Une nouvelle représentation vectorielle pour la classification sémantique

Salma Jamoussi

Multimedia, InfoRmation systems and Advanced Computing Laboratory, MIRACL.
Pôle technologique de Sfax, Route de Tunis Km 10 B.P. 242 SFAX 3021, Tunisie.
jamoussi@gmail.com

RÉSUMÉ. *L'idée que nous défendons dans cet article est qu'il est possible d'obtenir des concepts sémantiques significatifs par des méthodes de classification automatique. Pour ce faire, nous commençons par proposer des mesures permettant de quantifier les relations sémantiques entre mots. Ensuite, nous utilisons les méthodes de classification non supervisée pour construire les concepts d'une manière automatique. Nous testons alors deux méthodes de partitionnement : l'algorithme des* K-means *et les cartes de Kohonen. Ensuite, nous utilisons le réseau bayésien AutoClass conçu pour la classification non supervisée. Pour grouper les mots du vocabulaire en différentes classes, nous avons testé trois représentations vectorielles des mots. La première est une représentation contextuelle simple. La deuxième associe à chaque mot un vecteur de valeurs représentant sa similarité avec tous les mots du lexique. Enfin, la troisième représentation est une combinaison des deux premières.*

ABSTRACT. *The idea we defend in this paper is the possibility to obtain significant semantic concepts using clustering methods. We start by defining some semantic measures to quantify the semantic relations between words. Then, we use some clustering methods to build up concepts in an automatic way. We test two well known methods : the K-means algorithm and the Kohonen maps. Then, we propose the use of a Bayesian network conceived for clustering and called AutoClass. To group the words of the vocabulary in various classes, we test three vector representations of words. The first is a simple contextual representation. The second associates to each word a vector which represents its similarity with each word of the vocabulary. The third representation is a combination of the first and the second one.*

MOTS-CLÉS : *classification automatique, concepts sémantiques, représentation vectorielle des mots.*

KEYWORDS: *clustering, semantic concepts, word vector representation.*

TAL. Volume 50 – n° 3/2009, pages 23 à 57

1. Introduction

L'étiquetage sémantique est l'une des tâches les plus compliquées en recherche d'information. Elle repose généralement sur un travail manuel très lourd et très fastidieux. En effet, ce travail nécessite non seulement du temps et de l'effort, mais aussi beaucoup de méthodologie et d'objectivité. Cependant ce travail est nécessaire dès qu'il s'agit d'effectuer des recherches sémantiques sur des documents textuels, images ou vidéos (Benayed *et al.,* 2003), (Jamoussi *et al.,* 2004). Dans ce travail, on s'intéresse tout particulièrement aux documents textuels, puisque c'est le type de documents le plus répandu mais aussi sur lequel les recherches ont atteint une certaine maturité qui nous rend capable d'appréhender la tâche de leur indexation sémantique. La tâche d'étiquetage sémantique consiste à attribuer un sens à chaque partie du texte (mot, séquence de mots, phrase, etc.) et ce, en utilisant une représentation sémantique adéquate capable de traduire le plus fidèlement possible le sens du texte tout en étant simple et synthétique. Dans la littérature, les représentations sémantiques existantes varient d'une application à une autre. Elles peuvent être selon le cas, très proches du langage naturel ou alors exprimées à l'aide d'un langage formel similaire au langage de commandes.

Dans cet article on s'intéresse à une application assez particulière de l'étiquetage sémantique. Celle de la compréhension automatique de la parole. L'idée est de traduire une phrase prononcée en langage naturel en une suite de concepts sémantiques pour en déduire le sens. Dans ce cadre, nous traitons deux applications d'interrogation vocale de bases de données où le dialogue est finalisé et propre à un domaine donné. Cela signifie que les systèmes visés sont spécifiques à une tâche donnée et que le vocabulaire utilisé est limité au domaine de cette tâche. La représentation sémantique la plus utilisée dans ce cas est celle des *concepts sémantiques*. Ce choix a été adopté dans plusieurs autres travaux antérieurs (Pieraccini *et al.*, 1992), (Honkela *et al.*, 1997), (Bousquet-Vernhettes *et al.*, 1999), (Maynard et Lefèvre, 2002). Les concepts sémantiques sont utilisés pour représenter le sens d'un énoncé donné. Un concept est une classe de termes (simples ou complexes) traitant d'un même sujet et partageant des propriétés sémantiques communes. Par exemple, les termes *hôtel, chambre, auberge et studio* peuvent tous correspondre au concept « *hébergement* » dans une application touristique.

Le système CHRONUS (*Conceptuel Hidden Representation Of Natural Unconstrained Speech*) de AT&T (Pieraccini *et al.,* 1992) construit ses représentations sémantiques en deux étapes. La première étape consiste en une analyse lexicale de la phrase. Tout d'abord, une étape de lemmatisation est effectuée. Ensuite, les mots sont groupés en classes (noms de villes, noms d'avions, etc.) et enfin les séquences de mots sont attachés pour définir des expressions. Au niveau de la deuxième étape, un décodage conceptuel est mis en œuvre afin d'associer à chaque élément de la phrase le concept sémantique correspondant. Dans ce travail, les concepts sémantiques considérés sont au nombre de sept et ils sont très liés aux unités représentées dans la

base de données pour faciliter sa consultation.

Dans la tâche ATIS (*Air Travel Information Services*) (Minker, 1997), la représentation sémantique adoptée joue un rôle très important dans le processus de compréhension. Tout d'abord, une liste de concepts sémantiques est définie. Ensuite, dans chaque phrase on essaie de détecter un ou plusieurs mots de référence (sujet de la requête : réservation, horaire, tarif, réduction, etc.). Enfin, l'énoncé est présenté sous forme d'un « *schéma sémantique* » où, on identifie le concept clé et on définit ses attributs. Considérons par exemple la requête suivante : *Je voudrais aller de Boston à Denver*. Dans ce cas, le mot **aller** joue le rôle d'un mot référence, il permet d'identifier le concept « **vol** », les marqueurs **de** et **à** permettent d'identifier et d'instancier les attributs de ce concept, à savoir « ville_départ » et « ville_arrivée », tous les autres mots sont jugés inutiles pour l'interprétation de cet énoncé.

Pour la tâche ARISE (*Automatic Railway Information Systems for Europe*) (Baggia *et al.*, 1999), la représentation sémantique adoptée s'appelle CVR (concept/valeur représentation). Elle consiste à représenter chaque concept identifié dans une phrase par un triplet [**mode/concept : valeur**]. Dans ce cas, une phrase peut contenir plus qu'un concept (et non plus un seul concept clé) et on parle dans ce cas de sous-schémas. De la même manière la liste des concepts est fixée au préalable, tout ce qu'il reste à faire c'est identifier les concepts sémantiques présents dans la phrase puis attribuer à chacun une valeur et une modalité (affirmative, négative ou interrogative). Par exemple, nous pouvons trouver un concept sous la forme [*-/ville : Roissy*], ce qui signifie que : le concept considéré est **ville**, sa valeur est *Roissy* et sa modalité est négative (c'est-à-dire que la ville demandée n'est pas *Roissy*).

Dans l'environnement CACAO (compréhension automatique par segments conceptuels assistée par ordinateur) (Bousquet-Vernhettes *et al.*, 1999), la représentation sémantique utilisée s'appelle les segments conceptuels. Un segment conceptuel réunit les groupes de mots ayant un sens commun (par exemple des groupes de mots comme « partir pour Saint-Petersbourg » et « aller à Moscou » font partie du même segment conceptuel *direction*). Dans ce travail, chaque segment conceptuel est représenté sous forme d'un modèle de Markov caché. On admet que n'importe quel énoncé peut être décomposé en une suite de segments conceptuels à l'aide d'un module de décodage conceptuel qui utilise l'un des deux algorithmes $Viterbi$ ou A^*. À la fin de cette étape de décodage, seule la meilleure solution est retenue représentant ainsi le sens de l'énoncé.

Dans tous ces travaux les concepts sémantiques considérés sont identifiés manuellement par un expert. Cette tâche de détermination des concepts sous-jacents à l'application traitée devient vite fastidieuse et impraticable dès que le domaine de l'application s'élargit où que de nouvelles fonctionnalités sont intégrées dans le système. Il est

donc clair que la tâche de l'identification des concepts doit se faire automatiquement et ce, à l'aide d'un corpus d'apprentissage dédié à l'application en question.

L'idée que nous défendons ici est qu'il est possible d'obtenir des concepts sémantiques proches de ceux identifiés par l'expert à l'aide des méthodes de classification non supervisée. Il s'agit donc de grouper les mots du corpus en différentes classes sémantiques représentant chacune un concept précis. Cependant le problème qui reste posé, quant à la concrétisation de cette idée, est celui des représentations numériques des mots. En effet, la plupart des méthodes de classification traitent des données numériques pour calculer des distances, des moyennes, des centres, etc. Elles sont donc inexploitables quand il s'agit de mots (dans leur forme brut). Dans cet article, nous proposons des alternatives de représentations vectorielles des mots qui ont pour but de représenter numériquement les caractéristiques sémantiques de chaque mot. Ceci permettra de classer les mots en se fondant sur leurs propriétés sémantiques.

Le choix des représentations vectorielles sémantiques est un des problèmes les plus difficiles en traitement automatique du langage naturel. Les travaux traitant de ce type de problématique s'intéressent plutôt à la tâche de classification de documents et utilisent pour cela deux types de représentations vectorielles :

– des représentations linguistiques : utilisant des propriétés lexicales, syntaxiques ou grammaticales pour décrire les documents à classer (Bourigault *et al.*, 2005). Des informations d'ordre contextuel peuvent aussi être utilisées dans ce cas pour mieux caractériser les documents d'un point de vue sémantique (Nenadic *et al.*, 2002) ;

– des représentations statistiques : elles varient des plus simples qui sont à base de fréquences de mots au plus complexes utilisant des méthodes d'analyse de données comme les méthodes LSI et ACP. La méthode TF*IDF est sans conteste la méthode de représentation vectorielle qui a trouvé le plus de succès dans ce domaine.

Cependant, dans un contexte de compréhension de requêtes pour la consultation de bases de données, on ne peut pas travailler au niveau document mais seulement au niveau requête. Les présentations du type fréquence de mots, TF*IDF ou LSI ne peuvent être appliquées vu le nombre restreint de mots dans chaque phrase. Il faut donc chercher d'autres types de représentations qui peuvent s'associer à une indexation sémantique par mot.

Dans le travail de Laskri (Laskri et Meftouh, 2002), les vecteurs représentant les mots ont été élaborés à la main, en adoptant comme caractéristiques les propriétés sémantiques des mots. Ce travail manuel a présenté une charge de travail conséquente puisque ces caractéristiques dépendent aussi de la classe lexicale des mots. Par exemple pour les noms, il faut considérer les propriétés suivantes : animé, humain, féminin, unique, concret, etc. Ces vecteurs caractéristiques seront présentés à l'entrée d'un réseau de neurones récurrent qui doit décider du rôle que joue le mot en entrée. Les différents rôles possibles sont au nombre de 14, ils permettent de donner une représentation sémantique interne des phrases.

La charge du travail manuel que nécessite une représentation rigoureuse et efficace est assez importante ce qui complique encore la tâche. Afin de résoudre ce problème, une idée originale a été proposée par Miikkulainen dans (Miikkulainen et Dyer, 1991) inspirée des travaux de Elman (Elman, 1990). Il s'agit de construire à l'aide d'un réseau de neurones artificiels, une représentation distribuée interne des mots en utilisant l'algorithme de rétropropagation de l'erreur. Le réseau utilisé FGREP (*Forming Global Representations with Extended backPropagagtion*) est composé de trois couches : la couche d'entrée, la couche cachée et la couche de sortie. Durant l'apprentissage, le réseau rétropropage l'erreur déduite en comparant sa sortie à la sortie désirée. Et ainsi, il modifie progressivement en sens arrière les poids des connexions entre les neurones jusqu'à atteindre la couche d'entrée qu'il modifie aussi pour retrouver une nouvelle représentation de l'entrée. C'est cette nouvelle représentation qu'il va utiliser en entrée à la prochaine étape de son apprentissage. Ainsi de suite jusqu'à atteindre un seuil d'erreur acceptable. Enfin, les vecteurs obtenus par ce processus seront considérés comme des représentations sémantiques distribuées des mots présentés en entrée.

Quelques travaux arrivent à dégager des concepts sémantiques sans passer par une représentation vectorielle. Dans (Siu et Meng, 1999) par exemple, les auteurs ont exposé une méthode semi-automatique pour l'acquisition de structures sémantiques spécifiques d'un domaine. La distance de Kullback-Leibler a été utilisée pour calculer la similarité entre mots (ou groupes de mots) afin d'identifier des relations sémantiques entre termes. Dans ce genre de travaux, la distance entre mots suffit pour pouvoir, par exemple, identifier des règles d'association ou pour construire des hiérarchies de mots.

Cet article est organisé comme suit : la section suivante est dédiée à une description des mesures sémantiques existantes dans la littérature. Dans la section 3 nous proposons différentes représentations vectorielles des mots capables de rendre la tâche d'identification des concepts sémantiques plus efficace. Ces représentations sont fondées sur des métriques sémantiques déjà utilisées dans la littérature pour dégager les *triggers* (mots sémantiquement corrélés) dans un corpus donné. Dans la section 4, nous introduisons quelques méthodes de classification non supervisée qui peuvent être utilisées en classification sémantique : l'algorithme de *K-means* et les cartes de Kohonen. Ensuite, nous proposons d'utiliser les réseaux bayésiens pour la classification sémantique. Après avoir exposé les principes des réseaux bayésiens, nous présentons le réseau bayésien AutoClass conçu pour la classification non supervisée et qui nous a été très utile pour l'extraction automatique des concepts. Les résultats relatifs aux différentes méthodes de classification que nous avons employées sont exposés dans la dernière section.

2. Une métrique pour la sémantique

Dans le domaine du langage naturel, les données ne sont pas numériques. Afin de les traiter d'une manière automatique, il est indispensable de leur trouver une représentation numérique appropriée qui préserve leurs propriétés sémantiques et syntaxiques. Ensuite, il faut trouver des mesures efficaces permettant de comparer et de calculer

des distances sémantiques entre les mots. Bien que traitée dans la littérature, cette dernière question reste l'un des points les plus délicats dans le domaine de l'indexation sémantique. En fait, une mesure sémantique doit pouvoir exprimer quantitativement une similarité entre deux termes d'un point de vue sémantique. Lorsqu'il s'agit d'une distance elle doit alors satisfaire les conditions suivantes :

– cette mesure doit être exprimée numériquement afin de pouvoir être traitée d'une manière informatique. Elle permettra ainsi de quantifier la relation sémantique entre deux mots. De plus, attribuer une valeur numérique à cette mesure permet de lever toute ambiguïté et ne laisse que peu de place pour la subjectivité ;

– cette mesure doit être calculée à partir d'un corpus d'apprentissage qui illustre l'emploi sémantique de ses différents termes. Ainsi, nos calculs seront relatifs à une base d'apprentissage ce qui implique que les distances trouvées seront relatives à l'application traitée seulement ;

– de plus cette mesure doit satisfaire toutes les propriétés habituelles d'une distance mathématique : la symétrie ($distance(x, y) = distance(y, x)$), la séparation ($distance(x, y) = 0 \Leftrightarrow x = y$) et l'inégalité triangulaire ($distance(x, y) \leq distance(x, z) + distance(z, y)$).

Différentes méthodes existent pour trouver les relations sémantiques entre les mots en se fondant sur un corpus d'apprentissage. Dans la suite, nous présentons les mesures qui nous intéressent le plus et qui ont prouvé leur efficacité dans des tâches de comparaison sémantique (Siu et Meng, 1999).

2.1. *L'information mutuelle moyenne*

L'information mutuelle entre deux mots ou deux classes de mots représente une mesure quantitative de l'information fournie par ces deux entités en les considérant comme deux variables aléatoires X et Y. La formule de l'information mutuelle entre deux mots w_i et w_j est donnée par :

$$IM(w_i, w_j) = \log \frac{P(X = w_i, Y = w_j)}{P(X = w_i)P(Y = w_j)} = \log \frac{P(w_i, w_j)}{P(w_i)P(w_j)} \quad [1]$$

où $P(w_i, w_j)$ est la probabilité jointe de trouver les deux mots w_i et w_j dans un même contexte, $P(w_i)$ et $P(w_j)$ représentent respectivement la probabilité de rencontrer indépendamment w_i ou w_j. L'information mutuelle entre les mots constitue donc une mesure représentative de la cooccurrence de deux mots dans une même phrase ou dans un même paragraphe. Elle montre que cette co-occurrence n'est pas due au hasard. Si elle est importante c'est que ces deux mots se trouvent souvent ensemble et donc l'existence de l'un dépend de l'autre. En calculant l'information mutuelle entre les différents mots du texte à analyser nous pouvons extraire la liste de ses *triggers*. Un *trigger* est un couple de mots sémantiquement corrélés c'est-à-dire que l'apparition d'un des deux mots renforce la probabilité d'apparition de l'autre.

Une autre mesure utilisée dans le même but de sélection des mots corrélés est celle de l'information mutuelle moyenne. Cette dernière a l'avantage de mesurer en plus de l'information mutuelle classique, l'impact de l'absence d'un mot sur l'apparition d'un autre. En effet, la non-présence d'un mot dans l'historique peut aussi renforcer l'apparition ou non d'un autre. La formule de cette mesure est la suivante (Rosenfeld, 1994) :

$$\begin{aligned} IMM(w_i, w_j) = \quad & P(w_i, w_j) \log \tfrac{P(w_i,w_j)}{P(w_i)P(w_j)} + P(\overline{w}_i, \overline{w}_j) \log \tfrac{P(\overline{w}_i,\overline{w}_j)}{P(\overline{w}_i)P(\overline{w}_j)} + \\ & P(\overline{w}_i, w_j) \log \tfrac{P(\overline{w}_i,w_j)}{P(\overline{w}_i)P(w_j)} + P(w_i, \overline{w}_j) \log \tfrac{P(w_i,\overline{w}_j)}{P(w_i)P(\overline{w}_j)} \end{aligned} \quad [2]$$

où $\overline{w}_i$ et $\overline{w}_j$ désignent respectivement la non-présence des mots w_i et w_j. Dans cette formule on fait la sommation des informations mutuelles correspondant aux différentes possibilités de rencontrer ou non w_i et w_j dans le contexte. Si la probabilité de rencontrer w_i renforce la probabilité de rencontrer w_j (ou inversement), on aura $\log \frac{P(w_i,w_j)}{P(w_i)P(w_j)} > 0$ et $\log \frac{P(\overline{w}_i,\overline{w}_j)}{P(\overline{w}_i)P(\overline{w}_j)} > 0$ mais de l'autre côté les deux autres quantités (quoique faibles) seront négatives. Ce qui traduit bien le degré de corrélation entre w_i et w_j.

Pour calculer le nombre de cooccurrences de deux termes w_i et w_j, une fenêtre de mots est fixée. Nous supposons que deux mots sont rencontrés dans le même contexte, s'ils apparaissent tous les deux dans cette fenêtre de mots. La taille de cette fenêtre peut être statique ou dynamique. Dans notre cas, nous adoptons comme fenêtre la totalité de la phrase traitée. En effet, nous considérons que la phrase est une entité sémantique complète et qu'on ne peut juger de la relation entre les mots qu'en prenant toute la phrase comme fenêtre d'analyse. Nous justifions ce choix par le fait qu'une fenêtre plus petite que la phrase ne va pas considérer le sens complet de cette dernière. Par ailleurs, si la fenêtre adoptée est plus grande que la phrase, des éléments de la phrase suivante vont être pris en compte, or on peut changer complètement de sujet d'une phrase à une autre.

(Dagan *et al.*, 1999) propose une autre manière d'utiliser la mesure d'information mutuelle pour calculer la similarité entre deux mots w_i et w_j. Il exploite pour cela les valeurs d'information mutuelle entre les couples (w_i, w_k) et (w_j, w_k) pour tout mot w_k du vocabulaire. La similarité qu'il propose est évaluée de la manière suivante :

$$DS(w_i, w_j) = \frac{1}{2|V|} \sum_{k=1}^{|V|} \Big(\frac{min(IM(w_k, w_i), IM(w_k, w_j))}{max(IM(w_k, w_i), IM(w_k, w_j))} + \frac{min(IM(w_i, w_k), IM(w_j, w_k))}{max(IM(w_i, w_k), IM(w_j, w_k))} \Big)$$

où V est le vocabulaire et $IM(w_k, w_i)$ est l'information mutuelle simple entre les mots w_k et w_i.

2.2. *La distance de Kullback-Leibler*

La plupart des travaux qui s'intéressent à la sémantique utilisent la distance de Kullback-Leibler pour calculer des distances sémantiques entre mots (Siu et Meng, 1999 ; Bigi, 2003). Cette distance est utilisée en théorie de l'information pour déterminer la distance entre deux distributions de probabilités P_i et P_j. Ces deux distributions peuvent être relatives à deux mots différents w_i et w_j dont on calcule la distance en adoptant les équations suivantes :

$$D(P_i^d, P_j^d) = \sum_{w_k \in V} P_i^d(w_k) \log \frac{P_i^d(w_k)}{P_j^d(w_k)} \quad [3]$$

$P_i^d(w_k)$ n'est autre que $P(w_k|w_i)$. Elle exprime la probabilité conditionnelle de trouver le mot w_k sachant que nous avons déjà rencontré le mot w_i dans le contexte passé. Elle représente donc la probabilité de trouver w_k à droite de w_i d'où la notation $P_i^d(w_k)$. La distance de Kullback-Leibler ainsi définie n'est pas symétrique. Il faut donc définir une autre distance qui tient compte des deux termes $D(P_i^d, P_j^d)$ et $D(P_j^d, P_i^d)$. Cette distance s'appelle la divergence. Elle est donnée par l'équation suivante :

$$Div(P_i^d, P_j^d) = D(P_i^d, P_j^d) + D(P_j^d, P_i^d) \quad [4]$$

Afin de tenir compte des contextes gauche et droit des mots, on a besoin d'introduire aussi les distances de Kullback-Leibler relatives aux probabilités $P(w_i|w_k)$ et $P(w_j|w_k)$. Il s'agit donc de P_i^g et P_j^g. Ainsi, on peut définir une distance globale notée DKL. Cette distance est la somme des deux divergences gauche et droite. Elle est donnée par l'équation suivante :

$$DKL(w_i, w_j) = Div(P_i^d, P_j^d) + Div(P_i^g, P_j^g) \quad [5]$$

La sommation des divergences gauche et droite nous permet de travailler sur l'ensemble du contexte indépendamment de la position des mots. La formule de la distance de Kullback-Leibler a été mise au point dans (Kullback, 1959) et cette version symétrique (où on somme la divergence gauche et la divergence droite) est couramment utilisée dans la littérature (Siu et Meng, 1999 ; Bigi, 2003), etc.

Comme pour l'information mutuelle moyenne, la fenêtre de mots que nous avons fixée dans le cas de la distance de Kullback-Leibler est de taille dynamique. Elle correspond à la totalité de la phrase traitée. Ainsi, la distance de Kullback-Leibler, cherche à trouver des mots ou groupes de mots qui s'utilisent de la même manière (ayant les mêmes contextes) et donc qui ont la même sémantique. Elle peut donc nous être utile pour la comparaison sémantique des mots.

3. Représentation vectorielle des mots

Dans notre cas, nous souhaitons représenter numériquement des mots. Cette tâche s'avère compliquée puisqu'il est difficile de caractériser sémantiquement les mots en leur attribuant des nombres. D'un autre côté, les différentes méthodes de classification non supervisée que nous souhaitons appliquer ne peuvent pas fournir de bons résultats si les données à classer ne sont pas bien représentées. En effet, la représentation des données joue un rôle prédominant dans la tâche de classification surtout lorsqu'il s'agit d'une classification non supervisée.

Cette section est dédiée à l'étude de quelques représentations sémantiques utiles pour notre tâche d'extraction de concepts. Nous commençons par décrire une représentation contextuelle simple qui malheureusement, ne peut être appliquée qu'avec le classifieur bayésien AutoClass (voir section 4.3). Ensuite, nous exposons le modèle vectoriel qui est très utilisé dans la littérature de l'extraction de l'information. Enfin, nous proposons deux nouveaux types de représentations numériques des mots fondées sur les distances sémantiques décrites dans la section précédente. Ces dernières constituent des représentations vectorielles efficaces pour des fins de classification sémantique.

3.1. *Représentation contextuelle*

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 1 montre un exemple de la représentation contextuelle des mots. Dans cet exemple, nous disposons d'une phrase contenant quatre mots ; en fixant $N = 2$, nous représentons chaque mot W_i par les deux mots de ses contextes gauche et droit.

L'avantage de cette méthode c'est qu'elle est simple et intuitive. Cependant, son défaut majeur réside dans le fait qu'elle attribue un vecteur par occurrence et non pas par mot. Autrement dit, pour un seul mot w_i, cette méthode génère M vecteurs différents. Le vecteur numéro k, par exemple, représente les contextes gauche et droit de l'occurrence numéro k du mot w_i dans le corpus d'apprentissage. Ainsi, nous générons autant de vecteurs qu'il y a de mots dans le corpus. Ceci demande beaucoup d'espace mémoire et un temps de calcul considérable surtout lors de l'étape de classification.

Pour faciliter le travail de classification tout en améliorant les performances, nous avons ajouté à chaque vecteur une entrée plus discriminante qui correspond à la

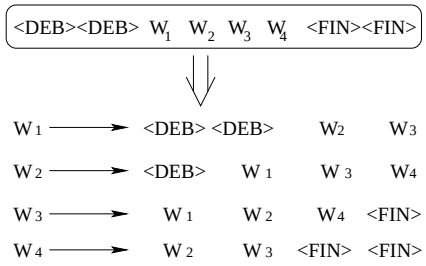

Figure 1. *Un exemple de représentation de mots par leurs contextes ($N = 2$).*

forme orthographique du mot. Ceci permet d'améliorer les résultats en fournissant des classes plus compactes.

3.2. *Le modèle vectoriel*

Dans le domaine de la recherche documentaire, plusieurs méthodes sont classées sous le nom de « modèles vectoriels ». Ces dernières consistent à associer à chaque document un vecteur de nombres réels qui permettent de le caractériser. Les méthodes les plus connues sont fondées sur le principe de « sac de mots », c'est-à-dire qu'elles utilisent un ensemble de mots considérés comme pertinents par rapport à l'application (ensemble des mots-clés de l'application) afin de représenter les documents traités.

Dans ce cas, les vecteurs caractéristiques adoptés contiennent les poids des différents mots-clés de l'application dans chaque document. Ainsi, un document sera présenté comme suit :

$$\overrightarrow{d_j} = \begin{bmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{ij} \\ \vdots \\ a_{mj} \end{bmatrix}$$

où a_{ij} désigne le poids du terme m_i dans le document d_j. Ce poids permet de refléter l'importance du terme m_i dans le document traité d_j. Il peut être calculé de diverses manières, la plus connue est la méthode TF-IDF (*Term Frequency, Inverse Document Frequency*) (Salton et Buckley, 1988).

Soit $tf(m_i, d_j)$ la fréquence du terme m_i dans le document d_j et $idf(m_i)$ l'inverse du nombre de documents qui contiennent le terme m_i. Dans le codage TF-IDF classique, les poids a_{ij} sont simplement calculés comme suit :

$$a_{ij} = tf(m_i, d_j) \times idf(m_i)$$

La mesure de similarité entre documents utilisée dans ce cas est la mesure cosinus définie par :

$$Sim(d_j, d_k) = \frac{\overrightarrow{d_j}.\overrightarrow{d_k}}{||\overrightarrow{d_j}|| \times ||\overrightarrow{d_k}||}$$

De très nombreuses variantes du codage TF-IDF et des fonctions de similarités ont été proposées dans la littérature. Elles ont fait l'objet d'un grand nombre de comparaisons expérimentales. Nous pouvons citer par exemple la méthode proposée par (Buckley *et al.*, 1992) qui introduit une pondération logarithmique des termes et celle de (Singhal *et al.*, 1996) qui, en plus de la pondération logarithmique, propose une normalisation par la moyenne de tous les $tf(m_i, d_j)$.

Ce modèle n'est pas applicable dans notre cas, car nous cherchons à représenter des mots et non pas des documents, cependant il nous a servi comme source d'inspiration. En effet, chercher la similarité entre les mots en accordant un poids d'importance à chaque terme du lexique est une idée intéressante et cohérente avec notre problème de classification. D'où notre approche d'utilisation des distances sémantiques présentée dans la section suivante.

3.3. *Les distances sémantiques*

L'idée de cette méthode consiste à combiner le modèle vectoriel et le calcul de similarité entre mots en une seule représentation. En effet, nous proposons de fusionner les deux opérations en un seul processus qui permet de livrer directement des vecteurs exprimant les degrés de similarité entre les mots du lexique.

La mesure de similarité que nous proposons d'utiliser est la mesure de l'information mutuelle moyenne donnée par l'équation 2. En effet, cette dernière prend en compte les différentes relations qui peuvent exister entre deux mots (co-occurrence dans la même phrase et absence des deux mots ou d'un seul mot dans une même phrase).

Dans cette approche, nous cherchons à associer à chaque mot un vecteur caractéristique qui contient autant d'éléments qu'il y a de mots dans le lexique de l'application. Notre proposition consiste à associer à l'élément numéro j de ce vecteur la valeur de l'information mutuelle moyenne entre le mot numéro j du lexique et le mot à repré-

senter, avec $j \in \{1 \dots M\}$. Ce vecteur exprime ainsi, le degré de similarité du mot en question avec tous les autres mots du corpus :

$$\mathbf{W_i} = \begin{bmatrix} IMM(w_1 : w_i) \\ IMM(w_2 : w_i) \\ \vdots \\ IMM(w_j : w_i) \\ \vdots \\ IMM(w_M : w_i) \end{bmatrix}$$

Comme nous l'avons déjà mentionné, cette méthode peut utiliser d'autres distances sémantiques que celle de l'information mutuelle moyenne. Dans le cadre de nos expérimentations, nous avons aussi essayé la distance de Kullback-Leibler (*cf.* section 2.2). Une comparaison entre les résultats donnés par ces deux distances est présentée à la section 6.

Avec une telle représentation vectorielle des mots, la tâche de classification sera beaucoup plus facile à faire que dans le cas des représentations contextuelles. De plus, le fait qu'un mot soit représenté par un seul vecteur résumant les propriétés de toutes ses occurrences permettra d'accélérer le processus de classification. Cependant, il ne faut pas oublier qu'avec cette méthode, nous avons perdu l'information contextuelle qui est très intéressante pour la caractérisation sémantique des mots. Dans la section suivante, nous essayons justement de récupérer ce type d'information en combinant l'approche courante avec celle fondée sur le contexte des mots.

3.4. *Une représentation matricielle mixte*

Afin d'améliorer encore la représentation vectorielle des mots, nous proposons de combiner l'approche contextuelle avec la méthode précédente. En effet, la première approche agit au niveau de l'occurrence et exploite directement les informations liées au contexte d'utilisation des mots, tandis que la seconde, 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.

La représentation vectorielle des mots proposée dans la section précédente est intéressante en termes de temps de calcul et d'espace mémoire. Elle permet d'associer à chaque mot, un seul vecteur. Or, si nous envisageons de la combiner avec l'approche contextuelle, il faut associer à chaque occurrence de mot un vecteur séparé. En effet, la combinaison la plus intuitive des deux approches consiste à utiliser les vecteurs contextuels des mots et à remplacer chaque élément par son vecteur de distances sémantiques.

Combiner ces deux méthodes consiste donc à représenter chaque occurrence d'un mot par une matrice d'information mutuelle moyenne de dimension $M \times 5$. Chaque

colonne représente le mot correspondant dans le vecteur contextuel. Ce travail est excessivement coûteux puisqu'il faut encore multiplier la taille de cette matrice ($M \times 5$) par le nombre de tous les mots du corpus.

L'idée que nous proposons consiste à faire une moyenne des distances sémantiques contextuelles des mots. D'une manière plus formelle, ceci repose sur l'utilisation d'une seule matrice par mot. Afin d'alléger encore cette représentation, nous nous sommes limités aux mots de contexte direct, c'est-à-dire les mots contigus au mot à représenter. Par conséquent, la matrice à utiliser contiendra trois colonnes seulement au lieu de cinq. La première colonne correspondra au vecteur sémantique relatif au mot en question comme précédemment défini (voir section 3.3). La deuxième colonne représentera une moyenne des distances sémantiques entre tous les mots du vocabulaire et le contexte gauche du mot à représenter. De même pour la troisième colonne, mais elle concernera le contexte droit.

Considérons, par exemple, l'information mutuelle moyenne comme distance sémantique à utiliser. Dans ce cas, la *j-ième* valeur de la deuxième colonne sera la moyenne pondérée des informations mutuelles moyennes entre le *j-ième* mot du vocabulaire et les mots du contexte gauche de W_i. Elle est calculée comme suit :

$$IMM_j(C_g^i) = \frac{\sum_{w_g \in \ contexte\ gauche\ de\ W_i} IMM(w_j, w_g) \times K_{wg}}{\sum_{w_g \in \ contexte\ gauche\ de\ W_i} K_{wg}} \qquad [6]$$

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. $IMM(w_j, w_g)$ 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. La même formule sera utilisée si nous choisissons de travailler avec une autre distance sémantique. Il suffira dans ce cas de changer $IMM(w_j, w_g)$ par la distance appropriée. De toutes les manières, le mot w_i sera représenté par une matrice $M \times 3$:

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

Au niveau de l'implémentation, cette représentation matricielle sera ordonnancée en une seule ligne (colonne par colonne) pour faciliter son exploitation par les différentes méthodes de classification utilisées.

La mesure de Dagan, décrite dans la section 2.1, utilise aussi les mêmes informations pour calculer la similarité entre deux mots. Cependant, la manière dont elle est

évaluée peut engendrer une grande perte d'information, chose que nous avons veillé à éviter dans notre cas. En effet, il est clair que cette similarité n'est pas aussi précise et complète que la matrice que nous proposons. Cette dernière exploite un maximum d'informations à propos du mot à représenter. Elle considère son contexte ainsi que sa similarité avec tous les autres mots du lexique sans leur faire subir de grandes transformations mathématiques. En la comparant avec l'approche précédente, cette représentation permettra sans doute d'améliorer les performances de la classification des mots afin d'extraire la liste de concepts relative à l'application traitée.

4. Méthodes de classification non supervisée

Dans le but d'extraire automatiquement les concepts sémantiques d'une application donnée, nous proposons dans cette section une approche fondée sur la classification non supervisée. Il s'agit d'utiliser des algorithmes de catégorisation connus dans la littérature pour pouvoir classer les mots de l'application en différentes classes sémantiques qui représentent les concepts recherchés.

Dans la suite, nous décrivons quelques méthodes de classification non supervisée en étudiant leur adéquation avec notre problématique. Nous commençons par la description des méthodes de partitionnement les plus connues, à savoir les *K-means* et les cartes auto-organisatrices de Kohonen. Ensuite, nous décrivons une approche fondée sur les réseaux bayésiens pour effectuer la tâche de regroupement des mots en concepts.

4.1. *L'algorithme des* K-means

L'algorithme des *K-means* est l'algorithme de *clustering* (ou partitionnement) le plus connu et le plus utilisé, tout en étant très efficace et simple. L'objectif de cet algorithme est de subdiviser l'ensemble des éléments donnés en entrée en un certain nombre de classes. *K-means* fait référence à l'utilisation de k centres de gravité (noyaux) pour partitionner les éléments d'entrée. Il a été proposé par (McQueen, 1967). Le principe de base de cet algorithme est assez simple. Il s'agit de choisir tout d'abord k centres de gravité d'une manière totalement arbitraire. Ensuite, on assigne aux k centres les éléments les plus proches selon une distance particulière en calculant à chaque fois les nouveaux centres de gravité des classes. Enfin, on s'arrête quand il n'y a plus de changement dans les centres de gravité. Les détails de ces différentes étapes sont donnés dans l'algorithme 1.

L'algorithme des *K-means* permet de traiter rapidement des données de taille importante. Cependant, l'un de ses principaux inconvénients réside dans le fait que la partition finale dépend de la partition initiale. En plus, le nombre k des différentes classes qui peuvent exister est déterminé par l'utilisateur. Cet algorithme tel qu'il est défini est incapable de le trouver d'une manière automatique.

Algorithme 1 Algorithme des *K-means*.

0- Choix des noyaux $(g_1, \ldots, g_K)$ des K classes $(c_1, \ldots, c_K)$ d'une manière aléatoire.
1- *Pour tout élément* e_i *de* e_1 *à* e_N *faire*
1.1- Chercher la classe c_k de l'élément en question (e_i).
$c_k = \arg \min_{j=1\ldots K} d(e_i, g_j)$
1.2- Recalculer le centre de gravité de la classe c_k :
$g_k = \frac{1}{n_k} \sum_{e_i \in c_k} e_i$
Fin Pour.
2- Arrêt s'il n'y a plus de changement dans les centres de gravité. Retourner en 1 sinon.

Dans notre travail, nous avons appliqué l'algorithme des *K-means* en utilisant la distance euclidienne. Cet algorithme nous permet de partitionner les mots du lexique en différentes classes représentant les concepts sémantiques de l'application traitée.

4.2. *Les cartes auto-organisatrices*

Les réseaux de Kohonen, ou cartes auto-organisatrices, ont été introduits en 1981 par Teuvo Kohonen (Kohonen, 1989). Ils sont généralement utilisés pour partitionner un ensemble de données et ils appartiennent à la classe des réseaux à compétition. Les neurones de la couche de sortie entrent en compétition de telle façon qu'un seul neurone en sortie soit activé pour une entrée donnée. Cette compétition entre les neurones est réalisée grâce aux connexions latérales.

Chaque neurone N_k de la carte de Kohonen est connecté à un nombre n d'entrées à travers n connexions de poids respectifs w_{ki}. Il existe aussi des connexions latérales de poids fixes. L'architecture d'un tel réseau est donnée par la figure 2. À la présentation d'une entrée, un neurone sur la carte est sélectionné. Il correspond le plus possible à cette entrée (minimisation d'une distance). Le modèle de réseau neuronal proposé par Kohonen montre des propriétés d'auto-organisation et de représentation topologique de l'espace d'entrée. Ces cartes s'organisent par rapport aux exemples présentés en respectant les contraintes topologiques de l'espace d'entrée. On trouve ainsi une mise en correspondance de l'espace d'entrée avec l'espace du réseau.

Lors de la phase d'apprentissage, le réseau s'auto-adapte, il modifie les poids de ses connexions en respectant une dérivée de la loi de Hebb qui peut être formulée par l'équation suivante :

$$w_{ki}(t+1) = w_{ki}(t) + \alpha(t).(e_k - w_{ki}(t))$$

où w_{ki} est le poids synaptique de la connexion entre le neurone k sur la carte et le neurone i en entrée, $\alpha(t)$ est un coefficient d'apprentissage et e_k représente l'entrée du neurone N_k.

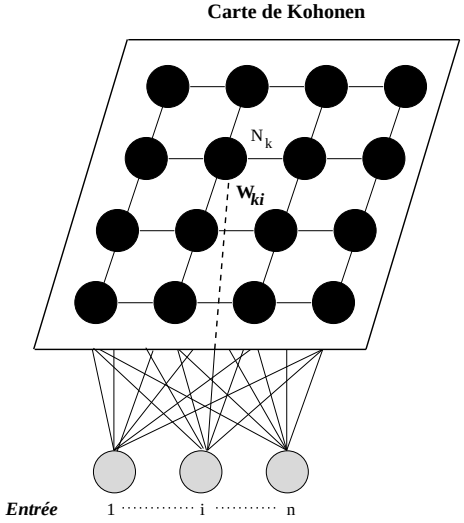

Figure 2. *Architecture d'une carte auto-organisatrice de Kohonen*

À chaque forme d'entrée correspond un neurone gagnant qui va modifier son poids et les poids de ses voisins en utilisant la loi de Hebb. La technique d'apprentissage au sein de ces réseaux est donnée par l'algorithme 2.

Dans notre travail, nous avons utilisé les cartes de Kohonen afin de visualiser la topologie des mots du lexique. Ceci nous a permis de regrouper ensemble les mots les plus proches construisant ainsi les concepts sémantiques relatifs aux applications traitées.

4.3. *Les réseaux bayésiens pour la classification*

Dans cette section, nous détaillons le principe de fonctionnement des réseaux bayésiens qui constituent une proposition originale pour l'extraction automatique des concepts sémantiques. Ces réseaux ont déjà fait leurs preuves dans plusieurs autres domaines liés au raisonnement et à l'apprentissage. Ils sont issus du mariage entre la théorie des graphes et celle des probabilités ce qui fait d'eux des outils intuitifs et naturels pour traiter les données complexes et incertaines.

Quelques réseaux bayésiens ont été conçus pour des problèmes de classification. Les plus connus, sont ceux fondés sur le modèle dit « *Naive Bayes* ». Ce dernier constitue une modélisation très simpliste du problème de classification supervisée. Plusieurs

Algorithme 2 Algorithme d'apprentissage dans une carte de Kohonen.

0- Initialisation des poids w_{ik} des K neurones de la carte à des valeurs aléatoires.
1- Présentation d'une entrée $E_l = (e_{l1}, e_{l2}, \ldots, e_{ln})$.
2- Calcul de la distance de chacun des neurones par rapport à l'entrée E_l :

$$x_k = ||E_l - w_k||$$

3- Sélection du neurone c le plus proche :

$$c = \arg \min_{k=1\ldots K} x_k$$

4- Modification des poids pour le neurone c et pour ses p plus proches voisins :

$$w_{ki}(t+1) = w_{ki}(t) + \alpha(t).(e_{lk} - w_{ki}(t))$$

5- Modification de la taille du voisinage p et du coefficient d'apprentissage $\alpha(t)$.
6- Tant que les performances sont insuffisantes :
sélection de l'exemple suivant dans la base d'apprentissage et retour à l'étape 2.

autres variantes de ce modèle ont été proposées dans la littérature. Nous allons dans le cadre de cette section décrire le réseau bayésien AutoClass qui constitue une extension des principes de *Naive Bayes* pour la classification non supervisée. En raison de son efficacité et de sa performance, nous avons employé le réseau AutoClass dans notre tâche de classification sémantique. Commençons tout d'abord par une présentation des bases théoriques des réseaux bayésiens.

4.3.1. *Principes*

Un réseau bayésien $B =< N, A, \Theta >$ est défini par un graphe dirigé et acyclique $G =< N, A >$ et une distribution de probabilité conditionnelle Θ définie pour chaque nœud du graphe. L'ensemble des nœuds N représente un jeu de variables aléatoires $N = X_1, \ldots, X_n$. La structure du graphe est représentée par l'ensemble de ses arcs dirigés A. Ces derniers encodent les dépendances conditionnelles entre les variables X_i. Ainsi, un arc allant d'un nœud X_a à un nœud X_b exprime le fait que X_b dépend directement de X_a (lien de cause à effet). L'absence d'arc ne renseigne alors que sur la non-existence d'une dépendance directe. Le jeu de probabilités Θ définit les probabilités conditionnelles locales associées à chaque variable. Pour une variable X_i donnée, sa probabilité conditionnelle locale est calculée comme étant $P(X_i|parents(X_i))$, par exemple pour la variable X_b nous associons la probabilité $P(X_b|X_a)$ s'il n'y a aucun autre lien vers X_b. Un réseau bayésien peut donc présenter deux aspects : un aspect qualitatif formulé par la structure du graphe G et qui exprime les relations entre les différentes variables du réseau et un aspect quantitatif qui permet de spécifier la distribution de probabilité conditionnelle de chaque variable et qui ainsi quantifie les relations entre les nœuds.

Afin de calculer la distribution de probabilité jointe des variables $X_1, X_2, \ldots, X_n$, nous pouvons utiliser la règle de chaîne qui permet d'obtenir la décomposition suivante :

$$\begin{aligned} P(X_1, X_2, \ldots, X_n) &= P(X_n|X_{n-1}, \ldots, X_2, X_1) \ldots P(X_2|X_1).P(X_1) \\ &= P(X_1) \prod_{i=2}^{n} P(X_i|X_{i-1}, \ldots, X_1) \end{aligned} \quad [7]$$

La structure d'un réseau bayésien permet de mieux factoriser cette probabilité en ne considérant que les probabilités $P(X_i|parents(X_i))$. Dans ce cas, nous pouvons écrire :

$$P(X_i|X_{i-1}, \ldots, X_1) = P(X_i|parents(X_i))$$

L'équation 7 devient alors :

$$P(X_1, X_2, \ldots, X_n) = \prod_i P(X_i|parents(X_i))$$

Cette formule permet de simplifier considérablement le calcul de la probabilité jointe de l'ensemble $X_1, X_2, \ldots, X_n$ en supposant que toute variable X_i ne dépend que de ses prédécesseurs directs (ses parents). En effet, la connaissance de tout autre variable constitue une information redondante et inutile.

4.3.2. *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. Il a la même structure qu'un réseau de type *Naive Bayes* (*cf.* figure 3) et il est fondé sur le théorème de Bayes :

$$P(H|D) = \frac{P(H)\ P(D|H)}{P(D)} \quad [8]$$

Dans ce type de réseau une donnée x_i est présentée sous la forme d'un vecteur à K attributs, $x_{ik},\ k \in \{1...K\}$. Une classe C_j est décrite, elle aussi, par K attributs, chacun est modélisé par une distribution gaussienne normale. $\overrightarrow{\theta_{jk}}$ est un vecteur paramètres décrivant le *k-ième* attribut de la *j-ième* classe C_j et il contient deux éléments, la moyenne μ_{jk} de la distribution considérée et son écart type σ_{jk}. Pour l'ensemble de la classe, ce vecteur est noté $\overrightarrow{\theta_j}$ et il contient les $\overrightarrow{\theta_{jk}}$ de tous ses attributs. La probabilité qu'une entrée x_i appartienne à la classe C_j est appelée *la probabilité de classe* notée π_j. Elle constitue aussi un paramètre descriptif de la classe C_j.

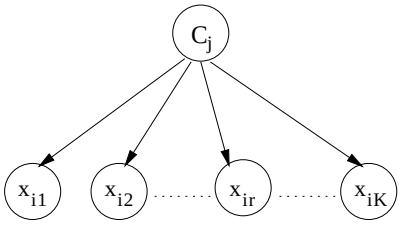

Figure 3. *La structure générale d'un réseau* Naive Bayes

Dans notre cas, nous définissons les données D comme l'ensemble des mots, représentés par un vecteur $\overrightarrow{x}$ à I éléments. L'hypothèse H correspond à la description des concepts[1], elle est représentée par trois éléments, le nombre de concepts J et les deux vecteurs $\overrightarrow{\pi}$ et $\overrightarrow{\theta}$ qui contiennent respectivement les π_j et les $\overrightarrow{\theta_j}$ de tous les concepts. En remplaçant, dans l'équation 8, D et H par leurs valeurs, on obtient :

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

où $p(\overrightarrow{\theta}, \overrightarrow{\pi} | J)$ est la probabilité *a priori* des paramètres de classification, son calcul suscite beaucoup de débats dans la littérature (*cf.* (Cheeseman et Stutz, 1996) pour une longue discussion sur le sujet). La probabilité *a priori* des mots $p(\overrightarrow{x} | J)$ peut être calculée directement. Elle est considérée simplement comme une constante de normalisation. $p(\overrightarrow{x} | \overrightarrow{\theta}, \overrightarrow{\pi}, J)$ représente la fonction de vraisemblance des données. Dans la suite on détaillera les étapes de son calcul.

On sait que $\overrightarrow{x}$ 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[2] comme le montre l'équation suivante :

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

$p(x_i | \overrightarrow{\theta}, \overrightarrow{\pi}, J)$ 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 :

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

1. On confond ici les mots « concept » et « classe » puisqu'un concept est une classe de mots.
2. Cette hypothèse est admise en considérant que les données sont mutuellement indépendantes.

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

$$p(x_i|x_i \in C_j, \overrightarrow{\theta_j}) = \prod_{k=1}^{K} p(x_{ik}|x_i \in C_j, \overrightarrow{\theta_{jk}})$$

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

$$p(x_{ik}|x_i \in C_j, \mu_{jk}, \sigma_{jk}) = \frac{1}{\sqrt{2\pi}\sigma_{jk}} \exp\left[-\frac{1}{2}\Big(\frac{x_{ik} - \mu_{jk}}{\sigma_{jk}}\Big)^2 \right] \quad [9]$$

Pour des attributs à valeurs discrètes, la distribution de classe est représentée par les ρ_{jkl} correspondants à toutes les valeurs possibles de l'attribut k. Pour l variant de 1 à L, ceci est noté :

$$\rho_{jkl} = p(x_{ik} = l|x_i \in C_j, \overrightarrow{\rho_{jk}}) \quad [10]$$

Une fois, la distribution de classe déterminée, il nous reste à chercher l'ensemble des paramètres de concepts qui maximisent la probabilité de départ $p(\overrightarrow{\theta}, \overrightarrow{\pi}|\overrightarrow{x}, J)$ et trouver ainsi l'ensemble des concepts optimaux relatifs à nos données. Pour ce faire, AutoClass utilise une variante bayésienne de l'algorithme EM (Expectation-Maximization) pour trouver les meilleurs paramètres de classes pour une valeur donnée de J. En fait, AutoClass divise le problème d'identification des concepts en deux parties : la détermination des paramètres de classification ($\overrightarrow{\pi}$ et $\overrightarrow{\theta}$) pour un nombre donné de concepts et la détermination du nombre de concepts J.

L'algorithme employé par AutoClass consiste à réestimer à chaque itération les distributions des classes définies par les équations 9 et 10. En notant w_{ij} la probabilité que l'entrée x_i appartienne à la classe C_j et W_j le poids de la classe C_j, la solution préconisée par AutoClass consiste à estimer les probabilités $\hat{\rho}_{jkl}$, dans le cas des attributs à valeurs discrètes, à :

$$\hat{\rho}_{jkl} = \frac{\sum_{i=1}^{I} w_{ij}\ \delta(l, x_{ik}) + w' - 1}{W_j + L(w' - 1)} \qquad \delta(l, x_{ik}) \equiv \begin{cases} 1, & \text{si } x_{ik} = l \\ 0, & \text{sinon} \end{cases}$$

avec

$$w_{ij} = p(x_i \in C_j|x_i, \hat{\theta}, \hat{\pi})$$

$$W_j = \sum_{i=1}^{I} w_{ij}$$

où w' représente la probabilité *a priori* qu'un attribut k appartienne à la classe C_j et L est le nombre de valeurs possibles que peut prendre l'attribut k.

Afin de déterminer le nombre de classes J, AutoClass utilise quelques approximations. Il considère qu'il est inutile d'inclure toute classe dont la probabilité π_j est négligeable. En effet, introduire une telle classe dans le modèle global va affaiblir la vraisemblance des données. De telles classes sont donc supprimées. Ainsi, AutoClass commence toujours par une valeur de J supérieure à celle proposée par l'utilisateur et il supprime ensuite toutes les classes avec des π_j faibles. Si au contraire, toutes les classes résultantes ont des probabilités significatives, alors la valeur de J est augmentée jusqu'à ce qu'on obtienne quelques classes inutiles, et ainsi de suite.

Dans notre cas, nous avons employé AutoClass pour l'extraction automatique des concepts. Il faut donc lui fournir les mots du vocabulaire représentés d'une manière vectorielle et récupérer une liste de classes en sortie. AutoClass fournit pour chaque mot ses probabilités d'appartenance aux différentes classes qu'il propose.

5. Applications traitées

Pour montrer la validité de notre approche, nous avons choisi de travailler sur deux corpus différents. Nous nous sommes donc intéressés à deux applications réelles : le dépouillement d'une liste de *bookmarks* et la consultation vocale de la bourse. Dans ce qui suit, nous décrivons ces deux applications tout en mettant l'accent sur les corpus relatifs à chacune d'elles.

5.1. *Consultation des favoris*

Il s'agit de la consultation et de l'interrogation d'une base de données musicale. On suppose qu'un utilisateur ayant consulté quelques titres musicaux les a sauvegardés comme étant ses titres favoris. La base de données développée vise à consulter cette liste de favoris en utilisant le langage naturel. Elle contient 71 287 requêtes toutes différentes exprimées en langage naturel et en langue française. Quelques exemples de ces requêtes sont donnés dans le tableau 1.

Notre but dans ce cas est de chercher les différents concepts sémantiques relatifs à cette application (désormais notée « Favoris ») d'une manière automatique. Dans une seconde étape d'étiquetage, il suffira d'attribuer à chaque mot l'étiquette relative à sa classe sémantique (concept auquel appartient le mot en question).

Après examen de toutes les requêtes présentes dans cette base de données, nous avons pu identifier manuellement treize concepts différents. Ces concepts sont représentatifs de toutes les requêtes de la base d'apprentissage. Leur petit nombre peut s'expliquer par : d'une part la base de données traitée concerne un même sujet, à savoir la consultation d'une liste de favoris, ce qui ne laisse pas un grand éventail de sens à exprimer. D'autre part, ces concepts sont assez généraux et peuvent êtres décomposés pour obtenir une liste plus importante. Cependant, cela ne nous intéresse pas car le niveau de précision que nous avons obtenu est parfaitement convenable pour une

Montre-moi le contenu de mes favoris.
Je voudrais savoir si tu peux me prendre le contenu que j'aime.
Est-il possible que tu me passes le premier de mes favoris.
Te serait-il possible de m'indiquer quelque chose de pareil.
Je me demande si tu pourrais me passer ce dernier.
Est-ce que tu pourrais me sélectionner celui-là.
Est-il possible que tu m'indiques celui d'après.
Je souhaiterais que tu me prennes le premier.
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.

Tableau 1. *Quelques exemples de requêtes du corpus Favoris*

tâche de consultation de base de données. La liste complète des concepts identifiés est donnée dans l'annexe A.

Pour mener à bien les étapes ultérieures de classification, nous avons effectué une étape de prétraitement du corpus qui consiste à éliminer les mots inutiles. Initialement ce corpus contenait 134 mots, nous n'en avons gardé que 87. Nous avons supprimé les mots outils de la langue et les mots à fréquence très faible (inférieure à 7). Cependant, nous n'avons effectué aucun traitement de lemmatisation car nous avons remarqué que chaque forme (d'un même lemme) correspond à une utilisation spécifique dans le corpus de l'application et donc à un concept différent. Par exemple, le verbe « *montrer* » est utilisé sous trois formes possibles (*montrer, montres et montre*) chacune correspondant à un concept différent (voir la liste de concepts établie manuellement dans l'annexe A).

5.2. *L'interrogation de la Bourse*

Le corpus « Bourse » est une deuxième application adoptée pour le test et la validation de notre approche. Ce corpus contient 51 864 requêtes toutes différentes exprimées en langue française. Chaque requête exprime une manière particulière d'interroger la base de données de la Bourse relative aux sociétés BNP et Alcatel. Des exemples de ces requêtes sont donnés dans le tableau 2.

De même que pour la première application, le but ici consiste à retrouver d'une manière automatique la liste des concepts relatifs à cette application. Ceci nous permettra d'effectuer une étape ultérieure d'étiquetage automatique. Les concepts identifiés manuellement avec cette base d'apprentissage sont au nombre de 11. De la même façon que pour la première application, nous pouvons expliquer le petit nombre de concepts obtenu par le fait que le corpus traité est très spécifique ce qui ne nécessite pas un grand nombre de concepts sémantiques. En plus, les concepts obtenus sont suffisamment généraux pour couvrir tout l'éventail de sens qui peuvent être exprimés dans les

Donnez-moi la hauteur du cours du groupe Alcatel.
Faxez-moi le cours de la BNP.
J'ai besoin de la progression du cours minimum de l'action 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.
Voudriez-vous me fournir la hauteur du cours du groupe Alcatel.
Je voudrais obtenir l'évolution du cours minimum d'Alcatel.
Je suis intéressé par la variation du cours moyen de la BNP.

Tableau 2. *Quelques exemples de requêtes du corpus Bourse*

requêtes de cette base d'apprentissage. La liste complète de ces concepts est donnée aussi dans l'annexe A.

De la même manière que pour l'application Favoris, nous avons commencé nos expérimentations avec le corpus Bourse en déterminant le lexique des mots significatifs. À partir des 141 mots que contenait le vocabulaire de cette application nous n'en avons gardé que 99.

6. Résultats et discussions

Cette section est consacrée à la présentation, l'évaluation et la comparaison des résultats des différentes méthodes de classification que nous avons utilisées. Par la suite, nous commençons par décrire brièvement une méthode d'extraction de concepts en utilisant la notion de *triggers*. Cette méthode est largement inspirée du travail de (Siu et Meng, 1999), nous la testons sur nos corpus afin de nous cituer par rapport aux travaux existants dans le domaine. Ensuite, nous définissons une mesure d'évaluation capable d'estimer la qualité d'un concept ou d'une liste de concepts. Enfin, nous exposons les résultats obtenus par les différentes méthodes avec les deux applications tout en comparant les performances réalisées.

6.1. *Les* triggers *pour la construction des concepts*

Dans ce travail, nous avons commencé par tester une méthode inspirée des travaux de (Siu et Meng, 1999) pour la construction de concepts sémantiques. Nous cherchons donc à extraire la liste des *triggers* d'un corpus donné et pour ce faire nous avons procédé en trois étapes. En premier lieu, nous avons construit le lexique de l'application. Ensuite, nous avons calculé la distance sémantique entre tous les couples des mots possibles de ce lexique. Enfin, nous n'avons gardé que les couples de mots sémantiquement proches. En utilisant la distance de Kullback-Leibler, nous ne gardons que

les couples de mots ayant une valeur de distance inférieure à un seuil fixé. Ainsi, nous obtenons pour chaque mot son ensemble de *triggers*, c'est-à-dire les mots qui lui sont fortement corrélés d'un point de vue sémantique.

Les *triggers* ainsi construits vont aider à extraire l'ensemble des concepts relatifs à l'application considérée. Notre méthode consiste à grouper ensemble tous les *triggers* complètement interconnectés. Autrement dit, si le couple de mots (A, B) est admis comme *trigger* et que (B, C) est un autre *trigger*, on ne peut construire le concept $\{A, B, C\}$ que si on a obtenu aussi le *trigger* (A, C). Et ainsi de suite jusqu'à former l'ensemble de tous les concepts relatifs à notre corpus d'apprentissage. À l'issue de cette étape nous avons été amenés à effectuer une légère étape manuelle pour éliminer ou déplacer quelques mots « parasites ».

Les listes de concepts obtenus sont données dans les tableaux 8 et 7 de l'annexe B. En analysant les résultats obtenus par cette méthode, nous pouvons remarquer que les mots formant un seul concept sont très corrélés. Ceci prouve que la distance sémantique utilisée est efficace. Cependant les concepts obtenus sont peu nombreux et ils ne couvrent pas la totalité du vocabulaire de l'application. Avec l'application Favoris, nous avons obtenu des concepts qui couvrent le vocabulaire à 44 % et à 48 % pour l'application Bourse. En conclusion, les concepts obtenus ne sont pas assez représentatifs des applications en question. Ils ne peuvent pas fournir une représentation sémantique complète et fidèle des requêtes présentes dans les corpus et ils sont donc insuffisants pour réaliser l'étape de compréhension. C'est pour ces raisons que nous proposons l'utilisation des méthodes de classification non supervisée pour extraire des concepts de meilleure qualité. Dans la section suivante nous introduisons une méthode d'évaluation de la qualité des concepts obtenus pour pouvoir comparer les performances achevées par les différentes méthodes de classification.

6.2. *L'efficacité pour l'évaluation des concepts*

L'évaluation des concepts sémantiques d'une manière totalement objective est une tâche très délicate. En effet, bien qu'un concept puisse apparaître cohérent et adéquat à l'application traitée, il peut être considéré comme partiellement faux si par exemple il ne contient pas tous les mots qui se rapportent au sens qu'il exprime. De la même manière, un concept regroupant plusieurs mots peut être vu comme un concept très général et par d'autres comme incohérent, etc. Quantifier ces avis et proposer une méthode d'évaluation objective constitue une difficulté majeure dans le domaine de l'extraction automatique des concepts.

Dans notre cas, nous avons choisi d'utiliser une version adaptée de la mesure d'efficacité puisque c'est la mesure la plus utilisée dans ce cas de figure (Rosenberg et Hirschberg, 2007). Cette méthode d'évaluation est fondée sur l'idée qu'une classe sémantique peut être vue comme le résultat de l'étiquetage d'une requête. Ce résultat sera

comparé à un étiquetage de référence, que nous assimilons dans ce cas à un concept de référence. Les mesures de rappel et de précision seront donc définies comme suit :

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

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

où n_{ij} est le nombre des mots présents à la fois dans le concept de référence C_i et dans le concept résultat C_j. N_i et N_j représentent respectivement le nombre total des mots dans les concepts C_i et C_j. Ainsi l'efficacité sera donnée par :

$$F(i,j) = \frac{2 \times précision(i,j) \times rappel(i,j)}{précision(i,j) + rappel(i,j)}$$

Or, $F(i,j)$ n'est relative qu'à un seul cas, celui de l'adéquation du concept C_j avec le concept C_i. Pour représenter tout le concept de référence C_i, il suffit de choisir l'efficacité maximale obtenue avec ce dernier :

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

L'efficacité globale de toute la liste des concepts sera enfin calculée de la manière suivante :

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

où p_i est un facteur de pondération qui représente le poids du concept C_i en divisant N_i par le nombre total des mots dans tous les concepts de référence. Il est donné par :

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

Dans la suite, nous utilisons la mesure d'efficacité F ainsi définie pour évaluer et comparer les concepts obtenus par les différentes méthodes de classification. Les listes des concepts de référence relatives à chaque application ont été établies manuellement. Elles sont données dans l'annexe A.

6.3. *Résultats expérimentaux*

Dans cette section nous présentons les résultats obtenus avec les deux applications traitées en faisant varier :

– la méthode de classification utilisée : comme décrit précédemment, nous avons comparé trois méthodes de classification non supervisée : l'algorithme des *K-means*, les cartes de Kohonen et le réseau bayésien AutoClass ;

– la représentation vectorielle adoptée : trois types de représentations vectorielles seront expérimentés. D'une part, la représentation contextuelle avec la méthode AutoClass et, d'autre part, les deux représentations vectorielles (simple et mixte) basées sur les distances sémantiques. Ces dernières seront employées avec toutes les méthodes de classification ;

– la distance sémantique considérée : nous utilisons dans nos expériences la distance de Kullback-Leibler (notée DKL) et l'information mutuelle moyenne (notée IMM) pour calculer les différentes représentations vectorielles des mots.

En conclusion, nous considérons trois méthodes de classification et cinq représentations vectorielles des mots qui sont : la représentation contextuelle, la représentation vectorielle simple utilisant la DKL et celle utilisant l'IMM et la représentation matricielle mixte utilisant la DKL et celle utilisant l'IMM. Dans la suite, nous décrivons les choix expérimentaux adoptés avec chacune des méthodes de classification employées. Enfin, nous présentons un tableau comparatif pour donner une synthèse des performances obtenues par chaque méthode et par chaque représentation vectorielle utilisée.

6.3.1. *L'algorithme des* K-means

En utilisant cet algorithme, nous sommes amenés à préciser un nombre de classes initial. Pour ce faire, nous avons effectué une série de tests et nous avons constaté que les meilleurs résultats sont obtenus en utilisant 20 noyaux de classes avec la première application Favoris et 15 avec l'application Bourse. Notons cependant qu'en fixant à N le nombre de classes nous n'obtenons pas forcément N concepts car quelques classes peuvent être vides. De plus, il y a des classes qui ne contiennent qu'un seul mot. De tels concepts sémantiques ne sont pas très significatifs.

En étudiant les concepts obtenus, nous pouvons remarquer que, à part les classes singletons, les autres classes sont sémantiquement significatives. En les comparant avec les concepts de référence, nous constatons que quelques concepts ont été divisés en deux ou en trois classes.

6.3.2. *Les cartes auto-organisatrices de Kohonen*

De même pour les cartes de Kohonen, nous avons réalisé plusieurs tests pour pouvoir décider des meilleures initialisations possibles de la carte. Les expériences nous ont montré que la meilleure carte à employer dans le cas de l'application Favoris est une carte de 20×40 neurones et pour l'application Bourse la meilleure carte est de 30×30 neurones. De telles cartes nous permettent d'obtenir une séparation optimale entre les différents mots du lexique.

La visualisation de la dispersion des mots sur la carte de Kohonen nous a permis de construire nos classes sémantiques révélées par des groupements de mots sur la carte. Parfois, ces groupements de mots ne sont représentés que par un seul neurone. Dans d'autres cas, les mots d'un même groupe sont représentés par un ensemble de

neurones contigus. En revanche, nous avons aussi trouvé plusieurs mots isolés sur la carte. Il s'agit là aussi de classes singletons.

6.3.3. *Le réseau bayésien AutoClass*

Avec AutoClass, nous avons fixé le nombre maximal de classes à 20. Avec les représentations vectorielles à base de DKL et d'IMM, le réseau a proposé des listes de concepts contenant entre 12 et 18 concepts pour l'application Favoris et entre 11 et 15 concepts pour l'application Bourse. Notons que AutoClass arrive à déterminer automatiquement le nombre de classes à chaque fois mais demande aussi de fixer un nombre de calsses maximal comme pour les deux méthodes précédentes.

La représentation contextuelle, quant à elle, a nécessité de fixer le nombre maximal de classes à 30 pour donner les meilleurs résultats. Après un temps d'apprentissage très long (250 000 données et 300 itérations), AutoClass a trouvé 26 classes sémantiques différentes pour l'application Bourse avec une efficacité de l'ordre de $74,5\ \%$, et 30 classes pour l'application Favoris avec une efficacité de l'ordre de $76\ \%$. Ces classes sont très volumineuses, elles se chevauchent et contiennent plusieurs répétitions de mots. Ce résultat est le moins bon obtenu par AutoClass.

Afin d'extraire les concepts les plus vraisemblables, nous n'avons considéré pour chaque mot w_i que la classe C_j qui maximise la probabilité $P(w_i|C_j)$ donnée par AutoClass. En examinant les listes de concepts proposées par AutoClass (*cf.* annexe C), nous constatons que nous n'avons plus de classes singletons et que les classes obtenues sont cohérentes et bien représentatives des différents sens exprimés dans les corpus d'apprentissage.

6.3.4. *Discussion*

Le tableau 3 résume les résultats obtenus par les différentes méthodes de classification avec l'application Favoris en adoptant deux types de représentations vectorielles des mots et en utilisant les distances sémantiques DKL et IMM. Ces résultats sont exprimés en termes d'efficacité globale calculée sur la totalité des concepts obtenus. Pour chaque valeur, nous précisons son intervalle de confiance calculé à $95\ \%$ selon (Rothman et Yankauer, 1986).

		K-means	**Kohonen**	**AutoClass**
Représentation vectorielle simple	DKL	70,5 $\pm$ 2,2	75,5 $\pm$ 2,1	81,3 $\pm$ 1,9
	IMM	74,1 $\pm$ 2,1	77,1 $\pm$ 2,1	84,7 $\pm$ 1,8
Représentation matricielle mixte	DKL	72,5 $\pm$ 2,2	76,7 $\pm$ 2,1	86,3 $\pm$ 1,7
	IMM	76,4 $\pm$ 2,1	80,4 $\pm$ 1,9	89,3 $\pm$ 1,5

Tableau 3. *Résultats, en pourcentage, relatifs à l'application Favoris obtenus par les différentes méthodes de classification en utilisant quatre types de représentations vectorielles*

Le tableau comparatif 3 montre que les meilleures performances sont obtenues par le réseau bayésien AutoClass et que la meilleure représentation vectorielle est celle qui représente un mot par une matrice de trois colonnes représentant le mot et ses contextes gauche et droit. Nous pouvons aussi constater que l'IMM fournit toujours de meilleurs résultats que la DKL. Par ailleurs, nous remarquons que la méthode qui donne les meilleurs résultats, parmi les deux méthodes de partitionnement testées, est celle des cartes de Kohonen. Ce résultat peut être expliqué par le fait que cette dernière offre une visualisation de la topologie des mots qui aide à construire des classes plus intéressantes.

En conclusion, les meilleurs concepts obtenus avec le corpus Favoris sont ceux donnés par AutoClass en utilisant la représentation matricielle mixte avec la distance sémantique IMM. Ces concepts ont une efficacité de $89,3\,\%$ par rapport aux concepts de référence. De plus, ils sont cohérents et très représentatifs.

Le tableau 4 résume les résultats obtenus par les différentes méthodes de classification en utilisant les quatre types de représentations vectorielles des mots avec l'application Bourse. Ce tableau indique que les cartes de Kohonen donnent toujours de meilleurs résultats que l'algorithme des *K-means*. Il confirme aussi que l'IMM est plus efficace que la DKL pour l'extraction automatique des concepts. Enfin, il montre que la représentation matricielle, bien qu'un peu compliquée, fournit toujours de meilleures performances que la représentation vectorielle simple.

Ces résultats ne peuvent qu'appuyer le choix de l'ajout de l'information contextuelle dans la représentation des mots. Dans notre travail, nous n'avons considéré que les mots se trouvant dans le contexte très proche du mot (premier mot à gauche et premier mot à droite). Nous pouvons aussi proposer d'étendre encore cette matrice pour prendre en compte différentes longueurs de contextes (de deux mots à la phrase entière). La seule limitation quant à la réalisation de cette proposition est la taille mémoire nécessaire. En effet, pour des applications à vocabulaires étendus ne nous pouvons pas envisager l'utilisation d'une matrice de grande taille pour représenter chaque mot du vocabulaire. Dans notre cas, les deux applications traitées ont un vocabulaire limité qui ne dépasse pas les 100 mots. La représentation matricielle avec trois colonnes a été tout à fait réalisable sans aucun souci de taille mémoire.

		K-means	**Kohonen**	**AutoClass**
Représentation vectorielle simple	DKL	62,5 $\pm$ 2,4	72,2 $\pm$ 2,2	79,8 $\pm$ 2,0
	IMM	63,2 $\pm$ 2,4	75,3 $\pm$ 2,1	82,0 $\pm$ 1,9
Représentation matricielle mixte	DKL	65,6 $\pm$ 2,3	74,5 $\pm$ 2,1	83,6 $\pm$ 1,8
	IMM	67,7 $\pm$ 2,3	77,1 $\pm$ 2,1	86,3 $\pm$ 1,7

Tableau 4. *Résultats, en pourcentage, relatifs à l'application Bourse obtenus par les différentes méthodes de classification en utilisant les quatre types de représentations vectorielles*

Signalons enfin que, compte tenu des résultats relatifs aux deux applications traitées Favoris et Bourse, les meilleurs concepts sont toujours obtenus par la méthode AutoClass. L'efficacité des listes de concepts proposées dans ce cas atteint respectivement les $89,3$ % et $86,3$ % en les comparant à des listes de concepts établies manuellement. Ceci montre l'intérêt de la classification bayésienne pour l'extraction automatique des concepts sémantiques.

7. Conclusion

Ce travail propose une contribution dans le domaine de l'extraction automatique des concepts sémantiques. Tout d'abord nous avons introduit quelques mesures qui peuvent servir pour calculer des distances sémantiques entre les mots. Ensuite, nous avons proposé l'utilisation de quelques méthodes de classification non supervisée pour regrouper efficacement les mots sémantiquement proches. Nous avons choisi donc d'utiliser les méthodes de partitionnement les plus connues : l'algorithme des *K-means* et les cartes auto-organisatrices de Kohonen. Ensuite, nous avons proposé de profiter des capacités de l'approche bayésienne pour la classification et ce, en utilisant un réseau bayésien pour la classification non supervisée, AutoClass. Ce dernier a permis en effet, d'atteindre de meilleurs résultats que les deux autres méthodes de partitionnement.

Afin d'utiliser les méthodes de classification mentionnées, nous avons été amenés à chercher des représentations numériques valides des mots. Nous avons donc exposé deux méthodes originales pour la représentation vectorielle des mots. Ces méthodes sont fondées sur les distances sémantiques. L'idée est de représenter chaque mot par un vecteur contenant ses valeurs de similarité avec tous les autres mots du lexique. Utilisées avec les différentes méthodes de classification, ces représentations ont prouvé leur efficacité quant à la représentation sémantique des mots.

Les résultats expérimentaux sont encourageants et montrent l'intérêt de l'idée d'extraction automatique des concepts dans des contextes de compréhension de requêtes exprimées en langage naturel. Ils nous incitent donc à limiter l'intervention humaine dans des tâches lourdes et fastidieuses telles que celle de l'identification des concepts. En effet, les résultats obtenus, et plus particulièrement ceux obtenus par AutoClass, montrent que les concepts trouvés sont cohérents, valides et très significatifs. Ils sont comparables à ceux établis manuellement.

Dans un futur proche, nous envisageons de tester notre approche d'extraction automatique des concepts dans le cadre de corpus plus généraux et avec des vocabulaires plus importants où les concepts sémantiques peuvent se chevaucher entre eux. Ceci nous amènera aussi à essayer des techniques hiérarchiques pour la construction des concepts sémantiques. Essayer d'adapter AutoClass pour ce genre de classification sera l'un des axes à suivre. Nous nous intéressons aussi à la recherche d'autres représentations vectorielles des mots qui soient aussi performantes que celles proposées

dans cet article et moins coûteuses à calculer. Compacter ces représentations constitue un enjeu important mais difficile car il risque d'engendrer une perte d'informations. Nous envisageons à plus long terme de trouver des techniques incrémentales pour la construction des concepts.

8. Bibliographie

Baggia P., Kellner A., Pérennou G., Popovici C., Sturm J., Wessel F., « Language Modeling and Spoken Dialogue Systems - the ARISE Experience », *Proceedings of the European Conference on Speech Communication and Technology*, p. 1767-1770, 1999.

Benayed Y., Fohr D., Haton J. P., Chollet G., « Confidence Measures for Keyword Spotting Using Support Vector Machines », *International Conference on Acoustics, Speech and Signal Processing*, p. 588-591, 2003.

Bigi B., « Using Kullback-Leibler Distance for Text Categorization », *Lecture Notes in Computer Science, Advances in Information Retrieval*, vol. Proceedings of the 25th European Conference on Information Retrieval Research, p. 305-319, 2003.

Bourigault D., Fabre C., Frérot C., Jacques M.-P., Ozdowska S., « Syntex, analyseur syntaxique de corpus », *Actes des 12^{e} journées sur le Traitement Automatique des Langues Naturelles*, Dourdan, France, 2005.

Bousquet-Vernhettes C., Vigouroux N., Pérennou G., « Stochastic Conceptual Model for Spoken Language Understeting », *International Workshop on Speech and Computer*, p. 71-74, 1999.

Buckley C., Salton G., Allan J., « Automatic Retrieval with Locality Information using SMART », *TREC-1 Proceedings*, p. 69-72, 1992.

Cheeseman P., Stutz J., « Bayesian Classification (AutoClass) : Theory and Results », *in* P. S. U. M. Fayyad, G. Piatetsky-Shapiro, R. Uthurusamy (eds), *Advances in Knowledge Discovery and Data Mining*, AAAI/MIT Press, p. 153-180, 1996.

Dagan I., Lillian L., Fernando C. N., « Similarity-Based Models of Word Coocurrence Probabilities », *Machine Learning*, vol. 34, p. 43-69, 1999.

Elman J. L., « Finding structure in time », *Cognitive Science*, vol. 14, p. 179-211, 1990.

Honkela T., Kaski S., Lagus K., Kohonen T., « WEBSOM self-organizing maps of document collections », *Proceedings of the workshop on Self-Organizing Maps*, p. 310-315, 1997.

Jamoussi S., Smaïli K., Haton J.-P., « A complete understanding speech system based on semantic concepts », *Fourth International Conference on Language Ressources and Evaluations (LREC'04)*, Lisbon, Portugal, 2004.

Kohonen T., *Self-Organization and Associative Memory*, 3ème edn, Spring-Verlag, 1989.

Kullback S., *Information theory and statistics*, John Wiley and Sons, NY, 1959.

Laskri M. T., Meftouh K., « Extraction automatique du sens d'une phrase en langue Française par une approche neuronale », *Journées internationales d'Analyse statistiques des Donneées Textuelles*, p. 413-422, 2002.

Maynard H., Lefèvre F., « Apprentissage d'un module stochastique de compréhension de la parole », *Journées d'Etude sur la parole*, p. 129-132, 2002.

McQueen J., « Some methods for classifcation and analysis of multivariate observations », *Proceedings of the Berkeley Symposium on Mathmatical Statistics and Probabilty*, p. 281-297, 1967.

Miikkulainen R., Dyer M. G., « Natural Language Processing With Modular PDP Networks and Distributed Lexicon », *Cognitive Science*, vol. 15, n° 3, p. 343-399, 1991.

Minker W., « Stochastically-Based Natural Language Understanding Across Tasks and Languages », *Proceedings of the European Conference on Speech Communication and Technology*, p. 1423-1426, 1997.

Nenadic G., Spasic I., Ananiadou S., « Automatic Discovery of Term Similarities Using Pattern Mining », *The second international workshop on computational terminology (COMPUTERM'02)*, 2002.

Pieraccini R., Tzoukermann E., Gorelov Z., Gauvain J. L., Levin E., Lee C. H., Wilpon J. G., « A Speech Understeting System Based on Statistical Representation of Semantics », *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, vol. 1, p. 193-196, 1992.

Rosenberg A., Hirschberg J., « V-Measure : A Conditional Entropy-Based External Cluster Evaluation Measure », *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 2007.

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

Rothman K. J., Yankauer A., « Confidence intervals vs significance tests : quantitative interpretation », *American Journal on Public Health*, vol. 75, p. 587-588, 1986.

Salton G., Buckley C., « Term weighting approaches in automatic text retrieval », *Information Processing and Management*, vol. 5, n° 24, p. 513-523, 1988.

Singhal A., Buckley C., Mitra M., « Pivoted Document Length Normalization », *ACM SIGIR*, p. 21-29, 1996.

Siu K. C., Meng H. M., « Semi-automatic acquisition of domain-specific semantic structures », *Proceedings of the European Conference on Speech Communication and Technology*, p. 2039-2042, 1999.

A. Concepts établis manuellement

Concept	Liste des mots formant les concepts
Souhait	possible, demande, souhaiterais, souhaite, faut, désire, désirerais, pourrais, voudrais, aimerais, veux, peux
Demande	montrer, indiquer, sélectionner, trouver, donner, afficher, présenter, prendre, passer, chercher, voir, connaître, savoir
Action	montres, indiques, sélectionnes, trouves, donnes, affiches, présentes, prennes, passes, cherches, fasses
Ordre	montre, indique, sélectionne, trouve, affiche, présente, prends, cherche, donne, fais
Favoris	favoris, choisi, apprécié, adoré, préféré, aimé
Objet	disques, titres, chansons, contenu
Mode	écouté, vu, regardé, utilisé
Liste	liste, inventaire, relevé
Période	matin, matinée
Date	décembre, 2001, fin, 2002
Similarité	similaire, semblable, pareil, équivalent, ressemblant, synonyme, proche, identique, rapproché
Limites	seulement, uniquement, exclusivement, simplement
Désignation	premier, deuxième, dernier

Tableau 5. *Concepts relatifs à l'application de consultation des favoris*

Concept	Liste des mots formant les concepts
Souhait	ambitionne, aspire, brûle, cherche, projette, peux, pourrais, veux, faut, envie, demande, désirerais, espoir, intérêt, souhait, aimerais, souhaiterais, voudrais, besoin, intéresse, aime, souhaite, voulu, espère, désire
Demande	obtenir, donner, fournir, procurer, récupérer, concernant, dus, réellement, vraiment, entendre, dises, disiez, obtenez, obtiens, dis, dites, donne, donnez, fais, faites, fournis, fournissez, connaître, donnes, donniez, savoir, faxe, faxez, maile, mailez
Politesse	pourriez, pouvez, voudriez, voulez
Action	percevoir, regarder, visualiser, voir, obtiennes, obteniez, attraper, choper, glaner, pêcher, récolter, apprenne, obtienne, sache
Valeur	hauteur, cours, montant, niveau, valeur, dernier, premier
Informations	informations, renseignements
Cours	groupe, titre, société, action
Nom	Alcatel, BNP
Critère	plus bas, plus haut, maximum, minimum, moyen
Évolution	évolution, progression, variation, limites
Envoi	fax, mail

Tableau 6. *Concepts relatifs à l'application d'interrogation de la Bourse*

B. Concepts trouvés par la méthode des *triggers*

Étiquette	**Liste des mots formant les concepts**
Action	affiches, cherches, trouves, donnes, fasses, passes, prennes, présentes, sélectionnes, montres, indiques
Demande	demande, peux, pourrais, aimerais, souhaiterais, voudrais, désirerais, désire, faut, souhaite, savoir
Critères	matin, matinée, écouté, utilisé, vu, regardé
Objet	contenu, favoris, préférés, liste, disques, chansons
Limites	exclusivement, seulement, simplement, uniquement

Tableau 7. *Liste des concepts relative à l'application Favoris*

Étiquette	**Liste des mots formant les concepts**
Ordre	apprenne, cherche, dus, obtienne, sache, dites, donne, donnez, faites, fais, faxe, faxez, fournis, fournissez, maile, mailez, obtenez, obtiens, projette, dis, sache, dises, disiez
Demande	aime, intéressé, désire, désirerais, espère, espoir, souhait, voulu, ambitionne, aspire, brûle, vraiment, réellement
Informations	valeur, cours, hauteur, dernier, maximum, minimum, montant, moyen, niveau, plus bas, plus haut, premier

Tableau 8. *Liste des concepts relative à l'application Bourse*

C. Meilleurs concepts obtenus automatiquement

Concept	Liste des mots formant les concepts
Concept n° 1 **Souhait**	possible, demande, souhaiterais, souhaite, faut, désire, désirerais, pourrais, voudrais, aimerais, veux, peux
Concept n° 2 **Politesse**	montrer, indiquer, sélectionner, trouver, donner, afficher, présenter, prendre, passer, chercher, voir, connaître, savoir
Concept n° 3 **Ordre**	montre, indique, sélectionne, trouve, affiche, présente, prends, cherche, donne, fais
Concept n° 4 **Action**	montres, indiques, sélectionnes, trouves, donnes, affiches, présentes, prennes, passes, cherches, fasses
Concept n° 5 **Objet**	disques, titres, chansons, contenu, premier, deuxième, dernier, liste, inventaire, relevé
Concept n° 6 **Utilisé**	écouté, vu, regardé, utilisé
Concept n° 7 **Favoris**	favoris, choisi, préférés
Concept n° 8 **Aimé**	apprécié, adoré, préféré, aimé
Concept n° 9 **Adverbe**	seulement, uniquement, exclusivement, simplement
Concept n° 10 **Date**	décembre, 2001, fin, 2002
Concept n° 11 **Période**	matin, matinée
Concept n° 12 **Similartié**	similaire, semblable, pareil, équivalent, ressemblant, synonyme, proche, identique, rapproché

Tableau 9. *Application de consultation des favoris : concepts obtenus avec AutoClass en utilisant la représentation matricielle mixte et la mesure d'IMM*

Concept	Liste des mots formant les concepts
Concept n° 1 **Politesse**	pourriez, pouvez, voudriez, voulez
Concept n° 2 **Souhait**	ambitionne, aspire, brûle, cherche, projette, peux, pourrais, veux, faut, envie, demande, désirerais, espoir, intérêt, souhait, aimerais, souhaiterais, voudrais, besoin, intéresse, aime, souhaite, voulu, espère, désire, obtenez, obtiens, dis
Concept n° 3 **Faire**	donnez, fais, faites, fournis, fournissez, connaître, donnes, donniez, savoir, faxe, faxez, maile, mailez, dites, donne
Concept n° 4 **Ordre**	obtenir, donner, fournir, procurer, récupérer, concernant, dus, réellement, vraiment, entendre, dises, disiez
Concept n° 5 **Action**	percevoir, regarder, visualiser, voir, obtiennes, obteniez, attraper, chopper, glaner, pêcher, récolter, apprenne, obtienne, sache
Concept n° 6 **Envoie**	fax, mail
Concept n° 7 **Informations**	informations, renseignements, hauteur, cours, montant, niveau, valeur, dernier, premier
Concept n° 8 **Valeur**	plus bas, plus haut, maximum, minimum, moyen
Concept n° 9 **Cours**	groupe, titre, sociéte, action
Concept n° 10 **Nom**	Alcatel, BNP
Concept n° 11 **Évolution**	évolution, progression, variation, limites

Tableau 10. *Application de l'interrogation de la Bourse : concepts obtenus avec AutoClass en utilisant la représentation matricielle mixte et la mesure d'*IMM