

## **Vers la compréhension automatique de la parole : extraction de concepts par réseaux bayésiens**

Salma Jamoussi, Kamel Smaili et Jean-Paul Haton

LORIA/INRIA-Lorraine

615 rue du Jardin Botanique, BP 101, F-54600 Villers-lès-Nancy, France

{jamoussi, smaili, jph}@loria.fr

### **Mots-clefs – Keywords**

Compréhension de la parole, concepts sémantiques, réseaux bayésiens, étiquetage sémantique, catégorisation automatique.

Speech understanding, semantic concepts, Bayesian networks, semantic labelling, automatic categorization.

### **Résumé - Abstract**

La compréhension automatique de la parole peut être considérée comme un problème d'association entre deux langages différents. En entrée, la requête exprimée en langage naturel et en sortie, juste avant l'étape d'interprétation, la même requête exprimée en terme de concepts. Un concept représente un sens bien déterminé. Il est défini par un ensemble de mots partageant les mêmes propriétés sémantiques. Dans cet article, nous proposons une méthode à base de réseau bayésien pour l'extraction automatique des concepts ainsi que trois approches différentes pour la représentation vectorielle des mots. Ces représentations aident un réseau bayésien à regrouper les mots, construisant ainsi la liste adéquate des concepts à partir d'un corpus d'apprentissage. Nous concluons cet article par la description d'une étape de post-traitement au cours de laquelle, nous étiquetons nos requêtes et nous générons les commandes SQL appropriées validant ainsi, notre approche de compréhension.

The automatic speech understanding can be considered as association problem between two different languages. At the entry, the request expressed in natural language and at the end, just before the stage of interpretation, the same request is expressed in term of concepts. One concept represents given meaning, it is defined by the 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 three different approaches for the vector representation of words. These representations help Bayesian network to build the adequate list of concepts for the considered application. We finish this paper by description of the post-processing step during which, we label our sentences and we generate the corresponding SQL queries. This step allows us to validate our speech understanding approach.

## 1 Introduction

Dans la littérature, plusieurs méthodes de compréhension de la parole ont été proposées. La plupart de ces méthodes se fondent sur des approches stochastiques de décodage conceptuel qui permettent d'approcher la compréhension automatique, réduisant ainsi le recours à l'expertise humaine. Cependant, ces méthodes nécessitent une étape d'apprentissage supervisé, ce qui signifie qu'il y a une étape antérieure d'annotation manuelle du corpus d'apprentissage (Maynard & Lefèvre, 2002; Bousquet-Vernhettes & Vigouroux, 2001; Pieraccini *et al.*, 1993).

Dans de telles approches fondées sur le décodage conceptuel, l'étape d'annotation consiste à segmenter les données d'apprentissage en des segments conceptuels représentant chacun un sens bien déterminé (Bousquet-Vernhettes & Vigouroux, 2001). Il s'agit donc de trouver tout d'abord les différents concepts relatifs au corpus, de segmenter ensuite les phrases de ce corpus, de les étiqueter en utilisant les concepts trouvés et de procéder enfin à l'apprentissage automatique. Faire tout ce travail d'une façon manuelle constitue sans doute une phase fastidieuse et coûteuse. De plus, l'extraction manuelle est sujette à la subjectivité et aux erreurs humaines. Automatiser cette tâche permettra donc de réduire ou d'annuler l'intervention humaine et surtout de pouvoir réutiliser ce même procédé lorsqu'on change de contexte (Siu & Meng, 1999).

Dans cet article, nous commençons par décrire l'architecture générale de notre système de compréhension de la parole, basée sur l'approche proposée dans (Pieraccini *et al.*, 1993). Ensuite, nous présentons une nouvelle approche pour extraire automatiquement les concepts sémantiques. Pour ce faire, nous utilisons un réseau bayésien pour la classification non supervisée, appelé AutoClass. Puis, nous exposons trois méthodes différentes pour la représentation vectorielle des mots afin de les regrouper pour former des concepts. Enfin, nous abordons la dernière étape du processus de compréhension, au cours de laquelle nous étiquetons les requêtes et nous générons les commandes SQL associées.

## 2 La compréhension automatique de la parole

Le problème de compréhension de la parole peut être vu comme un problème de mise en correspondance entre une chaîne de mots en entrée et une suite de mots dans un langage plus restreint véhiculant les idées principales d'une phrase. Il s'agit, dans un premier temps, d'associer les mots de la phrase en entrée du système à des messages dans un langage sémantique intermédiaire (souvent appelés concepts). Dans un second temps, afin de satisfaire la requête émise en entrée, on traduit les concepts obtenus en actions ou réponses et on parle dans ce cas de l'étape d'interprétation de la phrase.

L'entrée du système peut être donnée sous forme textuelle ou sous forme d'un signal de parole, sa sortie exprimée en tant qu'actions ou commandes n'est qu'une conversion d'une liste de concepts donnée par un module intermédiaire de traduction sémantique et fournissant le sens littéral de la phrase. Un concept est une classe de mots traitant d'un même sujet et partageant des propriétés communes. Par exemple, les mots *hôtel*, *chambre*, *auberge* et *studio* peuvent tous correspondre au concept "*hébergement*" dans une application touristique. Dans (Pieraccini *et al.*, 1993), les auteurs définissent un modèle général pour la compréhension automatique de la parole qui, en raison de sa simplicité et de son efficacité, a été repris dans plusieurs autres travaux (Maynard & Lefèvre, 2002; Bousquet-Vernhettes & Vigouroux, 2001). Nous avons

adopté la même architecture générale mais nous proposons des techniques différentes au sein de chacune de ses composantes. La figure 1 illustre l'architecture générale d'un tel système de compréhension de la parole.



Figure 1: Architecture générale d'un système de compréhension automatique de la parole.

Dans notre travail, nous nous intéressons à une application de consultation de pages "Favoris" (Bookmarks en anglais). Pour ce faire, nous utilisons un corpus du projet européen MIAMM dont l'objectif est de construire une plate-forme de dialogue oral multimodale. Le corpus contient 71287 requêtes différentes exprimées en langue française. Chaque requête exprime une manière particulière d'interroger la base. Des exemples de ces requêtes sont donnés dans la table 1. Ces requêtes sont fournies au système de compréhension sous leur forme textuelle. Notre but est de fournir à la fin les requêtes SQL correspondantes qui, en les exécutant, répondront aux demandes des utilisateurs.

Montre-moi le contenu de mes favoris. Je voudrais savoir si tu peux me prendre le contenu que j'aime. Est-ce que tu veux me sélectionner les titres que je préfère. Est-il possible que tu me passes le premier de mes favoris. Te serait-il possible de m'indiquer quelque chose de pareil. Tu peux faire voir uniquement décembre 2001. Il faut que tu me présentes la liste que j'ai utilisée tôt ce matin. Je te demande de me passer les chansons que j'ai écoutées ce matin. Je souhaiterais que tu me montres les disques que j'ai regardés dans la matinée. Je veux voir le deuxième que j'ai regardé dans la matinée.
---

Table 1: Quelques exemples de requêtes du corpus MIAMM.

### 3 Extraction des concepts : méthodes et résultats

Au cours de cette étape, nous cherchons à identifier les concepts sémantiques liés à notre application. La détermination manuelle de ces concepts est une tâche très lourde. Il nous faut donc trouver une méthode automatique qui, pourrait ne pas donner des résultats aussi performants que ceux obtenus par la méthode manuelle, mais qui, en contre partie, permet une automatisation complète du processus de compréhension.

Partant du principe d'automatisation de cette tâche de catégorisation, nous avons opté pour des méthodes de classification non supervisée. Notre but final étant de trouver des concepts cohérents de l'application, le meilleur moyen d'y parvenir est de regrouper les mots en fonction de leurs propriétés sémantiques. La méthode à utiliser va donc regrouper les mots du corpus en

différentes classes, construisant ainsi les concepts de l'application. Dans ce cas, il nous reste qu'à affecter un nom de concept approprié à chaque groupe de mots. Parmi les méthodes de classification non supervisée, nous avons implanté les cartes de Kohonen, les réseaux de neurones de Oja et Sanger, la méthode des K-means et quelques méthodes fondées sur la mesure d'information mutuelle moyenne entre les mots (Jamoussi *et al.*, 2002). Les concepts obtenus par ces méthodes étaient bien significatifs, mais contenaient du "bruit". Autrement dit, certains mots n'avaient pas leurs places dans le sens exprimé par ces concepts. Pour remédier à ce problème, nous avons exploré d'autres méthodes et avons adopté les réseaux bayésiens en raison de leur fondement mathématique fort et le mécanisme d'inférence puissant sous-jacent (Cheeseman & Stutz, 1996).

Dans la suite, nous présentons le principe de la théorie bayésienne sur laquelle se base l'outil que nous avons utilisé (AutoClass) et nous détaillons quelques étapes de calcul permettant de trouver les concepts correspondants au corpus d'apprentissage utilisé. Nous exposons aussi trois approches différentes pour la représentation vectorielle des mots. Cette représentation, qui doit être sémantiquement significative, constitue une étape clé au sein du système de compréhension puisqu'elle constitue l'entrée du réseau bayésien qui va décider des groupements des mots formant les concepts.

### 3.1 Principe du réseau bayésien "AutoClass"

AutoClass est un réseau bayésien pour la classification non supervisée qui accepte en entrée des valeurs réelles, mais aussi des valeurs non numériques comme des mots, des caractères etc. En résultat, il fournit des probabilités d'appartenance des éléments en entrée, aux classes trouvées. Il suppose l'existence d'une variable multinomiale cachée qui peut représenter les différentes classes auxquelles appartiennent les éléments en entrée. AutoClass est basé sur le théorème de Bayes exprimé par :

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

Dans notre cas,  $D$  représente les données et donc les mots à classer et  $H$  une hypothèse concernant le nombre de classes et leurs descriptions en terme de probabilités. Avec AutoClass, on cherche à maximiser la probabilité  $p(H|D)$ , c'est à dire que sachant  $D$ , les mots du corpus, on doit sélectionner  $H$ , l'ensemble des concepts, qui maximise cette probabilité.

Dans notre réseau bayésien représenté par la figure 2, un mot  $x_i$  est donné sous la forme d'un vecteur à  $K$  valeurs d'attributs,  $x_{ik}$ ,  $k \in \{1...K\}$ . Un concept  $C_j$  est décrit, lui aussi, par  $K$  attributs, chacun est modélisé par une distribution gaussienne normale.  $\vec{\theta}_{jk}$  est un vecteur paramètre décrivant le  $k$ ème attribut du  $j$ ème concept  $C_j$  et il contient deux éléments, la moyenne  $\mu_{jk}$  de la distribution considérée et son écart-type  $\sigma_{jk}$ . Pour l'ensemble du concept, ce vecteur est noté  $\vec{\theta}_j$  et il contient les  $\vec{\theta}_{jk}$  de tous ses attributs. La probabilité qu'un mot  $x_i$  appartienne au concept  $C_j$ , appelée *la probabilité de classe* et est notée  $\pi_j$  constitue aussi un paramètre descriptif du concept  $C_j$ .

Ainsi nous avons défini nos paramètres de travail, les données  $D$  sont bien les mots, représentés par le vecteur  $\vec{x}$  à  $I$  éléments englobant tous les  $x_i$ . L'hypothèse  $H$  correspondant à la description des concepts est représentée par trois éléments, le nombre de concepts  $J$  et les deux vecteurs  $\vec{\pi}$  et  $\vec{\theta}$  qui contiennent respectivement les  $\pi_j$  et les  $\vec{\theta}_j$  de tous les concepts. AutoClass divise le problème d'identification des concepts en deux parties : la détermination des

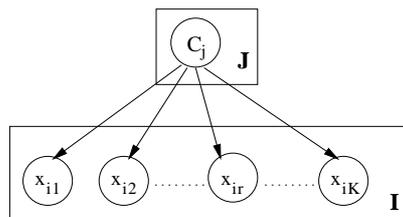


Figure 2: La structure générale du réseau bayésien AutoClass.

paramètres de classification ( $\vec{\pi}$  et  $\vec{\theta}$ ) pour un nombre donné de concepts et la détermination du nombre de concepts  $J$ . Ce dernier problème nécessite plusieurs approximations (pour les détails, voir (Cheeseman & Stutz, 1996)). Dans ce qui suit,  $H$  représente seulement les vecteurs  $\vec{\pi}$  et  $\vec{\theta}$ . En remplaçant, dans l'équation 1,  $D$  et  $H$  par leurs valeurs, on obtient :

$$p(\vec{\theta}, \vec{\pi} | \vec{x}) = \frac{p(\vec{\theta}, \vec{\pi}) p(\vec{x} | \vec{\theta}, \vec{\pi})}{p(\vec{x})} \quad (2)$$

où  $p(\vec{\theta}, \vec{\pi})$  est la probabilité *a priori* des paramètres de classification, son calcul est bien décrit dans (Cheeseman & Stutz, 1996). La probabilité *a priori* des mots,  $p(\vec{x})$  peut être calculée directement. Elle est considérée simplement comme une constante de normalisation. Dans ce qui suit, on s'intéresse au calcul de la probabilité  $p(\vec{x} | \vec{\theta}, \vec{\pi})$  qui représente la fonction de vraisemblance des données.

On sait que  $\vec{x}$  est un vecteur représentant tous les mots du corpus d'apprentissage, la vraisemblance de ce vecteur est donc calculée comme étant le produit des probabilités de tous les mots séparément comme le montre l'équation suivante :

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

$p(x_i | \vec{\theta}, \vec{\pi})$  est la probabilité d'observer le mot  $x_i$  indépendamment du concept auquel il appartient. Elle est donnée par la somme des probabilités que ce mot appartienne à chaque concept séparément, pondérée par les probabilités des classes comme indiqué par l'équation suivante :

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

Puisque le mot  $x_i$  est décrit par un ensemble de  $K$  attributs, avec la supposition, un peu forte, que ces attributs sont indépendants, la probabilité  $p(x_i | x_i \in C_j, \vec{\theta}_j)$  peut s'écrire donc, sous la forme suivante :

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

AutoClass modélise les attributs à valeurs réelles par une distribution gaussienne normale représentée par le vecteur  $\vec{\theta}_{jk}$  qui contient les deux paramètres  $\mu_{jk}$  et  $\sigma_{jk}$ . Dans ce cas,  $p(x_{ik} | x_i \in C_j, \vec{\theta}_{jk})$  qui correspond à la distribution de classe peut s'écrire sous la forme suivante :

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

Une fois, cette distribution de classe déterminée, il nous reste à chercher l'ensemble des paramètres de concepts qui maximisent la probabilité de départ  $p(\vec{\theta}, \vec{\pi} | \vec{x})$  et trouver ainsi l'ensemble des concepts optimaux relatifs à nos données.

## 3.2 Représentation vectorielle des mots

### 3.2.1 Contexte des mots

Un mot peut avoir plusieurs caractéristiques possibles, mais rares sont celles qui peuvent lui donner une représentation sémantique complète. Dans une première étape, nous avons décidé d'associer à chaque mot ses différents contextes d'utilisation en émettant l'hypothèse que si deux mots ont les mêmes contextes alors ils sont sémantiquement proches. Dans cette approche, un mot sera donc représenté par un vecteur à  $2 \times N$  éléments contenant les  $N$  mots de son contexte gauche et les  $N$  mots de son contexte droit.

La figure 3 présente un exemple de la représentation contextuelle des mots. Nous disposons d'une phrase contenant 4 mots ; en fixant  $N = 2$ , nous représentons chaque mot  $W_i$  par les deux mots de ses contextes gauche et droit.

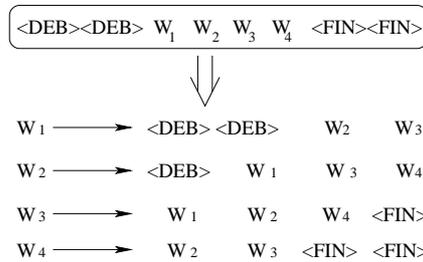


Figure 3: Un exemple de représentation de mots par leurs contextes ( $N = 2$ ).

La valeur de  $N$  a été fixée à 2 car les requêtes sont généralement courtes et concises. Par conséquent, les contextes sémantiquement significatifs sont constitués de peu de mots. Il faut rappeler aussi que les mots outils de la langue comme : *je, alors, le, mais, etc.*, ne sont pas considérés.

Les classes que nous avons obtenues par cette méthode représentent bien des concepts sémantiques, mais ont l'inconvénient de se chevaucher. De plus, nous avons eu des difficultés à contrôler le nombre de concepts. Quelques exemples de ces concepts sont donnés dans la table 2.

### 3.2.2 Similarité entre mots

Pour trouver des concepts plus homogènes, nous avons changé complètement la structure vectorielle de chaque mot. Nous avons utilisé la mesure de l'information mutuelle moyenne qui permet de trouver des similarités contextuelles entre mots.

Dans cette approche, nous associons à chaque mot un vecteur à  $M$  éléments, où  $M$  est la taille du lexique. L'élément numéro  $j$  de ce vecteur représente la valeur de l'information mutuelle moyenne entre le mot numéro  $j$  du lexique et le mot à représenter comme indiqué dans la suite:

$$W_i = [I(w_1 : w_i), I(w_2 : w_i), \dots, I(w_j : w_i), \dots, I(w_M : w_i)] \quad (7)$$

Ce vecteur exprime le degré de similarité du mot en question avec tous les autres mots du corpus. La formule de l'information mutuelle moyenne (Rosenfeld, 1994) entre deux mots  $w_a$

Concept	Groupe de mots
<b>Favoris_1</b>	Préfééré, favoris, préférés, choisi, apprécié, aimé, adoré
<b>Favoris_2</b>	Favoris, préférés, écouté, vu, utilisé, regardé
<b>Favoris_3</b>	Favoris, préférés, choisi, apprécié, aimé, adoré, préféré, similaire, semblable, pareil, équivalent, ressemblant, synonyme, proche, identique, rapproché
<b>Demande_1</b>	Possible, demande, veut, voudrais, aimerais, souhaite, souhaiterais, faut, désire, désirerais
<b>Demande_2</b>	Peux, pourrais, veux, voudrais, possible, cherche, demande, aimerais, souhaite, souhaiterais, faut, désire, désirerais, fais
<b>Ordre</b>	Montrer, indiquer, sélectionner, trouver, donner, afficher, voir, presser, prendre, passer, chercher

Table 2: Exemples de concepts obtenus avec la représentation basée sur le contexte des mots.

et  $w_b$  est donnée par :

$$I(w_a : w_b) = P(w_a, w_b) \log \frac{P(w_a|w_b)}{P(w_a)P(w_b)} + P(w_a, \bar{w}_b) \log \frac{P(w_a|\bar{w}_b)}{P(w_a)P(\bar{w}_b)} + P(\bar{w}_a, w_b) \log \frac{P(\bar{w}_a|w_b)}{P(\bar{w}_a)P(w_b)} + P(\bar{w}_a, \bar{w}_b) \log \frac{P(\bar{w}_a|\bar{w}_b)}{P(\bar{w}_a)P(\bar{w}_b)} \quad (8)$$

où  $P(w_a, w_b)$  est la probabilité de trouver les deux mots  $w_a$  et  $w_b$  dans la même phrase,  $P(w_a | w_b)$  est la probabilité de trouver le mot  $w_a$  sachant qu'on a déjà rencontré le mot  $w_b$ ,  $P(w_a)$  est la probabilité de trouver le mot  $w_a$  et  $P(\bar{w}_a)$  est la probabilité de ne pas avoir rencontré le mot  $w_a$  etc.

En représentant ainsi chaque mot du corpus, et en utilisant le réseau bayésien, nous avons obtenu des classes sémantiques homogènes, une classe étant constituée de mots partageant les mêmes propriétés sémantiques. Le nombre de ces classes est très cohérent avec notre application. Cette représentation nous a permis aussi de résoudre le problème des chevauchements entre concepts. Dans la table 3, nous donnons quelques exemples des concepts ainsi construits, où l'on remarque bien qu'il n'y a plus de chevauchement, mais qu'il existe encore quelques imperfections comme dans le cas des concepts *Demande\_1*, *Demande\_2* et *Demande\_3* qui auraient dû être regroupés ensemble.

Concept	Groupe de mots
<b>Favoris</b>	Favoris, préférés, choisi, apprécié, adoré, préféré
<b>Mode</b>	Écouté, vu, regardé, utilisé
<b>Similarité</b>	Dernier, similaire, semblable, pareil, équivalent, ressemblant, synonyme, proche, identique, rapproché
<b>Demande_1</b>	Pourrais, veux, voudrais
<b>Demande_2</b>	Possible, aimerais, souhaiterais
<b>Demande_3</b>	Souhaite, faut, désire, désirerais
<b>Ordre</b>	Montrer, indiquer, sélectionner, trouver, donner, afficher, présenter, prendre, passer, chercher

Table 3: Quelques exemples de concepts obtenus en utilisant la représentation basée sur l'information mutuelle moyenne entre les mots.

### 3.2.3 Combinaison : contexte et similarité

Nous avons combiné les deux approches précédentes en vue d'améliorer les résultats. En effet, la première approche travaillant au niveau occurrence, exploite directement les informations liées au contexte d'utilisation des mots, tandis que la deuxième, utilise une mesure pour chercher des similarités entre deux mots. On peut aisément comprendre que les informations utilisées au niveau de ces deux méthodes sont différentes et complémentaires.

Combiner ces deux méthodes, consiste donc à représenter chaque mot par une matrice d'information mutuelle moyenne à dimension  $M \times 3$ . La première colonne correspond au vecteur d'information mutuelle moyenne précédent (voir section 3.2.2), la deuxième colonne représente l'information mutuelle moyenne entre un mot quelconque du vocabulaire et le contexte gauche du mot à représenter. Idem pour la troisième colonne mais concernant le contexte droit.

La  $j$ ème valeur de la deuxième colonne est la moyenne pondérée des informations mutuelles moyennes entre le  $j$ ème mot du vocabulaire et le vecteur constituant le contexte gauche du mot  $W_i$  en question. Elle est calculée comme suit :

$$IMM_j(C_g^i) = \frac{\sum_{w_g \in \text{contexte gauche de } W_i} I(w_j : w_g) \times K_{w_g}}{Nb\_occ} \quad (9)$$

Où  $IMM_j(C_g)$  représente l'information mutuelle moyenne entre le mot  $w_j$  du lexique et le contexte gauche du mot  $W_i$ .  $I(w_j : w_g)$  représente l'information mutuelle moyenne entre le mot numéro  $j$  du lexique et le mot  $w_g$  qui appartient au contexte gauche du mot  $W_i$ .  $K_{w_g}$  est le nombre de fois où le mot  $w_g$  est trouvé comme contexte gauche du mot  $W_i$  et  $Nb\_occ$  est le nombre total d'occurrence du mot  $W_i$  dans le corpus. Le mot  $W_i$  sera donc représenté par une matrice comme le montre la figure 4.

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

Figure 4: Représentation du mot  $W_i$  par la méthode combinée.

La matrice utilisée pour représenter un mot du corpus exploite un maximum d'informations sur ce mot. Elle considère son contexte ainsi que sa similarité avec tous les autres mots du lexique. Une telle représentation des mots a pu aidé le réseau bayésien dans sa tâche de classification et nous a permis d'améliorer considérablement nos résultats. Nous obtenons alors une liste de concepts bien cohérente qu'on va utiliser dans la suite de nos traitements. Des exemples de ces résultats sont donnés au niveau de la table 4.

## 4 Étiquetage et post-traitement

La dernière étape consiste à fournir les commandes SQL associées aux requêtes textuelles émises en entrée. C'est au cours de cette phase que nous entamons l'étape d'interprétation

Concept	Groupe de mots
<b>Favoris</b>	Favoris, préférés, choisi, apprécié, adoré, aimé
<b>Mode</b>	Écouté, vu, regardé, utilisé
<b>Similarité</b>	Similaire, semblable, pareil, équivalent, ressemblant, synonyme, proche, identique, rapproché
<b>Demande</b>	Souhaite, faut, désire, désirerais, peux, pourrais, veux, voudrais, possible, aimerais, souhaiterais
<b>Ordre</b>	Montrer, indiquer, sélectionner, trouver, donner, afficher, présenter, prendre, passer, chercher

Table 4: Quelques exemples de concepts obtenus en utilisant l’approche combinée.

des requêtes. En effet, disposant de l’ensemble des concepts qui régissent notre application, nous pouvons attribuer à chaque requête les concepts appropriés. Il s’agit de l’étape de “Traduction sémantique”, la première composante de l’architecture générale de notre système de compréhension (voir la figure 1). Pour ce faire, il nous suffit d’étiqueter nos phrases en associant à chaque mot dans la phrase sa classe sémantique correspondante. Puisque nos concepts ne se chevauchent pas entre eux, étiqueter ainsi les requêtes ne présente aucun risque d’ambiguïté.

Ensuite, nous pouvons passer à la deuxième composante de notre modèle, le “Convertisseur de représentation”, où il s’agit de convertir les concepts trouvés en commandes SQL permettant d’extraire l’information requise de notre base de données. Pour ce faire, nous avons réalisé un moteur d’inférence qui à chaque concept, fait correspondre une ou plusieurs sous-requêtes génériques. Dans une requête SQL générique, les concepts interviennent au niveau des conditions. Ainsi, par exemple, si nous trouvons le concept “Date”, nous ne connaissons pas la valeur de cette date mais, nous pouvons indiquer dans la requête générée qu’il y a une condition sur la date. Ce moteur d’inférence prend en compte bien sûr les répétitions, les oublis, les demandes multiples et implicites ainsi d’autres phénomènes de la parole spontanée. Dans la phase suivante, nousinstancions chaque concept, dans la requête générique obtenue, par sa valeur qui est déduite en revenant à la phrase initiale. Ainsi, nous obtenons une vraie commande SQL que nous pouvons exécuter pour extraire les pages recherchées. Dans la figure 5, nous donnons un exemple illustrant les différentes étapes suivies afin d’aboutir à une commande SQL finalisée. Les résultats obtenus sont encourageants, en effet, en terme de requêtes SQL correctes, nous obtenons un taux de 100% avec le corpus d’apprentissage et un taux de 92.5% avec un corpus de test contenant 400 phrases différentes.

## 5 Conclusion

Dans cet article, nous sommes partis du principe que le problème de la compréhension automatique est un problème d’association entre deux langages différents, le langage naturel et le langage des concepts. Les concepts sont des entités sémantiques regroupant un ensemble de mots qui partagent les mêmes propriétés sémantiques et qui expriment une certaine idée. Nous avons proposé trois méthodes différentes pour l’extraction automatique des concepts, ainsi qu’une approche d’étiquetage et de génération automatique des requêtes SQL correspondantes aux demandes des utilisateurs.

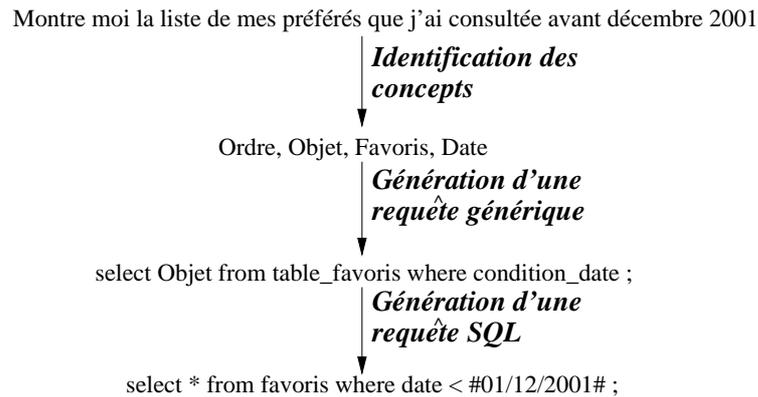


Figure 5: Chaîne de traitement appliquée à une requête en langage naturel.

Les tâches d'extraction de concepts et d'étiquetage sont d'habitude réalisées manuellement. Elles constituent la phase la plus délicate et la plus coûteuse dans le processus de compréhension. Les méthodes proposées dans cet article ont permis d'éviter ce recours à l'expertise humaine et peuvent servir à plusieurs autres domaines qui touchent à la classification sémantique, comme les domaines de catégorisation de texte, d'extraction d'information et de fouille de données. La méthode combinée a donné les meilleurs résultats grâce aux informations qu'elle a su exploitées pour représenter au mieux un mot du corpus. Pour notre application de consultation de pages favorites, les concepts qu'elle a trouvés sont très satisfaisants. Il nous ont permis ensuite, de mener à bien l'étape d'étiquetage sans rencontrer des difficultés notables.

Nous envisageons d'étendre le module de post-traitement de façon à ce qu'il puisse réagir face à de nouveaux mots clés non pris en compte par les concepts. Pour ce faire, il faut adapter notre modèle à la phase d'exploitation pour que nous puissions ajouter des mots aux concepts. Nous souhaitons aussi intégrer notre module de compréhension dans un système de reconnaissance automatique de la parole afin de réaliser une application interactive exploitable.

## Références

- BOUSQUET-VERNHETTES C. & VIGOUROUX N. (2001). Context use to improve the speech understanding processing. In *International Workshop on Speech and Computer, SPECOM'01*, Moscow.
- CHEESEMAN P. & STUTZ J. (1996). Bayesian classification (autoclass): Theory and results. In *Advances in Knowledge Discovery and Data Mining*. U. Fayyad, G. Shapiro, P. Smyth, R. Uthurusamy.
- JAMOUCSI S., SMAÏLI K. & HATON J. (2002). Neural network and information theory in speech understanding. In *International Workshop on Speech and Computer, SPECOM'02*, St. Petersburg.
- MAYNARD H. & LEFÈVRE F. (2002). Apprentissage d'un module stochastique de compréhension de la parole. In *24èmes Journées d'Étude sur la parole*, Nancy.
- PIERACCINI R., LEVIN E. & VIDAL E. (1993). Learning how to understand language. In *Proceedings of the 4rd European Conference on Speech Communication and Technology*, Berlin.
- ROSENFELD R. (1994). *Adaptive Statistical Language Modeling: A Maximum Entropy Approach*. PhD thesis, School of Computer Science Carnegie Mellon University, Pittsburgh, PA 15213.
- SIU K.-C. & MENG H. M. (1999). Semi-automatic acquisition of domain-specific semantic structures. In *Proceedings of the 6th European Conference on Speech Communication and Technology*, Budapest.