A complete understanding speech system based on semantic concepts

Salma Jamoussi, Kamel Smaïli, Dominique Fohr, Jean-Paul Haton

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

Abstract

In this work, we present a complete speech understanding system based on our speech recognizer: ESPERE. The input signal is processed and the best sentence is then proposed to the understanding module. In our case, the understanding problem is considered as a matching process between two different languages. At the entry, the request expressed in natural language and at the output the corresponding SQL form. The SQL request is obtained after an intermediate step in which the entry is expressed in terms 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 classifier to automatically extract the underlined concepts. We also propose a new approach for vector representation of words. Then, we describe the postprocessing step during which, we label our sentences and we generate the corresponding SQL queries. We conclude our paper by describing the integration step of our understanding module in a complete platform of human-machine oral intercation.

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. To acheive this learning step we have first to manually annotate the training data which is a very heavy task (Bousquet-Vernhettes and Vigouroux, 2001; Lefèvre and Bonneau-Maynard, 2002; Pieraccini et al., 1993).

The data annotation step consists in segmenting the data into conceptual segments where each segment represents an underlined meaning (Bousquet-Vernhettes and Vigouroux, 2001). Within this step, we have to find first of all the list of concepts related our 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 manually this work constitutes a very expensive task. 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 giving the general architecture of our understanding system based on the approach suggested in (Pieraccini et al., 1993). Then, we present a new approach to automatically extract the semantic concepts of the considered application. For this, we use a Bayesian method for unsupervised classification, called AutoClass, we expose then a new method to represent 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, we generate the associated SQL queries and we integrate our understanding module in a speech recognition system.

2. Automatic speech understanding

A speech understanding system could be seen as a machine that produces an action as the result of an input sentence. Thus, the understanding problem could be considered as a translation process, it translates a signal into a special form that represents the meaning convoyed by the sentence. First, the recognized sentence is labelled by a list of conceptual entities (often called concepts). Second, this representation is used to interpret semantically the input query.

The used concepts constitute a useful intermediate representation which must be simple and representative. A concept is related to a given meaning, it is represented 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. Within the step of interpretation, we convert the obtained concepts to an action to be executed as a final response to the user’s query. In order to achieve such a goal, we have to generate a target formal command (e.g. an SQL query, a shell command, etc.).

Figure 1 illustrates the general architecture of such speech understanding system, this model was given in (Pieraccini et al., 1993) and it was included in several other works because of its effectiveness and its simplicity (Bousquet-Vernhettes and Vigouroux, 2001; Lefèvre and Bonneau-Maynard, 2002). We also, adopt the same general architecture but we propose new techniques within each component. Moreover, we extract automatically the appropriate concepts in a preliminary step.

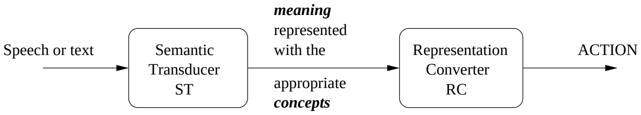

Figure 1: General architecture of a speech or text understanding system.

In our detailed system architecture shown in figure 2, we consider three principal components. The first one extracts the appropriate list of concepts by using a Bayesian classifier. 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. The role of the concept transducer is to assign to each word or phrase its concept. The main objective of the SQL converter is to produce an SQL request which corresponds to the speech or text entry. Two components are used, the first one is dedicated to produce a generic SQL query whereas the second generates the real SQL query corresponding to the initial entry.

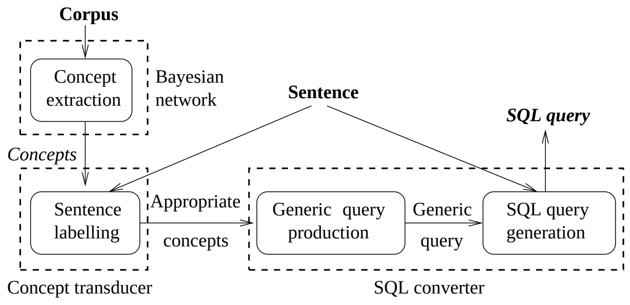

Figure 2: Our detailed understanding system architecture.

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. To build up the appropriate concepts, the corpus words have to be gathered in several classes.

To reach our goal we used an unsupervised classification technique. Among the unsupervised classification methods, we tried the Kohonen maps (Ritter and Kohonen, 1989), the Oja and Sanger neural networks (Memmi and Delichère, 2002), the K-means method (Ripley, 1996) and some other methods based on the mutual information measure between words (Jamoussi et al., 2002). 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 classifier technique because of its mathematical base and its powerful inference mechanism. Therefore, we use AutoClass a Bayesian network for unsupervised classification, it accepts real and discrete values as input. As result, it provides for each input, its membership probabilities to each retrieved class. AutoClass is based on the Bayes theorem and it supposes that there is a hidden multinomial variable which represents the different classes of the input data (Cheeseman and Stutz, 1996).

3.1. Vector representation of words

AutoClass has to classify the corpus words into several classes, each class will represent one semantic concept. We then propose to find an efficient word representation for which a maximum number of information related to a given word are exploited. Two kinds of information semantically significant are used : the word context and the similarity of this word with all the other lexicon words.

In our case, we use the average mutual information measure to compute 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 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 (Rosenfeld, 1994):

$$\begin{aligned} I(w_a : w_b) = \quad & P(w_a, w_b) \log \frac{P(w_a|w_b)}{P(w_a)P(w_b)} + \\ & P(w_a, \overline{w}_b) \log \frac{P(w_a|\overline{w}_b)}{P(w_a)P(\overline{w}_b)} + \\ & P(\overline{w}_a, w_b) \log \frac{P(\overline{w}_a|w_b)}{P(\overline{w}_a)P(w_b)} + \\ & P(\overline{w}_a, \overline{w}_b) \log \frac{P(\overline{w}_a|\overline{w}_b)}{P(\overline{w}_a)P(\overline{w}_b)} \end{aligned} \qquad (1)$$

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 a 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:

$$IMM_j(C_l^i) = \frac{\sum_{w_l \in L_{W_i}} I(w_j : w_l) \times K_{wl}}{\sum_{w_l \in L_{W_i}} K_{wl}} \qquad (2)$$

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. L_{W_i} is a set of words beloging to the left context of W_i. $I(w_j : w_l)$ represents the average mutual information between the word j of the lexicon and the word w_l belonging to the left context of W_i. K_{wl} is the occurrence of the word w_l found in the left context of W_i. The word W_i is thus represented by the matrix shown in the figure 3.

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

Figure 3: The matrix representation of the word W_i.

3.2. Experimental results

In our work, we are interested in bookmark request 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. Some examples of these queries are given in the table 1.

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.

Table 1: Some examples of queries in the MIAMM corpus.

Using the matrix representation of words as described previously, 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 SQL query which corresponds to the input request. The obtained meaningful concepts can help us to reach such a goal. Figure 4 illustrates the association between the concepts and the different fields of a SQL request. At the bottom of this hierarchy we have a generic request and the level above we have the concepts used for this application.

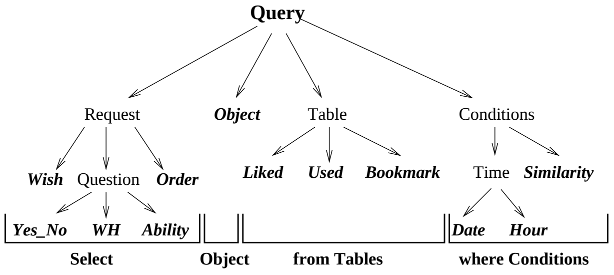

Figure 4: Obtained concept hierarchy.

4. Postprocessing step

The last step consists in providing the real SQL queries which can be used by a database system. During this phase, each word is assigned to its conceptual term. The obtained sentence is then used to generate the SQL request.

4.1. Generic query representation

The generic query production constitutes the first step in the "SQL 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 are replaced by tables and conditions as shown in figure 5. For instance, if the concept "Date" is encontered ; at this step no value is associated to it but we can guess that the query is concerned by a condition on date. Therefore, the generated query can be written as :

select *Object*
from *table_bookmark*
where *condition_date*;

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. At the end of this stage, we obtain a well formed SQL query that we can carry out to extract the required bookmarks (c.f. figure 5).

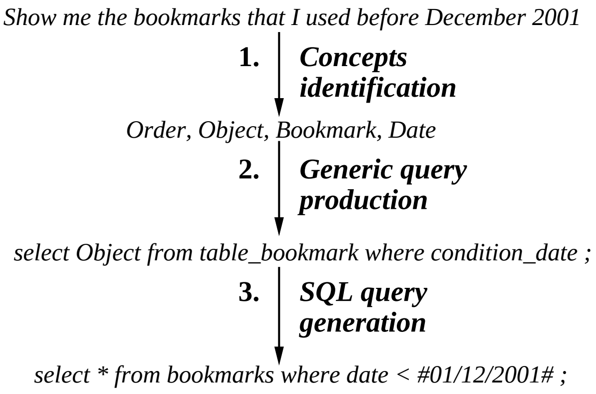
Show me the bookmarks that I used before December 2001
1. Concepts identification
Order, Object, Bookmark, Date
2. Generic query production
select Object from table_bookmark where condition_date ;
3. SQL query generation
select * from bookmarks where date < #01/12/2001# ;

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

5. A finalised speech understanding system

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 the next sections, we briefly present the recognition engine and its experimental conditions. Then, we discuss the results obtained before and after the recognition step.

5.1. Experimental conditions

For our experiments, we use the automatic speech recognition system ESPERE (Engine for SPEech REcognition) developed in our team (Fohr et al., 2000) based on a Hidden Markov Model (HMM). We choose the following acoustic parameterization: 35 features, namely 11 static mel-cepstral coeffections (C0 was removed), 12 delta and 12 delta delta. The chosen HMM is 3 states multigaussien context independant. A bigram language model has been trained on MIAMM corpus (950K words) and on a small vocabulary of 150 words.

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

Thus, 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 200 sentences pronounced by 4 different speakers. It worths to be mentioned that the test sentences are very different from those used in the training step.

5.2. Results and discussion

The obtained results are encouraging, with a speech entry 76.5% of concepts are well detected and 78% of correct SQL requests have been achieved. 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 speech understanding system developed by Pieraccini (Pieraccini et al., 1992) on ATIS coprus (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.

In spite of the recognition errors the understanding speech system we developped 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.

6. Conclusion

In this article, we consider that the automatic speech understanding problem can be seen as a matching problem between two different languages, the natural language and the SQL query. Concepts are used as an intermediate stage to reach this objective. They are considered as semantic entities which gather sets of words sharing the same semantic properties and expressing the same 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. We obtain a rate of 92% of correct SQL queries on the test corpus. 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 rate of 62%, we achieve a performance of 78% in terms of understanding. This result shows that the understanding process we developed is robust with the speech recognition system errors.

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 will be able to add new words to the appropriate concepts within the exploitation step.

7. References

Bousquet-Vernhettes, C. and N. Vigouroux, 2001. Context use to improve the speech understanding processing. In *International Workshop on Speech and Computer (SPECOM 2001)*. Moscow, Russia.

Cheeseman, P. and J. Stutz, 1996. Bayesian classification (autoclass): Theory and results. In *Advances in Knowledge Discovery and Data Mining*. American Association for Artificial Intelligence, Menlo Park, CA, USA, pages 153–180.

Fohr, D., O. Mella, and C. Antoine, 2000. The automatic speech recognition engine espere experiments on telephone speech. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP 2000)*. Beijing, China.

Jamoussi, S., K. Smali, and J.P. Haton, 2002. Neural network and information theory in speech understanding. In *International Workshop on Speech and Computer, (SPECOM 2002)*. St. Petersburg, Russia.

Lefèvre, F. and H. Bonneau-Maynard, 2002. Issues in the development of a stochastic speech understanding system. In *Proccedings of the International Conference on Spoken Language Processing (ICSLP 2002)*. Denver - Colorado, USA.

Memmi, D. and M. Delichère, 2002. Neural dimensionality reduction for document processing. In *European Symposium on Artificial Neural Networks (ESANN 2002)*. Bruges, Belgium.

Pieraccini, R., E. Levin, and E. Vidal, 1993. Learning how to understand language. In *Proceedings 4rd European Conference on Speech Communication and Technology (EuroSpeech 1993)*. Berlin, Germany.

Pieraccini, R., E. Tzoukermann, Z. Gorelov, J.-L. Gauvain, E. Levin, C.-H. Lee, and J. G. Wilpon, 1992. A speech understanding system based on statistical representation of semantics. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP 1992)*. San Francisco, CA, USA.

Ripley, B. D., 1996. *Pattern Recognition and Neural Networks*. Cambridge University Press, Cambridge, United Kingdom.

Ritter, H. and T. Kohonen, 1989. Self-organizing semantic maps. *Biological Cybernetics*, 61:241–254.

Rosenfeld, R., 1994. *Adaptive Statistical Language Modeling: A Maximum Entropy Approach*. Ph.D. thesis, School of Computer Science Carnegie Mellon University, Pittsburgh, PA 15213.