Structure et proximité pour la recherche documentaire

Michel Beigbeder

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

RÉSUMÉ. *Notre étude compare les performances d'un système de recherche d'information basé sur la proximité des occurrences des termes de la requête dans les documents avec un système classique de modèle de langue avec lissage de Dirichlet et le modèle Okapi BM25. Notre modèle basé sur la proximité calcule en chaque position du document une valeur d'autant plus grande que des occurrences de tous les termes de la requête sont proches de cette position. De plus pour le modèle à proximité nous testons dans le cas de documents structurés l'hypothèse que les termes apparaissant dans les titres doivent être considérés comme proches des positions de toute la section correspondant à ce titre.*

ABSTRACT. *Our study compares the effectiveness of an information retrieval system based on the proximity of the query term occurrences in the documents and an IRS based on a language model with Dirichlet smoothing and with the Okapi BM25 model. Our proximity based model computes at each position in the document a value much higher as some occurrences of all the query terms are close to this position. Moreover for the proximity based model we are testing the assumption that the title terms are to be considered as close to all the positions of the whole corresponding section.*

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

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

1. Introduction

La plupart des modèles de recherche d'information n'utilisent que des informations statistiques sur les documents pour leur attribuer un score de similarité avec la requête. Ainsi deux documents qui utilisent le même vocabulaire avec la même distribution du nombre des occurrences des termes ne sont pas distinguables par les fonctions d'attribution de score de ces modèles. Nous présentons ici notre modèle de correspondance qui utilise les positions des occurrences des termes. Pour une requête demandant les termes A et B, un modèle statistique calcule le même score pour deux documents qui contiennent une occurrence de chacun de ces termes quelles que soient leur position. Notre modèle calcule un score d'autant plus élevé que ces deux occurrences de ces deux termes sont proches. Il s'agit ici de l'utilisation de la première structure de tout texte qui est celle de la séquentialité des termes. Par ailleurs de nombreux documents, et en particulier tous les documents scientifiques et techniques, ont une structure logique hiérarchique composée de sections avec des titres. Ces titres décrivent d'une certaine façon le contenu de toute la section qui leur correspond. Cette dernière hypothèse rapportée dans un sytème de recherche d'information à base statistique va faire donner plus d'importance aux mots des titres, importance qui va se traduire en terme de poids. Dans notre modèle de proximité, la traduction de cette hypothèse est que les termes lorsqu'ils apparaissent dans un titre ont une influence qui se propage à toute la section qui correspond à ce titre.

Dans cet article, nous détaillons notre modèle à base de proximité pour les textes plats et son extension pour prendre en compte la propagation de l'influence des termes des titres. Nous présentons ensuite l'implémentation et la collection de test qui nous permettent de mettre en place une expérience de recherche d'information. Enfin nous présentons les résultats de performance de ce système dans les configurations plate et structurée et les comparons avec ceux obtenus par deux systèmes basés sur des statistiques, le modèle de Dirichlet et le modèle Okapi BM25.

2. Modèle de proximité

2.1. *Proximité à un terme*

Nous modélisons les textes comme des fonctions qui associent des positions — des entiers naturels — avec des termes du vocabulaire. Ce qui revient à numéroter les occurences des termes au fur et à mesure de la lecture du texte. Ce sont ces entiers qui servent à la numérotation que nous appelons dans la suite des *positions*.

Notre idée de base pour utiliser la proximité dans la recherche d'information consiste à définir une fonction définie pour chaque position d'un texte. L'idée est que la valeur de cette fonction soit d'autant plus grande que l'on est proche de tous les termes de la requête. Nous allons développer cette idée en examinant successivement le cas d'un terme puis de la combinaison des termes dans la requête.

Considérons d'abord le cas d'un terme de la requête et d'une occurrence de ce terme. La valeur à attribuer à une position doit être décroissante par rapport à la distance x à cette occurrence de ce terme, et de plus nous la gardons positive. En choisissant la fonction $x \mapsto \frac{\max(k-x,0)}{k}$ cette valeur peut être interprétée comme une proximité *floue* à l'occurrence du terme en cette position. La valeur du paramètre k définit la demie-base du triangle de cette fonction « triangulaire », il s'interprète donc comme la *portée* de l'occurence, puisque pour des positions à une distance plus grande que k la valeur de proximité floue est nulle. Dans nos expériences, nous prenons $k = 200$, ce qui correspond à une longueur moyenne de paragraphe. (Ce paragraphe comporte environ 120 mots).

Toujours pour un terme de la requête, lorsque plusieurs occurrences du terme sont considérées, la valeur de proximité doit être le maximum des valeurs de proximité à toutes les occurrences de ce terme ; avec la condition de décroissance pour la proximité par rapport à une occurrence du terme cela revient à dire que la proximité à un terme en une position est la proximité floue à la plus proche des occurrences de ce terme.

2.2. *Combinaison de proximités*

Lorsqu'une requête comporte plusieurs termes se pose la question de leur combinaison. Dans la majorité des modèles de recherche d'information cette combinaison est plutôt disjonctive. Ce qui signifie que le modèle n'impose pas que tous les termes soient présents dans les documents pour être retrouvés. Les outils de recherche sur le Web ont plutôt un comportement conjonctif, exigeant donc la présence de tous les termes ; ceci est probablement dû à l'avalanche de documents qui vérifient déjà cette contrainte. Nous avons choisi dans notre modèle de reporter ce choix au niveau de la requête en travaillant avec des requêtes booléennes. Ce choix est aussi naturel par rapport à la notion de proximité, puisqu'il y a une différence entre être proche de A et de B d'une part, et être proche de A ou de B d'autre part. La traduction de cette différence dans notre modèle est simple, pour une combinaison disjonctive, nous prenons le max des fonctions de proximité, et pour une combinaison conjonctive, nous prenons leur min ; ce qui correspond aux fonctions de combinaison des opérateurs de réunion et d'intersection des ensembles flous. Nous pouvons aussi considérer le cas des négations : la proximité à la négation d'un terme (d'une requête) se modélise comme 1 moins la proximité à ce terme (à cette requête).

2.3. *Modèle de requête*

Nous travaillons donc avec des requêtes booléennes avec les traditionnels opérateurs pour la conjonction, la disjonction et la négation. Nous pouvons aussi traiter des expressions (*phrases* en anglais). Pour celles-ci, nous considérons qu'une expression a une occurrence d'apparition à la position de son dernier terme.

Si une requête est une simple liste de termes, on peut la considérer au choix soit comme une requête purement disjonctive soit comme purement conjonctive. Puisque l'idée de notre modèle à travers les fonctions de pertinence locale que nous avons introduites est de donner des scores plus élevés aux documents lorsque les différents termes de la requête sont proches, le choix en l'absence d'opérateurs explicites doit se porter sur la conjonction.

2.4. *Score de pertinence d'un document*

La fonction que nous avons définie permet d'attribuer en chaque position du document une valeur qui représente la proximité à la totalité de la requête. Pour le cas conjonctif, cette proximité est donc grande quand une position est proche de tous les termes de la conjonction. Indirectement, on mesure donc la distance des occurrences des termes à cette position.

On peut interpréter cette valeur de proximité en une position comme une pertinence de cette position par rapport à la requête, et nous l'appelerons *pertinence locale*. En effet, pour le cas trivial d'un terme, on ne peut trouver de meilleur endroit pertinent à la requête dans un document que les endroits où se trouvent précisément ce terme. C'est d'ailleurs utilisé par de nombreux outils de recherche qui mettent en évidence les termes de la requête lors de la présentation des réponses, typiquement en surlignant en couleurs vives leurs occurrences.

Avec cette interprétation, pour donner une valeur de pertinence au document nous calculons la moyenne des pertinences locales de toutes ses positions. Cela revient à intégrer (sommer) toutes les valeurs de pertinence locale sur la longueur du document, puis à normaliser cette intégrale.

2.5. *Implémentation du calcul de la pertinence locale à un terme*

D'un point de vue pratique, une requête booléenne est représentée par un arbre, et nous avons déjà indiqué comment remonter les fonctions de pertinences locales sur les nœuds internes grâce aux opérateurs booléens des ensembles flous. La fonction de proximité locale est renseignée pour chaque feuille de l'arbre de la requête de la façon suivante. Pour une feuille de l'arbre de la requête un tableau dimensionné à la taille du document est alloué et initialisé à zéro. Pour chaque occurrence du terme (ou de l'expression) associé(e) à la feuille, la fonction triangulaire qui atteint son maximum 1 à la position de l'occurrence est combinée par la fonction max dans le tableau. Lorsque toutes les occurrences ont été prises en compte, le tableau contient la fonction de proximité floue (ou de pertinence locale) au terme.

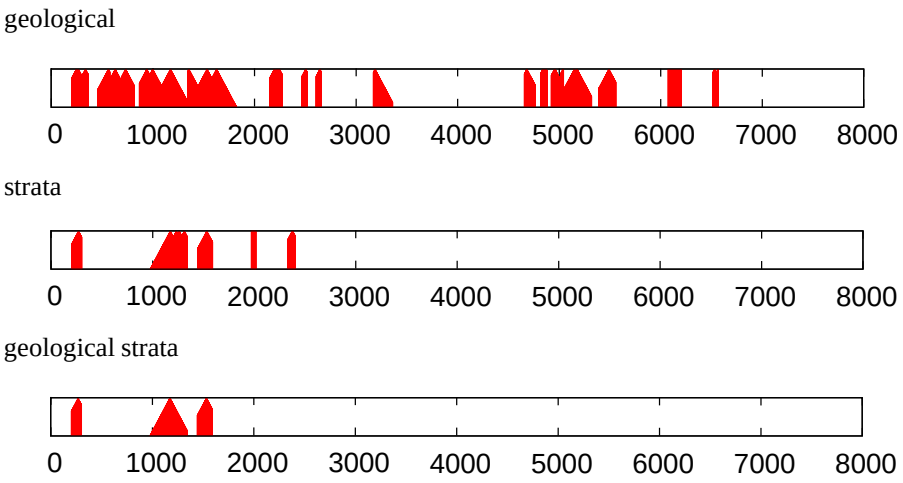

Figure 1. *Les pertinences locales au terme* `geological`*, au terme* `strata` *et à la combinaison conjonctive* `geological & strata` *dans le fichier 543667 de INEX, en prenant en compte les éléments sectionnants et propageants.*

2.6. *Propagation des termes de titre*

Nous présentons dans cette section une idée pour prendre en compte la structure des documents dans le modèle à base de proximité que nous avons présenté dans la section précédente dans le cas des textes plats. De la structure des documents nous ne considérons que ce qui concerne la hiérarchie, typiquement les chapitres, section, sous-sections, . . ., paragraphes. Nous ignorons donc d'autres aspects comme les listes, les tableaux, etc. Cette structure hiérarchique nous donne donc une arborescence liée à l'imbrication des éléments de niveaux inférieurs dans les éléments de niveaux supérieurs. Associé à cette notion de hiérarchie il y a la notion de titre, lequel décrit — certes d'une façon plus ou moins explicite — le contenu de la section qu'il intitule.

Du point de vue de la proximité, avec l'hypothèse que ces mots du titre décrivent le contenu de la section, l'idée est que les termes qui apparaissent dans un titre soient « proches » des termes qui apparaissent dans le corps de la section. Nous avons vu précédemment que la proximité était une conséquence d'une pertinence locale qui décroit avec la distance aux occurrences. Réciproquement pour être proche sur tout le corps d'une section l'idée est donc d'avoir une pertinence locale maximale — de valeur 1 — pour un terme du titre sur tout le corps de la section correspondante.

La structure effective des documents que nous manipulons contient bien d'autres types d'éléments que ceux de la hiérarchie et des titres. Nous classons donc les balises en trois catégories, celles qui définissent les éléments qui interviennent dans la hiérarchie — les *éléments sectionnants* —, celles des éléments qui contiennent un titre — les *éléments propageants* —, et enfin les autres balises.

2.7. *Implémentation du calcul de la pertinence locale à un terme*

Prendre en compte la structure du document dans les calculs des fonctions de pertinence locales n'intervient que pour les feuilles. En effet, pour tous les nœuds internes, il ne s'agit comme précédemment que de combiner les fonctions de pertinence locales des fils du nœud.

Comme pour le cas du texte plat, on considère successivement pour une feuille toutes les occurrences du terme associé. Pour une occurrence il faut trouver dans quel élément elle se trouve. Pour cela, nous utilisons une représentation de la structure où pour chaque élément sont conservées les positions des occurrences des premiers et derniers termes appartenant au contenu textuel de l'élément, y compris si ces contenus textuels sont en fait dans des éléments fils. Nous utilisons la structure de stockage de ces informations décrites dans [BEI 08].

Par un parcours récursif dans l'arbre de la structure du document, on trouve donc l'élément auquel appartient une occurrence. À partir de cet élément on remonte dans l'arbre jusqu'à trouver soit un élément sectionnant, soit un élément propageant. Si l'élément ainsi trouvé est un élément sectionnant, on introduit dans le tableau de la fonction de proximité locale le triangle en le tronquant aux limites de l'élément sectionnant. Si au contraire l'élément trouvé est un élément propageant, on continue de remonter dans l'arbre de la structure jusqu'au prochain élément sectionnant. Le tableau de la fonction de proximité locale est alors rempli avec la valeur maximale 1 sur toute la longueur de cet élément sectionnant.

La figure 1 montre trois fonctions de pertinence locale dans le document 543667 de la collection INEX (cf. infra). La première fonction montre la pertinence locale (la proximité floue) au terme `geological`. On y voit des triangles dont certains se recouvrent, donnant l'aspect en « montagne ». On y voit aussi des triangles tronqués, indiquant que la portée d'une occurrence du terme `geological` a été limitée par son appartenance à un élément sectionnant. On y voit enfin des rectangles, indiquant qu'une occurrence de `geological` est apparue dans un titre et que la portée a été étendue uniformément à tout l'élément sectionnant englobant. La deuxième fonction montre la pertinence locale (la proximité floue) au terme `strata` et la troisième est le min des deux précédentes, puisque les deux termes sont combinés conjonctivement.

3. Implémentation

3.1. *Indexation*

La base de l'implémentation est bien sûr d'indexer les documents en gardant les positions de toutes les occurrences des termes dans les documents au fur et à mesure de leur analyse. Pour celà nous utilisons le logiciel *zettair*[1] qui a cette fonctionnalité.

1. http://www.seg.rmit.edu.au/zettair/

Nous avons implanté dans ce logiciel l'analyse et le stockage au moment de l'indexation de la structure arborescente du XML.

3.2. *Langage de requête*

Notre modèle de calcul de scores est basé sur des requêtes booléennes. Nous avons donc besoin d'un langage de requête qui permet d'introduire les opérateurs de conjonction, de disjonction et de négation par rapport aux termes et aux expressions que l'on retrouvera dans les feuilles. Le caractère '|' sert d'opérateur de disjonction, les caractères ' ' et '&' servent d'opérateur de conjonction, et le caractère '−' pour la négation. De plus les parenthèses permettent de contrôler les priorités entre les opérateurs et les guillemets '"' d'introduire les expressions. Nous avons écrit un analyseur pour ce langage de requête et l'avons inséré dans *zettair*. Cet analyseur construit un arbre pour la requête qui est ensuite passé à l'implémentation de notre modèle de calcul de score.

Le langage de requête de la version initiale de *zettair* reconnaît à la base les opérateurs '−' pour indiquer une préférence pour l'absence du terme qui suit dans les réponses (une sorte de négation, bien que moins forte que la négation du modèle booléen) et l'opérateur '"' pour introduire les expressions. Par ailleurs, il ignore les autres caractères non alphanumériques. Ainsi, une requête booléenne dans le langage de requête défini ci-dessus peut être utilisée par *zettair* avec ses modèles classiques (Okapi, Dirichlet, etc.). Simplement les '|', '&' et les parenthèses sont ignorés.

Réciproquement, notre choix d'utiliser aussi l'espace comme opérateur de conjonction nous permet d'utiliser quasiment telles quelles des requêtes usuelles pour *zettair* avec notre modèle basé sur la proximité, avec toutefois une interprétation conjonctive au lieu de l'interprétation disjonctive des modèles classiques.

3.3. *Interrogation*

Avec ces données, nous avons implanté toujours dans ce logiciel le modèle de correspondance basé sur la proximité que nous venons de décrire. Comme nous l'avons déjà mentionné, la différence entre la version qui prend en compte la structure et celle qui travaille sur le texte plat n'est que dans le remplissage des tableaux représentant les fonctions de proximité floue des feuilles de l'arbre de la requête booléenne.

De plus notre modèle lorsqu'il est utilisé en mode conjonctif est extrêment sélectif, puisque pour qu'un document ait un score non nul il faut non seulement qu'il contiennent TOUS les termes de la requête conjonctive mais en plus qu'il y ait des occurrences de ces termes proches dans les documents. Lorsqu'il est utilisé seul, les listes de réponses sont généralement très courtes. Pour les compléter jusqu'à la limite des *runs* que l'on peut soumettre dans les campagnes de recherche d'information (typiquement 1000 ou 1500 selon les campagnes) elle sont complétées par les résultats obtenus par le modèle de base de *zettair* (modèle de langue avec lissage de Dirichlet).

4. Expérience

Nous avons construit notre expérience selon la méthodologie traditionnelle en recherche d'information ad'hoc avec une collection de test. Les données de cette collection sont celles utilisées dans l'édition 2008 de la campagne INEX. Les documents sont ceux issues d'une collecte de l'encyclopédie collaborative *Wikipedia* de 2006. Cet ensemble de documents a été utilisé dans les campagnes INEX depuis 2006. Il est composé de 659 388 documents qui correspondent à autant d'entrées dans l'encyclopédie en ligne. Ces documents sont formattés en XML et utilisent 1 263 types d'éléments différents.

Parmi ces éléments nous avons sélectionné manuellement ceux qui sont propageants : `name`, `template`, `title` et `caption` ; et ceux qui sont sectionnants : `article`, `body`, `section`, `figure`, `image`, `page` et `div`.

La campagne 2008 de INEX a utilisé 285 besoins d'information dont 135 ont été développés par les participants pour cette campagne, les 150 autres sont issus du journal d'un moteur d'interrogation de la *Wikipedia*. Les besoins d'informations sont composés des trois champs : `<title>`, `<description>` et `<narrative>` ; et parfois un quatrième : `<castitle>`. Les trois premiers sont analogues à ceux utilisées dans les jeux de *topics* utilisés dans les campagnes TREC. Le quatrième est spécifique à INEX et contient une requête dans le langage NEXI qui permet d'introduire des aspects de structure des documents dans la formalisation du besoin d'information.

Pour construire les requêtes pour notre système nous nous sommes basés sur le champ `<title>` qui est une requête composée de mots-clés, avec quelques opérateurs, `'+'` qui renforce la demande de présence du terme qui le suit dans un document retrouvé, `'−'` qui recommande l'absence du terme qui le suit, et les guillemets qui permettent d'introduire des expressions dans le besoin d'information.

Avec le langage de requête que nous avons défini ci-dessus, le champ `<title>` est directement utilisable comme requête pour notre implémentation aussi bien en mode de recherche par proximité que dans les modes de recherche classiques. Dans quelques cas, nous avons enrichi ce champ pour construire les requêtes en introduisant manuellement des regroupements de mots en expressions, des variations de formes sur certains termes avec l'opérateur `'|'`, et dans de rares cas des variations liées à la synonymie. Par exemple, voici les champs `<title>` pour les *topics* numéro 553 et 564 :

```
spanish classical +guitar players
criticism limitation null hypothesis significance test
```

et les requêtes que nous avons utilisées :

```
spanish (classical | classic) guitar players
(criticism | limitation) "null hypothesis" "significance test"
```

Enfin pour établir des références pour la comparaison de notre modèle à base de proximité, nous avons aussi lancé les requêtes avec le modèle par défaut de *zettair*, le modèle de langue avec lissage de Dirichlet et avec le modèle Okapi avec les paramètres $k_1 = 1.2$, $k_3 = \infty$ et $b = 0.75$.

Les jugements de pertinence utilisé dans la campagne 2008 de INEX sont plus riches que ce qui nous est utile. En effet, pour un document et une requête les assesseurs devaient surligner dans le texte les passages pertinents et donner de plus le meilleur point d'entrée (*Best Entry Point*) pour la lecture d'un document qui contient au moins un passage pertinent. Toutes ces informations se retrouvent dans le fichier des jugements avec un format qui étend celui utilisé par `trec_eval`[2]. On y trouve successivement le numéro de *topic*, un champ ignoré, le numéro de document jugé, la somme de la longueur des passages jugés pertinents (et donc 0 si le document n'est pas pertinent), la longueur du document. Si la somme n'est pas nulle on trouve ensuite la position du *Best Entry Point* et la liste des passages avec des couples `<position>:<longueur>`. Toutes les positions et les longueurs sont exprimées en caractères. Voici deux lignes extraites de ce fichier, la première pour un document non pertinent et la deuxième pour un document contenant deux passages pertinents :

```
544 Q0 682628 0 2055
544 Q0 177316 572 4798 1915 1915:299 3711:273
```

Nous avons donc très simplement transformé ce fichier en ne gardant que les quatre premières colonnes, ce qui en fait un fichier utilisable par l'outil d'évaluation `trec_eval`.

Il faut noter que seules 70 des 285 *topics* ont été jugés.

5. Résultats

Nous avons donc quatre *runs* avec : le modèle Okapi BM 25, le modèle de langue avec lissage de Dirichlet, le modèle avec proximité sur les textes plats, le modèle avec proximité et propagation de l'influence des termes de titre. Le tableau 1 montre la sortie de l'outil `trec_eval` pour ces quatre *runs*. Les mesures calculées par cet outil sont classiques en recherche d'information et largement documentées par ailleurs. La sortie de `trec_eval` est complétée par des colonnes intitulées % qui indique le pourcentage de la différence par rapport à la méthode de Dirichlet. Pour beaucoup de ces mesures le classement du meilleur au moins bon est Dirichlet, puis proximité avec propagation, puis Okapi, puis proximité. Les performances du modèle de proximité sans propagation sont honorables et proches de celles du modèle Okapi. Les performances du modèle de proximité avec propagation sont très proches de celles du modèle de langue avec lissage de Dirichlet.

2. http://trec.nist.gov/trec_eval/

modèle :	**prox.**	%	**Okapi**	%	**prox. et struct.**	%	**Dirichlet**
Total number of documents over all queries							
Retrieved :	96023		94013		93190		94150
Relevant :	4850		4850		4850		4850
Rel_ret :	3743		3402		3785		3503
Interpolated Recall - Precision Averages :							
at 0.00	0.7433	−15.44	0.8341	−5.11	0.8786	−0.05	0.8790
at 0.10	0.5431	−14.58	0.6138	−3.46	0.6183	−2.75	0.6358
at 0.20	0.4731	−9.40	0.5152	−1.34	0.5003	−4.19	0.5222
at 0.30	0.3813	−8.71	0.3802	−8.98	0.3911	−6.37	0.4177
at 0.40	0.3058	−8.00	0.3059	−7.97	0.3033	−8.75	0.3324
at 0.50	0.2608	−5.30	0.2339	−15.07	0.2620	−4.87	0.2754
at 0.60	0.1979	−2.27	0.1723	−14.91	0.2139	5.63	0.2025
at 0.70	0.1447	12.43	0.1108	−13.91	0.1605	24.71	0.1287
at 0.80	0.0796	11.64	0.0671	−5.89	0.1057	48.25	0.0713
at 0.90	0.0442	20.77	0.0287	−21.58	0.0661	80.60	0.0366
at 1.00	0.0103	0.98	0.0030	−70.59	0.0205	100.98	0.0102
Average precision (non−interpolated) over all rel docs							
	0.2671	−7.93	0.2688	−7.34	0.2881	−0.69	0.2901
Precision :							
At 5 docs	0.5286	−11.06	0.5543	−6.73	0.5629	−5.28	0.5943
At 10 docs	0.4671	−8.16	0.4857	−4.50	0.4929	−3.09	0.5086
At 15 docs	0.4448	−3.30	0.4343	−5.59	0.4486	−2.48	0.4600
At 20 docs	0.4186	0.00	0.4050	−3.25	0.4164	−0.53	0.4186
At 30 docs	0.3781	1.67	0.3643	−2.04	0.3838	3.20	0.3719
At 100 docs	0.2284	2.61	0.2104	−5.48	0.2289	2.83	0.2226
At 200 docs	0.1501	5.85	0.1369	−3.46	0.1484	4.65	0.1418
At 500 docs	0.0843	9.91	0.0747	−2.61	0.0827	7.82	0.0767
At 1000 docs	0.0498	7.33	0.0448	−3.45	0.0493	6.25	0.0464
R−Precision (precision after R (= num_rel for a query) docs retrieved) :							
Exact :	0.3091	−6.19	0.3105	−5.77	0.3226	−2.09	0.3295

Tableau 1. *La sortie de* `trec_eval` *pour les quatre runs : proximité dans le texte plat, Okapi, proximité avec propagation des termes de titres, et modèle de langue avec lissage de Dirichlet.*

6. Travaux reliés et conclusion

La très grande majorité des travaux sur les modèles de recherche d'information qui donnent un score aux documents de façon à pouvoir les classer (*ranking*) portent sur des modèles vectoriels et probabilistes où les seules données sont des données statistiques de comptage. Ces données ignorent donc la position des occurrences des termes.

La plupart des travaux qui ont essayé de prendre en compte la proximité des termes l'ont fait en modifiant les classiques modèles vectoriels ou probabilistes. Beaucoup de

ces travaux ont ajouté des contributions relatives à la présence d'expressions dans la suite des travaux de Fagan [FAG 87]. Le travail conclusif sur cette méthode a été fait par [MIT 97] et nous reprenons ici leur conclusion : si on part d'un système de base médiocre (par exemple *lnc.ltc* en notation SMART), cette méthode améliore les résultats ; mais si on part d'un système correct (*pivoted cosine* dans leurs expériences [SIN 96]) il n'y a pas d'amélioration, voire dégradation.

D'autres auteurs [RAS 03, BUT 06] en relâchant la contrainte d'expressions exactes et en demandant à ce que les termes de l'expression apparaissent dans un fenêtre ont obtenus des améliorations par rapport à la méthode Okapi BM25. De bonnes améliorations ont été obtenues par Hearst en filtrant les documents classés par un classique modèle vectoriel et en ne gardant que ceux qui vérifient une contrainte booléenne qui doit être vérifiée par au moins un des passages du document [HEA 96]. Cette contrainte de passage est bien moins forte (de l'ordre de 100 à 300 mots) que celle sur les expressions, même avec relaxation.

À notre connaissance, seuls Clarke et al. [CLA 00] et Hawking et al. [HAW 95] ont proposé des méthodes de notation des documents vraiment nouvelles basées sur la proximité des termes de la requête. Clarke *et al.* proposent d'utiliser une algèbre qui recherche la famille des intervalles (*extents*) les plus petits qui vérifient telle ou telle containte. Ils utilisent cette algèbre pour chercher les intervalles les plus courts qui contiennent tous les termes de la requête. Ils donnent ensuite un score à chaque intervalle d'autant plus grand que l'intervalle est petit. Le score du document est la somme des scores des intervalles retrouvés. Ils montrent de bonnes performances pour des requêtes courtes. Hawking *et al.* ont aussi modélisé et implémenté pour la campagne 1995 de TREC des idées très proches de celles de Clarke *et al.* avec des intervalles et un score d'intervalle décroissant avec la longueur de l'intervalle.

Il y a plusieurs points communs entre notre méthode et celles de Clarke *et al.* et de Hawking *et al.* : une interprétation conjonctive des requêtes, une prise en compte de proximité, un score de document calculé par sommation de pertinences partielles.

Nous avons décrit un modèle de recherche d'information qui calcule le score d'un document en prenant en compte les positions des occurrences des termes de la requête. Avec une requête conjonctive, les scores calculés sont d'autant plus grand que les termes de la requête (qui doivent être tous présents dans le document) ont des occurrences proches dans le texte du document. Nous avons présenté une extension de ce modèle pour prendre en compte une structure logique hiérarchique qui propage l'influence des termes des titres. Les expériences que nous avons présentées montrent que ce modèle de calcul de score avec proximité permet d'obtenir des performances comparables à celles des meilleurs modèles de recherche d'information connus dans l'état de l'art. Il serait intéressant de comparer cette méthode sur les mêmes bases expérimentales avec les méthodes qui prennent en compte la position des occurrences des termes que nous avons citées dans l'état de l'art.

Remerciements

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

7. Bibliographie

[BEI 08] BEIGBEDER M., « Compression de structure XML pour la recherche d'information structurée », *actes de CORIA'08, Cinquième Conférence en Recherche d'Information et Applications*, 2008.

[BUT 06] BUTTCHER S., CLARKE C. L. A., LUSHMAN B., « Term proximity scoring for ad-hoc retrieval on very large text collections », *SIGIR '06 : Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, New York, NY, USA, 2006, ACM, p. 621–622.

[CLA 00] CLARKE C. L. A., CORMACK G. V., TUDHOPE E. A., « Relevance ranking for one to three term queries », *Information Processing and Management*, vol. 36, 2000, p. 291-311.

[FAG 87] FAGAN J., « Experiments in automatic phrase indexing for document retrieval : A comparison of syntactic and non-syntactic methods », PhD thesis, Cornell University, 1987.

[HAW 95] HAWKING D., THISTLEWAITE P., « Proximity Operators - So Near And Yet So Far », Department of Commerce, National Institute of Standards and Technology, 1995.

[HEA 96] HEARST M. A., « Improving full-text precision on short queries using simple constraints », *Proceedings of the 5th Annual Symposium on Document Analysis and Information Retrieval (SDAIR)*, 1996, p. 217–232.

[MIT 97] MITRA M., BUCKLEY C., SINGHAL A., CARDIE C., « An analysis of statistical and syntactic phrases », *Proceedings of RIAO-97, 5th International Conference "Recherche d'Information Assistee par Ordinateur"*, 1997, p. 200–214.

[RAS 03] RASOLOFO Y., SAVOY J., « Term Proximity Scoring for Keyword-based Retrieval Systems », *ECIR 2003 proceedings*, n° 2633 LNCS, Springer, 2003, p. 207–218.

[SIN 96] SINGHAL A., BUCKLEY C., MITRA M., « Pivoted Document Length Normalization », *SIGIR '96, Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, 8 1996, p. 21-29.