Une approche de représentation de l'information en RI basée sur les sous- arbres

Mustapha Baziz, Mohand Boughanem, Henri Prade

Une approche de représentation de l'information en RI basée sur les sous-arbres

Mustapha Baziz*, Mohand Boughanem*, Henri Prade*

** IRIT, Campus universitaire ToulouseIII*
118 rte de Narbonne,
F-31062 Toulouse Cedex 4, France
{baziz, boughane, prade}@irit.fr

RÉSUMÉ. *Ce papier propose une approche de recherche d'information basée sur l'utilisation d'une structure conceptuelle pour indexer les documents. La structure conceptuelle est hiérarchique. Elle est représentée par un sous-arbre pondéré. Un sous-arbre est obtenu d'abord en projetant document et requête sur une ressource conceptuelle externe, puis en appliquant une méthode de complétion via des nœuds intermédiaires extraits de cette ressource en vue d'avoir une représentation hiérarchique. Dans cette approche, l'évaluation des requêtes se fait par la comparaison entre le sous-arbre de la requête et celui correspondant à chaque document dans la collection. La similarité document-requête est basée sur le calcul d'un degré d'inclusion multi-valuée traduisant jusqu'à quel point le sous-arbre de la requête est inclus dans celui du document. Différentes méthodes d'inclusion sont évaluées sur la base de leur sémantique respective. L'approche que nous proposons généralise l'approche de recherche d'information basée sur la logique floue, une évaluation sur une collection de test est également présentée.*

ABSTRACT. *The paper proposes an approach to information retrieval based on the use of a conceptual structure both for indexing document and expressing user queries. The conceptual structure is hierarchical and it is formally represented as a weighted tree. In this approach, the evaluation of queries is based on the comparison of minimal sub-trees containing the two sets of nodes corresponding to the concepts expressed in the document and the query respectively. The comparison is based on the computation of a multiple-valued degree of inclusion. Some candidate implications are discussed on the basis of their respective semantics. The proposed approach generalizes standard fuzzy information retrieval and its evaluation on benchmark example is also presented.*

MOTS-CLÉS : *Recherche d'Information, Représentation Sémantique de documents, sous-arbres, indexation conceptuelle, ontologies, WordNet.*

KEYWORDS: *Information Retrieval, Semantic Representation of documents, sub-trees, Conceptual indexing, ontologies, WordNet.*

1. Introduction

Un objectif majeur de la recherche d'information est de retrouver des documents traitant du thème exprimé dans une requête utilisateur (Rijsbergen, 1979). Or, la majorité des systèmes existants ont été implémentés de sorte à retrouver les documents ayant les mêmes mots que ceux de la requête. Ces systèmes, souvent basés sur la notion de « sac de mots », font "implicitement" l'hypothèse qu'il ya une correspondance stricte entre les mots et les sens. Ils ne traitent donc pas des problèmes connus des linguistes, notamment la polysémie et la synonymie. Ce décalage entre l'objectif et la méthode a poussé des chercheurs à proposer de nouvelles approches tentant d'utiliser la sémantique en RI. Ces approches se basent souvent sur des ressources sémantiques externes telles les ontologies et les hiérarchies de concepts. Dans ces approches, des groupes nominaux sont projetés sur les concepts qu'ils dénotent (Cucchiarelli *et al.*, 2004, Khan *et al.,* 2002). Un concept[1] correspond souvent à un nœud de l'ontologie ou de la ressource sémantique utilisée et est représenté par un ou plusieurs termes définis de manière non ambigüe. Par ce modèle, un document est représenté par un ensemble de concepts. Une structure sémantique peut être représentée en utilisant différentes structures de données : arbres, réseaux sémantiques, graphes conceptuels, etc. Ces structures peuvent être des dictionnaires, thesaurus ou ontologies (Mihalcea *et al.*, 2000). Elles sont soit manuellement ou automatiquement générées (Maedche *et al.*, 2000) ou alors, elles peuvent pré-exister. WordNet et EuroWordNet sont des exemples de thesaurus basés sur des bases de données lexicales. Elles sont assimilées à des ontologies et largement utilisés pour améliorer l'efficacité des Systèmes de RI. Gonzalo et ses collègues (1998), proposent une méthode d'indexation basée sur les synsets de WordNet: le modèle d'espace vectoriel est utilisé, en prenant les synsets comme espace d'indexation au lieu des mots clés. Les auteurs constatent une amélioration de +29% lorsque les synsets de WordNet sont utilisés comme espace d'indexation à la place du modèle classique basé sur les mots simples. Dans (Cucchiarelli *et al.*, 2004), les auteurs proposent dans leur système (OntoLearn) une méthode appelée *structural semantic interconnection* pour désambiguïser les mots dans le texte via les glossaires de WordNet (les définitions des concepts). Cette technique est aussi utilisée pour l'expansion de requêtes.

La similarité entre concepts dans les réseaux sémantiques a fait l'objet de divers travaux, différentes métriques et méthodes ont été proposées dans la littérature. Les principales sont de (Hirst *et al.*, 1998, Resnik, 1999, Banerjee *et al.*, 2002). Resnik (1999) utilise la sous hiérarchie de WordNet formée par la relation is-a (généralisation) pour calculer la similarité entre deux concepts donnés. La mesure qu'il propose est basée sur le plus spécifique des concepts (synset de WordNet) qui subsume deux concepts à comparer. Banerjee (2002) utilise l'algorithme de Lesk

[1] Dans notre cas, un concept est représenté par un Synsets de WordNet qui est un ensemble de termes synonymes dans un contexte donné auquel est rajouté une définition plus un ou plusieurs exemples du monde réel.

(1986) adapté, basé sur le recouvrement des glossaires de WordNet pour désambiguïser les mots dans le texte (Word Sense Disambiguation ou WSD). La plupart de ces mesures sont utilisées pour la désambiguïsation de mots dans le texte. Dans notre cas, la finalité n'est pas spécialement le WSD — pour le lecteur intéressé par ces méthodes, un état de l'art complet peut être trouvé dans (Budanitsky *et al.*, 2001) sur l'utilisation des mesures et (Sanderson, 2000) sur l'application du WSD en RI — cependant nous l'utilisons comme une étape (projection d'un texte sur l'ontologie) pour détecter dans un document/requête, les entrées de l'ontologie (que nous appelons ici concepts ou nœuds) et qui sont le point de départ de l'approche que nous décrivons dans ce papier. Cette méthode de projection est décrite dans (Baziz *et al*., 2005).

Nous proposons dans ce papier une nouvelle approche de RI basée sur les concepts avec une proposition originale pour la représentation des documents et des requêtes. Cette représentation se base sur une structure de sous-arbre. Un sous-arbre est le résultat de la projection d'un document/requête sur une ressource conceptuelle externe (ontologie, hiérarchie de concept, thesaurus, etc.) où les nœuds représentent les concepts de la ressource externe qui sont identifiés dans le document/requête et les arcs le lien de subsomption (est-un).

L'organisation de l'article est structurée comme suit : nous commençons par donner une vue globale de l'approche (section 2) en énumérant les différentes étapes. Nous passons ensuite à la description des détails (section 3) en commençant par décrire les modèles de représentation conceptuelle adoptés, d'abord, par représentation séparée (sections 3.1) puis commune de la requête et du document (3.2), puis nous passons au modèle de recherche utilisant un appariement basé sur des opérateurs flous (section 3.3). Dans la section 3.4, une méthode d'élargissement de la représentation de la requête et du document par rapport à un référentiel commun est donnée, puis, un exemple est développé en section 3.5. Nous proposons par la suite, une amélioration de l'approche (par simplification du modèle) pour passer outre les contraintes qui peuvent le rendre incalculable en pratique (section 3.6). L'évaluation de l'approche avec une discussion des résultats obtenus sont données en section 4. Enfin une conclusion et des perspectives sont données en section 5.

2. Vue globale de l'approche

Le principe de l'approche est de projeter la requête et le document sur une ressource sémantique externe ayant une structure hiérarchique. Pour cela, nous utilisons le sous ensemble de WordNet[2] constitué de la relation de subsomption (*IS-A*) que nous avons représentée dans notre cas par le lien d'Hypéronymie.

[2] WordNet n'est pas à proprement parler une ontologie étant donné qu'elle ne contient pas de couche formelle (déduction). Elle est qualifiée d'ontologie « LightWeigh » ou légère.

Ceci est motivé par le fait qu'il puisse exister des ressources sémantiques où seule la subsomption est utilisée. C'est le cas notamment des taxonomies ou encore, par rapport au web, dans la façon dont les catégories sont organisées dans *Yahoo!*. Dans une arborescence, les concepts les plus génériques sont représentés dans la partie supérieure de la hiérarchie et les concepts les plus spécifiques en bas de la hiérarchie (feuilles de l'arbre). Un concept c_1 subsume un autre concept c_2, si la notion à laquelle le concept c_1 renvoie, encapsule ou englobe celle de c_2. Le concept c_2 est alors un cas particulier du concept c_1. Le nœud *racine* est un nœud particulier. Il désigne le nœud qui subsume tous les autres nœuds de l'arbre et lui-même n'est subsumé par aucun autre nœud.

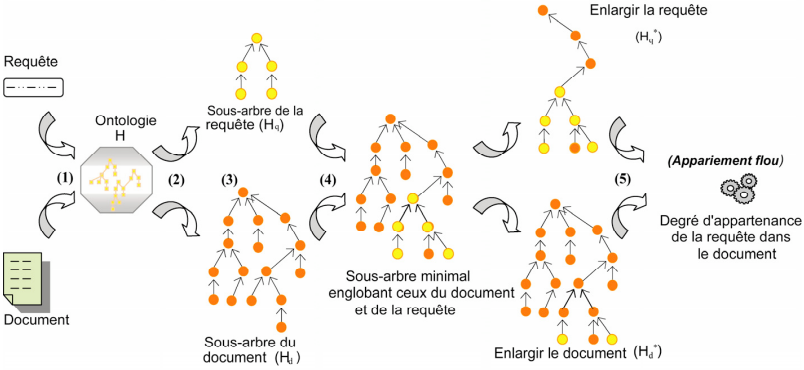

Figure 1. *Schéma général de l'approche basée sur les sous-arbres.*

Dans l'approche conceptuelle que nous avons adoptée, la requête et le document sont représentés par des sous hiérarchies formées par les concepts qu'ils contiennent et qui apparient ceux de l'ontologie. Après un traitement sur les deux représentations, les deux sous arbres sont comparés pour évaluer jusqu'à quel point le document traite du sujet de a requête. La comparaison des deux sous arbres se fait avec des opérateurs flous pour éviter la rigidité qu'une simple comparaison nœud-nœud pourrait donner. En effet, même si les nœuds des deux sous arbres ne sont pas identiques, un degré d'appariement est calculé en prenant en compte leurs éventuels parents communs qui se trouvent dans l'ontologie.

De façon générale, comme le schématise la **Figure 1**, l'approche comprend quatre étapes. **(1)** La première étape consiste à projeter la requête ou le document sur l'ontologie pour détecter les termes qui peuvent représenter des concepts de l'ontologie. Dans la même étape, une phase de désambiguïsation est nécessaire pour sélectionner pour chaque terme extrait, le concept de l'ontologie qui représente au mieux son sens dans le contexte du document. Cette première étape est décrite dans (Baziz *et al.*, 2005). De manière globale, la désambiguïsation est basée sur un calcul de proximité sémantique (score) entre les différents sens possibles des termes d'un document en utilisant différentes relations sémantiques de l'ontologie telles

l'hyperonymie, l'hyponymie, la méronymie, l'holonymie, etc. Pour chaque terme du document, son sens ayant cumulé le plus grand score est ensuite retenu. Ceci nous permet d'avoir pour chaque document les entrées dans l'ontologie qui sont appelées aussi les nœuds ou les concepts. La deuxième phase **(2)** concerne la construction du sous arbre lui-même. Si un document ou une requête ne contient pas tous les nœuds de sorte à former une arborescence (s'il y a discontinuité donc), les nœuds intermédiaires manquants, sont rajoutés. Ainsi, on obtient dans tous les cas un sous arbre où chaque nœud peut être remonté jusqu'à sa racine (la racine d'un sous arbre est propre à chaque document et requête et peut être un nœud intermédiaire de l'ontologie). **(3)** La requête et le document sont représentés par rapport à un même référentiel. Ce référentiel est le sous arbre minimal qui contient les deux sous arbres, celui de la requête et celui du document. **(4)** dans l'étape quatre, les sous arbres de la requête et du document sont élargis en rajoutant les nœuds du référentiel en le parcourant à partir des feuilles. **(5)** enfin, dans l'étape cinq, les deux représentations sont comparées en utilisant des opérateurs flous. Une valeur de pertinence est ensuite calculée à l'aide de ses fonctions. Cette valeur exprime jusqu'à quel point le document contient les thèmes exprimés dans la requête.

3. Description détaillée

3.1 Représentations séparées du document et de la requête

Soit la hiérarchie de concepts *H* d'une ontologie où tous les arcs sont représentés par le lien est-un (is-a) et soit un document initial d_{init} composé de *n* mots.

$$d_{init} = \{t_1, t_2, .., t_n\} \quad (1)$$

Après extraction des concepts et leur pondération (Baziz *et al.*, 2005), le document sera représenté par *m* concepts (*m*<=*n*) avec leurs poids respectifs :

$$d = \{(c_1, w_1), (c_2, w_2), .., (c_m, w_m)\} \quad (2)$$

La représentation de *d* dans *H* se fait alors en définissant *N(H, d)* comme l'ensemble des concepts de *d* qui correspondent à des concepts de *H* :

$$N(H, d) = \{(n_j, y_j), j = 1, .., m(d)\} \quad (3)$$

Puisque l'extraction des concepts dans le texte du document démarre de l'ontologie (pas d'utilisation de méthodes syntaxiques ou de cooccurrence), le risque de se retrouver avec de "faux concepts", c'est à dire, ceux non reconnus par cette dernière, est écarté. On est donc dans le cas où *d* et *N(H,d) (et m et m(d)*) sont confondus.

Les nœuds de *N(H,d),* une fois projetés sur *H*, peuvent ne pas former un sous arbre de *H*. On les complète alors par des nœuds intermédiaires pour former le *sous arbre minimal* de *d*, nommé H_d. Dans H_d, les nœuds appartenant à *N(H,d)* gardent leur poids inchangé, et ceux rajoutés ont un poids nul dans un premier temps. Nous verrons que dans le cas de l'élargissement (section 3.4) le poids de ces nœuds est modifié pour enrichir la représentation.

Considérons maintenant une requête q comprenant des concepts de H et des poids. La requête peut être obtenue soit en sélectionnant des nœuds dans H ou alors après un traitement identique à celui du document :

$$N(H,q) = \{(l_k, \delta_k), k = 1,.., r(k)\} \quad (4)$$

On suppose que la requête est comme une conjonction (pondérée) de concepts. Elle est aussi modélisée par un sous arbre minimal H_q de H contenant les nœuds de q. De même que pour le document, ici aussi les nœuds de H_q n'appartenant pas à q ont un poids nul dans un premier temps.

3.2 Représentation arborescente commune du document et de la requête

Une fois les sous arbres minimaux correspondant au document et à la requête construits, on cherche à les représenter par rapport à un même référentiel. Il s'agit donc de trouver le sous arbre minimal, H_E, qui contient au même temps H_d et H_q.

On aura alors les deux sous arbres H_d^* et H_q^* qui représenteront respectivement les extensions de H_d et H_q dans H_E en mettant à zéro le poids des nœuds respectivement de H_E-H_D et H_E-H_q.

3.3 Appariement requête-document basé sur le sous arbre minimal

L'évaluation de la requête se fait en comparant les deux sous ensembles de nœuds de H, H_d^* et H_q^*, l'un correspondant à la requête et l'autre au document courant. Un degré d'inclusion est alors calculé pour évaluer jusqu'à quel point le document contient le thème de la requête.

Pour une requête conjonctive q, le degré de pertinence d'un document d par rapport à cette requête est calculé comme suit :

$$Rel_{conj}(d; q) = min_{n \in H_E} \mu_{H_q^*}(n) \rightarrow \mu_{H_d^*}(n) \quad (5)$$

Où $\mu_{H_d^*}(n)$ (resp. $\mu_{H_q^*}(n)$) représente le poids associé au nœud n dans H_d^* (resp. H_q^*) et $\rightarrow$ une implication multivaluée "connective" exprimant le fait que tous les concepts de la requête doivent apparaître dans la description du document. Ici les différentes implications connues dans le domaine du flou peuvent être utilisées :

- ✓ L'implication de Dienes : $a \rightarrow b = \max(1 - a, b)$.
- ✓ L'implication de Gödel : $a \rightarrow b = 1$ si $a \leq b$ $a \rightarrow b = b$ si $a > b$.
- ✓ L'implication de Lukasiewicz : $a \rightarrow b = \min(1, 1 - a + b)$.

Dans le cas de requête disjonctive, il n'est pas nécessaire d'avoir tous les concepts de la requête dans un document pour le retourner. L'évaluation est calculée alors comme suit :

$$Rel_{disj}(d; q) = max_{n \in H_E} min(\mu_{Hd^*}(n), \mu_{Hq^*}(n)) \quad (6)$$

Une autre façon de calculer la pertinence autre que la conjonction et la disjonction, serait d'utiliser une simple somme. Ceci peut se traduire par le souci de permettre à tous les concepts de la requête d'intervenir dans le calcul de la valeur finale de pertinence. Ce qui a pour conséquence de chercher le "meilleur appariement", qui est adapté à la RI, au lieu du min et du max dans les deux cas précédents qui sont eux beaucoup plus adaptés à la recherche de données (Data Retrieval) (Rijsbergen, 1979). La pertinence est calculée alors comme suit :

$$Rel_{som}(d; q) = \sum_{n \in H_E} \mu_{H_q^*}(n) \rightarrow \mu_{H_d^*}(n) \quad (7)$$

3.4 Elargissement du document et de la requête

Dans les étapes précédentes, des noeuds ont été rajoutés à deux reprises à la représentation du document (resp. requête). Le premier ajout concerne le passage de la représentation en nœuds isolés *N(H,d)* (resp. *N(H,q)*) à la représentation en sous arbre minimal du document (resp. la requête). Le deuxième concerne l'élargissement de H_d (resp. H_q) dans H_E, pour arriver à la représentation finale en H_d^* (resp. H_q^*). Dans les deux cas, ces nœuds ne sont pas exploitables étant donné que le poids des nœuds intermédiaires rajoutés est mis à zéro. Il serait intéressant de les utiliser lors de la comparaison de la requête avec le document étant donné qu'ils ont un lien (is-a) avec les nœuds d'origine.

Posons w_i^s et w_i^{s+1}, les poids des niveaux s et s+1 dans la hiérarchie (les numéros de niveau sont descendants et la racine a le niveau 0). Etant donné qu'un nœud avec un poids nul peut se trouver à n'importe quel endroit du sous arbre, l'idée est de changer récursivement le poids des nœuds en démarrant des feuilles comme suit :

$$w_{i,new}^s = \max(w_i^s, (\max_i w_{i,new}^{s+1}) * fact(s)) \quad (8)$$

Où *fact(s)* représente un facteur qui dépend du niveau de la hiérarchie. Pour favoriser les nœuds spécifiques situés en bas de la hiérarchie, ce facteur doit être inférieur à 1. Ce facteur sert aussi à atténuer l'effet de l'expansion en affectant aux nœuds rajoutés un ratio du poids des nœuds de départ.

3.5 Exemple

Considérons une portion d'ontologie, *H*, comme illustré dans la Figure 2, où chaque nœud, n_i représente un concept. Soit *d* un document indexé avec les concepts suivants :

d = {(natural science, 0.1), (geology, 1), (geography, 0.5), (geophysics, 0.8)}

Considérons maintenant une requête *q* représentée par les concepts suivants :

q = {(earth science, 1), (geography 1)}

Les projections respectives du document d et de la requête q sur H sont les suivants:

$N(H,d)$ = {(n4, 0.1), (n12, 1), (n14, 0.5), (n16, 0.8)},

$N(H,q)$ = {(n9, 1), (n14, 1)}

Les arbres minimaux Hd et Hq peuvent alors être facilement construits à partir de $N(H, d)$ et $N(H, q)$ comme suit :

Hd = {(n4, 0.1), (n9, 0), (n12, 1), (n14, 0.5), (n16, 0.8)},
Hq = {(n9, 1), (n14, 1)}

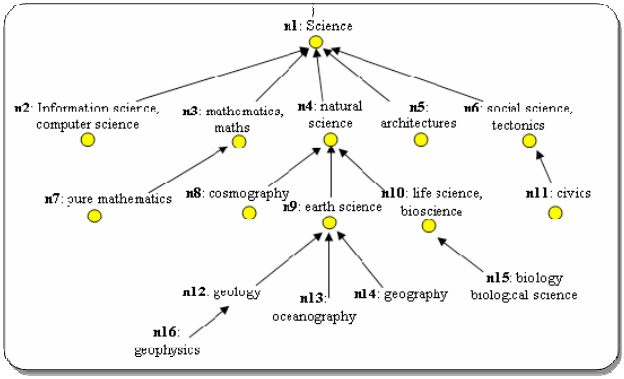

Figure 2. Exemple simplifié de l'ontologie des sciences.

Ainsi, l'arbre minimal non pondéré (H_E) qui contient les deux sous arbres Hd et Hq est représenté par les nœuds suivant :

H_E = {n4, n9, n12, n14, n16}

Pour voir l'impact de l'élargissement, évaluons la requête en utilisant l'implication de Dienes pour les deux cas, avec et sans élargissement. Les extensions de Hd et Hq dans HE, dans le cas sans élargissement sont comme suit :

Hd^* = {(n4, 0.1), (n9, 0), (n12, 1), (n14, 0.5), (n16, 0.8)}

Hq^* = {(n4, 0), (n9, 1), (n12, 0), (n14, 1), (n16, 0)}

L'évaluation de la requête conjonctive est alors donnée par :

$Rel_{conj}(d , q)$ = min(max(1-0, 0.1), max(1-1, 0), max(1-0, 1), max(1-1, 0.5), max(1-0, 0.8))=0

Notons que l'évaluation du cas disjonctif donne une valeur non nulle :

$Rel_{disj}(d, q)$ = max(min(0, 0.1), min(1, 0), min(0, 1), min(1, 0.5), min(0, 0.8))=0.5.

Si on considère maintenant le cas avec élargissement en fixant *fact(s)* à 0.7 (constante), *Hd** et *Hq** sont alors :

*Hd** = {(n4, 0.49), (n9, 0,7), (n12, 1), (n14, 0.5), (n16, 0.8)}
*Hq** = {(n4, 0), (n9, 1), (n12,0), (n14, 1), (n16, 0)}

D'où, $Rel_{conj}(d, q)$=0.5.

On peut noter que l'élargissement permet une identification meilleure des concepts d'un document. En effet, dans cet exemple et sans élargissement, "earth science" (le nœud n_9 dans *H*) n'apparaît pas dans le document *d*. l'appariement entre ce document et la requête donne alors une valeur nulle; par conséquent, le document n'est pas sélectionné. Cependant, on peut déduire, du fait que le document parle de "geology", "geography" et "natural science", qu'il traite de "earth science". Ainsi, la procédure d'élargissement permet de palier ce problème, le noeud n_9 devient non nul dans H_{d*}. Ce qui permet de sélectionner le document *d*.

3.6 Suppression des nœuds abstraits

En pratique, lors de l'élargissement d'un document (requête), seule une partie des nœuds intermédiaires est rajoutée à la représentation de celui-ci (celle-ci). Ceci peut être justifié par deux raisons. La première raison a trait au temps de calcul qui sera sensiblement réduit en ne considérant que la partie basse de l'arbre. La deuxième raison est due au fait que les nœuds à supprimer, situés dans la partie supérieure de la hiérarchie représentent des concepts abstraits. En effet, plus on monte dans l'arbre, plus les nœuds sont abstraits. Comme ces nœuds subsument la majorité des nœuds de l'arbre, ils reviennent souvent et ne représentent donc pas un facteur discriminant pour les documents. Inversement, plus on se rapproche des feuilles de l'arbre, plus les nœuds sont spécifiques et permettent de différencier les documents. Cette phase supplémentaire de suppression des nœuds abstraits utilise deux informations pour décider si un nœud peut être rajouté à la représentation du document (requête). La première information est la position de la racine, *depth*, du sous arbre contenant les nœuds d'origine dans H. La deuxième, *length*, est le nombre de nœuds de la sous branche courante (de la sous hiérarchie à étendre) contenant les nœuds intermédiaires candidats à être rajoutés à la description du document (requête). Ainsi pour une branche B_i d'un sous arbre donné à étendre, le nombre *Nb* de nœuds à rajouter réellement est calculé comme suit :

$$Nb(B_i) = min\ [(length\ (B_i)\text{-}1 + depth+1)/2,\ length(B_i)\text{-}1] \qquad (9)$$

Ainsi, une valeur importante de *depth* indique que la racine du sous arbre se situe près des feuilles et permet de rajouter un nombre de nœuds spécifiques; alors q'une petite valeur de *depth* permet de supprimer les nœuds abstraits se trouvant au voisinage immédiat de la racine de *H* (dans le cas de la racine, *depth*=0).

Exemple

Pour illustrer la sélection des nœuds à utiliser lors de l'élargissement d'un document (requête), considérons le cas simple d'un sous arbre avec deux branches. Les deux concepts suivants sont extraits d'un document (nommé Arthroskopie.00130047 dans la collection utilisée) : *alternative#n#1* qui représente le sens 1 du nom "alternative", et *amount#n#1,* le sens 1 du nom "amount". Le sous arbre formé par les deux concepts est comme dans la Figure 3.

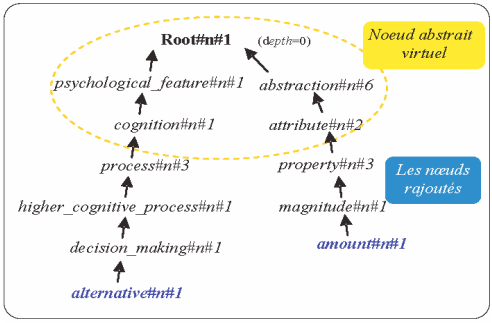

Figure 3. Le sous arbre contenant les deux concepts *alternative#n#1*et *amount#n#1*

La position du nœud racine du sous arbre (*depth)* est égale à 0 étant donné que dans ce cas, ce nœud représente aussi la racine de l'arbre. Posons B_1 et B_2 les deux branches comme représentées dans la Figure 3 :

B_1 = alternative#n#1 → decision_making#n#1 → higher_cognitive_process#n#1 → process#n#3 → cognition#n#1 → psychological_feature#n#1

B_2 = amount#n#1 → magnitude#n#1→ property#n#3→ attribute#n#2 → abstraction#n#6

Ainsi, length (B_1)=6 et length (B_2)=5.

Le nombre de nœuds à ajouter (en démarrant des feuilles) pour la branche B_1 est :

$Nb(B_1)=min[(length(B_1)\text{-}1+depth)/2, length(B_1)\text{-}1]$ =3 noeuds.

Qui sont : decision_making#n#1, higher_cognitive_process#n#1 et process#n#3.

Et pour la 2ème branche B_2 :
$Nb(B_2)=min[(length(B_2)\text{-}1+depth)/2, length(B_2)\text{-}1]$ =2 noeuds

Qui sont : magnitude#n#1 et property#n#3.

Les nœuds restants sont effectivement des nœuds abstraits et ne sont pas utilisés pour élargir la représentation du document. Ils représentent ce que nous appelons un *nœud abstrait virtuel*.

4. Evaluation

Le but de l'évaluation est de mesurer l'efficacité de l'approche basée concept (ici représentation avec les sous arbres) par rapport à l'approche classique basée mots clés. Plus spécialement, deux principales contributions proposées dans l'approche sont évaluées :

- Quel est l'apport de l'indexation basée concepts, telle que nous l'avons défini par rapport à une indexation classique?
- Quel est l'apport de l'élargissement des descriptions des documents et des requêtes comparé au classique?

Un module de représentation tel que décrit dans les sections 3.2, 3.4, 3.6 et de recherche, décrit en section 3.3, ont été implémentés. Comme ontologie, nous avons utilisé la hiérarchie de concepts qui correspond au sous ensemble du réseau conceptuel de WordNet, définie par la relation *IS-A*. La collection de test est issue du projet MuchMore[3] (Buitelaar *et al.*, 2004). Cette collection inclut 7823 documents (résumés d'articles) obtenus à partir du site de SpringerLink, 25 sujets types (topics en anglais) à partir desquels les requêtes sont extraites ainsi que les jugements de pertinence réalisés par des experts du domaine. Voici un exemple de requête (numéro 29 et étiquetée 109) extrait de la collection :

Query 109 : Treatment of sensorineural hearing loss (SNHL)

Après la detection du concept multi mots *sensorineural_hearing_loss* défini dans WordNet par :

sensorineural hearing loss, nerve deafness -- (hearing loss due to failure of the auditory nerve)

la requête finale est :

109 *treatement* 1	109 *hear 1*
109 *sensorineural_hearing_loss 1*	109 *loss 1*
109 *sensorineur 1*	109 *snhl 1*

Les documents de la collection traitent du domaine médical. Cependant le vocabulaire utilisé est assez général et est largement couvert par WordNet. Comme indiqué dans le Tableau 1, le taux de couverture représente 87% du vocabulaire des documents et 77% du vocabulaire utilisé dans les requêtes.

Nous avons choisi d'utiliser cette collection relativement réduite à cause du temps de calcul que nécessite le calcul des sous-arbres pour l'ensemble des documents de la collection. La collection de test utilisée est la base *MuchMore*. Cependant seules 20 requêtes parmi les 25 que contient la collection sont utilisées ici. Les cinq autres requêtes ne peuvent pas être représentées étant donné que le vocabulaire qu'y est utilisé n'est pas (ou pas suffisamment) couvert par WordNet.

[3] http://muchmore.dfki.de/

	NOMBRE MOYEN DE TERMES	
	Documents	Requêtes
TOUS LES TERMES (INDEX CLASSIQUE)	56,01	4,05
TERMES DE WORDNET SEULEMENT	48,89	3,1
%	87%	77%

Tableau 1. *Nombre moyen de termes couverts par WordNet dans la collection MuchMore*

4.1 Méthode d'évaluation

Pour évaluer l'approche, deux ensembles d'expérimentations ont été construits. Le premier ensemble est basé sur l'indexation classique et le deuxième sur la représentation en sous arbres, telle que définie précédemment. Dans l'approche classique, les documents sont d'abord indexés en sélectionnant les mots simples occurrant dans les documents, les mots outils sont éliminés en utilisant une liste standard (Salton, 1984), enfin les mots restants sont radicalisés en utilisant Porter (Porter, 1980). Le même processus est appliqué aux requêtes. Le système Mercure (Boughanem *et al.*, 1998) est alors utilisé pour rechercher les documents. Nous avons utilisé ce cas comme référence (baseline).

Dans l'approche basée concepts, les documents et les requêtes sont indexés comme suit. Les termes (simples ou composés) sont d'abord extraits pour chaque document (requête) suivant la méthode décrite dans (Baziz *et al.*, 2005). Ils sont ensuite désambiguïsés localement (c'est à dire, par rapport aux autres termes du document) en utilisant la mesure de Resnik (1999). Ce qui permet d'affecter à chaque terme extrait, un nœud (concept) unique dans la hiérarchie. Le résultat de cette phase est que chaque document (requête) est représenté par un ensemble de concepts (*N(H,d) et N(H,q)*). Une fois que les nœuds représentant les documents (requêtes) sont identifiés, les sous arbres correspondants, H_d^* et H_q^* sont construits en calculant récursivement les nœuds qui subsument les nœuds extraits jusqu'à avoir un sous arbre unique qui couvre tous les concepts extraits. Il peut arriver que les nœuds qui se trouvent en haut des sous arbres construits soient situés dans le voisinage immédiat de la racine de WordNet (*entity*). Dans ce cas, étant donné que ces nœuds sont trop abstraits, ils ne sont pas utilisables, ils sont donc enlevés de la représentation en utilisant la méthode d'élimination des nœuds abstraits décrite précédemment en section 3.6.

L'évaluation des requêtes est basée sur les implications de Lukasiewicz, Gödel et Dienes en utilisant trois (3) fonctions d'agrégation : min, max et la somme. La méthode d'évaluation est la même que celle suivie par le protocole de TREC (Voorhees *et al.*, 1998) : pour chacune des 20 requêtes utilisées, on calcule les

valeurs P5, P10, P15, P30 représentant les précisions à 5, 10, 15 et 30 premiers documents restitués ainsi que la MAP (Mean Average Precision), la précision moyenne.

4.2 Résultats et Discussion

Les résultats les plus représentatifs sont résumés dans le Tableau 2. Le premier cas, *classique*, décrit les résultats de l'approche classique. Les autres cas sont ceux combinant les implications de Lukasiewicz, Dienes et Gödel avec la fonction d'agrégation *max* (pour une évaluation disjonctive des requêtes) qui sont *CO_Lukas_dis*, *CO_Dienes_dis* et *CO_Godel_dis*; ceux utilisant *min* (pour le cas conjonctif), *CO_Lukas_conj*, *CO_Dienes_conj*, *CO_Godel_conj*; et ceux utilisant la somme, *CO_Lukas_sum*, *CO_Dienes_sum* et *CO_Godel_sum*. La seconde colonne du Tableau 2 indique si un élargissement (complétion qui consiste à modifier le poids des nœuds à 0) est effectué pour les requêtes et/ou les documents.

Premièrement, on peut remarquer que le type d'implication n'a pas d'impact sur les résultats. En effet, les résultats du Tableau 2 montrent que pour une fonction d'agrégation donnée, les résultats sont comparables. Cependant, on peut remarquer que la fonction d'agrégation est déterminante. D'après le Tableau 2, les résultats de la méthode classique sont fortement supérieurs à ceux de l'évaluation disjonctive et conjonctive de la requête sur tous les points de précision considérés. Le cas conjonctif est meilleur que le cas disjonctif du fait qu'il n'évalue que les documents qui contiennent tous les termes de la requête, ce qui est le cas de douze (12) requêtes sur les vingt utilisées.

En fait, l'évaluation en elle-même, si elle peut expliquer partiellement le problème du cas conjonctif (tous les termes de la requête doivent apparaître dans le document pour qu'il soit sélectionné), elle n'explique pas celui de la fonction disjonctive. En effet, en analysant la liste des documents retournés, on a remarqué que des documents pertinents ont été sélectionnés mais sont mal ordonnés. Ceci est dû au fait que les fonctions de disjonction et de conjonction ne sont pas adaptées pour ordonner les documents résultats (ranking). Dans le cas de la fonction disjonctive, un document ayant par exemple, les valeurs de pertinence (0.6, 0, 0, 0) passe avant un autre document ayant plus de termes communs mais avec des poids légèrement inférieurs (0.5, 0.5, 0.5, 0.5). Pour la fonction conjonctive, le problème est différent. Cette fonction qui effectue un appariement exact, est surtout adaptée à la Recherche de Données où un document est sélectionné si et seulement s'il contient tous les termes de la requête, alors qu'un appariement au mieux (best matching) est utilisé en RI (Rijsbergen, 1979). Enfin, on remarque aussi que lorsqu'un élargissement est appliqué aux descriptions des requêtes et des documents, les résultats restent inférieurs par rapport au cas classique.

Quand la somme (*sum*) est utilisée comme fonction d'agrégation (*CO_Lukas_sum*, *CO_Dienes_sum* et *CO_Godel_sum*), les résultats sont d'un autre

ordre. On peut voir dans ce cas que la somme, sans élargissement, rapporte des résultats meilleurs que les fonctions d'agrégation de conjonction et de disjonction. Les résultats sont dans la même échelle que ceux du classique avec un léger bénéfice pour l'approche classique. Cependant, lorsqu'un élargissement est appliqué aux requêtes et/ou aux documents, les résultats sont améliorés sur tous les points de précision avec un réel bénéfice lorsque l'élargissement est appliqué aussi bien aux requêtes qu'aux documents. En effet, pour ce dernier cas, toutes les valeurs de précision obtenues sont meilleures que la méthode classique.

		P5	P10	P15	P30	MAP
(1) Classique (Baseline)		**0,3600**	**0,3500**	**0,3200**	**0,2230**	**0,2560**
(2) C0_Lukas_conj	Sans élargissement	0.2300	0.1750	0.1267	0.0233	0.0376
	Elargis. des documents	0.2800	0.2300	0.1733	0.0967	0.1245
	Elargis. des requêtes	0.0300	0.0250	0.0233	0.0233	0.0376
	Elargis. des requêtes et des documents	0.2300	0.1750	0.1267	0.0783	0.0994
(3) C0_Lukas_dis	Sans élargissement	0.0600	0.0650	0.0667	0.0675	0.0719
	Elargis. des documents	0.0600	0.0600	0.0633	0.0600	0.0701
	Elargis. des requêtes	0.0900	0.1000	0.0867	0.0767	0.0755
	Elargis. des requêtes et des documents	0.0900	0.0950	0.0767	0.0567	0.0675
(4) CO_Dienes_dis	Sans élargissement	0.0600	0.0650	0.0667	0.0667	0.0719
	Elargis. des documents	0.0400	0.0600	0.0467	0.0583	0.0661
	Elargis. des requêtes	0.0900	0.0900	0.0933	0.0650	0.0801
	Elargis. des reqs. et docs	0.1500	0.1600	0.1500	0.1517	0.1249
(5) CO_Godel_dis	Sans élargissement	0.0600	0.0650	0.0667	0.0667	0.0719
	Elargis. des documents	0.0600	0.0600	0.0633	0.0600	0.0701
	Elargis. des requêtes	0.0900	0.1000	0.0867	0.0767	0.0755
	Elargis. des reqs. et docs	0.0900	0.0950	0.0767	0.0675	0.0675
(6) CO_Lukas_sum	Sans élargissement	0.4600	0.4200	0.3833	0.2633	0.2686
	Elargis. des documents	0.4700	0.4300	0.3833	0.2550	0.2741
	Elargis. des requêtes	0.4200	0.4150	0.3833	0.2733	0.2598
	Elargis. des reqs. et docs	**0.4700**	**0.4550**	**0.4233**	**0.2850**	**0.2996**
(7) CO_Dienes_sum	Sans élargissement	0.4600	0.4200	0.3833	0.2633	0.2686
	Elargis. des documents	0.4600	0.4250	0.3767	0.2550	0.2707
	Elargis. des requêtes	0.3800	0.3650	0.3200	0.2317	0.2346
	Elargis. des reqs. et docs	**0.4700**	**0.4250**	**0.3867**	**0.2850**	**0.2920**
(8) CO_Godel_sum	Sans élargissement	0.4600	0.4200	0.3833	0.2633	0.2686
	Elargis. des documents	0.4700	0.4300	0.3833	0.2550	0.2741
	Elargis. des requêtes	0.4200	0.4150	0.3833	0.2733	0.2598
	Elargis. des reqs. et docs	**0.4700**	**0.4550**	**0.4233**	**0.2833**	**0.3005**

Tableau 2. *Comparaison entre l'approche classique et les approches conceptuelles*

Un autre résultat important de l'évaluation concerne l'élargissement des documents. En effet, à notre connaissance, la plupart des approches "élargissement/expansion" existantes dans les systèmes de RI sont utilisées pour les requêtes mais pas pour les documents. Ces expériences préliminaires, tendent à indiquer que l'élargissement des requêtes et des documents peut servir comme moyen pour identifier les concepts importants qui n'apparaissent pas explicitement dans le texte des documents et des requêtes, fournissant ainsi, une possibilité de recherche contextuelle.

4.3 Conclusion

Nous avons présenté un modèle de recherche d'information guidée par ontologie et basé sur une représentation hiérarchique de l'information textuelle. Dans ce modèle, les documents et les requêtes sont représentés à l'aide d'une structure arborescente où les nœuds représentent les concepts extraits (entrées de l'ontologie) et les arcs la relation de subsomption (*IS-A*). Après identification des concepts, un traitement préliminaire est d'abord effectué pour rajouter d'éventuels nœuds intermédiaires, afin que l'arbre du document (requête) soit une sous hiérarchie de l'ontologie. Puis, un référentiel commun représenté par le sous arbre minimal qui englobe celui du document et de la requête est construit. Enfin les sous arbres du document et de la requête sont élargis par rapport à ce référentiel en utilisant un facteur de propagation de poids. Un modèle de recherche basé sur un appariement flou permet ensuite de calculer une valeur de pertinence qui traduit jusqu'à quel point le document inclut les concepts de la requête. Différents types d'implications et fonctions d'agrégation sont évalués. Les résultats ont montré que notre approche de représentation conceptuelle donne de meilleurs résultats que l'approche classique (sur tous les différents points de précision), notamment quand la somme est utilisée comme fonction d'agrégation.

Une perspective envisageable à ce travail concerne la généralisation de l'approche à d'autres ressources conceptuelles externes de domaines spécifiques (telle UMLS) ainsi que l'étude de l'optimisation des algorithmes pour évaluer le passage à l'échelle (ex. : les collections Gygabyte de TREC).

Référence :

Banerjee, S. and Pedersen, T.: "An adapted Lesk algorithm for word sense disambiguation using Word-Net". In *Proc. of the Third International Conference on Intelligent Text Processing and Computational Linguistics*, Mexico City, February 2002.

Baziz, M., Boughanem, M., Aussenac-Gilles, N., Chrisment, C., "Semantic Cores for Representing Documents in IR". In *Proceeding of the 2005 ACM Symposium on Applied Computing*, vol2 pp. 1011-1017, Santa Fe, New Mexico, USA, March 2005.

Boughanem M., Dkaki, T. Mothe J and C. Soulé-Dupuy "Mercure at TREC-7". In Proceeding of Trec-7, (1998).

Budanitsky, A., Hirst, G. “Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures”, in: *Workshop on WordNet and Other Lexical Resources*, Second meeting of the ACL, Pittsburgh, 2001, pp. 29–34.

Buitelaar, P., Steffen D., Volk, M., Widdows, D., Sacaleanu, B., Vintar, S., Peters, S., Uszkoreit, H., *Evaluation Resources for Concept-based Cross-Lingual IR in the Medical Domain. In Proc. of LREC2004,* Lissabon, Portugal, May 2004.

Cucchiarelli, R. Navigli, F. Neri, P. Velardi. “Extending and Enriching WordNet with OntoLearn”. *Proc. of The Second Global Wordnet Conference 2004 (GWC 2004), Brno, Czech Republic*, January 20-23rd, 2004

Gonzalo, J., Verdejo, F., Chugur, I., Cigarrán, J., “Indexing with WordNet synsets can improve text retrieval”, in *Proc. the COLING/ACL '98 Workshop on Usage of WordNet for Natural Language Processing*, 1998.

Hirst, G., and St. Onge, D.: *Lexical chains as representations of context for the detection and correction of malapropisms*. In C. Fellbaum, editor, WordNet: An electronic lexical database, pages 305–332. MIT Press, 1998.

Khan, L., and Luo, F.: “Ontology Construction for Information Selection”. In *Proc. of 14th IEEE International Conference on Tools with Artificial Intelligence*, pp. 122-127, Washington DC, November 2002.

Lesk, M.: “Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from a ice cream cone”. In *Proc. of SIGDOC ’86*, 1986.

Maedche A. and Staab S., 2000. “Semi-automatic Engineering of Ontologies from Text”. *Proceedings of the Twelfth International Conference on Software Engineering and Knowledge Engineering*.

Mihalcea, R. and Moldovan, D.: “Semantic indexing using WordNet senses”. In *Proceedings of ACL Workshop on IR & NLP*, Hong Kong, October 2000.

Porter M., "An algorithm for Suffix Stripping", Program, Vol. 14(3), pp 130-137, 1980.

Resnik, P., "Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language", *Journal of Artificial Intelligence Research (JAIR)*, 11, pp. 95-130, 1999.

Salton G., and M.J. Mc. Gill, *Introduction to Modern Information retrieval*. McGraw-Hill Int. Book Co., 1984.

Sanderson, M., “Retrieving with good senses”. In *Information Retrieval, Vol. 2(1)*, pp. 49-69, 2000.

Van Rijsbergen, C. J. (1979). *Information retrieval*. Butterworths, London, 2nd edition.

Vorhees, E. M., and Harman, D. K., “Overview of the sixth Text REtrieval Conference (TREC-6)”, in Vorhees, E. M., and Karman, D. K. (eds.), *Proc. Of TREC-6*, (1998).