Un système de compréhension automatique de la parole pour l'interrogation orale d'une base de données de bourse

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

LORIA/INRIA-Lorraine
615 rue du Jardin Botanique, BP 101, F-54600 Villers-lès-Nancy, France
Tél : + 33 3 83 59 30 00 - Fax : + 33 3 83 27 83 19
Mél : {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

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, nécessitant une étape antérieure d'annotation manuelle du corpus d'apprentissage [1, 2, 5]. Cette étape d'annotation consiste à segmenter les données d'apprentissage en des segments conceptuels représentant chacun un sens bien déterminé [5]. 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. Réaliser ce travail d'une façon manuelle constitue une opération fastidieuse et coûteuse. De plus, la définition manuelle des concepts est sujette à subjectivité et aux erreurs humaines. Automatiser cette tâche permettrait de réduire ou d'annuler l'intervention humaine et de simplifier un changement d'application.

Dans cet article, nous commençons par décrire l'architecture de notre système de compréhension de la parole, fondée sur l'approche proposée dans [1]. Ensuite, nous présentons une nouvelle approche pour déterminer automatiquement des concepts sémantiques. Pour ce faire, nous utilisons un réseau bayésien pour la classification non supervisée, appelé AutoClass. Puis, nous présentons une nouvelle méthode pour la représentation vectorielle des mots. Enfin, nous abordons la dernière étape du processus de compréhension, au cours de laquelle nous étiquetons les requêtes, générons les commandes SQL associées et intégrons le module de compréhension dans une plate-forme réelle de compréhesnsion orale homme-machine.

2. LA COMPRÉHENSION AUTOMATIQUE DE LA PAROLE

Un système de compréhension de la parole peut être considéré comme une machine qui traduit une chaîne de mots en une ou plusieurs actions. 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 répondre à la requête d'entrée, on traduit les concepts obtenus en actions ou réponses au cours d'une 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 terme d'actions ou commandes n'est qu'une conversion d'une liste donnée de concepts par un module intermédiaire de traduction sémantique. 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 [1], 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 travaux [2, 5]. Nous avons adopté la même architecture générale (voir figure 1), mais nous proposons des techniques différentes au sein de chacune de ses composantes. De plus, nous proposons de construire la liste des concepts d'une façon automatique.

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

L'architecture du système de compréhension que nous proposons est composée de trois composantes principales. Le but du premier module est d'extraire à partir du corpus

textuel la liste des concepts relatifs à l'application considérée. Ceci est réalisé à l'aide d'un réseau bayésien utilisé pour la catégorisation automatique. Le résultat de ce module est ensuite utilisé dans toutes les autres étapes du traitement.

La deuxième et la troisième composantes correspondent à celles déjà définies dans la figure 1, à savoir le traducteur sémantique et le convertisseur de représentation. Dans notre cas, le traducteur sémantique sert à étiqueter les phrases en utilisant les concepts trouvés par le premier module. Le module de conversion des représentations comprend deux sous-parties. La première sert à générer une requête SQL conceptuelle et la deuxième génère une requête directement interprétable par un SGBD.

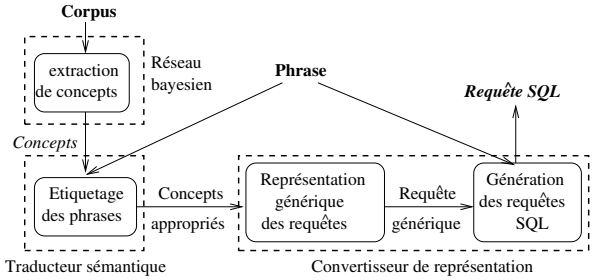

FIG. 2: Architecture détaillée de notre système de compréhension.

3. EXTRACTION AUTOMATIQUE DES CONCEPTS

Dans cette étape, nous cherchons à identifier les concepts sémantiques liés à une application. La détermination manuelle de ces concepts est une tâche très lourde que nous proposons d'automatiser.

Afin d'atteindre ce but, nous avons opté pour des méthodes de classification non supervisées. Pour obtenir des concepts cohérents, il est judicieux de regrouper les mots en fonction de leurs propriétés sémantiques. La méthode à utiliser regroupe les mots du corpus en différentes classes, construisant ainsi les concepts de l'application. Pour ce faire, nous utilisons les réseaux bayésiens. Ce choix est dû principalement au fondement mathématique de cette théorie et au mécanisme d'inférence puissant sous-jacent. Le réseau bayésien utilisé est celui de l'outil AutoClass qui accepte en entrée des valeurs réelles, mais aussi des valeurs non numériques comme des mots, des caractères etc [3]. En résultat, le réseau fournit les probabilités d'appartenance d'un élément en entrée à chacune des classes qu'il propose. Il suppose l'existence d'une variable multinomiale cachée qui peut représenter les différentes classes auxquelles appartiennent les éléments en entrée.

3.1. Représentation vectorielle des mots

Il nous reste à trouver une représentation adéquate des mots. En effet, un mot peut être représenté par plusieurs traits possibles, mais rares sont ceux qui peuvent lui donner une représentation sémantique complète. Nous utilisons deux types d'information : le contexte d'un mot et sa similarité avec les autres mots du lexique. Afin d'exprimer cette similarité, nous utilisons la mesure de l'information mutuelle moyenne [4]. Nous associons à chaque mot un vecteur à M éléments, où M est la taille du lexique. L'élément j de ce vecteur représente la valeur de l'information mutuelle moyenne entre le mot j du lexique et le mot à représenter. La formule de l'information mutuelle moyenne entre deux mots w_a et w_b est donnée par [7] :

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

Où $P(w_a, w_b)$ est la probabilité de trouver les deux mots w_a et w_b dans la même phrase, $P(w_a \mid w_b)$ 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 $P(\overline{w}_a)$ est la probabilité de ne pas avoir rencontré le mot w_a etc.

Afin d'utiliser le contexte et la mesure d'information mutuelle, nous représentons désormais chaque mot par une matrice d'information mutuelle moyenne de dimension $M \times 3$. La première colonne correspond au vecteur d'information mutuelle moyenne décrit précédemment, la deuxième colonne représente l'information mutuelle moyenne entre un mot quelconque du vocabulaire et le contexte gauche du mot à représenter, de même pour la troisième colonne mais en prenant en compte 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 :

$$IMM_j(C_g^i) = \frac{\sum_{w_g \in \, contexte\ gauche\ de\ W_i} I(w_j : w_g).K_{wg}}{\sum_{w_g \in \, contexte\ gauche\ de\ W_i} K_{wg}} \tag{2}$$

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_{wg} est le nombre de fois où le mot w_g est trouvé comme contexte gauche du mot W_i. Le mot W_i sera donc représenté par une matrice comme le montre la figure 3.

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

FIG. 3: Représentation matricielle du mot W_i.

TAB. 1: Quelques exemples de requêtes du corpus d'apprentissage.

Donnez-moi la hauteur du cours du groupe Alcatel.
Est-ce que tu pourrais me fournir la valeur du cours d'Alcatel.
Faxez-moi le cours de la BNP.
Il faut que j'ai le cours de la société Alcatel.
J'ai besoin de la progression du cours minimum de l'action BNP.
J'aimerais avoir l'évolution du cours moyen du titre la BNP.
Je souhaiterais la hauteur du cours d'Alcatel par mail .
Je serais intéressé par avoir les limites de la BNP.
Quel est le cours le plus haut du groupe Alcatel.
J'ai besoin d'avoir les limites du cours le plus bas de l'action BNP.

TAB. 2: Quelques exemples de concepts obtenus en utilisant la représentation matricielle.

Concept	Groupe de mots
Information	Informations, renseignements, cours, montant, hauteur, niveau, valeur, dernier, premier
Valeur	Plus-bas, plus-haut, moyen, maximum, minimum,
Cours	Groupe, titre, societe, action
Evolution	Evolution, progression, variation, limites
Nom	Alcatel, BNP

3.2. Résultats expérimentaux

Nous nous intéressons à une application de consultation du cours de la bourse. Pour ce faire, nous utilisons un corpus développé dans le cadre du projet RNRT IVOMOB dont l'objectif est de construire une interface vocale pour les services accessibles par les utilisateurs depuis des véhicules mobiles. Le corpus contient 51864 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.

En utilisant AutoClass avec la représentation matricielle des mots décrite ci-dessus, nous avons obtenu une liste de 10 concepts cohérents et parfaitement liés à notre application. Quelques exemples de ces concepts sont donnés dans la table 2.

L'objectif final est de fournir les requêtes SQL permettant de répondre aux demandes des utilisateurs. Les concepts obtenus sont très pertinents et vont nous aider à atteindre ce but. La hiérarchisation des concepts (figure 4) permet en fait de représenter une requête. La racine de cet arbre est le méta-concept "Requête" et chaque nœud représente une partie de la requête. Les feuilles de cet arbre sont représentées par les concepts eux-mêmes.

4. ÉTIQUETAGE ET POST-TRAITEMENT

La dernière étape consiste à fournir les commandes SQL associées aux requêtes émises en entrée. C'est au cours de cette phase que l'interprétation des requêtes est effectuée. En effet, disposant de l'ensemble des concepts qui régissent l'application, nous pouvons attribuer à chaque

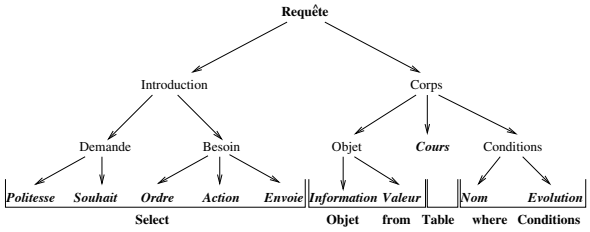

FIG. 4: Hiérarchie des concepts obtenus.

requête les concepts appropriés. Cette étape ne pose pas de problèmes particuliers puisque le procédé d'étiquetage est déterministe.

4.1. Représentation générique des requêtes

La représentation générique des requêtes constitue la première étape du "Convertisseur de représentation" (c.f. figure 2). La fonction principale de ce module est de transformer la représentation conceptuelle de la phrase en une représentation proche d'une requête SQL. En fait, il s'agit d'une requête SQL dont les objets ne sont pas encore identifiés. Ainsi, dans cette étape on génère une requête (c.f. figure 5) du style :

select *Information, Valeur*
from *Cours*
where *Condition_Nom* ;

Les objets en italiques de cette requête ne sont pas encore instanciés à ce stade. Cette génération de requête SQL dite générique est assurée par un moteur d'inférence.

4.2. Génération des requêtes SQL

Dans cette dernière étape de compréhension, nous instancions chaque concept, dans la requête générique obtenue, par sa valeur extraite de la phrase initiale. Ceci est réalisé à l'aide d'un mécanisme de mise en correspondance entre les mots de la phrase initiale et les concepts de la requête générique. Ensuite, une requête SQL est générée. Dans la figure 5, nous donnons un exemple illustrant les différentes étapes du processus de compréhension, allant du texte/signal jusqu'à l'obtention de la requête SQL finale.

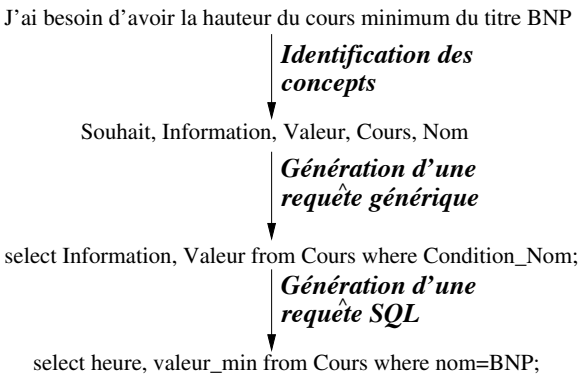

FIG. 5: Chaîne de traitement appliquée à une requête en langage naturel.

5. UN SYSTÈME DE COMPRÉHENSION DE LA PAROLE FINALISÉ

La dernière étape de ce travail consiste à intégrer le module de compréhension dans une plate-forme réelle de reconnaissance automatique de la parole. Pour cela, nous avons utilisé la sortie d'un système de reconnaissance comme entrée de notre module de compréhension. Dans la suite, nous présentons brièvement le moteur de reconnaissance utilisé et les conditions de nos expérimentations. Ensuite, nous discutons des résultats obtenus avant et après l'étape de reconnaissance.

5.1. Conditions expérimentales

Nous utilisons le système de reconnaissance automatique de la parole ESPERE (Engine for SPEech REcognition) développé dans l'équipe [6]. Le logiciel ESPERE constitue une boîte à outils conçue autour d'un moteur de reconnaissance markovien. Pour nos expérimentations, nous avons choisi la paramétrisation acoustique suivante : 35 coefficients (12 MFCC et leurs coefficients de régression du premier et deuxième ordre, en omettant C0).

Les modèles de Markov choisis pour la reconnaissance sont des modèles phonétiques à 3 états multigaussiens indépendants du contexte. Le modèle de langage utilisé est de type bigramme, développé sur le vocabulaire de l'application qui contient environ 150 mots.

Enfin pour adapter le système à notre plate-forme expérimentale, nous avons ajouté quelques fonctionnalités pour faciliter son utilisation réelle dans ce processus de compréhension. Nous avons donc veillé à ce que l'entrée du système de reconnaissance contienne le moins de bruit possible en détectant le début et la fin de la parole afin que le système n'insère pas des mots incorrects. En plus, nous avons réalisé une interface de commandes pour pouvoir communiquer plus librement avec le système.

Ainsi, notre système de compréhension orale est exploitable et opérationnel. Il permet de prendre un signal acoustique en entrée et de délivrer en sortie une requête SQL. Nous considérons que la compréhension a abouti lorsque la requête SQL générée correspond à ce qui a été prononcé. Autrement dit, si la requête SQL générée est exécutée par un système de gestion de base de données, la réponse obtenue doit répondre exactement à ce qui a été demandé par le locuteur.

5.2. Résultats et discussion

Les résultats obtenus par notre sytème de compréhension ont montré une robustesse par rapport aux erreurs de reconnaissance. En effet, le taux de reconnaissance obtenu sur notre corpus de test qui contient 100 phrases différentes est égale à 66.4%, cependant en terme de concepts l'identification atteint les 80.9%. Enfin, en terme de requêtes SQL correctes, nous obtenons un taux de 82% qui correspond au taux de compréhension. Si on prend en entrée le même corpus mais sous forme textuelle, le taux de compréhension atteint les 92%. Ceci montre que malgré les erreurs de reconnaissance, le système arrive dans plusieurs cas à surmonter ces limites et à proposer une requête SQL correcte répondant à la demande de l'utilisateur. En effet, afin de comprendre une requête exprimée en langage naturel, le système se base sur un ensemble de mots clés constituant les concepts sémantiques de l'application. Le fait de passer par les concepts permet de réduire les erreurs grâce à la généralisation apportée par les concepts.

6. CONCLUSION

Dans cet article, nous avons considéré la compréhension automatique comme un problème d'association entre deux langages différents, le langage naturel et le langage des concepts. Nous avons proposé une nouvelle méthode pour l'extraction automatique des concepts, ainsi qu'une approche d'étiquetage et de génération automatique des requêtes SQL correspondant aux demandes des utilisateurs. Les tâches d'extraction de concepts et d'étiquetage, d'habitude réalisées manuellement, constituent la phase la plus délicate et la plus coûteuse dans le processus de compréhension. La méthode proposée dans cet article a permis d'éviter ce recours à l'expertise humaine et nous a permis d'avoir 92% de bonnes réponses sur un corpus de test de 100 requêtes.

Nous avons aussi intégré notre module de compréhension dans un système de reconnaissance automatique de la parole afin de réaliser une application interactive complète qui a montré une robustesse vis à vis des erreurs de reconnaissance et qui a donné un taux de 82% de requêtes SQL correctes avec un taux de reconnaissance de 66.4% mots.

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 cela, il faut utiliser les nouveaux mots souvent rencontrés pour enrichir les concepts initiaux ou pour créer d'autres concepts.

RÉFÉRENCES

[1] R. Pieraccini et E. Levin et E. Vidal. Learning how to understet language. In *Proceedings 4rd European Conference on Speech Communication et Technology*, Berlin, 1993.

[2] H. Maynard et F. Lefèvre. Apprentissage d'un module stochastique de compréhension de la parole. In *24èmes Journées d'Étude sur la parole*, Nancy, Juin 2002.

[3] P. Cheeseman et J. Stutz. Bayesian classification (autoclass) : Theory et results. In *Advances in Knowledge Discovery et Data Mining*. U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy, 1996.

[4] S. Jamoussi et K. Smaïli et J.P. Haton. Neural network et information theory in speech understeting. In *International Workshop on Speech et Computer, SPECOM'02*, St. Petersbourg, Septembre 2002.

[5] C. Bousquet-Vernhettes et N. Vigouroux. Context use to improve the speech understeting processing. In *International Workshop on Speech et Computer, SPECOM'01*, Moscow, Octobre 2001.

[6] D. Fohr et O. Mella et C. Antoine. The automatic speech recognition engine espere experiments on telephone speech. In *Proceedings of the International Conference on Spoken Language Processing*, Beijing, Octobre 2000.

[7] R. Rosenfeld. *Adaptive Statistical Language Modeling : A Maximum Entropy Approach*. PhD thesis, School of Computer Science Carnegie Mellon University, Pittsburgh, PA 15213, Avril 1994.