Expansion sémantique des requêtes pour un modèle de recherche d'information par proximité

Bissan AUDEH, Philippe BEAUNE, Michel BEIGBEDER

Institut Henri FAYOL
Ecole des Mines de Saint Étienne
158 Cours Fauriel 42000 FRANCE
{audeh, beaune, mbeig}@emse.fr

RÉSUMÉ. *L'utilisation d'une ressource lexicale pour reformuler automatiquement les requêtes est une approche d'expansion qui attire souvent les chercheurs. Malheureusement, l'utilité de cette approche, ne peut pas être généralisée à tous les contextes de recherche d'information. Dans cet article, nous prenons le cas d'un modèle de recherche fondé sur la proximité. Ce modèle obtient une très bonne précision dans les campagnes d'évaluation, mais il renvoie peu de résultats. Nous proposons de compléter les résultats des requêtes originales par celles des requêtes étendues, afin d'augmenter le rappel tout en préservant la bonne précision du modèle. Dans nos expériences, nous montrons que dans ce contexte, il est possible d'améliorer à la fois le rappel et la précision tout en utilisant WordNet comme une ressource sémantique pour la désambiguïsation du sens et pour l'extraction de termes d'expansion*

ABSTRACT. *Using a lexical thesauri to modify queries is an approach of automatic query expansion that often attracts researchers. Unfortunately the usefulness of this approach, on its own, couldn't be generalized to all information retrieval contexts. In this paper, we take the case of a proximity based retrieval model that has a very high precision but returns few results. Based on the specificity of this model we propose to extend the results of users' query by results of the expanded query, in order to enhance the recall while preserving the original good precision. In our experiments we show that in this context it is possible to enhance both the recall and the precision while using only WordNet as a semantic resources for word sense disambiguation and for new terms extraction*

MOTS-CLÉS : *Expansion sémantique des requêtes, Modèle de recherche par proximité, désambiguïsation, WordNet, Ressource lexicale*

KEYWORDS: *Semantic Query Expansion, Proximity-Based Retrieval Model, WordSense Disambiguation, WordNet, Lexical Thesaurus*

1. Introduction

La notion de proximité n'est pas nouvelle en recherche d'information. Plusieurs travaux ont trouvé une relation importante entre la proximité entre les mots de la requête dans un document, et la pertinence de ce document. Depuis les années 80 l'opérateur NEAR a été utilisé pour exprimer la proximité entre certains mots de la requête (Salton et McGill, 1986). Le défaut majeur de cet opérateur est qu'il ne peut pas lier plus de deux mots à la fois (Mitchell, 1973). Pour cette raison, d'autres études se fondent sur le calcul des intervalles qui contiennent les termes de la requête dans un document (Hawking et Thistlewaite, 1996; Rasolofo et Savoy, 2003). Ces méthodes permettent de généraliser la notion de proximité pour plusieurs termes en sélectionnant des intervalles de texte contenant les termes de la requête puis en donnant un score à ces intervalles. Beigbeder et al. (Beigbeder et Mercier, 2005) ont proposé un modèle de recherche par proximité qui utilise des requêtes booléennes et la logique floue pour calculer la proximité.
Dans cet article, nous cherchons à savoir si l'expansion sémantique des requêtes peut améliorer la performance d'un modèle de recherche basé sur la proximité. Pour cela, nous choisissons le modèle de Beigbeder et al. car ce modèle a une bonne précision mais il renvoie trop peu de documents : il est donc un bon candidat pour l'expansion des requêtes. Nous commençons d'abord par présenter le modèle de recherche par proximité en section 2, puis nous présentons notre proposition en section 3 suivie par nos expériences et résultats. Nous terminons cet article en section 5 par une conclusion et des idées pour nos prochains travaux.

2. Le modèle FPIRM (Fuzzy Proximity Information Retrieval Model)

Le modèle de recherche par proximité floue (Beigbeder et Mercier, 2005) profite de la simplicité des requêtes booléennes et de la flexibilité de la logique floue pour interpréter la proximité. Dans les campagnes d'évaluation (TREC2005, CLEF2005 et 2006), FPIRM a réussi à dépasser les modèles précédents fondés sur la proximité. L'idée de ce modèle est de créer une *zone d'influence* autour de chaque terme de la requête dans le document à évaluer. Pour créer ces zones, une fonction floue est utilisée pour attribuer une valeur réelle dans l'intervalle [0,1] à chaque position dans le document. Cette valeur est maximale dans les positions qui contiennent des mots de la requête, puis décroit progressivement selon la fonction floue utilisée. Selon les opérateurs booléens utilisés entre les termes de la requête, les valeurs d'influence dans chaque position sont combinées en utilisant le min (si l'opérateur est AND) ou le max (si l'opérateur est OR). Le score final du document est la somme des valeurs d'influences à chaque position dans le document. Par défaut, le modèle de proximité floue considère une relation conjonctive entre les termes de la requête si l'opérateur OR n'est pas mentionné explicitement. Pour cette raison, ce modèle est très sélectif, ce qui lui permet d'avoir des réponses très précises par rapport à BM25 pour la première partie de la liste de résultats. Malheureusement, cette haute sélectivité est la raison pour laquelle ce modèle obtient des listes de résultats relativement courtes. Par

conséquent, le rappel de ce modèle est faible. Une solution qui permet d'augmenter le nombre de résultats est d'élargir la taille des zones d'influence. Ce choix a un effet direct sur la bonne précision de ce modèle, car il peut remplacer les documents pertinents du début de la liste avec d'autres moins pertinents. Une autre solution est l'approche d'expansion de la requête, qui permet de trouver des documents pertinents même s'ils ne contiennent pas les mots exacts de la requête. Dans cette article nous étudions l'effet de cette approche sur la performance de FPIRM.

3. L'expansion des requêtes pour le modèle FPIRM

L'expansion de requêtes est une approche souvent utilisée en recherche d'information pour améliorer le rappel. Plusieures méthodes existent dans la littérature pour effectuer cette approche (Manning *et al.*, 2008). Nous étudions dans ce papier une méthode d'expansion utilisant la ressource sémantique WordNet[1], qui organise les termes en groupes de synonymes appelés les Synsets.
Notre processus contient les étapes suivantes : tout d'abord, les termes de la requête sont désambiguïsés afin de choisir un sens unique pour chaque terme de la requête. Une fois que les sens sont choisis, nous cherchons les termes d'expansion liés à chaque sens. Ces termes sont ensuite intégrés à la requête originale tout en essayant d'interpréter correctement les relations booléennes entre eux. Finalement, les résultats de la requête originale sont complétés avec ceux rendus par la requête étendue afin de construire la liste finale de résultats. Dans ce travail, nous désambiguïsons uniquement les mots isolés de la requête car nous supposons que si l'utilisateur a précisé explicitement une phrase ou une expression dans sa requête, c'est qu'il souhaite voir les mots exacts de cette expression dans un document pertinent.

3.1. *Désambiguïsation*

Il existe différentes méthodes de désambiguïsation qui dépendent de l'analyse des documents et des requêtes, comme la classification automatique et les graphes de cooccurence. Dans la présente étude, puisque nous utilisions WordNet pour l'expansion des requêtes, il nous a semblé cohérent d'utiliser aussi WordNet pour la désambiguïsation. Pour cela nous avons adopté une méthode structurelle fondée sur une distance sémantique entre les concepts selon la formule suivante (Navigli, 2009) :

$$\hat{S} = \text{argmax}_{S \in Senses(w_i)} \sum_{w_j \in T : w_i \neq w_j} \max_{S' \in Senses(w_j)} Score(S, S') \qquad [1]$$

où T est l'ensemble des termes de la requête, w_i est le terme qu'on souhaite désambiguïser, $Senses(w_i)$ est l'ensemble des concepts candidats pour le terme w_i, ce qui correspond dans WordNet aux synsets qui contiennent ce terme, et $Score(S, S')$ est la fonction utilisée pour mesurer la similarité entre deux concepts S et S'. Plusieurs

1. http ://wordnet.princeton.edu/

méthodes existent pour mesurer la similarité entre deux concepts S et S' : nous choisissons dans cet article, suite à plusieurs expériences de comparaison, une approche basée sur le parcours des arêtes du graphe (Palmer et Wu, 1994). Cette approche suppose que la similarité entre deux concepts dépend de la profondeur des nœuds concernés et leur ancêtre commun (*Least Common Concept*)[2] par rapport à un noeud racine dans la ressource. Cette mesure de similarité s'applique aux synsets qui se trouvent dans la même taxonomie, ainsi la présence d'un verbe ou d'un adjectif dans la requête, qui est rare pour nos requêtes de test, est ignorée pendant la désambiguïsation des noms. Pour les verbes et adjectifs, nous choisissons le Synset qui signifie le sens le plus fréquent selon WordNet.

3.2. *La sélection des termes*

Une fois faite la désambiguïsation des termes de la requête, l'étape suivante essaie de trouver les termes d'expansion les mieux adaptés pour chaque mot original. Notre première option est de chercher les synonymes dans le Synset sélectionné par l'étape précédente. Dans certains cas, le Synset choisi ne contient aucun terme que celui qu'on essaie d'étendre. Dans ces cas, nous prenons les termes qui se trouvent dans de l'hypéronyme [3] du terme à étendre.

3.3. *La reformulation de la requête*

La performance d'un modèle de recherche d'information est fortement influencée par la qualité de la requête. Bien que la plupart des modèles proposent un langage de requêtes contenant une variété d'opérateurs pour mieux exprimer les besoins d'information, les utilisateurs préfèrent souvent fournir les requêtes sous forme de sac à mots, laissant ainsi le modèle de recherche interpréter les liens entre les termes selon sa conception. Ce comportement est fréquent dans la recherche d'information sur le Web, c'est aussi le cas de la plupart des campagnes d'évaluation comme TREC et INEX qui présentent les titres de leurs besoins d'information sous forme de sac de mots. FPIRM ajoute l'opérateur AND entre les termes de la requête si aucun autre opérateur n'est précisé. Pour cette raison, nous proposons une nouvelle technique qui permet de choisir le « bon » opérateur entre les termes lors de l'intégration des nouveaux termes d'expansion dans la requête. Notre algorithme de sélection vérifie, à chaque fois qu'on essaie d'ajouter un terme, si ce terme a été déjà considéré en tant que synonyme d'un terme déjà traité. Si c'est le cas, nous ne rajoutons pas ce terme à la requête même s'il est un terme original. Cette opération va détecter les termes originaux qui sont synonymes entre eux, et va donc transformer la relation conjonctive implicite entre eux en OR. La version actuelle de cet algorithme traite uniquement les requêtes plates, où l'opérateur AND est l'opérateur par défaut si aucun autre choix

2. *Least Common Concept* est le premier ascendant commun entre deux nœuds.
3. L'hypéronyme d'un terme est son prédécesseur direct dans la taxonomie IS-A

n'est précisé explicitement. Bien que cette technique ne résolve pas le problème d'ambiguïté, elle est quand même utile pour les requêtes longues et moyennes. Par exemple, la requête `chemists physicists scientists alchemists table` sera interprétée par l'algorithme précédent comme `(chemists OR scientist OR physicist) AND (alchemists OR intellect) AND (table OR board).`

3.4. *Complétion des résultats*

Dans notre approche, le modèle de RI est exécuté deux fois : une fois avec la requête originale, et la deuxième fois avec la requête reformulée. Les deux listes de résultats de ces deux exécutions sont combinées comme suit : les documents renvoyés par la requête originale sont privilégiés et sont donc mis au début de la liste finale, les documents résultant de la requête reformulée son ajoutés à la liste précédente s'ils n'y sont pas déjà. Les scores des documents de la deuxième liste sont normalisés par rapport au score du dernière document de la première liste en utilisant la formule suivante :

$$newScore(d'_i) = \frac{score(d'_i)}{score(d'_1)}(score(d_n) - \lambda) \quad [2]$$

où d'_i est un document dans la liste de résultats de la requête reformulée, d'_i est le premier document dans cette liste, et d_n est le dernier document renvoyé par la requête originale. Un constante λ est utilisée pour que le première document dans la deuxième liste obtienne un score légèrement inférieur au dernier document de la première liste (le cas où $d'_i = d'_1$).

4. Expériences

Pour tester notre approche, nous avons utilisé la collection de test INEX2009 (tâche ad-hoc) qui contient 2 600 000 articles en anglais issus de Wikipedia. Ces documents ont été annotés en utilisant l'ontologie YAGO. 63 requêtes de la tache ad-hoc d'INEX2009 ont été utilisées. Les documents et les requêtes ont été lemmatisées en utilisant l'algorithme de Porter. La largeur de la zone d'influence de FPIRM (Beigbeder et Mercier, 2005) est 300, cette valeur a été choisie après plusieurs expériences avec des valeurs qui varient de 100 à 600 où la longueur moyenne des documents de notre collection est 660. Pour la mesure de similarité entre concepts, nous avons utilisé notre propre implémentation en Java qui a été testée et validée en donnant les mêmes valeurs que la bibliothèque Perl `Wordnet::Similarity`(Pedersen, 2004).
Il est clair que notre approche ne va pas détruire la précision originale du modèle. Elle peut par contre améliorer le rappel, ce qui est le but de ce travail. Pour cette raison, le but de nos expériences n'est pas de voir si nous avons amélioré la performance, mais plutôt de savoir à quel point nous pouvons améliorer le rappel. Ces expériences comparent les trois ensembles de résultats obtenus en utilisant FPIRM comme un modèle de base :

	FPIRM-Base	FPIRM-ExpandQuery	FPIRM-ReformulateQuery
MAP	0.1155	0.1256	0.1309

Tableau 1. *Le MAP pour les trois cas de test.*

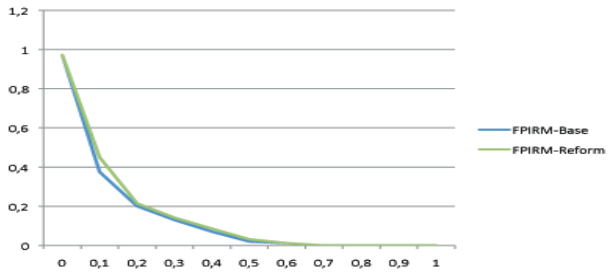

Figure 1. *La courbe Rappel-Précision de FPIRM-Base et FPIRM-ReformulateQuery*

– FPIRM-Base : les résultats des requêtes originales sans l'expansion des requêtes (les titres des requêtes INEX2009),

– FPIRM-ExpandQuery : les résultats des requêtes originales sont complétées par celles des requêtes étendues. L'expansion dans ce cas se fait en ajoutant les nouveaux termes aux termes orignaux par l'opérateur OR.

– FPIRM-ReformulateQuery : comme dans le cas précédent, la liste originale de résultat est complétée avec celle des requêtes étendues. Par contre, dans ce cas les nouveaux termes sont intégrés dans la requête originale en utilisant l'algorithme de la section 4.3 qui modifie la structure booléenne de larequête.

4.1. *Résultats*

En observant le tableau 1 nous constatons que la MAP est amélioré de 8,74% en utilisant l'expansion simple des requêtes, alors que cette amélioration atteint 13.33% en utilisant l'algorithme de reformulation booléen. L'utilisation de FPIRM-Expand Query et FPIRM-ReformulateQuery a amélioré 40 requêtes sur 63 par rapport à FPIRM-Base. De l'autre côté, FPIRM-ReformulateQuery a amélioré 7 requêtes uniquement par rapport à FPIRM-ExpandQuery. Ces 7 requêtes ont été améliorées suffisamment pour augmenter de 4,21% le MAP de FPIRM-ExpandQuery. Après la clarification de cette différence entre FPIRM-ExpandQuery and FPIRM-ReformulateQuery, nous comparons par la suite les résultats de FPIRM-Base par rapport à FPIRM-Reformulate Query uniquement. L'analyse de la courbe Rappel-Précision de FPIRM-Base et FPIRM-ReformulateQuery (cf. Fig. 1) montre une amélioration de 19,97% de la précision interpolée au niveau 10% du rappel, cette amélioration diminue pour les niveaux supérieurs à ce point.

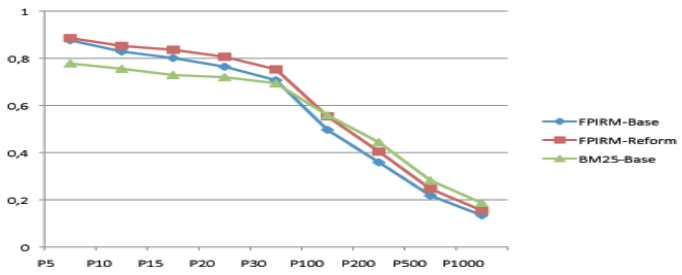

Figure 2. *La précision aux différentes positions de la liste de résultats pour FPIRM-Base, FPIRM-ReformulateQuery et BM25*

Pour comprendre le comportement de notre méthode, nous présentons également la courbe de la précision aux différentes positions de la liste de résultats de nos expériences par rapport aux résultats de BM25 sur les requêtes originales. La figure Fig. 2 montre que nous avons réussi à dépasser FPIRM-Base et BM25 pour les 100 premières positions. Après ce point, notre approche est toujours meilleure que FPIRM-Base mais elle devient légèrement moins bien que BM25.

4.2. *Discussion*

Les résultats de la section précédente ont montré que FPIRM-ExpandQuery et FPIRM-ReformulateQuery ont améliorécbhn la MAP de FPIRM-Base. La complétion des résultats de FPIRM-Base a protégé la bonne précision du modèle quand les requêtes originales ont obtenu assez de résultats. Quand les requêtes originales obtiennent peu (ou pas) de résultats, les requêtes transformées se sont bien comportées. Cela peut être constaté par la Figure 3 qui montre que FPIRM-ReformQuery a une meilleure précision que FPIRM-Base pour les 100 premiers documents. D'un autre côté, bien que la transformation des requêtes en utilisant l'algorithme de la section 4.3 n'ait pas amélioré les 63 requêtes, elle était quand même capable d'augmenter le MAP total. Nous pensons que cet algorithme est utile quand les requêtes ne sont pas bien formulées, c'est-à-dire quand un AND est utilisé alors que la vraie relation est OR, ce qui n'était pas souvent le cas pour les requêtes ad-hoc d'INEX2009. L'amélioration importante du MAP en utilisant cette technique, malgré le petit nombre de requêtes améliorées, est générée à cause du fait que cette technique réduit la haute sélectivité du modèle en remplaçant les conjonctions avec des disjonctions dans certains endroits de la requête. Il est important de noter que BM25 n'est pas un modèle conjonctif, il interprète les relations implicites entre les termes de la requête en tant que OR, alors que FPIRM les interprète comme AND. Ce fait explique pourquoi le rappel de notre méthode n'a pas dépassé celui de BM25.

5. Conclusion et perspectives

Dans cet article nous avons étudié l'utilisation d'une ressource lexicale pour la désambiguïsation et pour l'expansion automatique des requêtes pour le cas d'un modèle de recherche par proximité FPIRM qui est un modèle très sélectif. Le but de notre approche était d'améliorer la performance du modèle sans prendre le risque de détruire sa bonne précision. Nos expériences ont montré qu'une simple utilisation de WordNet peut améliorer considérablement le MAP et le rappel du modèle. La précision a été également améliorée pour les requêtes qui obtiennent très peu de résultats. La prochaine étape de ce travail sera d'abord de comparer les méthodes de similarité entre concepts pour la phase de désambiguïsation, pour mesurer l'efficacité des méthodes basées sur le contenu en information. Nous pensons tester notre approche avec d'autres collections de test et/ou avec d'autre ressources sémantiques comme YAGO par exemple. Un autre aspect qui nous intéresse est d'intégrer la notion de pondération des termes dans le modèle FPIRM, et d'étudier la différence entre cette idée et l'approche de complétion de résultats présentée dans cet article.

Références

Beigbeder M., Mercier A., « An information retrieval model using the fuzzy proximity degree of term occurrences », *Proceedings of SAC '05*, , 2005, page 1018, ACM Press.

Hawking D., Thistlewaite P., « Relevance Weighting Using Distance Between Term Occurrences », rapport, 1996.

Manning C. D., Raghavan P., Schütze H., *Introduction to Information Retrieval*, Cambridge University Press, 2008.

Mitchell P. C., « A note about the proximity operators in information retrieval », *SIGIR Forum*, vol. 9, 1973, p. 177–180, ACM.

Navigli R., « Word sense disambiguation : A survey », *ACM Computing Surveys*, vol. 41, 2009, p. 1-69, ACM Press.

Palmer M., Wu Z., « VERB SEMANTICS AND LEXICAL SELECTION », *ACL*, Association for Computational Linguistics Stroudsburg, PA, USA, 1994.

Pedersen T., « WordNet : : Similarity-Measuring the Relatedness of Concepts », *Demonstration Papers at HLT-NAACL*, , 2004, p. 38-41, Association for Computational Linguistics Stroudsburg, PA, USA.

Rasolofo Y., Savoy J., « Term Proximity Scoring for Keyword-Based Retrieval Systems », *ECIR'03*, Springer-Verlag Berlin, Heidelberg, 2003, p. 207-218.

Salton G., McGill M. J., *Introduction to Modern Information Retrieval*, McGraw-Hill, Inc. New York, 1986.