40}$$

2. Induction
$$q_{\beta}^{\Omega_t}\left(S_i^t\right) = \sum_{S_j^{t+1}\subseteq\Omega_{t+1}} \left(m_{\beta\circledast b}^{t+1}\left(S_j^{t+1}\right) * q_a^{\Omega_t}\left[S_j^{t+1}\right]\left(S_i^t\right)\right) \tag{3.41}$$

where $m_{\beta\circledast b}^{t+1} = m_{\beta}^{t+1} \circledast m_b^{t+1}$, such that $\circledast$ denotes the conjunctive operator which can be the CRC or the CCRC. $q_a^{\Omega_t}\left[S_j^{t+1}\right](S_i^t)$ is derived from $q_a^{\Omega_t}[S_i^t]\left(S_j^{t+1}\right)$ by the mean of the GBT (see section 1.6) using the relation 1.48.

3.4.2.2 Decoding problem

The goal of the Viterbi procedure, in the probabilistic case, is to define the best state sequence given the observation sequence, then, at time t we should found the best state s_j such that we know the state sequence from the first instant until $t-1$. Many solutions are proposed to extend this algorithm to the TBM (Ramasso and al, 2007; Ramasso, 2009; Serir and al, 2011). All of them search to maximize the state sequence plausibility. According to the definition given in (Serir and al, 2011), the plausibility of a sequence of singleton states $S = \left\{s^1, s^2, \ldots, s^T\right\}$, $s^t \in \Omega_t$ is given by:

$$pl_{\delta}(S) = pl_{\pi}\left(s^1\right) . \prod_{t=2}^{T} pl_a^{\Omega_t}\left[s^{t-1}\right]\left(s^t\right) . \prod_{t=1}^{T} pl_b\left(s^t\right) \tag{3.42}$$

Hence, we can choose the best state sequence by maximizing this plausibility. A further proposal consists to use the probabilistic Viterbi algorithm with plausibilities on singletons as inputs (Ramasso, 2009; Serir and al, 2011). Then inputs of the probabilistic Viterbi will be the following:

- The prior plausibility $pl_{\pi}^{\Omega_1}\left(s_i^1\right)$, in the case where we have no available prior information, we have $pl_{\pi}^{\Omega_1}\left(s_i^1\right) = 1$ for all states.

- Transitions plausibilities $pl_a^{\Omega_t}\left[s_i^{t-1}\right]\left(s_j^t\right)$ defined conditionally to singleton states.
- Observations plausibilities $pl_b^{\Omega_t}\left[O_t\right]\left(s_j^t\right)$ defined on singleton states.

This solution is optimal in term of the complexity, because it has the same complexity as the probabilistic Viterbi. Despite the optimality of this algorithm, it reduces belief functions to singletons. That is why, the third solution is interesting.

The ***credal Viterbi algorithm*** is proposed by Ramasso (Ramasso, 2007; Ramasso, 2009) to extend the probabilistic version into belief functions. It can be used either with distinct body of evidence or with non distinct body of evidence. It needs as inputs $m_a^{\Omega_t}\left[S_i^{t-1}\right]\left(S_j^t\right)$ and $m_b^{\Omega_t}\left[O_t\right]\left(S_j^t\right)$. Then the algorithm follows these steps:

1. Initialization
   (a) $m_\delta^{\Omega_1}\left(\Omega\right)=1$, the Viterbi BBA initialized to vacuous.
   (b) $\psi_1^{'}\left(s_i^1\right)=0$, $\forall s_i^1\in\Omega_1$, this variable stores the best predecessor of the current state. At $t=1$, the first state has not a predecessor,so we initialize the variable to zero.
   (c) $Q_1\left(s_*^1,\lambda\right)=1$, the propagation metric, calculated at each t, used in the termination step to choose the best state sequence.
   (d) $A^1=\emptyset$, the set of predecessor defined at $t-1$, initialized to the empty set because at time $t=1$ we have no predecessors. The use of this variable is justified by the fact that at time t two different states can have the same predecessor at time $t-1$.
2. Recursion: $2\leq t\leq T-1$
   (a) $q_\delta^{\Omega_t}\left(S_j^t\right)=\left(\sum_{S_i^{t-1}\subseteq A^{t-1}} m_\delta^{\Omega_t}\left(S_i^{t-1}\right).q_a^{\Omega_t}\left[S_i^{t-1}\right]\left(S_j^t\right)\right)\circledast q_b^{\Omega_t}\left(S_j^t\right),\forall S_j^t\subseteq\Omega_t$, the operator $\circledast$ can be replaced either by the CRC or by the CCRC.
   (b) Use equation 1.17 to obtain $m_\delta^{\Omega_t}$.
   (c) Calculate $m_\delta^{\Omega_t}\left[s_i^{t-1}\right]$ using the conditioning rule (formulas 1.34).
   (d) $P_t\left[s_i^{t-1}\right]\left(s_j^t\right)=\text{BetP}\left\{m_\delta^{\Omega_t}\left[s_i^{t-1}\right]\right\}\left(s_j^t\right)$, the pignistic transformation can be replaced by the plausibility criteria (see section 1.8 for more details).
   (e) $\psi_t\left(s_j^t\right)=\arg\max_{s_i^{t-1}\in\Omega_{t-1}}\left[\left(1-m_\delta^{\Omega_t}\left[s_i^{t-1}\right]\left(\emptyset\right)\right).P_t\left[s_i^{t-1}\right]\left(s_j^t\right)\right]$.
   (f) $Q_t\left(s_*^t,\lambda\right)=Q_{t-1}\left(\psi_t\left(s_j^t\right),\lambda\right).pl_\delta^t\left(s_j^t\right)$.
   (g) $A^t=\bigcup_{s_j^t\in\Omega_t}\psi_t\left(s_j^t\right)$.
3. Termination
   (a) $s_*^T=\text{argmax}_{s_j^T\in\Omega_T}Q_T\left(s_*^T,\lambda\right)$
4. Path backtracking: $t=T-1,T-2,\ldots,1$
   (a) $s_*^t=\psi_{t+1}\left(s_*^{t+1}\right)$

3.4.2.3 Learning problem

The learning problem is encapsulated in estimating three parameter sets which are:

1. The observation models (generally Gaussian mixture model GMM) that relate the set of observations to the states.
2. The credal transition matrix.
3. The prior BBA.

In the probabilistic case, the Baum-Welch algorithm resolves this problem. In the credal case, Ramasso and Serir have proposed some solutions to estimate these parameters, we discusses them in the following.

Observation models estimation In the probabilistic case, we associate a mixture of Gaussian models to every state. GMMs produce the likelihood of observations at every time instance. In the credal case, (Ramasso, 2007) assimilates these likelihoods to plausibilities as:

$$pl_b\left[s_j^t\right](O_t) \equiv b_j(O_t) \tag{3.43}$$

Then he applies the generalized Bayesian theorem in order to obtain $pl_b^{\Omega_t}[O_t]\left(S_j^t\right)$ using formula 1.45.

Credal transition matrix estimation A first solution that mimic the probabilistic one is introduced in (Ramasso, 2007; Ramasso and al, 2007), this solution is based on estimating the credal γ variable and the credal ξ variable. This last one is defined online and offline. These variable estimations are given by:

$$q_\gamma^{\Omega_t}\left(S_j^t\right) = q_\alpha^{\Omega_t}\left(S_j^t\right) \cap q_\beta^{\Omega_t}\left(S_j^t\right) \tag{3.44}$$

$$q_{\xi_{on}}^{\Omega_{t-1}\times\Omega_t}(S) = q_\alpha^{\Omega_{t-1}\uparrow\Omega_{t-1}\times\Omega_t}(S).q_a^{\Omega_{t-1}\times\Omega_t}(S).q_b^{\Omega_t}[O_t]^{\uparrow\Omega_{t-1}\times\Omega_t}(S) \tag{3.45}$$

$$q_{\xi_{off}}^{\Omega_t\times\Omega_{t+1}}(S) = q_{\xi_{on}}^{\Omega_t\times\Omega_{t+1}}(S).q_\alpha^{\Omega_{t+1}\uparrow\Omega_t\times\Omega_{t+1}}(S) \tag{3.46}$$

$q_a^{\Omega_{t-1}\times\Omega_t}$ is obtained from $q_a^{\Omega_t}\left[S_i^{t-1}\right]\left(S_j^t\right)$ using formula 1.51. Then the credal transition matrix can be estimated as:

$$q_{\overline{a}}^{\Omega_{t+1}}\left[s_i^t\right] = \left(\frac{1}{T}\sum_{t=1}^{T} q_{\xi_{off}}^{\Omega_t\times\Omega_{t+1}}\right)\left[s_i^t\right]^{\downarrow\Omega_{t+1}} \tag{3.47}$$

The last formula is calculated in three steps, the first one consists on calculating the average of $q_{\xi_{off}}^{\Omega_t\times\Omega_{t+1}}$, then the resulting average is conditioned to all singleton states at t. The last step consists on using the marginalization operator (formula 1.42).

The problem with this solution is that it uses many combination rules (four conjunctive combination rules). This can lead to the loss of interest of belief functions, in fact conjunctive

combination promotes the focal elements that have low cardinality, this effect is called ***specialization effect***. To avoid this problem, (Ramasso, 2009) proposes to estimate the credal transition matrix independently from the transitions themselves. He uses the observation BBAs as:

$$m_{\overline{a}_0}^{\Omega_t \times \Omega_{t+1}} \propto \frac{1}{T-1} \cdot \sum_{t=1}^{T} \left(m_b^{\Omega_t} \left[O_t\right]^{\uparrow \Omega_t \times \Omega_{t+1}} \cap m_b^{\Omega_{t+1}} \left[O_{t+1}\right]^{\uparrow \Omega_t \times \Omega_{t+1}} \right) \tag{3.48}$$

where $m_b^{\Omega_t} \left[O_t\right]^{\uparrow \Omega_t \times \Omega_{t+1}}$ and $m_b^{\Omega_{t+1}} \left[O_{t+1}\right]^{\uparrow \Omega_t \times \Omega_{t+1}}$ are computed using the vacuous extension operator (formula 1.38). The formula 3.48 estimates a BBA function defined on the product space $\Omega_t \times \Omega_{t+1}$, to obtain the credal transition matrix, this BBA is conditionned to all subsets of Ω_t, then it is marginalized on Ω_{t+1}.

This estimation formula is used by (Serir and al, 2011) as an initialization for ***ITS (Iterative Transition Specialization)*** algorithm. ITS is an iterative algorithm that uses the credal forward algorithm to improve the estimation results of the credal transition matrix. It stops when the conflict metric (formula 3.38) converged. It takes as inputs the resultant BBA $m_{\overline{a}_0}$ of the formula 3.48 and a convergence threshold ϵ used in the stopping criteria. At the first iteration, the credal forward is applied having $m_{\overline{a}_0}$ and m_b as inputs. The resultant m_α and the conflict metric are then used in the next iteration. The forward BBA is used instead of m_b in the formula 3.48 in order to reestimate the transition used in the next iteration instead of $m_{\overline{a}_0}$ and the conflict metric is used to test the convergence. When the algorithm achieves the last iteration, i.e. the difference between the value of conflict of the current iteration and the previous one is greater than epsilon, we obtain $m_{\overline{a}_*}$ the resultant transition matrix.

The problem with this estimation is that we will obtain a transition matrix with high values on singletons and low values on subsets (doubt). This problem is due to the conjunctive combination used by the forward propagation. In order to solve this problem (Serir and al, 2011) proposed to take the result of the following formula as final credal transition matrix:

$$m_{\overline{a}_*} = \left(m_{\overline{a}_*} + m_{\overline{a}_0}\right)/2 \tag{3.49}$$

Prior estimation The advantage of belief function is that we can define our prior BBA to vacuous which indicates the case of total ignorance. Also, (Ramasso, 2007) gives another way to estimate this BBA using the result of the credal transition as:

$$q_{\overline{\pi}}^{\Omega_1} = q_{\overline{a}}^{\Omega_1 \times \Omega_2 \downarrow \Omega_1} \tag{3.50}$$

3.5 Belief HMM recognizer

Our goal is to create a speech recognizer using the belief HMM instead of the probabilistic HMM. HMM recognizer uses an acoustic model to recognize the content of the signal. Then,

we seek to mimic this model in order to create a belief HMM based one. We should note that existing parameter estimation methods presented for the belief HMM cannot be used to estimate model parameters using multiple observation sequences. This fact should be taken into account when we design our belief acoustic model.

3.5.1 Belief acoustic model

In the probabilistic case, we use an HMM for each acoustic unit, its parameters are trained using multiple speech realization of the unit. In the credal case, a similar model cannot be used. In fact, belief HMM cannot be trained using multiple observation sequences. Hence, we present an alternate method that takes this fact into account.

Let K be the number of the speech realization of a given acoustic unit. These speech realizations are transformed into feature vectors. Hence, we obtain K observation sequences. Our training set will be:

$$O = \left[O^1, O^2, \ldots, O^K\right] \tag{3.51}$$

where $O^k = \left(O_1^k, O_2^k, \ldots, O_{T_k}^k\right)$ is the k^{th} observation sequence of length T_k. These observations are supposed to be independent to each other. So instead of training one model for all observation set O, we propose to create a belief model for each observation sequence O^k. These K models will be used to represent the given acoustic unit in the recognition process.

Like the acoustic model based on the probabilistic HMM, we have to make some choices in order to have a good belief acoustic model. In the first place, we choose the acoustic unit. The same choices of the probabilistic case can be adopted for the belief case. In this document, we consider the case of isolated word recognition problem, hence words are chosen as acoustic units. In the second place, we choose the model. We should note that we cannot choose the topology of the belief HMM, this is due to the estimation process of the credal transition matrix. In other words, the resultant credal observation model is used to estimate the credal transition matrix, then, we cannot choose the topology of our resultant model. Consequently, choosing the model in the credal case consists on choosing the number of states and the number of gaussian mixtures. In our case, we fix the number of states to three and we choose the number of gaussian mixtures experimentally.

The belief acoustic model is designed to recognize isolated words. For each word, we create a set of models that are used together in the recognition process. Figure 3.5 presents the form of our belief recognizer.

3.5.2 Speech recognition process

The belief acoustic model is used in the speech recognition process. As we said in the previous sub-section, we want to recognize isolated words and we design the model for this purpose.

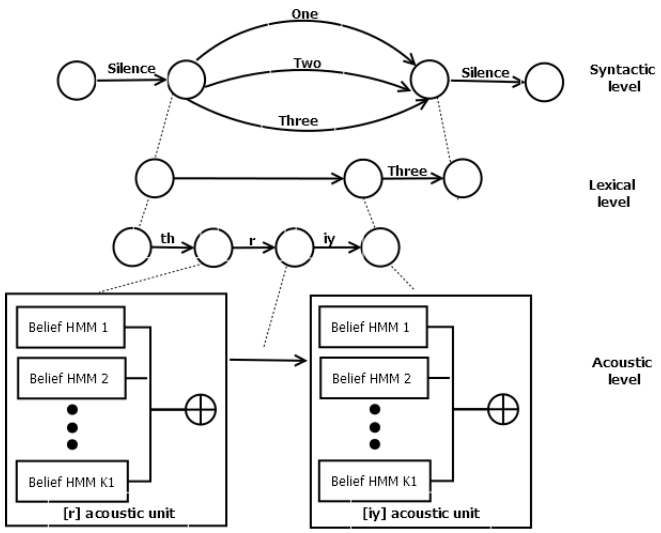

Figure 3.5: Belief HMM recognizer

Now, we explain how the resultant model (shown in figure 3.5) will be used for recognizing speech signal.

Let S be our speech signal to be recognized. Recognizing S consists on finding the most likely set of models. The first step, is to transform S into a sequence of acoustic vectors using the same feature extraction method used for training, then we obtain our sequence of observation O. This last is used as input for all models. The credal forward algorithm is then applied. As shown in figure 3.5, each model gives us an output which is the value of the conflict metric (calculated using the formula 3.38). An acoustic unit is presented by a set of models, every model gives a value for the conflict metric. Then we calculate the arithmetic mean of the resulting values. Finally, as we use several models for each unit, we choose those that optimizes the ***average*** of the conflict metric instead of optimizing the conflict metric, as proposed by (Ramasso, 2009), (using formula 3.39). Hence our observation is recognized.

3.6 Conclusion

Existing belief HMM cannot be trained on multiple observation sequences. Hence the probabilistic acoustic model should be modified in order to be extended to belief functions. An alternate solution is proposed, it takes into account the special characteristics of the belief HMM. In the next chapter, experiments and results will be presented.

4

Experiments and results

4.1 Introduction

To study the effect of belief function theory on the speech recognition process, we compare our belief HMM recognizer to HTK (Young and al, 2006) (an optimized probabilistic HMM speech recognizer). We focus on the isolated speech recognition task. In this chapter, we present our experiments in order to validate our approach. Before that, we introduce HTK, our evaluation method and our speech corpus.

4.2 HMM toolkit (HTK)

HTK is a toolkit for hidden Markov model, developed at Cambridge University Engineering Department (CUED). It builds tools for speech processing and it is optimized for the HMM speech recognition process. HTK provides tools for (Young and al, 2006):

- ***Data manipulation***: HCopy, HQuant, HLEd, HHEd, HDMan and HBuild.
- ***Data visualization***: HSLab, HList and HSGen.
- ***Training***: HCompV, HInit, HRest, HERest, HEAdapt and HSmooth.
- ***Recognition***: HLStats, HParse, HVite and HResults.

These tools are run using a traditional command-line.

4.2.1 HTK training

As inputs, HTK needs a speech corpus and its corresponding transcription files. We will describe the process that we use in our experiments. Figure 4.1 presents a summary of the important steps (Young and al, 2006). Then the training process in HTK follows these steps:

- ***Data preparation***: a feature extraction method is used in order to transform the training speech corpus into sequences of acoustic vectors which are saved in training files (according to Figure 4.1). This step is made by the mean of HCopy tool.

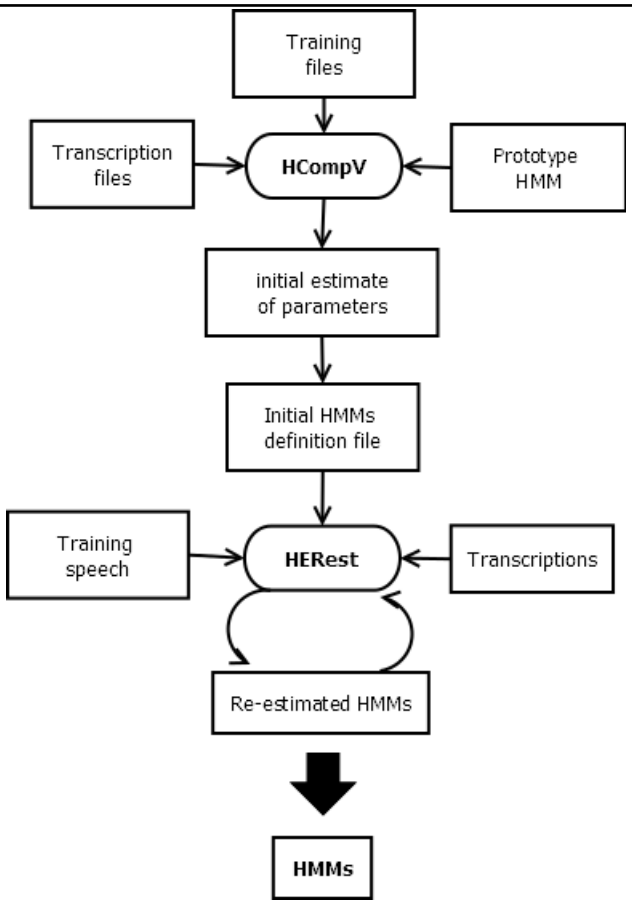

Figure 4.1: Training HMMs with HTK

- ***HMM initialization***: a prototype of HMM is used in order to describe the topology of the HMM (number of states, number of mixtures, etc). Then HCompV tool is run taking as inputs the HMM prototype, the transcription and the training files (as shown in Figure 4.1). HCompV calculates a global speech means and covariances that are used to initialize the set of HMMs and then the HMMs definition file is created (contains an HMM for every acoustic unit).
- ***Parameters estimation***: the tool HERset is used many times. Each run of this tool performs a single re-estimation of HMMs parameters.

4.2.2 HTK testing

The HTK recognizer includes three components:

- A set of HMM: defined in the HMM definition file resulting of the training process.

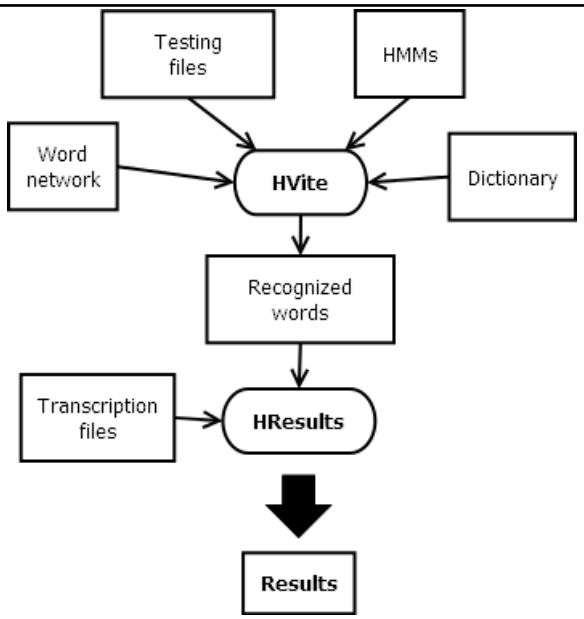

Figure 4.2: Testing process of HTK

- A dictionary: contains the set of words that can be recognized and their phonetic transcription.
- A word network: created by the tool HBuild that takes the dictionary as input.

Then we can test the performance of our recognizer as shown in figure 4.2 (Young and al, 2006). The tool HVite takes as inputs the recognizer and a set of testing file (obtained from a testing corpus by using the same method of feature extraction). This tool runs a Viterbi decoder in order to find the most likely sequence of words. Finally, recognition statistics can be obtained using the HResults tool that compares the sequence of recognized words and the transcription files (that correspond to the testing corpus).

4.3 Evaluation

Generally there exists three types of errors for the speech recognition task (Young and al, 2006):

- Substitution errors (S): we have a substitution if the recognized label and the real one are different.
- Deletion errors (D): we have a deletion if we have an omitted segmentation mark, i.e. for example we obtain two segments for a signal that contains three.
- Insertion errors (I): we have an insertion if we have an additional segmentation mark (the opposite of the deletion).

Let N be the total number of acoustic units in test corpus. Then we can calculate:

$$\text{Percent correct} = \frac{N - S - D}{N} * 100\% \tag{4.1}$$

$$\text{Percent accuracy} = \frac{N - S - D - I}{N} * 100\% \tag{4.2}$$

Percent correct measure does not take into account the number of insertion, it calculates the percentage correctly recognized acoustic units. Percent accuracy takes into account the number of insertions and it better evaluates the recognizer performance.

In the case of our isolated word recognizer, we choose words as acoustic units. Hence, there is no deletion or insertion errors. Then we will have:

$$\text{Percent correct} = \text{Percent accuracy} = \frac{N - S}{N} * 100\% \tag{4.3}$$

This measure can also be called the Percent of correctly clasified acoustic units (words).

4.4 Speech corpus description

We use an isolated word speech corpus that contains speech realization of seven different words which are: apple, banana, kiwi, lime, orange, peach and pineapple. We have fifteen examplary of each word.

In our experiments, this corpus is devided into two, the first one is used for training and the second one is used for test.

4.5 Belief HMM recognizer vs probabilistic HMM recognizer

In this section we present experiments in order to validate our approach. We compare our belief HMM recognizer to the probabilistic HMM recognizer.

We use MATLAB to implement the belief HMM, for the fast mobius transforms, the BBA normalization and the pignistic transformation we use the FMT MATLAB toolbox for belief functions[1] . For the probabilistic HMM recognizer we use HTK.

We use MFCC (Mel Frequency Cepstral Coefficient) as feature vectors. Also, we use a three state HMM and two Gaussian mixtures. Finally, to evaluate our models we calculate the percent of correctly recognized acoustic units. Results are shown in Table 4.1.

[1]Developped by Philippe Smets and available on: http://iridia.ulb.ac.be/~psmets/

Table 4.1: Results summary: the Influence of the number of observations on the recognition rate

Number of observations per acoustic unit	1	2	3
Belief HMM recognizer	**85,71**	71.43	71.43
Probabilistic HMM recognizer (HTK)	13.79	10	84.52

The lack of data for training the probabilistic HMM leads to a very poor learning and the resulting acoustic model cannot be efficient. Then using a training set that contains only one exemplary of each acoustic unit leads to have a bad probabilistic recognizer. In this case our belief HMM based recognizer gives a recognition rate equal to 85.71% against 13.79% for the probabilistic HMM which is trained using HTK (Young and al, 2006) (see Figure 4.3). This result shows that the belief HMM recognizer is insensitive to the lack of data and we can obtain a good belief acoustic model using only one observation for each unit. In fact, the belief HMM models knowledge by taking into account doubt, imprecision and conflict which lead to a discriminative model in the case of the lack of data.

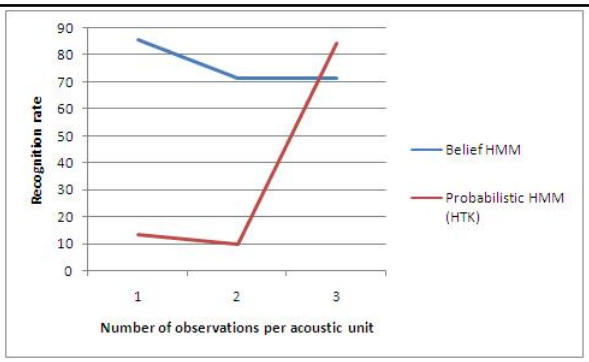

Figure 4.3: Influence of the number of observations on the recognition rate

HTK is a toolkit for HMMs and it is optimized for the HMM speech recognition process. It is known to be powerful under the condition of having many exemplary of each acoustic unit. Hence, it needs to use several hours of speech for training. Having a good speech corpus is very expensive which influence the cost of the recognition system. Then, the speech recognition systems are very expensive. Consequently, using the belief HMM recognizer can greatly minimize the cost of these systems.

4.6 Conclusion

To sum up, using belief functions theory and belief HMM in the speech recognition process can give interesting results. Results prove that our belief HMM recognizer can be trained using only one exemplary of each acoustic unit and, in this case, it gives better recognition rate than the probabilistic HMM.

Conclusion and perspectives

Conclusion

Speech Recognition aims to label the speech signal. In other words, it searches to predict the spoken words automatically. The speech is a very uncertain signal because of the variations with the speaker voices. Therefore, that is a real problem to build good models to recognize the speech. In literature, there exist many methods for speech recognition among them the HMM based speech recognition which is widely used because it guarantees a good recognition rate. It allows us to recognize about 80% of a given speech signal, but this recognition rate still not yet satisfying.

Hidden Markov Models (HMM) have shown their capacity and performance to treat large speech corpus for several years. It is a statistical approach that is used for speech recognition. It is performed in two steps. The first is the learning step in which models parameters are learned using a pre-segmented speech signal. The second is the decoding or alignment step, in which HMMs are used to find the sequence of acoustic units in the signal and their boundaries.

Belief Hidden Markov Model is an extension of HMM to the theory of belief functions. This theory is one of the most popular among the quantitative approaches because it can be seen as a generalization of others. Its strength lies in: its richer representation of uncertainty and imprecision compared to the probability theory, and its higher ability to combine pieces of information.

We proposed the Belief HMM recognizer. We showed that incorporating belief functions theory in the speech recognition process is very beneficial, in fact, it reduces considerably the cost of the speech recognition system.

Perspectives

In this master thesis, we developed a new speech recognition system based on the use of the belief HMM. We showed that our is very powerful in the case of lack of data, in fact it can be trained using only one speech realization of each acoustic unit.

As future works, we suggest to focus on the following points:

- ***Continuous speech recognition task***: Isolated word recognition is a simple task generally used to validate speech recognizer. Continuous speech recognition task is a more compliacated problem and its basic goal is to recognize connected words. Hence, the real challenge is to develop a belief speech recognizer for continuous speech.

- ***Noisy speech signal***: Our experiments are made using a non-noisy speech signal. However, in real word applications, speech recognizer has to be robust to noise. Hence, building a model that can recognize noisy speech signal will be very interesting.

- ***Speech recognizer independent to speaker***: Speaker independent recognizer can label speech signals that have different sources. Such a system needs a very big speech corpus for training. Using our belief HMM recognizer can be very beneficial in this case.

Bibliography

André-Obrecht R. (1988). A new statistical approach for the automatic segmentation of continous speech signals. IEEE Transactions on Acoustics, Speech and Signal Processing, volume 36, pages 29–40.

Basseville M. and Benveniste A. (1983). Sequential detection of abrupt changes in spectral characteristics of digital signals. IEEE Transactions on Information Theory, volume 29, pages 709–724.

Bouma C. A. (2009). Speech Processing (part 1). INSIDE COLLECTION (COURSE): Purdue Digital Signal Processing Labs (ECE 438). Available on http://cnx.org/content/m18086/latest/?collection=col10593/latest.

Brugnara F. Falavigna D. and Omologo M. (1993). Automatic segmentation and labeling of speech based on hiddenMarkov models. Speech Communication, volume 12, pages 370–375.

Carvalho P. Oliveira L. C. Trancoso I. M. and Céu Viana M. (1998). Concatenative Speech Synthesis for European Portuguese. 3rd ESCA/COCOSDA Workshop on Speech Synthesis, NSW Australia. pages 159-163.

Cox S. Brady R. and Jackson P. (1998). Techniques for accurate automatic annotation of speech waveforms. Proc. ICASSP. Sydney, Australia. pages 1947-1950.

De Cheveigne A. and Kawahara H. (2002). YIN, a fundamental frequency estimator for speech and music. J. Acoust. Soc. Am, volume 111, pages 1917–1930.

Dempster, A. P. (1967). Upper and Lower probabilities induced by a multivalued mapping. Annals of Mathematical Statistics, 38, pages 325-339.

Denoeux T. and Ben Yaghlane A. (2002). Approximating the Combination of Belief Functions using the Fast Moebius Transform in a coarsened frame. International Journal of Approximate Reasoning, Vol. 31, No. 1-2, pages 77-101.

Denoeux T. (2007). Conjunctive and disjunctive combination of belief functions induced by non distinct bodies of evidence. Artificial Intelligence, vol. 172, pages. 234–264.

Drugman T. and Alwan A. (2011). Joint Robust Voicing Detection and Pitch Estimation Based on Residual Harmonics. Proceedings of Interseech. Florence, Italy. pages 1973-1976.

Dubois D. and Prade H. (1986).The principle of minimum specificity as a basis for evidential reasoning. In B. Bouchon and R. R. Yager, editors, Uncertainty in Knowledge-Based Systems. Springer-Verlag, Berlin. pages 75–84.

Dutoit T. (1999). A Short Introduction to Text-to- Speech Synthesis. Available on http://scgwww.epfl.ch/courses/Traitement_de_la_parole-2009-2010-pdf/11-Short-Intro-to-TTS.pdf.

Dutoit T. ouvreur L. Malfrère F. Pagel V. and Ris . (2002). Synthèse vocale et reconnaisance de la parole : droites gauches et mondes parallèles. Actes du 6è Congrès Français d'Acoustique, Lille, France, pages 8–11.

Hsia Y. T. (1991). Characterizing belief with minimum commitment. In Proceedings of the Inter- national Joint Conference on Artificial Intelligence, IJCAI-91, Morgan Kaufman, San Mateo, CA. pages 1184–1189.

Hunt A. and Black A. (1996). Unit selection in a concatenative speech synthesis system using a large speech database. Proc. of ICASSP, Atlanta, Georgia, pages 373-376.

Jarifi S. Pastor D. and Rosec O. (2005). Brandt's GLR method & refined HMM segmentation for tts synthesis application. 13th European Signal Processing Conference (EUSIPCO). Antalya,Turkey.

Jarifi S. (2007). Segmentation automatique de corpus de parole continue dédiés à la synthèse vocale. Thèse de doctorat, Université de Rennes I.

Jarifi S. Pastor D. Rosec O. (2008). A fusion approach for automatic speech segmentation of large corpora with application to speech synthesis. Speech Communication, volume 50, pages 67–80.

Klatt H. D. (1980). Software for a cascade/parallel synthesizer. Journal of the Acoustical Society of America (JASA), volume 67, pages 971–995.

Klement E. P. Mesiar R., and Pap E. (2000). Triangular norms. Kluwer Academic Publishers, Dordrecht.

Kouemou G. L. (2011). History and Theoretical Basics of Hidden Markov Models. Hidden Markov Models, Theory and Applications. Edited by: Przemyslaw Dymarski.

Maeda S. (1990). Compensatory articulation during speech : evidence from the analysis and synthesis of vocal-tract shapes using an articulatory model. Speech Production and Speech Modelling, Kluwer Academic, pages 131–149.

Malfrère F. Deroo O. Dutoit T. and Ris C.. (2003). Phonetic alignment : speech synthesis based vs. viterbi-based. Speech Communication, volume 40, pages 503–515.

Martin A. and Mauuary L. (2003). Voicing Parameter and Energy Based Speech/Non-Speech Detection for Speech Recognition in Adverse Conditions. Eurospeech, Geneva, Switzerland. pages 3069–3072.

Muda L. Begam M. and Elamvazuthi I. (2010). Voice Recognition Algorithms using Mel Frequency Cepstral Coefficient (MFCC) and Dynamic Time Warping (DTW) Techniques. Journal of computing, volume 2, issue 3, pages 138-143.

Nefti S. (2004). Segmentation automatique de la parole en phones. Correction d'étiquetage par l'introduction de mesures de confiance. Thèse de doctorat, Université de Rennes I.

Peccei J. (2006). A Beginner's Guide to Phonetics. http://jcarreras.homestead.com/rrphonetics1.html.

Petek B. Andersen O. and Dalsgaard P. (1996). On the robust automatic segmentation of spontaneous speech. ICSLP, volume 2, pages 913–916.

Pierrehumbert J. (1990). Phonological and phonetic representation. Journal of Phonetics, volume 18, pages 375-394.

Rabiner L. (1989). A tutorial on hidden markov models and selected applications in speech recognition. Proceedings of IEEE, volume 77, No. 2, pages 257-286.

Rabiner L. Juang B. H. (1993). Fundamentals of speech recognition. Prentice-Hall, Inc. Upper Saddle River, NJ, USA.

Ramasso E, Rombaut M, and Pellerin D. (2007). Forward-Backward-Viterbi procedures in theTransferable Belief Model for state sequence analysis using belief functions. ECSQARU, Hammamet: Tunisie pages. 405–417.

Ramasso E. (2007). Reconnaissance de séquences d'états par le Modèle des Croyances Transférables Application à l'analyse de vidéos d'athlétisme. Thèse de Doctorat. Université Joseph Fourier.

Ramasso E. (2009). Contribution of belief functions to HMM with an application to fault diagnosis. IEEE International Workshop on Machine Learning and Signal Processing, Grenoble, France, Sept. 2-4. pages 1-6.

Riley M. D. (1987). Time-Frequency Representations for Speech Signals. Available on http://hdl.handle.net/1721.1/6827. Seen on 31/03/2012.

Serir L. Ramasso E. and Zerhouni N. (2011). Time-Sliced Temporal Evidential Networks: the case of Evidential HMM with application to dynamical system analysis. IEEE International Conference on Prognostics and Health Management. Denver, Colorado, USA. pages 1-10.

Shafer G. (1976). A mathematical theory of evidence. Princeton University Press.

Smets P. (1990). The Combination of Evidence in the Transferable Belief Model. IEEE Transactions on Pattern Analysis and Machine Intelligence, 12(5) pages 447-458.

Smets P. (1993). Beliefs functions: The Disjunctive Rule of Combination and the Generalized Bayesian Theorem. International Journal of Approximate Reasoning, 9 pages 1–35.

Smets P. and Kennes R. (1994). The Transferable Belief Model. Artificial Intelligence, 66(2) pages 191–234.

Smets P. (1995). The canonical decomposition of a weighted belief. in Morgan Kaufman. ed. International Joint Conference on Artificial Intelligence, pages. 1896-1901.

Smets P. (2000). Belief functions and the Transferable Belief Model. Available on www.sipta.org/documentation/belief/belief.ps.

Smets P. (2005). Decision making in the TBM : The necessity of the pignistic transformation. Int. Jour. of Approximate Reasoning, 38 pages 133–147.

Talkin D. (1995). A robust algorithmfor pitch tracking (RAPT). Speech coding and synthesis, pages 495–518. Elsevier.

Toledano D. T. Hernandez Gomez L. A. and Grande L. V. (2003). Automatic phonetic segmentation. IEEE Trans. Speech, Audio Processing, volume 11, no. 6, pages 617-625.

Vorstermans A. Martens J. P. and Van Coile B. (1996). Automatic segmentation and labelling of multi-lingual speech data. Speech Comm. vol. 19, pages 271-293.

The Voice Problem Website. (2003). Understanding How Voice is Produced. http://www.voiceproblem.org/anatomy/understanding.php. Seen on 31/03/2012.

Wikipedia. (2012). Vowel. Wikipedia, the free encyclopedia. http://en.wikipedia.org/wiki/Vowel. Seen on 31/03/2012.

Young S. Evermann G. Kershaw D. Moore G. Odell J. Ollason D. Valtchev V. and Woodland P. (2006). The HTK book (for HTK version 3.4). Microsoft Corporation and Cambridge University Engineering Department.