From speech to SQL queries : a speech understanding system

Salma Jamoussi, Kamel Smaïli, Jean-Paul Haton

From speech to SQL queries : a speech understanding system

Salma Jamoussi and Kamel Smaïli and Jean-Paul Haton

LORIA/INRIA-Lorraine
615 rue du Jardin Botanique, BP 101, F-54600 Villers-lès-Nancy, France

Abstract

In this paper, we describe our speech understanding system and we test it on two different applications. The proposed system is a task specific one and it concern especially oral database consultation tasks. In this work, we consider that the automatic speech understanding problem could be seen as an association problem between two different languages. At the entry, the request expressed in natural language and at the end, just before the interpretation stage, the same request is expressed in term of concepts. A concept represents a given meaning, it is defined by a set of words sharing the same semantic properties. In this paper, we propose a new Bayesian network based method to automatically extract the underlined concepts. We also propose and compare three approaches for the vector representation of words. We finish this paper by a description of the post-processing step during which we generate corresponding SQL queries to the pronounced sentences and we connect our understanding system to a speech recognition engine. This step allows us to validate our speech understanding approach by obtaining with the two treated applications the rates of 78% and 81% of well formed SQL requests.

Key words:
Speech understanding, semantic measures, automatic extraction of concepts, word vector representation, Bayesian networks, AutoClass.

1 Introduction

Language and speech recognition processing become very important research areas and their applications are more and more present in our daily life. These

Email address: {jamoussi, smaili, jph}@loria.fr (Salma Jamoussi and Kamel Smaïli and Jean-Paul Haton).

interactive applications must be able to process users spoken queries. It means they have to recognize what has been uttered, extract its meaning and give suitable answers or execute right corresponding commands. In such applications, the speech understanding component constitutes a key step. Several methods were proposed in the literature to clean up this problem and the majority of them is based on stochastic approaches. These methods allow to reduce the need of human expertise, however they require a supervised learning step which means a former stage of manual annotation of the training corpus [1,4,5].

The data annotation step consists in segmenting the data into conceptual segments where each segment represents an underlined meaning [1]. Within this step, we have to find first of all the list of concepts which are related to the considered corpus. Then, we can use these concepts to label the segments of each sentence in the corpus and finally, we can launch the training step. Doing all this in a manual way constitutes a tiresome and an expensive phase. Moreover, the manual extraction is prone to subjectivity and to human errors. Automating this task will thus reduce the human intervention and will especially allow us to use the same process when context changes. Our purpose in this paper is to fully automate the understanding process from the input signal until the SQL request generation step.

In this paper, we start by giving a brief description of the statistical approach which constitutes the most used method for resolving the speech understanding problem. We present then the detailed architecture of our understanding system where we propose a new approach to automatically extract the semantic concepts of the considered application. For this, we use a Bayesian network for unsupervised classification, called AutoClass and we expose three methods for the vector representation of words, these representations aim to help the Bayesian network to build up efficient concepts. We test this method on two applications data and we compare the Bayesian network performances with those obtained by the Kohonen maps and the K-means algorithm. Then, we will describe the last stage of our understanding process, in which we label the user requests and we generate the associated SQL queries. Finally, we use a speech recognition system to be able to treat sentences given in their signal forms. Tow kinds of results are given in this paper. The first results are obtained when the system input is speech and the second ones concern the textual entry form.

2 The statistical approach for the speech understanding problem

A speech understanding system could be considered as a machine that produces an action as the result of an input sentence. Thus, the understanding

problem could be seen as a translation process, it translates a signal (represented by a sequence of words) into a special form that represents the meaning convoyed by the sentence. In a first time, the sentence is labelled by a list of conceptual entities (often called concepts), these labels constitute a useful intermediate representation which must be simple and representative. In a second time, this representation will be used to interpret semantically the sentence.

The speech understanding problem can be seen then as an association problem, where we have to associate inputs (e.g. speech or text) to their respective meanings represented by a list of concepts. A concept is related to a given meaning, it is given by a set of words expressing the same idea and sharing the same semantic properties. For example, the words *plane, train, boat, bus* can all correspond to the concept "*transport means*" in a travel application.

The step of interpretation consists in converting the obtained concepts to an action to be done as a final response to the user. In order to achieve such a goal, we have to convert these concepts into a target formal command (e.g. an SQL query, a shell command, etc.). The figure 1 illustrates the general architecture of such speech understanding system, this model was given in [5] and it was included in several other works because of its effectiveness and its simplicity [1,4].

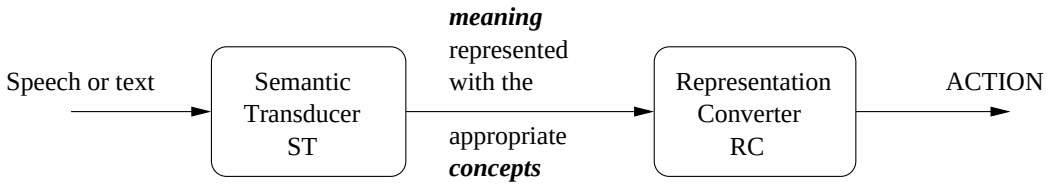

Fig. 1. General architecture of a speech understanding system.

The semantic transducer component is considered as the main module in a speech understanding system while it translates a given sentence to its conceptual form. In the literature, this step is often achieved by using the Hidden Markov Models technique. The aim is to find the concept list maximizing the likelihood $P(C|A)$ where C represents a set of concepts and A is the acoustic representation of the pronounced sentence S :

$$\hat{C} = \arg\max_{C} P(C|A)$$

Using the Bayes formula we can transform this last equation into :

$$\hat{C} = \arg\max_{S,C} P(A|S)P(S|C)P(C)$$

The terms of this equation represent three particular models :

- The acoustic model represented by the probability of the acoustic observation A given the sequence of the sentence words S : $P(A|S)$
- The syntaxic model which is given by $P(S|C)$: the probability that we have a word sequence S given some meaning C. C represents a certain sequence of concepts.
- The semantic model given by the probability $P(C)$ of a concept sequence C.

The acoustic model is often maximized using some speech recognition techniques. For the understanding task only the tow last models are considered. In this case we approximate these models to :

$$P(S|C) \simeq \prod_i P(w_i|w_{i-1}, \ldots, w_{i-n+1}, c_i)$$

$$P(C) \simeq \prod_i P(c_i|c_{i-1}, \ldots, c_{i-m+1})$$

Where the w_i are the words of the sentence S and the c_i are the concepts composing C. To simplify these equations, we often consider that $n = 1$ and $m = 2$ to obtain :

$$P(Ph|C) \simeq \prod_i P(w_i|c_i)$$

$$P(C) \simeq \prod_i P(c_i|c_{i-1})$$

In this case, the HMM states will correspond to the concepts and $P(c_i|c_{i-1})$ is the transition probability between the two states c_i and c_{i-1}. $P(w_i|c_i)$ represents the probability of observing the word w_i at the HMM state c_i. A very simple case of this modelisation problem is given by the next figure (figure 2).

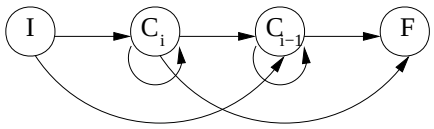

Fig. 2. A simple conceptual modelisation example using HMM.

The problem with this method is the needed data for the training step. In fact, to compute our model probabilities we have to provide huge corpus of annotated data. The annotation step is achieved manually and it is considered as the most difficult task. In this paper, we try to automate all the understanding process especially to avoid these very heavy manual steps.

3 Our understanding system architecture

In our work, we adopt the same general architecture as given in the figure 1 but we propose new techniques within each component. Moreover, we try to extract automatically our concepts in a preliminary step.

Our detailed system architecture is shown in the figure 3. It is composed of three principal components. The first one is a corpus processing module, where we try to automatically extract the appropriate list of concepts by using a Bayesian network. This step is the more crucial one, because we will use its output in all the other steps. The second and the third ones are those already defined in the figure 1. In our case the "Semantic Transducer" is a sentence labelling module where we associate to each word its semantic class. The "Representation Converter" is divided into two steps. The first one is the "Generic query production" stage, it uses the sentence concepts to build up a generic query for this sentence. The second one is the "SQL query generation" stage, it uses the initial sentence and the generic query to set concepts and to provide the final corresponding SQL query.

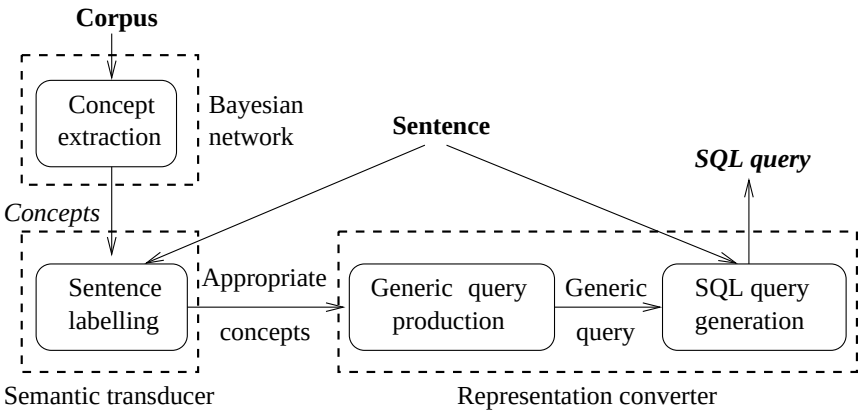

Fig. 3. Our detailed understanding system architecture.

4 Bayesian approach for automatic concept extraction

The aim of this step is to identify the semantic concepts related to our application. The manual determination of these concepts is a very heavy task, so we should find an automatic method to achieve such a work. The method to be used must be able to gather the words of the corpus in various classes in order to build up the list of the appropriate concepts.

To reach our goal we used an unsupervised classification technique. Among the unsupervised classification methods, we tried the Kohonen maps and the K-means method. The obtained concepts were quite significant, but contained

some "noise", it means that we found many words which did not have their place in the meaning expressed by these concepts. To solve this problem, we explored other methods and adopted the Bayesian network technique because of its mathematical base and its powerful inference mechanism [2]. A comparison between these methods performances will be given in the section 6.2. In this section, we describe the Bayesian theory principle used for the clustering problem and we detail some calculation stages allowing us to find the concepts related to our training corpus.

In this paper, we use a Bayesian network conceived for the clustering problem and called "AutoClass". It accepts real and discrete values as input. As result, it provides for each input, its membership probabilities in all the found classes. AutoClass supposes that there is a hidden multinomial variable which represents the various classes of the input data. It is based on the Bayes theorem expressed by :

$$p(H|D) = \frac{p(H)\, p(D|H)}{p(D)} \tag{1}$$

In our case, D represents the observed data, that means, the words to be classified. H is a hypothesis concerning the number of classes and their descriptions in term of probabilities. AutoClass tries to maximize the probability $p(H|D)$, i.e. given D (the words of the corpus), we must select H (the set of concepts) which maximizes this probability.

In our Bayesian network, a word x_i is given by a vector of K attributes, $x_{ik},\ k \in \{1...K\}$. A concept C_j is also described by K attributes, each one is modeled by a normal Gaussian distribution. $\overrightarrow{\theta}_{jk}$ is a vector parameter describing the attribute number k of the concept number j, C_j. It contains two elements, the distribution mean μ_{jk} and the variance σ_{jk}. For the whole concept, this vector is noted $\overrightarrow{\theta_j}$ and it contains the $\overrightarrow{\theta_{jk}}$ of all the attributes of the concept C_j. The probability that a word x_i belongs to the concept C_j, called *the class probability* is noted π_j and it also constitutes a descriptive parameter of the concept C_j.

Thus, we defined our network parameters, the data D represents the words as a vector $\overrightarrow{x}$ with I elements including all the x_i. The hypothesis H corresponds to the description of the concepts and it is represented by three elements, the number of concepts J and the two vectors $\overrightarrow{\pi}$ and $\overrightarrow{\theta}$ which contains respectively π_j and $\overrightarrow{\theta}_j$ of all the concepts. AutoClass divides the concept identification problem into two parts : the determination of the classification parameters ($\overrightarrow{\pi}$ and $\overrightarrow{\theta}$) for a given number of concepts and the determination of the number of concepts J. This last problem requires several approximations which are explained in [2]. In what follows, H will only represent the vectors

$\overrightarrow{\pi}$ and $\overrightarrow{\theta}$. Replacing D and H by their values in the equation 1 we obtain :

$$p(\overrightarrow{\theta}, \overrightarrow{\pi}|\overrightarrow{x}) = \frac{p(\overrightarrow{\theta}, \overrightarrow{\pi})\ p(\overrightarrow{x}|\overrightarrow{\theta}, \overrightarrow{\pi})}{p(\overrightarrow{x})} \tag{2}$$

Where $p(\overrightarrow{\theta}, \overrightarrow{\pi})$ is the prior distribution of the classification parameters, its calculation is well described in [2]. The prior probability of words, $p(\overrightarrow{x})$ can be computed directly, it is simply considered as a normalizing constant. Here, we are interested in the calculus of the probability $p(\overrightarrow{x}|\overrightarrow{\theta}, \overrightarrow{\pi})$ which represents the likelihood function of the data.

It is known that $\overrightarrow{x}$ is a vector representing all the words of the training data, the probability of this vector is obtained by the product of the probabilities of all the words separately as shown in the following equation :

$$p(\overrightarrow{x}|\overrightarrow{\theta}, \overrightarrow{\pi}) = \prod_{i=1}^{I} p(x_i|\overrightarrow{\theta}, \overrightarrow{\pi}) \tag{3}$$

$p(x_i|\overrightarrow{\theta}, \overrightarrow{\pi})$ is the probability of observing the word x_i independently of the concept to which it belongs. It is given by the sum of the probabilities that this word belongs to each concept separately, weighted by the class probabilities as indicated by the following equation :

$$p(x_i|\overrightarrow{\theta}, \overrightarrow{\pi}) = \sum_{j=1}^{J} \pi_j\ p(x_i|x_i \in C_j, \overrightarrow{\theta}_j) \tag{4}$$

Since the word x_i is described by K attributes, with the strong assumption that these attributes are independent, the probability $p(x_i|x_i \in C_j, \overrightarrow{\theta}_j)$ can thus be written in the following form :

$$p(x_i|x_i \in C_j, \overrightarrow{\theta}_j) = \prod_{k=1}^{K} p(x_{ik}|x_i \in C_j, \overrightarrow{\theta}_{jk}) \tag{5}$$

AutoClass models each real attribute by a normal Gaussian distribution represented by the vector $\overrightarrow{\theta}_{jk}$ which contains two parameters μ_{jk} and σ_{jk}. In this case, the class distribution $p(x_{ik}|x_i \in C_j, \overrightarrow{\theta}_{jk})$, can be written like this :

$$p(x_{ik}|x_i \in C_j, \mu_{jk}, \sigma_{jk}) = \frac{1}{\sqrt{2\pi}\sigma_{jk}} \exp\Big[-\frac{1}{2}\Big(\frac{x_{ik} - \mu_{jk}}{\sigma_{jk}}\Big)^2 \Big] \tag{6}$$

Once this class distribution is determined, we only have to seek for the concept parameters which maximize the starting probability $p(\overrightarrow{\theta}, \overrightarrow{\pi}|\overrightarrow{x})$ and find the

optimal concepts related to our data [2].

5 Vector representations of words

In this section we present three different approaches to represent words in vectorial aspect. This representation, which must be semantically significant, constitutes a key stage in the understanding process. In fact, according to this representation, the Bayesian network will decide of words to group in the same class in order to build up the needed list of concepts.

5.1 Word context

One word can have several features but only few of them are relevant for a good semantic representation. In a first step, we decided to associate to each word its different contexts. We consider that if two words have the same contexts then they are semantically similar. In this approach, a word will be represented by a vector of $2 \times N$ elements containing the N left context words and the N right context words. Figure 4 shows how we associate for each word its left and right bigram contextual representation.

<BG> <BG> W_1 W_2 W_3 W_4 <ED> <ED>

BG	BG	W_1	W_2
BG	W_1	W_2	W_3
W_2	W_3	W_4	ED
W_3	W_4	ED	ED

Fig. 4. The bigram contextual representation of words.

Using this vector representation of words with our Bayesian network, we obtain many classes representing good semantic concepts, but an important overlapping has been noticed. Moreover, we had difficulties in controlling the number of concepts.

5.2 Similarity vector representation

To find more homogeneous concepts, we completely changed the vector structure of each word. We used the average mutual information measure which tries to find contextual similarities between words.

In this approach, we associate to each word a vector with M elements, where M is the size of the lexicon. The jth element of this vector represents the average mutual information between the word number j of the lexicon and the word to be represented (equation 7).

$$W_i = [I(w_1 : w_i), I(w_2 : w_i), \ldots, I(w_j : w_i), \ldots, I(w_M : w_i)] \tag{7}$$

This vector expresses the similarity degree between the word to represent and all the other words of the corpus. The formula of the average mutual information between two words w_a and w_b is given by :

$$\begin{aligned} I(w_a : w_b) = P(w_a, w_b) \log \tfrac{P(w_a|w_b)}{P(w_a)P(w_b)} + P(w_a, \overline{w}_b) \log \tfrac{P(w_a|\overline{w}_b)}{P(w_a)P(\overline{w}_b)} + \\ P(\overline{w}_a, w_b) \log \tfrac{P(\overline{w}_a|w_b)}{P(\overline{w}_a)P(w_b)} + P(\overline{w}_a, \overline{w}_b) \log \tfrac{P(\overline{w}_a|\overline{w}_b)}{P(\overline{w}_a)P(\overline{w}_b)} \end{aligned} \tag{8}$$

Where $P(w_a, w_b)$ is the probability to find w_a and w_b in the same sentence, $P(w_a \mid w_b)$ is the probability to find w_a knowing that we already met w_b, $P(w_a)$ is the probability of the word w_a and $P(\overline{w}_a)$ is the probability of any other word except w_a.

By using this vector representation, the Bayesian network achieves homogeneous semantic classes. A class is made up of words sharing the same semantic properties. The number of classes is very coherent with our application. This representation also enables us to solve the problem of the overlapping between concepts. However some imperfections are still present and we will try to avoid them with the next proposed word representation.

5.3 Combinaison : context and similarity

In this approach we combined the two preceding representations in order to improve results. In the first approach we work on the occurrence level where we directly exploit information related to the word context. In the second one, we use a measure to seek for similarities between words. We can easily notice that the information used in these two methods is different but complementary.

To combine these two methods, we decided to represent each word by a matrix $M \times 3$ of average mutual information measures. The first column of this matrix corresponds to the preceding vector of average mutual information (see section 5.2), the second column represents the average mutual information measures between the vocabulary words and the left context of the word to be represented. The third column is determined by the same manner but it concerns the right context. The jth value of the second column is the weighted

average mutual information between the jth word of the vocabulary and the vector constituting the left context of the word W_i. It is calculated as follows:

$$IMM_j(C_l^i) = \frac{\sum_{w_l \in L_{W_i}} I(w_j : w_l) \times K_{wl}}{\sum_{w_l \in L_{W_i}} K_{wl}} \quad (9)$$

Where $IMM_j(C_l^i)$ is the average mutual information between the word w_j of the lexicon and the left context of the word W_i. $I(w_j : w_l)$ represents the average mutual information between the word number j of the lexicon and the word w_l which belongs to the left context of the word W_i and K_{wl} is the number of times where the word w_l is found in the left context of the word W_i. The word W_i thus represented by the matrix shown in the figure 5.

$$W_i = \begin{bmatrix} I(w_1 : w_i) & IMM_1(C_l^i) & IMM_1(C_r^i) \\ I(w_2 : w_i) & IMM_2(C_l^i) & IMM_2(C_r^i) \\ \vdots & \vdots & \vdots \\ I(w_j : w_i) & IMM_j(C_l^i) & IMM_j(C_r^i) \\ \vdots & \vdots & \vdots \\ I(w_M : w_i) & IMM_M(C_l^i) & IMM_M(C_r^i) \end{bmatrix}$$

Fig. 5. Representation of the word W_i by the combined method.

The matrix used to represent a word in the corpus exploits a maximum number of information that can be related to this word. It considers its context and its similarity with all the other words of the lexicon. Such a word representation could help the Bayesian network to classify the words and allows us to considerably improve results. We obtain a coherent list of concepts. We decided to keep these ones for the rest of the understanding treatment.

6 Experimental conditions

In our work, we are interested in two kinds of applications. The first one is a bookmark consultation application where we use the corpus of the European project MIAMM. The aim of this project is to build up a platform of an oral multimodal dialogue. The corpus contains 71287 different queries expressed in French. Each query expresses a particular manner to request the database. Some examples of these queries are given in the table 1.

The second application concerns the purse and it is related to a french project called IVOMOB that aims to industrialize the techniques of speech recognition and speech understanding. The training corpus that we used contains 51864

queries written in French and expressing many manners to request the purse database. Some examples of these queries are also given in table 2.

Table 1
Some examples of queries in the MIAMM corpus.

Show me the contents of my bookmarks.
I would like to know if you can take the contents that I prefer.
Do you want to select the titles that I prefer.
Is it possible that you select the first of my bookmarks.
Is it possible to indicate me a similar thing.
Can you show me only December 2001.
It is necessary that you print the list that I used early this morning.

Table 2
Some examples of queries in the IVOMOB corpus.

Give me the course level of the Alcatel company.
Can you provide me the action level of Alcatel company.
I need the progression of the minimum level of the Alcatel group course.
I would like to know the BNP course evolution.
I need the course level of the BNP company by mail.
Can you give me by fax the most high course level of BNP.
Just the maximum of the Alcatel action.

Each training corpus contains almost one hundred vocabulary words after eliminating all the tool words as well as the words having weak frequency. Our aim in this step is to cluster all these vocabulary words to form the semantic concepts of the treated application.

6.1 The evaluation method for the concept extraction step

Making an objective evaluation of semantic concepts is a very hard task. In fact, if we want to decide if a concept is correct or not, we can found as much responses as asked people. This kind of question depends on many factors like the application context, the asked people, the words nature, etc.

In our case, we use an evaluation measure well known and used in some similar tasks which is the efficacity measure. This latter aims to find a kind of percentage of correct words in each found concept. Consequently, we need to draw up manually a reference list of concepts that we consider perfect and where the words are very well classified. In our work, we define 13 reference concepts

for the MIAMM application and 11 reference concepts for the IVOMOB application. We assume that each reference concept is noted C_i. To compute the efficacity of an obtained list of n concepts $(C_{j,1\leq j\leq n})$, we first need to define the recall and the precision measures as :

$$recall(i,j) = \frac{n_{ij}}{N_i}$$

$$precision(i,j) = \frac{n_{ij}}{N_j}$$

Where n_{ij} is the number of words present in both concepts C_i and C_j. N_i is the total number of words of the concept C_i and respectively for N_j.

Thus the efficacity $F(i,j)$ will be defined as :

$$F(i,j) = \frac{2 \times precision(i,j) \times recall(i,j)}{precision(i,j) + recall(i,j)}$$

This efficacity measure concerns only two concepts C_i and C_j. To compute the efficacity related to the whole reference concept C_i we just seek for the maximum efficacity value obtained with this latter :

$$F(i) = \max_j F(i,j)$$

For the whole list of the reference concepts the final value of efficacity is computed as a weighted mean :

$$F = \sum_i p_i \times F(i)$$

Where p_i represents the weight of the concept C_i and it is given by :

$$p_i = \frac{N_i}{\sum_k N_k}$$

6.2 Results

As said above, we consider that the task of extracting the semantic concepts of an application consists in clustering its vocabulary words to build up various

classes which represent the concepts. We then test two clustering methods and we compare their performances with those achieved by the Bayesian network. The tested methods are the Kohonen maps and the K-means algorithm.

To evaluate and compare the concepts obtained by each clustering method we use the efficacity measure that we defined in the preceding section.

We also compare the three vector representations of words that we proposed in th e section 5. In the next table (table 3), we only give results obtained by the similarity vector representation and the combined matrix representation. In fact, context word representation can be only used by the Bayesian network and we can not determine its results with the Kohonen and the K-means methods. We just notice that when using these kind of representation with AutoClass we obtain many overlapping concepts some times not semantically representatives. The achieved efficacity values are 76% and 74.5% respectively with the applications MIAMM and IVOMOB.

		K-means	**Kohonen**	**AutoClass**
Similarity	*IVOMOB*	63.2%	75.3%	82.0%
vector representation	*MIAMM*	74.1%	77.1%	84.7%
Combined	*IVOMOB*	67.7%	77.1%	86.3%
matrix representation	*MIAMM*	76.4%	80.4%	89.3%

Table 3
Efficacities values obtained with the concepts found by each clustering method and each vector representation with MIAMM and IVOMOB applications.

Finally, we notice that using the combined matrix representation of words, the Bayesian network give the best results and finds a coherent lists of concepts which are perfectly related to the considered applications. In fact, using this method we achieve good results and we obtain 86.3% and 89.3% as efficacity rates respectively with the MIAMM and the IVOMOB applications.

7 Post-processing step

The last step consists in providing the SQL queries associated with the input textual requests. During this phase, we start by the request interpretation. In fact, if we have all the concepts which govern our application, we can affect to each request its suitable list of concepts by associating to each word its corresponding semantic class. Since our concepts do not overlap, labelling the requests does not present any risk of ambiguity.

Our goal is to provide at the end the corresponding SQL query which can answer the user request. In the literature this task is usually ignored or done in an ad-hoc manner. In these cases, very complicated inference mechanisms are implemented to obtain final SQL queries.

In our case, we choose to divide this tasks into two steps. In the first one, we try to automatically build up a kind of generic queries having the same structures than those written in SQL.

This generic query production constitutes the first step in the "Representation Converter" (see figure 3). The principal function of this module is to translate the conceptual representation of the sentence into a query representation where the concepts take the places of tables and conditions as shown in the figure 8. For example, if we find the concept "Date", we don't know the value of this date but, we can indicate in the generated query that there is a condition on the date. Therefore, the generated query can be written as :

select *Object*
from *Table_Bookmark*
where *Condition_Date*;

To achieve this task with an automatic manner, we define a concept hierarchy which is quite similar to the general structure of an SQL query. The figures 6 and 7 show how we can organize the MIAMM and IVOMOB concepts in a same hierarchy to fit the SQL queries structure. In these figures, we place the obtained concepts at the tree leaf level and we draw up the matching between each hierarchy part with an SQL query phrase. In this way, we can automatically build generic queries by replacing each concept by its corresponding SQL phrase.

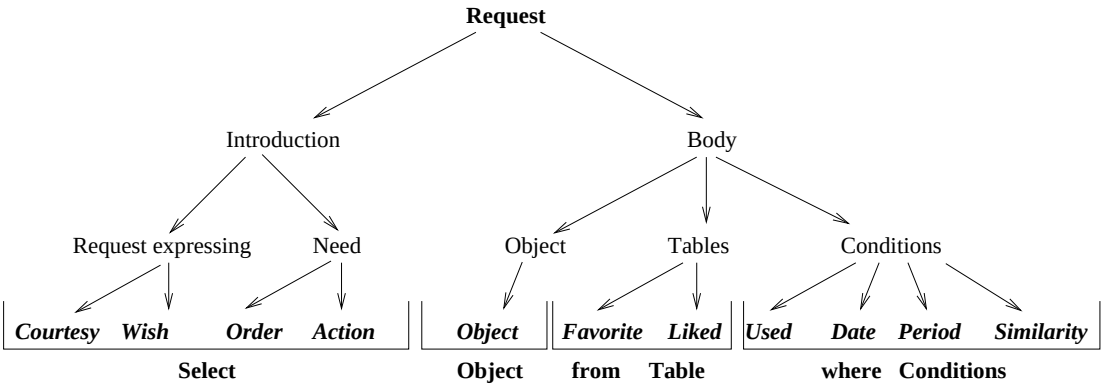

Fig. 6. Hierarchy of the MIAMM application concepts.

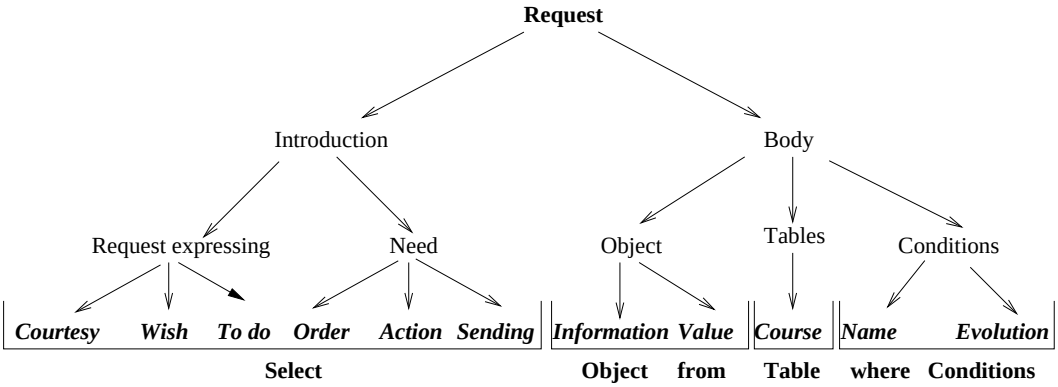

Fig. 7. Hierarchy of the IVOMOB application concepts.

7.2 SQL query generation

As a last phase, we set each concept, in the generic request, by its value deduced by going back to the initial sentence. This is done by a pattern matching mechanism which retrieves the proper object from the sentence and replaces it by the needed database attribute in the final SQL query. This module is a task specific one. In fact, such inference mechanism is very specific one and can not be automated because it has to respect the database structure and the initial request of the user. Obviously, this task can't be done in an automatic way. That's why we implement a simple inference engine to obtain at the end well formed SQL queries that we can carry out to extract the required data. This is also shown as a last step in the figure 8.

7.3 The speech recognition step

The last step of this work consists in integrating the understanding module in a real platform of automatic speech recognition. For that, we use the recognition output as an input for our understanding module. In our experiments, we use the automatic speech recognition system ESPERE (Engine for SPEech REcognition) developed in our team [7] and based on a Hidden Markov Model (HMM). We choose the following acoustic parameterization: 35 features, namely 11 static mel-cepstral coeffetions (C_0 was removed), 12 delta and 12 delta delta. The chosen HMM is 3 states multigaussien context independent. Two bigram language models have been trained on the MIAMM and the IVOMOB corpora.

To adapt the system to our experimental platform, we added some functionalities to use it in a real context. We also remove noise at the beginning and at the end of each sentence. By this way, we decrease the insertion rate of the recognition step.

8 Results and discussion

At the end, our oral understanding system is operational. As input, queries can be given as a signal or a text. The output of this palteform is a SQL query which fits perfectly the user's request. In other words, we consider that a system understands what has been uttered if the answer retrieved from the database via the SQL command corresponds to what the user asked for. For test we use 400 sentences pronounced by 4 different speakers for the MIAMM application and 200 sentences pronounced by 2 speakers for the IVOMOB application. It worths to be mentioned that the test sentences are very different from those used in the training step.

To illustrate the various stages followed in order to generate a good SQL query, an example is given in figure 8 concerning a bookmark consultation request (MIAMM application).

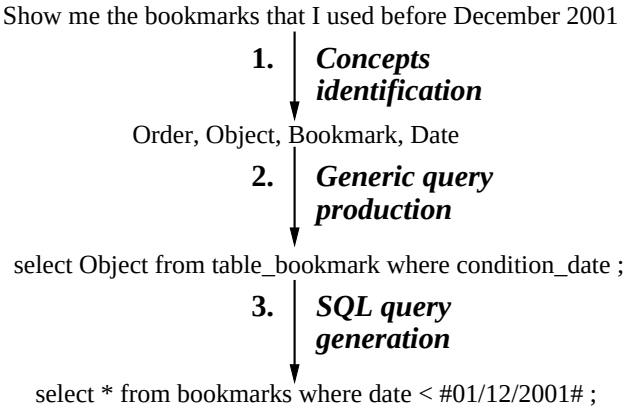

Fig. 8. Treatment sequence : from a natural language request to the corresponding SQL query.

The obtained results are very encouraging. In fact, with the MIAMM application we achieve a rate of 76.5% of concepts well detected and a rate of 78% of correct SQL requests. Although these results are quite high, the speech recognition system gives only a performance of 62%. When entry is text, the understanding performance reaches 92%. The same remarks can be done with the IVOMOB results where we achieve a rate of 79.1% of concepts well detected and a rate of 81% of correct SQL requests when the speech recognition system gives only a performance of 65.2%. With textual entries, the understanding performance reaches 92.5%. These results are resumed in the table 4.

The speech understanding system developed by Pieraccini [8] on ATIS corpus (Air-Travel Information Services) correctly answers 141 queries from a total test set of 195 sentences which is over 72% success rate. With a speech input of the same test set, the system gives more than 50% as understanding rate.

	MIAMM	IVOMOB
The test corpus size	400	200
Number of speackers	4	2
Recognition rate	62%	65.2%
Concept detection rate	76.5%	79.1%
Understanding rate	78%	81%
Understanding rate with text input	92%	92.5%

Table 4
Obtained results with the two applications MIAMM and IVOMOB within each understanding step.

In spite of the recognition errors the understanding speech system we developed yields a good result. So many works have to be done in order to improve the results and to obtain similar results to those with a text entry. Obviously, efforts have to be done on both speech recognition and understanding process.

9 Conclusion

In this article, we consider that the automatic speech understanding problem can be seen as an association problem between two different languages, the natural language and the concept language. Concepts are semantic entities gathering a set of words which share the same semantic properties and which express a given idea. We proposed a Bayesian network based method to automatically extract the concepts, as well as an approach for automatic sentence labelling and an engine for generating SQL queries corresponding to the user requests.

The concept extraction and the sentence labelling tasks are usually carried out manually. They constitute then, the most delicate and the most expensive phase in the understanding process. The method suggested in this article allows us to avoid the need for the human expertise and gives good results in terms of concepts viability and relevant retrieved SQL requests. At the end, we obtain 92% and 92.5% of correct SQL queries on the test corpora of the two treated applications. The proposed method can also be used for several other research fields that use the semantic classification : text categorization, information retrieval and data mining.

We also integrated our understanding module with a speech recognition system in order to carry out a complete interactive application. In spite of a speech recognition rates of 62% and 65.2%, we achieve final understanding

performances respectively of 78% and 81%. These results show that the understanding process we developed is robust with the speech recognition system errors.

We plan to extend the post-processing module to make it able to react vis-a-vis new key words not included in the concepts. It is then necessary that our model be able to add new words to the appropriate concepts within the exploitation step. We also plan to integrate our understanding module in a real plate-form of man-machine oral dialogue.

References

[1] C. Bousquet-Vernhettes and N. Vigouroux, "Context use to improve the speech understanding processing", Int. Workshop on Speech and Computer, Russia, 2001.

[2] P. Cheeseman and J. Stutz, "Bayesian classification (AutoClass): theory and results", "Advances in Knowledge Discovery and Data Mining", 1996.

[3] S. Jamoussi and K. Smaïli and J.P. Haton, "Neural network and information theory in speech understanding", Int. Workshop on Speech and Computer, Russia, 2002.

[4] F. Lefèvre and H. Bonneau-Maynard, "Issues in the development of a stochastic speech understanding system", Int. Conf. on Spoken Language Processing, Denver, 2002.

[5] R. Pieraccini and E. Levin and E. Vidal, "Learning how to understand language", Proc. 4rd European Conf. on Speech Communication and Technology, Germany, 1993.

[6] R. Rosenfeld, "Adaptive statistical language modelling: a maximum entropy approach", School of Computer Science Carnegie Mellon University, Pittsburgh, 1994.

[7] D. Fohr and O. Mella and C. Antoine, "The automatic speech recognition engine ESPERE : experiments on telephone speech", Int. Conf. on Spoken Language Processing, Beijing, 2000.

[8] R. Pieraccini and E. Tzoukermann and Z. Gorelov and J.-L. Gauvain and E. Levin and C.-H. Lee and J. G. Wilpon, "A speech understanding system based on statistical representation of semantics", Int. Conf. on Acoustics, Speech and Signal Processing, 1992.