Recherche d'information dans des documents structurés par proximité des termes

Michel Beigbeder

École Nationale Supérieure des Mines de Saint-Etienne
158, cours Fauriel
F-42023 Saint-Etienne cedex 2
mbeig@emse.fr

RÉSUMÉ. *Nous présentons une méthode pour calculer un score d'un élément quelconque d'un document structuré qui prend en compte la proximité des termes de la requête dans le texte du document. Plus précisément nous définissons autour de chaque occurrence d'un terme de la requête une fonction d'influence. Pour une occurrence qui apparaît dans le texte proprement dit, cette fonction d'influence décroit linéairement de 1 à 0 selon la distance à l'occurrence. Lorsqu'un terme de la requête apparaît dans le titre d'une (sous-)section d'un document structuré, son influence est uniformément 1 du début à la fin de la (sous-)section. Nous utilisons des requêtes booléennes et les fonctions d'influence sont combinées à chaque nœud ET et OU de l'arbre de la requête selon le modèle de la logique floue. Le score d'une partie quelconque de document est la normalisation de la somme de la fonction d'influence résultante à la racine de l'arbre de la requête sur l'intervalle associé à cette partie. Nous présentons et commentons les résultats obtenus dans le cadre de la tâche ad'hoc de la campagne INEX 2006.*

ABSTRACT. *We present a method for scoring any part of a structured document that takes into account the proximity of the query terms in the text of the document. More precisely we define around each occurrence of a query term an influence function. For an occurrence appearing in the text itself, this influence function is linearly decreasing from 1 to 0 depending on the distance to the occurrence. When a query term happens to appear in a (sub-)section title of a structured document its influence is uniformly 1 from the beginning to the end of the (sub-)section. We use boolean queries and these influence functions are combined at each node AND or OR of the query tree by using fuzzy logic. The score of any part of a document is the normalization of the summation of the resulting influence function at the root of the query tree on the range of this part. We present and comment the results obtained within the 2006 INEX ad'hoc track.*

MOTS-CLÉS : *Recherche d'information, documents structurés, proximité des termes, logique floue.*

KEYWORDS: *Information retrieval, structured documents, term proximity, fuzzy logic.*

1. Introduction

Le besoin d'outils pour la recherche d'information est maintenant bien établi, et les outils disponibles sont maintenant largement acceptés et connus des utilisateurs. Cependant la très grande majorité de ces outils et des méthodes sous-jacentes sont dédiés à des documents *plats* alors que la majorité des documents sont créés avec une structure, au moins implicite.

De plus la plupart des méthodes utilisées en recherche d'information sur des textes plats ne prennent pas en compte la structure basique d'un texte, à savoir sa linéarité, autrement dit la juxtaposition des mots. En fait, ces méthodes utilisent des données statistiques comme la fréquence des termes (à la fois dans chaque document et à l'intérieur de la collection) et la longueur des documents. Cependant il y a eu quelques propositions pour utiliser la position des occurrences des termes dans le texte, soit par l'utilisation explicite d'opérateurs de proximité dans le langage de requête ou par une attribution de score qui se base sur la proximité des termes de la requête dans les documents. Nous présenterons un tour d'horizon sur ces méthodes en section 2. D'un point de vue qualitatif, ces méthodes de recherche d'informations donnent des résultats comparables à la méthode Okapi BM-25. L'une d'entre elles obtient de meilleures performances, avec l'appui de requêtes booléennes, et elle est à la base de l'extension aux documents structurés que nous proposons ici.

Pour ce qui concerne la structure logique qui est la structure à laquelle il est fait habituellement allusion lorsqu'on utilise l'expression « documents structurés », ce n'est que récemment qu'il existe une représentation suffisamment répandue – à savoir le langage XML – de sorte que de grandes collections de documents structurés aient été créées et diffusées grâce aux campagnes de recherche d'informations INEX[1]. Ainsi il est possible d'expérimenter en vraie grandeur des idées pour la recherche d'information dans des collections de documents structurés et de créer de nouvelles méthodes.

Dans cet article, nous présentons une extension à la recherche d'information structurée d'une méthode d'attribution de score basée sur la proximité des occurrences des termes de la requête originellement définie pour les documents plats. Ce modèle permet de calculer un score pour n'importe quel segment de texte. En particulier, dans le cas des documents structurés, on peut attribuer un score à tout élément d'un document hiérarchique, quel que soit son niveau. Dans la section 3, nous présentons le modèle de document avec lequel cette méthode opère, et dans la section 4 la méthode elle-même. Les expériences menées dans le cadre de la campagne INEX 2006 et les résultats obtenus font l'objet de la section 5.

2. Méthodes de recherche d'information utilisant la proximité

L'idée d'utiliser la proximité des termes de la requête pour classer les documents a été d'abord implémentée dans les systèmes booléens avec les opérateurs ADJ et

1. http://inex.is.informatik.uni-duisburg.de/

NEAR, le premier demandant que les termes qu'il connecte soient adjacents, le deuxième qu'ils soient dans un intervalle de longueur bornée. L'introduction de ces opérateurs dans les systèmes booléens était motivée par le besoin de donner dans les requêtes des expressions strictes (avec ADJ) ou relaxées (avec NEAR). Ces opérateurs sont encore utilisés dans les outils dédiés à la recherche dans les catalogues de bibliothèques. Cependant du point vue de la modélisation, ils souffrent de deux handicaps qui ont sans doute freiné leur utilisation dans les outils qui travaillent sur le texte intégral. Le premier est qu'ils sont intimement liés au modèle de recherche booléen qui ne permet pas de classer les documents retrouvés. Le second est qu'ils ne s'adaptent pas bien au modèle de requête booléen lui-même parce que ces opérateurs ne permettent de connecter que des termes et qu'ils ne peuvent pas être étendus de façon consistante [MIT 74] à des sous-expressions booléennes.

Des idées plus récentes pour utiliser la proximité des mots clés ont été développées et n'ont pas ces limitations. Pour ce qui concerne la deuxième, les requêtes reconnues par le modèle de requête sont soit basées sur des sacs de termes, soit n'utilisent que les opérateurs booléens basiques (ET et OU). Pour la première limitation, toutes ces méthodes donnent un score aux documents sur la base des positions des occurrences des termes en prenant en compte leur proximité. Nous allons maintenant présenter le principe de ces méthodes.

2.1. *Méthodes basées sur les intervalles*

Pour leur participation à la campagne TREC-4, les équipes Clarke *et al.* [CLA 95] et Hawking *et al.* [HAW 95] ont développé des méthodes similaires pour classer les documents en prenant en compte la proximité des termes. L'idée est de sélectionner des intervalles de texte qui contiennent tous les mots clés ; d'attribuer un score partiel à ces intervalles (plus l'intervalle est court, plus le score est élevé) ; et d'additionner les contributions de tous les intervalles sélectionnés pour donner un score au document.

Leurs deux méthodes diffèrent dans les critères de sélection des intervalles : pour Clarke *et al.* les intervalles ne doivent pas être imbriqués les uns dans les autres. Pour cela seuls les plus courts sont conservés. Pour Hawking *et al.*, chaque occurrence d'un mot clé est le début d'un intervalle sélectionné, là encore le plus court à partir de cette position qui contient tous les mots clés. Ainsi, s'il y a deux occurrences successives d'un même terme sans occurrence d'un autre mot-clé entre elles, deux intervalles imbriqués sont sélectionnés pour cette deuxième méthode, alors que le premier de ces intervalles serait éliminé par la première méthode.

Les deux méthodes diffèrent aussi dans le calcul du score d'un document. Clarke *et al.* choisissent un score inversement proportionnel à la longueur de l'intervalle, et Hawking *et al.* basent le score d'un intervalle sur l'inverse du carré de la longueur de l'intervalle.

L'idée d'utiliser les intervalles a été revisitée par Rasolofo *et al.* [RAS 03]. Leur proposition est basée sur la méthode *Okapi* et ils ajoutent un score supplémentaire à

la probabilité Okapi. Ce score additionnel est basé sur les intervalles qui contiennent une paire de termes. Chaque intervalle de longueur plus courte qu'un seuil, fixé à 6 dans leurs expérimentations, et qui contient des occurrences d'au moins deux termes de la requête contribue à ce score additionnel.

2.2. *Modèle à zone d'influence*

Beigbeder *et al.* [BEI 05] ont dévelopé un modèle de recherche d'information basé sur la *proximité floue des mots clés*. Plus précisemment, chaque occurrence d'un mot clé a une influence sur son voisinage. Cette influence atteint son maximum de 1 à la position du mot clé et décroit avec la distance à cette position. La fonction la plus simple avec ce comportement est une fonction triangulaire. De plus il est aisé de contrôler une telle fonction pour définir la largeur de la zone d'influence d'une occurrence de terme avec la largeur de la base du triangle. Nous appellerons k la demi largeur de cette base.

Étant données les fonctions d'influence de toutes les occurrences d'un terme dans un document, elles sont combinées avec un opérateur $\max$. Si la fonction d'influence est symétrique, cela consiste à considérer que l'influence du terme en une position quelconque du texte est l'influence de la plus proche des occurrences.

Le modèle de langage de requête est le modèle booléen classique avec les opérateurs ET, OU et NON (ni de ADJ, ni de NEAR). Les fonctions d'influence des différents termes sont combinées dans l'arbre de la requête booléenne avec l'interprétation floue des opérateurs booléens : respectivement le $\max$ et le $\min$ pour les opérateurs d'union et d'intersection.

Prenons l'exemple du document `X X X X A X X X B X X X X X` où il y a une occurrence du terme A (resp. B) à la position 5 (resp. 9) et où X dénote n'importe quel terme qui n'est ni A ni B. La figure 1 montre les fonctions d'influence aux termes A et B dans ce document exemple (avec $k = 5$) et leur combinaison avec un $\min$ correspondant à une requête conjonctive entre ces termes A et B.

Finalement, le score d'un document est la sommation de la fonction d'influence sur toutes les positions dans le texte. Cela consiste à évaluer l'aire de la surface située en dessous de la fonction d'influence associé à la racine de l'arbre de la requête. Dans notre exemple, cette surface est le triangle défini par la courbe A ET B.

D'un point de vue qualitatif, ce modèle a été comparé [MER 06] avec les modéles décrits dans la section 2.1 et avec le modèle Okapi BM-25. Sur la collection utilisée dans la campagne CLEF[2] 2004 les méthodes à base de proximité obtiennent des résultats analogues à Okapi lorsqu'elles sont utilisées avec des requêtes générées automatiquement à partir des champs `title` des besoins d'informations. Les résultats de ces méthodes deviennent meilleurs comparés à Okapi lorsque des requêtes sont construites manuellement à partir des champs `title` et `descr`; en particulier

2. http://clef.isti.cnr.it/

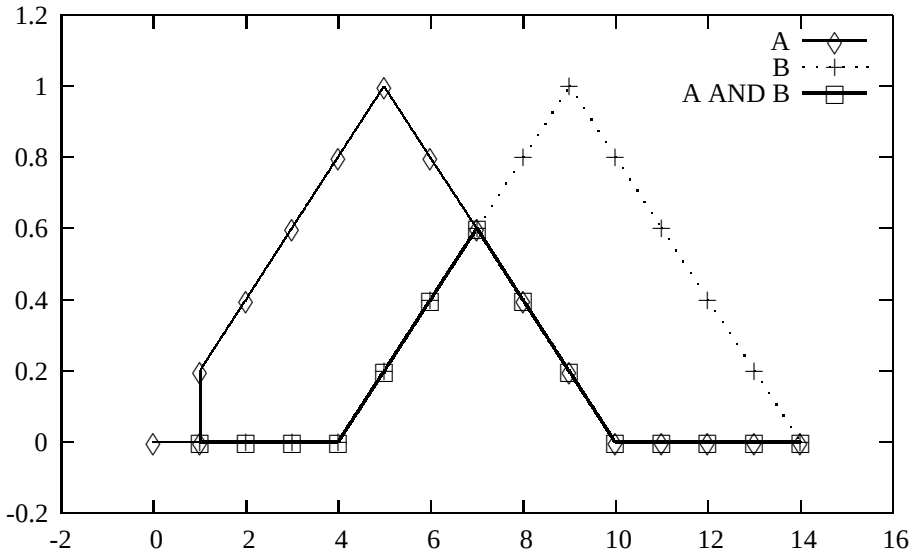

Figure 1. *Proximités aux termes A et B et leur combinaison pour la requête A ET B : en abscisse la position des mots dans le texte, en ordonnée la proximité floue.*

les requêtes construites pour le modèle à zone d'influence sont vraiment des requêtes booléennes.

Dans la section suivante, nous présentons une extension de ce modèle à zone d'influence aux documents structurés.

3. Modèle de document struturé

Notre travail est pragmatique en ce qui concerne la structure des documents. Nous voulons prendre en compte la structure de base de beaucoup de modèles de documents, à savoir la structure hiérarchique avec des sections imbriquées et associées à des titres. C'est la base pour les articles scientifiques et les documents techniques mais aussi pour beaucoup de documents moins formalisés. Nous ignorons tout autre type de structure, que ce soient les listes ou les emphases par exemple. Comme un cas particulier, nous considérons qu'un document dans sa totalité est situé au plus haut niveau dans la hiérarchie de sectionnement.

Un autre aspect est que les notions de sections et de titres sont le plus souvent totalement liées, à tel point que dans les styles LaTeX il n'existe que des commandes de sectionnement (`\section`, `\subsection`, ...) et les titres sont des paramètres de ces commandes.

Ainsi la base pour nos modèles de documents est la famille de documents qui pourraient être codés dans les styles LaTeX uniquement avec les commandes de sectionnement. En voici un exemple :

```
\title{titre 1}

bla bla
  \section{title 2}
  bla bla
    \subsection{title 3}
    bla bla
  \section{title 4}
  bla bla
```

Cet exemple pourrait être codé en XML ainsi :

```
<section><title>title 1</title>
bla bla
  <section><title>title 2</title>
  bla bla
    <section><title>title 3</title>
    bla bla
    </section>
  </section>
  <section><title>title 4</title>
  bla bla
  </section>
</section>
```

Formellement, la grammaire de notre modèle de documents est :

```
document        = section
section         = '<section>''<title>' title_text '</title>'
                     section_content
                  '</section>'
section_content = (section_text | section)*
```

4. Influence des occurrences de termes

Dans le modèle présenté en section 2.2, l'influence d'un terme était définie uniquement dans le cadre d'un texte plat. Avec notre modèle de document structuré, nous devons modéliser l'influence d'une occurrence d'un terme en fonction de la partie

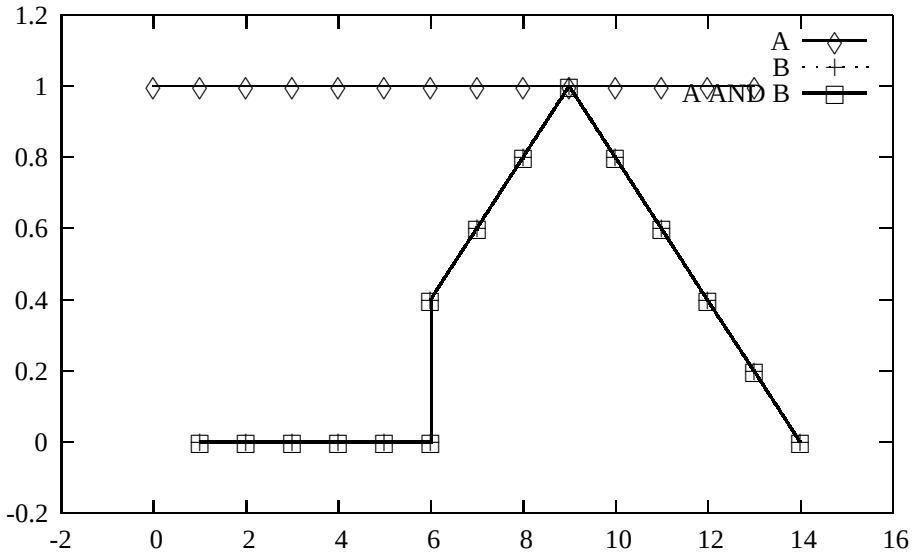

Figure 2. *Limitation de l'influence d'une occurrence à la partie* $section\_text$ *dans laquelle elle apparaît (proximité à B) ; propagation d'un terme du titre (proximité à A) : en abscisse la position des mots dans le texte, en ordonnée la proximité floue.*

structurelle du document dans laquelle elle apparaît. Comme notre modèle de document est très simple, il n'y a que deux cas possibles : l'occurrence apparaît dans une partie $section\_text$ ou dans une partie $title\_text$.

Pour une occurrence de terme qui apparaît dans une partie $section\_text$, nous définissons l'influence comme dans le cas du texte plat : une décroissance linéaire en fonction de l'éloignement de l'occurrence. Cependant une nouvelle contrainte est prise en considération, l'influence est limitée *strictement* à la partie $section\_text$ dans laquelle l'occurrence apparaît. L'influence ne traverse donc pas les frontières des sections que ce soit sur les parties $section\_text$ des sections englobantes ou des sections englobées.

Prenons un exemple de document avec le même contenu textuel que l'exemple de la section 2.2 mais avec quelques balises structurelles : `<section><title>X X X A</title> X X X B X X X X X</section>`. L'occurrence du terme B est dans la partie $section\_text$ de la section. La figure 2 montre l'application de cette limitation à la section sur la fonction d'influence associée à l'occurrence du terme B.

Pour les occurrences de terme qui apparaissent dans les parties $title\_text$, leur influence est au contraire étendue à tout le contenu de la section et récursivement aux sous-sections qu'elle contient avec une valeur de 1 sur toute la portée de la section

concernée. L'influence est aussi limitée strictement à la section, et donc la fonction d'influence prend la valeur 0 en dehors de la portée de la section.

Toujours avec notre exemple de document, l'occurrence du terme A est dans la partie $title\_text$ de la section. La figure 2 montre la propagation de l'influence de cette occurrence à toute la section.

Autrement, comme dans le modèle présenté en section 2.2, nous utilisons un langage de requête booléen et nous combinons les fonctions d'influence avec des $\min$ et $\max$ sur les nœuds de l'arbre de la requête booléenne. Un score est calculé par sommation de la fonction d'influence à la racine de l'arbre sur l'intervalle associé au document entier ou à une partie du document. Ce score est normalisé par le score maximum qui pourrait être atteint par cette partie, autrement dit sa longueur. Ce maximum peut être effectivement atteint, par exemple si tous les termes de la requête apparaissent dans le titre de la section.

5. Expérimentation et implémentation

Nos expérimentations ont été faites dans le cadre de la campagne d'évaluation INEX 2006 [3].

5.1. *Conversion des documents*

Les documents utilisés dans cette campagne d'évaluation sont écrits en XML. Cependant la struture de ces documents est plus complexe que celle de notre modèle de documents de la section 3, par exemple 1506 balises différentes y sont utilisées.

La majeure partie de la conversion consiste à conserver le texte et les balises de sectionnement et de titre avec leur balise de fermeture correspondante. Cela peut être facilement mis en œuvre avec un processeur `xslt`, mais des choix non triviaux doivent être faits à propos du contenu textuel, en particulier en ce qui concerne les espaces. Malheureusement, au niveau syntaxique aucun choix correct ne peut être fait à cause de l'inconsistance dans l'utilisation de certaines balises. Par exemple, le document numéro 1341796 contient l'extrait suivant (les attributs des balises `collectionlink` ont été supprimés) :

```
This is a
<emph3>List of
<collectionlink ...>poison</collectionlink>ings</emph3>in
alphabetical order of victim. It also includes confirmed attempted
and fictional poisonings. Many of the people listed here committed
or attempted to commit
<collectionlink ...>suicide</collectionlink>by
poison;
```

3. `http://inex.is.informatik.uni-duisburg.de/2006/`

```
others were poisoned by others.
```

La question concerne l'insertion ou non d'un espace après la balise fermante `collectionlink`. Si un espace est inséré, le texte suivant est obtenu (l'erreur est en italique) :

```
This is a List of poison ings in alphabetical order of victim.
It also includes confirmed attempted and fictional poisonings.
Many of the people listed here committed or attempted to commit
suicide by poison ; others were poisoned by others.
```

Ce qui est correct pour la deuxième instance de cette balise, mais pas pour la première. Au contraire, si des espaces ne sont pas ajoutés après ces balises fermantes, voici le texte obtenu avec déplacement de l'erreur :

```
This is a List of poisonings in alphabetical order of victim. It
also includes confirmed attempted and fictional poisonings. Many
of the people listed here committed or attempted to commit suicideby
poison ; others were poisoned by others.
```

De plus un choix doit être fait pour tous les types de balise, et dans les exemples ci-dessus, les balises `emph3` ont été remplacées par des espaces.

Comme aucun choix consistant ne pouvait être fait, nous remplaçons chaque instance de balise par un espace.

Du point de vue de la sructure, bien que les documents de la Wikipedia ne suivent pas une DTD, nous avons testé la présence de titres avec les sections. Sur les 1 610 197 sections qui apparaissent dans les différents documents, seulement environ 1% n'ont pas de titre et aucune n'en a plusieurs. Autrement dit 99% des sections ont bien un titre et un seul. Pour les sections sans titre, il n'y a tout simplement pas de propagation des termes du titre.

5.2. *Indexation*

L'implémentation de notre système est basée sur l'outil LUCY[4] dans sa version 0.5.4. Bien que ce soit une version ancienne et rendue obsolète par différentes versions de son successeur ZETTAIR[5], ses capacités sont suffisantes pour le volume de la collection Wikipedia. C'est de plus une bonne base pour le modèle de recherche d'information que nous proposons car cet outil conserve dans son index les positions de toutes les occurrences de tous les termes trouvés dans les documents. De plus son analyseur syntaxique reconnaît la syntaxe des balises XML. Dans la phase d'indexation, nous avons ajouté le code nécessaire pour garder la trace des positions et de l'emboîtement des balises `section` et `title` (toutes les autres balises ont été supprimées dans la phase précédente lors de la conversion des documents dans notre modèle purement hiérarchique). Il n'y a aucune élimination de termes (mots vides) ni de lemmatisation.

4. http://www.seg.rmit.edu.au/lucy/
5. http://www.seg.rmit.edu.au/zettair/

5.3. *Construction des requêtes*

Il serait possible et facile de construire automatiquement un jeu de requêtes booléennes en combinant conjonctivement tous les termes qui apparaissent dans le champ `title` des besoins d'information. Comme notre méthode est très sélective il y aurait très peu de résultats dans les listes de documents retrouvés par le système. Il faut donc soit relaxer les contraintes de ces requêtes purement conjonctives, soit avoir une méthode moins sélective. En conservant les requêtes purement conjonctives, il serait possible d'agrandir l'ensemble des résultats retournés en utilisant une lemmatisation à la fois lors de l'indexation et lors de l'analyse des requêtes. Nous n'avons pas testé cette solution mais avons choisi de construire des requêtes relaxées d'une façon manuelle.

Sur la base des requêtes construites par conjonction des termes du champ `title`, parfois quelques termes ont été supprimés, mais le plus souvent la relaxation a été faite en étendant les requêtes par des disjonctions sur des variations sur les termes initiaux. Ces variations sont soit simplement des variations flexionnelles (pluriel et singulier, par exemple) ou dérivationnelle (verbe, nom, adjectif, etc.) et même dans certains cas sémantiques par adjonction de synonymes ou de concepts reliés.

Par exemple le champ titre du besoin d'information numéro 289 est :

```
emperor "Napoleon I" Polish
```

Par simple conjonction, la requête serait (le symbole '&' est utilisé pour dénoter l'opérateur conjonctif) :

```
emperor & Napoleon & I & Polish
```

Différentes relaxations peuvent être introduites, par exemple :

```
emperor & Napoleon & Polish
Napoleon & Polish
Napoleon & (Polish | Poland)
```

En utilisant de plus les termes trouvés dans les champs `description`, `narrative` et `ontopic_keywords`, d'autres requêtes peuvent être formulées, par exemple :

```
Napoleon & (Polish | Poland | Laczynska | Malewski | Poniatowski)
```

Nous avons construit deux jeux de requêtes, le jeu de requêtes courtes dans lesquelles nous avons utilisé seulement les termes du titre avec peu de relaxation, et le jeu de requêtes étendues pour lesquelles des termes de tous les champs des besoins d'information ont été utilisés avec des relaxations de toutes les sortes que nous avons évoquées : flexionnelles, dérivationnelles et sémantiques.

5.4. *Runs*

Étant donnée une requête et une valeur pour le paramètre k, notre méthode est à même de calculer l'influence des termes de la requête pour toutes les feuilles de l'arbre

et de les combiner récursivement jusqu'à la racine de l'arbre. Avec nos deux jeux de requêtes, nous avons lancé notre système avec deux valeurs de k, 50 et 200. Comme seulement trois soumissions pouvaient être faites dans la campagne INEX, nous avons soumis ces quatre combinaisons à l'exception de celle avec le jeu de requêtes étendues et $k = 50$.

Pour la participation à la tâche THOROUGH dans laquelle on doit rendre des éléments indépendamment les uns des autres, nous avons calculé le score de tous les documents, de leurs sections et sous-sections et nous avons classé tous ces éléments selon leur score.

Pour la tâche BEST IN CONTEXT il faut rendre un seul élément pour un document pertinent ou contenant au moins un élément pertinent. Comme notre fonction d'influence à la racine de l'arbre de la requête mesure pour chaque position la proximité avec la requête, nous cherchons la position où le maximum de cette fonction est atteint. À partir de cette position nous descendons dans la structure jusqu'à l'élément le plus petit — donc le plus spécifique — qui contient cette position.

Dans la tâche FOCUSED il ne faut rendre pour un document donné que des éléments qui ne se recouvrent pas les uns les autres. Nous avons adopté une approche très conservative pour cela en ne rendant, comme dans la tâche BEST IN CONTEXT qu'un seul élément par document. Nous trions d'abord les documents selon leur score et choisissons parmi ses éléments celui qui a le score maximal.

6. Résultats

Quelque soit la tâche, nos meilleurs résultats ont été obtenus avec le jeu de requêtes étendues. Avec le jeu de requêtes courtes, les deux listes de résultats obtenus respectivement avec $k = 50$ et $k = 200$ sont très similaires et donnent des résultats nettement moins bons que ceux issus du jeu de requêtes étendues. Cela montre l'importance du mécanisme de relaxation des requêtes par expansion. Concernant le réglage du paramètre k, d'autres expériences devront être menées pour pouvoir tirer des conclusions. Dans la suite nous commentons les résultats obtenus avec le jeu de requêtes étendues et $k = 200$.

La figure 3 montre les résultats tels qu'ils sont résumés par la mesure ep-gr [LAL 07] pour le jeu de requêtes étendues et $k = 200$. Aux premiers niveaux de rappel les résultats sont plutôt bons mais il y a une décroissance rapide de qualité et cette dernière est quasiment à zéro après 0,1. Le même comportement se trouve aussi pour la tâche FOCUSED que ce soit mesuré avec *overlap On* (cf. fig. 4) ou *overlap Off* (cf. fig. 5). La mesure utilisée pour tracer ces figures se base sur la liste des 5 premiers documents retournés par les systèmes. Lorsque la mesure utilise plus de documents notre méthode se compare de moins en moins favorablement par rapport aux autres méthodes testées dans le cadre de la campagne INEX 2006. Nous développons quelques explications pour ce comportement dans la suite.

Tableau 1. *Distribution des requêtes en fonction de la longueur de leur liste de résultats.*

			0	1 à 10	10 à 100	100 à 1000	plus de 1000
Thorough	k=50	courtes	13	21	50	37	4
Thorough	k=200	courtes	7	20	46	44	8
Thorough	k=200	étendues		7	53	55	10
Foc./Best	k=50	courtes	13	40	48	22	2
Foc./Best	k=200	courtes	7	30	60	26	2
Foc./Best	k=200	étendues		15	75	32	3

Nos requêtes sont des conjonctions de termes qu'un document (ou une partie de document) doit contenir pour obtenir un score non nul. De plus les occurrences de ces différents termes doivent être proches les unes des autres. De ce fait cette méthode est très sélective, et beaucoup plus que celles qui, par exemple basées sur un modèle vectoriel, ont un comportement plutôt disjonctif. Le tableau 1 montre la distribution des requêtes en fonction de la longueur de la liste de résultats fournie par notre méthode. Ce tableau met en évidence que ces listes sont plutôt courtes. La longueur de ces listes augmente lorsque k croît — ce qui est trivial à expliquer — et aussi lors de l'utilisation du jeu de requêtes étendues du fait que ces dernières relâchent les contraintes sur les documents à retrouver. On peut aussi voir sur ce tableau qu'une très large majorité des listes de résultats sont bien plus courtes que la limite de 1500 imposée par les organisateurs de INEX.

La simplification de la structure que nous avons mise en œuvre ne nous est pas favorable dans le cadre d'une comparaison avec d'autres systèmes qui ont travaillé sur la structure initiale des documents. En particulier dans la tâche THOROUGH, nous ne rendons que des éléments de type `section` alors que les résultats des autres participations contiennent potentiellement beaucoup plus d'éléments à classer.

Enfin notre mécanisme de propagation a été perturbé par des documents « étranges ». Par exemple le document numéro 192509 contient beaucoup de texte entre autre dans des listes elles-mêmes incluses entre des balises `<title>` et `</title>`. Du fait de l'élimination des balises de listes, notre simplification de structure construit un titre très long et le mécanisme de propagation fait que l'influence de ces très nombreux termes est étendue à tout ce titre et sa section associée. S'il se trouve que pour une requête, tous les termes apparaissent dans ce titre erroné la section atteint le score maximal de 1 sans être toutefois pertinente. Indépendamment du côté bogué de ce document, cela met en évidence que le mécanisme de propagation que nous avons choisi peut donner trop d'importance aux termes qui apparaissent dans les titres.

7. Conclusion

Nous avons présenté dans cet article les idées utilisées pour notre participation à la campagne INEX 2006 dans la tâche *ad'hoc*. Notre méthode est basée sur la proximité

des mots clés dans le texte des documents et sur la propagation des termes des titres à toute la section associée à ce titre. Les résultats obtenus sont plutôt bons pour ce qui concerne la précision aux premiers niveaux de rappel. Les mesures faites par les métriques sont à notre désavantage parce que nos listes de résulats sont très courtes, à la fois parce que notre méthode est très sélective et aussi parce qu'elle travaille sur un modèle de document simplifié où nous ne considérons que certains types d'éléments dans les documents de la collection.

Nous avons aussi montré que cette méthode nécessite un mécanisme d'expansion de requête pour relâcher les contraintes sur les documents à retrouver. De nouvelles expériences doivent être menées pour étudier l'influence du paramètre k qui contrôle la largeur de la zone d'influence d'une occurrence de terme dans le texte des documents. Il faut aussi améliorer le mécanisme de propagation des mots du titre qui est très simple et engendre un comportement qui favorise trop les documents à titre long.

Remerciements

Ces travaux sont soutenus par le projet *Web Intelligence* du cluster *ISLE* financé par la région Rhône-Alpes.

8. Bibliographie

[BEI 05] BEIGBEDER M., MERCIER A., « An Information Retrieval Model Using the Fuzzy Proximity Degree of Term Occurences », HADDAD H., LIEBROCK L. M., OMICINI A., WAINWRIGHT R. L., Eds., *SAC*, ACM, 2005, p. 1018–1022.

[CLA 95] CLARKE C. L. A., CORMACK G. V., BURKOWSKI F. J., « Shortest Substring Ranking (MultiText Experiments for TREC-4) », Harman [HAR 95].

[HAR 95] HARMAN D. K., Ed., *The Fourth Text REtrieval Conference (TREC-4)*, Department of Commerce, National Institute of Standards and Technology, 1995.

[HAW 95] HAWKING D., THISTLEWAITE P., « Proximity Operators - So Near And Yet So Far », Harman [HAR 95].

[LAL 07] LALMAS M., KAZAI G., KAMPS J., PEHCEVSKI J., PIWOWARSKI B., ROBERTSON S., « INEX 2006 evaluation measures », FUHR N., LALMAS M., TROTMAN A., Eds., *Comparative Evaluation of XML Information Retrieval Systems*, n° 4518 Lecture Notes in Computer Science, Springer-Verlag, 2007, p. 20–34.

[MER 06] MERCIER A., BEIGBEDER M., « Calcul de pertinence basée sur la proximité pour la recherche d'information », *Document numérique*, vol. 9, n° 1, 2006, p. 43–60.

[MIT 74] MITCHELL P. C., « A note about the proximity operators in information retrieval », *Proceedings of the 1973 meeting on Programming languages and information retrieval*, ACM Press, 1974, p. 177–180.

[RAS 03] RASOLOFO Y., SAVOY J., « Term Proximity Scoring for Keyword-based Retrieval Systems », SEBASTIANI F., Ed., *ECIR*, vol. 2633 de *Lecture Notes in Computer Science*, Springer, 2003, p. 207-218.

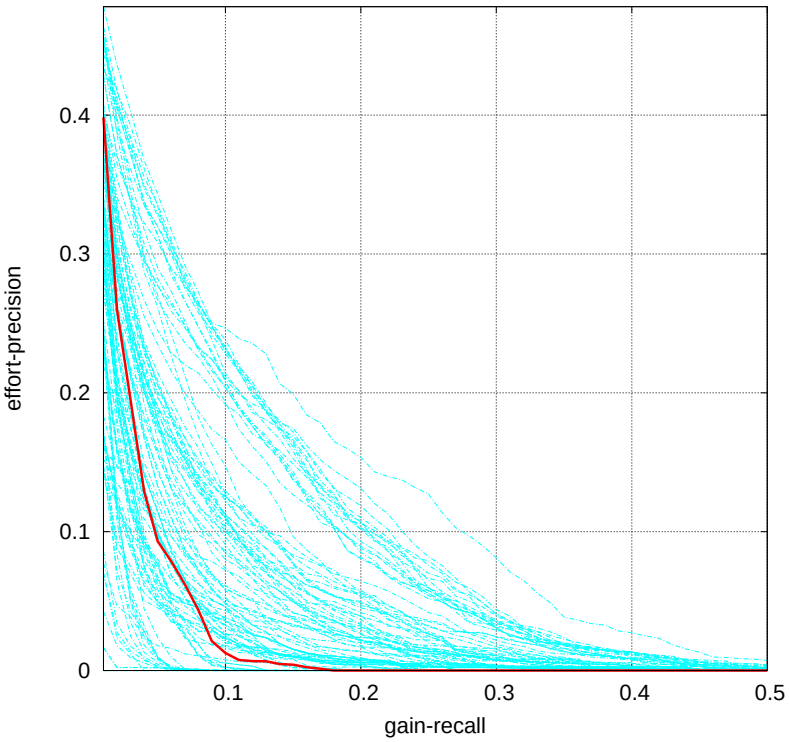

Figure 3. *INEX 2006 — Métrique : ep-gr, Quantisation : gen, Tâche : thorough, Run : title_Q_Prox200NT02.*

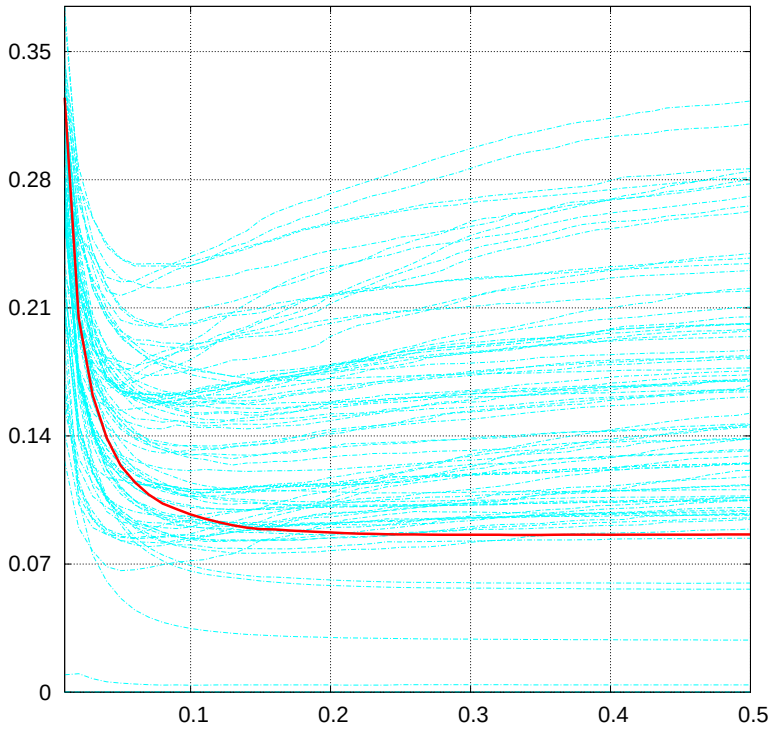

Figure 4. *INEX 2006 — Métrique : nxCG[5], Quantisation : gen, Tâche : focused, Overlap=On, Run : title_Q_Prox200NF02.*

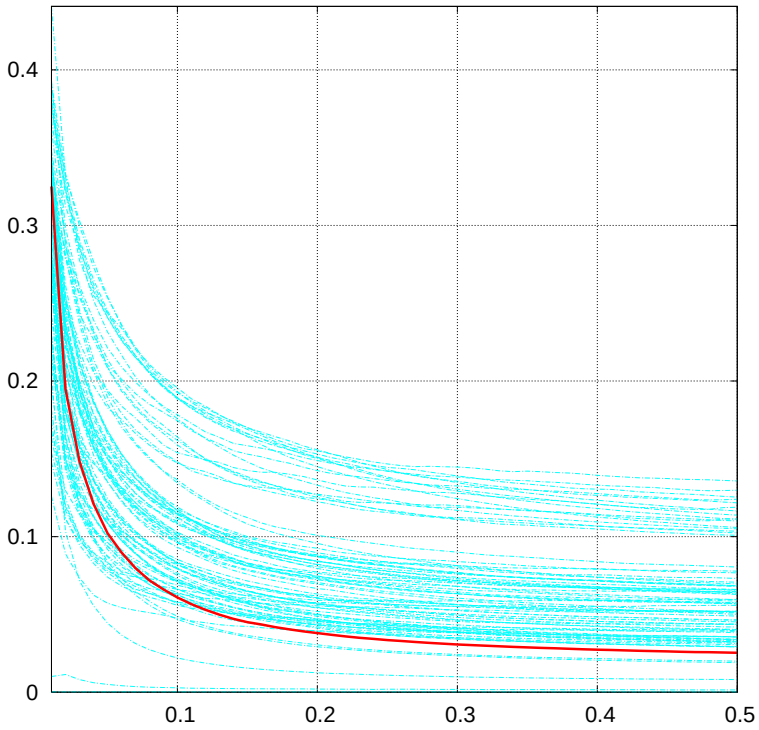

Figure 5. *INEX 2006 — Métrique : nxCG[5], Quantisation : gen, Tâche : focused, Overlap=Off, Run : title_Q_Prox200NF02.*