Understanding process for speech recognition

Salma Jamoussi, Kamel Smaïli, Jean-Paul Haton, Kamel Sma¨ılisma¨ıli

Understanding process for speech recognition

Salma Jamoussi, Kamel Smaïli and Jean-Paul Haton

LORIA/INRIA-Lorraine
615 rue du Jardin Botanique, BP 101, F-54600 Villers-lès-Nancy, France
{jamoussi, smaili, jph}@loria.fr

Abstract

The automatic speech understanding problem could be considered 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 a new approach for the vector representation of words. We finish this paper by a description of the postprocessing step during which, we label our sentences and we generate the corresponding SQL queries. This step allows us to validate our speech understanding approach by obtaining good results. In fact, a rate of 92.5% of well formed SQL requests has been achieved on the test corpus.

1. Introduction

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 the following, we start by describing the general architecture of our understanding system based on the approach suggested in [5]. Then, we present 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 a new method for the vector representation of words, this representation will help the Bayesian network to build up efficient concepts. Finally, we will describe the last stage of our understanding process, in which we label the user requests and we generate the associated SQL queries.

2. Automatic speech understanding

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]. We also, adopt the same general architecture but we propose new techniques within each component. Moreover, we try to extract automatically our concepts in a preliminary step.

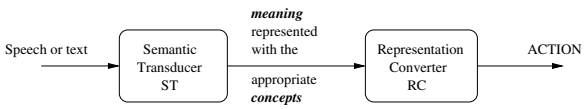

Figure 1: General architecture of a speech understanding system.

In our detailed system architecture shown in the figure 2, we consider 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.

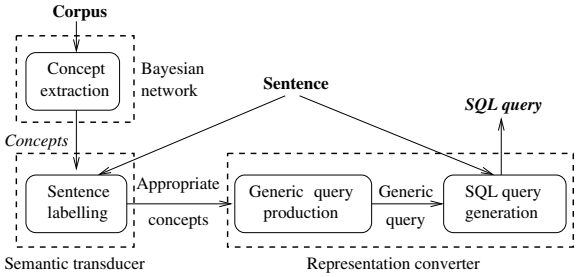

Figure 2: Our detailed understanding system architecture.

3. 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, the Oja and Sanger neural networks, the K-means method and some other methods based on the mutual information measure between words [3]. The obtained concepts were quite significant, but contained a lot of "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].

In the next paragraphs we present the Bayesian theory principle since the tool we used (AutoClass) is based on. We also detail some calculation stages allowing us to find the concepts related to our training corpus. We then expose a new approach to represent words in a 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. Obtained concepts and their hierarchy are given at the end of the current section.

3.1. The Bayesian network principle

AutoClass is a Bayesian network for unsupervised classification, 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 modelled 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 :

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

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.

3.2. Vector representation of words

One word can have several features but only few of them are relevant for a good semantic representation. In this stage of work we try to find an efficient word representation where we exploit the maximum number of information related to a given word. We decide to consider two kinds of information : the word context and the similarity of this word with all the other lexicon words.

In our case, the similarity of a word with others is expressed by the means of the average mutual information measure which tries to find contextual similarities between words. We then 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. The formula of the average mutual information between two words w_a and w_b is given by [6] :

$$\begin{aligned} I(w_a : w_b) = \quad & P(w_a, w_b) \log \tfrac{P(w_a|w_b)}{P(w_a)P(w_b)} + \\ & P(w_a, \overline{w}_b) \log \tfrac{P(w_a|\overline{w}_b)}{P(w_a)P(\overline{w}_b)} + \\ & P(\overline{w}_a, w_b) \log \tfrac{P(\overline{w}_a|w_b)}{P(\overline{w}_a)P(w_b)} + \\ & P(\overline{w}_a, \overline{w}_b) \log \tfrac{P(\overline{w}_a|\overline{w}_b)}{P(\overline{w}_a)P(\overline{w}_b)} \end{aligned} \quad (7)$$

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 w_a and $P(\overline{w}_a)$ is the probability of any other word except w_a.

To combine context and mutual information vector, we decide 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, the second column represents the average mutual information measures between the vocabulary words and the left context of the represented word. The third column is determined in 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:

$$IMM_j(C_l^i) = \frac{\sum_{w_l \in W_i \; left \; context} I(w_j : w_l) \times K_{wl}}{Nb\_occ} \quad (8)$$

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. K_{wl} is the number of times the word w_l is found in the left context of the word W_i and Nb_occ is the total number of occurrences of the word W_i in the corpus. The word W_i is thus represented by the matrix shown in the figure 3.

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

Figure 3: The matrix representation of the word W_i.

3.3. Experiments and results

In our work, we are interested in a bookmark consultation application, for that 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. These queries are provided to the understanding system in their textual form.

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 1: Some examples of queries in the MIAMM corpus.

Using the matrix representation of words like previously described, the Bayesian network finds a coherent list of 12 concepts which are perfectly related to the considered application. Some examples of these results are given in the table 2.

Concept	Group of words
Liked	Favourites, preferred, chosen, appreciated, adored, liked
Similarity	Similar, equivalent, resembling, identical, synonymous, near, close
Wish	Wish, wishes, can, wants, like, possible, would wish, would like
Order	Show, select, find, give, post, pass, seek, present, indicate, take

Table 2: Some examples of the obtained concepts.

Our goal is to provide at the end the corresponding SQL query which can answer the user request. The obtained concepts are very efficient and can help us to reach such a goal. In fact, in the figure 4, we can distinguish that their hierarchy is quite similar to the SQL query structure. In this figure, the concepts are represented at the tree leaf level and the matching between them and the SQL query structure is given at the bottom.

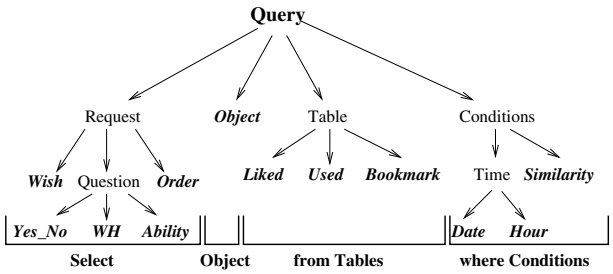

Figure 4: Obtained concept hierarchy.

4. Postprocessing 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.

4.1. Generic query representation

The generic query production constitutes the first step in the "Representation Converter" (see figure 2). 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 5. 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. This module is implemented as an inference engine which takes into account repetitions, lapses, multiple and implicit requests and others phenomena of the spontaneous speech.

4.2. 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 can include some more complicated functions namely when we need grammars to determine a date or an hour. At the end of this stage, we obtain a well formed SQL query that we can carry out to extract the required bookmarks, this is also shown as a last step in the figure 5.

4.3. Results

To illustrate the various stages followed in order to generate a good SQL query, an example is given in figure 5. The obtained results are very encouraging, in fact, in term of correct SQL queries, we obtain a rate of 100% with the training corpus and a rate of 80% with the test corpus which contains 400 sentences. To improve these results we associate to each base word all its morphological variants, for example, the words : *show, showing, shows*, are all represented by the word ***show***. Like this, we decreased the number of out of vocabulary words and the postprocessing inference engines became more efficient. Thanks to this method we achieve a rate of 92.5% of good results with the same test corpus.

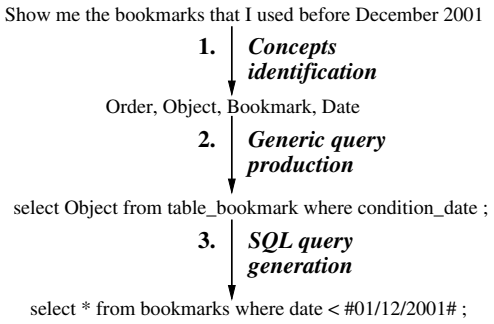

Figure 5: Treatment sequence : from a natural language request to the corresponding SQL query.

5. 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 a rate of 92.5% of correct SQL queries on the test corpus. The proposed method can also be used for several other research fields which use the semantic classification : text categorization, information retrieval and data mining.

We plan to extend the postprocessing 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.

6. 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.