

Application de la logique floue à un modèle de recherche d'information basé sur la proximité

Fuzzy set theory applied to a proximity model for information retrieval

Michel BEIGBEDER¹

Annabelle MERCIER¹

¹ École Nationale Supérieure des Mines de Saint Étienne
Centre Génie industriel et Informatique

158 cours Fauriel 42023 Saint Etienne Cedex 2

Annabelle.Mercier@emse.fr, Michel.Beigbeder@emse.fr

Résumé :

La détection et le classement des documents pertinents par rapport au besoin d'information d'un utilisateur est une motivation principale dans le domaine de la recherche documentaire. Après avoir rappelé le principe des modèles de recherche d'information utilisant la logique floue, nous définissons un modèle basé sur la proximité des termes. Notre étude repose sur l'hypothèse que plus les occurrences des termes d'une requête se retrouvent proches dans un document alors plus ce document doit être positionné en tête de la liste de réponses retournées par un système de recherche d'information. Notre modèle de requête est celui du modèle booléen classique de la recherche d'information qui utilise les opérateurs AND et OR. Pour ces requêtes nous proposons donc une méthode pour calculer le score de pertinence d'un document en fonction de la position des termes dans ce document.

Mots-clés :

Recherche documentaire, modèle booléen, ensembles flous, proximité des termes.

Abstract:

The detection and the ranking of the relevant documents compared to the user's information needs is one principal motivation in the information retrieval domain. We recall the foundation of the fuzzy information retrieval models, then we define a model based on the term proximity. Our study is based on the assumption that the more occurrences of the query terms are found close to each other in a document then the more this document should be in the top of the list of documents retrieved by an information retrieval system. Our query model is that of the classical Boolean model which uses AND and OR operators. Given such a query, we thus propose a method to compute the document relevance score according to the position of the query terms in this document.

Keywords:

Information retrieval, boolean model, fuzzy set, term proximity.

1 Introduction

Les systèmes de recherche d'information répondent aux besoins d'information des utilisateurs en retournant un ensemble de documents traitant des informations qu'ils recherchent, c'est ce que réalisent les moteurs de recherche du Web. La quantité de données numériques augmente quotidiennement et les moteurs de recherche ont besoin de développer de nouvelles techniques pour être plus précis et performants.

Au cours du processus de recherche d'information, la pertinence d'un document est jugée par l'utilisateur. L'apparition des mots-clés posés dans la requête intervient dans ce jugement mais il est clair que l'utilisateur ne décide pas de la pertinence du document en évaluant strictement une requête booléenne pour chaque document mais prend aussi en compte le sens et le contexte des termes de la requête. Notre méthode, en tenant compte de la position et de la densité des termes de la requête dans un document tente d'approcher ce processus. En effet dans une recherche sur les « structures de données », les documents dans lesquels ces deux mots n'apparaissent pas proches l'un de l'autre ne sont pas pertinents. De plus, l'apport de la logique floue permet d'introduire une progressivité dans la notion de proximité et évite d'être contraint par une évaluation strictement

binaire de cette dernière telle qu'elle est faite dans certains systèmes de recherche d'information booléen.

Dans la suite nous présentons d'abord dans la section 2 le modèle booléen et son extension classique basée sur la théorie des sous-ensembles flous. Ensuite dans la section 3, nous expliquons comment la notion de proximité est utilisée dans de nombreuses variantes du modèle booléen. Ceci nous permet d'introduire et de définir la notion de proximité floue. Enfin nous exposons dans la section 4 notre modèle de recherche d'information basé sur cette proximité avant de conclure dans la dernière section.

2 État de l'art

Un système de recherche d'information utilise une méthode d'indexation pour représenter les documents. Le plus souvent, cette indexation s'appuie sur les occurrences des termes trouvés dans les documents. Dans la suite, nous appellerons T l'ensemble des termes et D l'ensemble des documents.

Un modèle classique en recherche d'information est le modèle booléen où chaque document est représenté par l'ensemble des termes qui le composent et la requête de l'utilisateur est formulée à l'aide d'une expression booléenne. Une requête booléenne est représentée par un arbre où les feuilles sont des termes et les nœuds sont les opérateurs AND et OR. Par exemple, la requête $(A \text{ AND } B) \text{ OR } C$ est représentée par l'arbre de la figure 1.

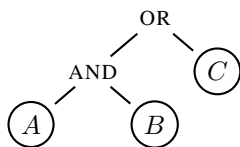


Figure 1 – Arbre de la requête $(A \text{ AND } B) \text{ OR } C$

La fonction de correspondance μ_q qui associe un document à la requête q prend des valeurs

binaires, ainsi

$$\mu_q : D \rightarrow \{0, 1\}.$$

Dans ce modèle, les résultats sont construits en prenant des unions (pour le OR) et des intersections (pour le AND) dans l'ensemble des documents contenant les termes de la requête. Une modélisation mathématique de ce modèle consiste à considérer des fonctions $\mu_t : D \rightarrow \{0, 1\}$ pour chaque terme $t \in T$. La fonction μ_t prend la valeur 1 si et seulement si le terme t apparaît dans le texte du document d . Ces fonctions μ_t sont associées aux feuilles des requêtes.

Dans l'arbre de la requête q , l'ensemble des documents répondant à un nœud $\mu_{q'} \text{ OR } \mu_{q''}$, où q' et q'' sont deux sous-arbres, est la réunion des documents répondant à $\mu_{q'}$ et de ceux répondant à $\mu_{q''}$, ce qui peut s'exprimer par

$$\mu_{q' \text{ OR } q''}^{-1}(1) = \mu_{q'}^{-1}(1) \cup \mu_{q''}^{-1}(1).$$

Ce résultat peut être obtenu en posant

$$\mu_{q' \text{ OR } q''} = \max(\mu_{q'}, \mu_{q''}).$$

Il s'agit ici d'une définition récursive et lorsqu'on arrive aux feuilles, on a, par exemple, $\mu_q = \mu_t$. De même, l'opérateur AND est associé à l'intersection, et on pose

$$\mu_{q' \text{ AND } q''} = \min(\mu_{q'}, \mu_{q''}).$$

Dans ce modèle, le critère de décision de pertinence est binaire, le score attribué au document étant pris dans l'ensemble $\{0, 1\}$. Donc les documents retournés ne peuvent pas être classés. Pour graduer ce score dans le cadre des modèles ensemblistes de la recherche d'information, plusieurs modèles basés sur la théorie des sous-ensembles flous ont été développés [5].

Dans les modèles basés sur la logique floue, à chaque terme $t \in T$ est associée une fonction μ_t traduisant le degré d'appartenance d'un document à l'ensemble flou correspondant au terme t

$$\begin{aligned} \mu_t : D &\rightarrow [0, 1] \\ d &\mapsto \mu_t(d). \end{aligned}$$

Notons que dans cette modélisation, on considère qu'un document appartient (plus ou moins) à un terme, alors que dans le langage courant on dirait plutôt qu'un terme appartient à (apparaît dans) un document. Cette formulation correspond à une modélisation orientée vers la requête, dans la mesure où le processus de recherche d'information est initié à partir des termes.

Dans ce modèle, une requête est aussi représentée par un arbre, un nœud avec l'opérateur OR (resp. AND) est évalué en prenant le maximum (resp. minimum) sur les valeurs de ses fils, ce qui correspond à la réunion (resp. intersection) floue des sous-ensembles flous correspondant à ses fils. Ce modèle permet d'obtenir un score de pertinence pour un document dans l'intervalle $[0,1]$, ce qui permet cette fois de classer les documents.

La figure 2 montre un exemple d'évaluation de la pertinence de trois documents d_1, d_2, d_3 pour la requête de la figure 1.

$$\begin{array}{l}
 T = \{A, B, C\} \quad \mu_A : D \rightarrow [0, 1] \\
 D = \{d_1, d_2, d_3\} \quad d_1 \mapsto 0.08 \\
 \quad \quad \quad \quad \quad d_2 \mapsto 0.05 \\
 \quad \quad \quad \quad \quad d_3 \mapsto 0.79 \\
 \\
 \mu_B : D \rightarrow [0, 1] \quad \mu_C : D \rightarrow [0, 1] \\
 d_1 \mapsto 0.12 \quad d_1 \mapsto 0.27 \\
 d_2 \mapsto 0.04 \quad d_2 \mapsto 0.03 \\
 d_3 \mapsto 0.76 \quad d_3 \mapsto 0.80 \\
 \\
 q = (A \text{ AND } B) \text{ OR } C \quad \mu_q(d_1) = 0.12 \\
 \quad \quad \quad \mu_q(d_2) = 0.03 \quad \mu_q(d_3) = 0.79
 \end{array}$$

Figure 2 – Exemple avec trois documents

Indépendamment des extensions « floues » que nous venons de citer, une extension classique du modèle booléen permet de prendre en compte la proximité entre les occurrences des termes de la requête dans le document en introduisant un opérateur NEAR [7]. Les travaux de Keen [3] indiquent que l'usage de cet opérateur per-

met d'améliorer la précision¹ pour l'ensemble des documents retournés tout en restant dans un cadre purement booléen.

D'autres approches plus récentes prennent en compte les intervalles de texte qui contiennent les occurrences des termes de la requête dans le document. Après une phase de sélection de ces intervalles, chacun reçoit un score qui décroît avec sa longueur, ces scores sont ensuite additionnés. Selon les méthodes, les intervalles sélectionnés ne sont pas les mêmes. La méthode de Clarke et al. [1] sélectionne les intervalles les plus courts qui contiennent tous les termes de la requête², ainsi les intervalles sélectionnés ne sont pas emboîtés les uns dans les autres. Dans la méthode Hawking et al. [2], pour chaque occurrence d'un terme de la requête, l'intervalle le plus court débutant sur cette occurrence et contenant tous les termes est sélectionné. Enfin Rasolofo et al. [6] choisissent de sélectionner les intervalles contenant deux termes de la requête, avec la contrainte supplémentaire que leur longueur soit de moins de cinq mots. Les résultats obtenus avec ces méthodes sont meilleurs que ceux des modèles traditionnels de recherche d'information. Nous allons donc utiliser cette idée de proximité avec la théorie des sous-ensembles flous pour obtenir un score pour chaque document qui dépende aussi de la proximité des occurrences des termes.

Enfin, il existe d'autres approches que les modèles ensemblistes en recherche d'informations. Le modèle le plus utilisé dans les outils est le modèle vectoriel [7] où les requêtes et les documents sont représentés par des vecteurs dont les composantes sont les poids pour les termes $t \in T$. Classiquement, le poids $w(d, t)$ du terme t dans le document d dépend de la

¹La précision est une mesure traditionnelle en recherche d'informations. Elle est définie comme le rapport entre le nombre de documents pertinents et retrouvés et le nombre de documents retrouvés, ou encore comme la proportion de documents pertinents parmi les documents retrouvés.

²Cette contrainte est relaxée si le nombre de documents retrouvés ne satisfait pas l'utilisateur.

fréquence du terme dans ce document et de la fréquence documentaire de ce terme, c'est-à-dire du nombre de documents où le terme t apparaît. La valeur de similarité entre un document et une requête est le plus souvent calculée avec la méthode du cosinus. À noter que l'on pourrait aussi utiliser les poids $w(d, t)$ du modèle vectoriel comme valeurs de $\mu_t(d)$ à condition de normaliser ces poids dans l'intervalle $[0, 1]$.

3 L'opérateur de proximité

3.1 Proximité binaire

L'utilisation de la proximité existe dans les systèmes booléens qui implantent l'opérateur NEAR. Ce dernier permet de préciser une distance maximale entre deux termes de la requête comme par exemple dans $A \text{ NEAR } 5 B$. Pour cet exemple, un système qui plante l'opérateur NEAR évalue la requête à la valeur vraie si et seulement si au moins une occurrence du terme A est à moins de 5 mots d'au moins une occurrence du terme B . L'opérateur NEAR se comporte donc basiquement comme un opérateur AND avec une contrainte supplémentaire sur les positions des occurrences des termes concernés.

3.2 Proximité floue

Nous souhaitons « flouifier » la notion de proximité, une première approche est de donner une interprétation floue à NEAR. Pour cela, nous modélisons un document d comme une suite finie de longueur l de termes de T , $(t_0, t_1, t_2, \dots, t_{l-1}) \in T^l$, c'est-à-dire, une fonction $d : \mathbb{N} \rightarrow T$ dont l'ensemble de définition est un intervalle de \mathbb{N} commençant en 0. Avec cette notation, $d^{-1}(t)$ désigne l'ensemble des positions où apparaît le terme t .

Si on cherche par exemple A et B proches, nous donnons une valeur de proximité à la requête $\text{NEAR}(A, B)$ dans le document d avec

$$\mu_{\text{NEAR}(A, B)}(d) = \max_{\substack{i \in d^{-1}(A) \\ j \in d^{-1}(B)}} \left(\max \left(\frac{k - |j - i|}{k}, 0 \right) \right)$$

où k est une constante fixant la portée d'une occurrence, pour les exemples, nous prendrons $k = 10$. La valeur que nous attribuons ainsi est liée à la distance séparant les deux plus proches occurrences de A et B dans le document d . La valeur maximale est atteinte lorsque la valeur absolue $|j - i|$ est minimale. Comme A et B ne peuvent pas apparaître à la même position, on a forcément $i \neq j$. La valeur minimale de $|j - i|$ est donc 1 et est atteinte lorsqu'il y a une occurrence de A qui est voisine dans le texte d'une occurrence de B .

3.3 Opérateur de proximité et arbre de requête

L'opérateur NEAR que nous venons de présenter dans son usage binaire habituel en section 3.1 et que nous avons étendue vers une notion de proximité floue en section 3.2, s'applique à deux termes. Sa généralisation pour l'appliquer à des sous-arbres pose des problèmes. Considérons la requête $(A \text{ OR } B) \text{ NEAR } C$. Elle peut s'interpréter comme trouver les documents dans lesquels il existe une occurrence de A ou une occurrence de B proche d'une occurrence de C . Autrement dit, on est amené à considérer les deux ensembles de positions $d^{-1}(A) \cup d^{-1}(B)$ et $d^{-1}(C)$, et ici l'opérateur OR se traduit par une union de l'ensemble des positions des occurrences de A avec l'ensemble des positions des occurrences de B . Par contre si l'on considère la requête $(A \text{ AND } B) \text{ NEAR } C$ où l'on a remplacé le OR par un AND, on a à considérer l'ensemble $d^{-1}(A) \cap d^{-1}(B)$, qui est toujours vide. Une autre interprétation de la dernière requête consiste à rendre l'opérateur NEAR distributif par rapport à l'opérateur AND et à trouver les documents dans lesquels i . A est proche de C , et ii . B est proche de C . Malheureusement, comme démontré par Mitchell [4], cette approche est inconsistante.

On ne peut donc pas généraliser l'opérateur NEAR et le faire intervenir n'importe où dans un arbre de requête du modèle booléen au même titre que les opérateurs AND et OR. Notre

x	-2	-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	...
d	...	ϵ	ϵ		A		B		C		A	B	C	C	ϵ	ϵ	...
$\mu_A^d(x)$	0.7	0.8	0.9	1.	0.9	0.8	0.7	0.7	0.8	0.9	1.	0.9	0.8	0.7	0.6	0.5	
$\mu_B^d(x)$	0.5	0.6	0.7	0.8	0.9	1.	0.9	0.8	0.7	0.8	0.9	1.	0.9	0.8	0.7	0.6	
$\mu_C^d(x)$	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.	0.9	0.8	0.9	1.	1.	0.9	0.8	
$\mu_{A \text{ ET } B}^d(x)$	0.5	0.6	0.7	0.8	0.9	0.8	0.7	0.7	0.7	0.8	0.9	0.9	0.8	0.7	0.6	0.5	
$\mu_{(A \text{ ET } B) \text{ OU } C}^d(x)$	0.5	0.6	0.7	0.8	0.9	0.8	0.8	0.9	1.	0.9	0.9	0.9	1.	1.	0.9	0.8	

Figure 3 – Un document et les valeurs des fonctions de proximité.

modèle ne va donc pas transposer l'opérateur décrit en 3.2 au modèle de recherche d'information à base d'ensembles flous présenté en section 2. Nous allons par contre généraliser la notion de proximité aux opérateurs AND et OR dans la section suivante.

4 Le modèle de proximité

4.1 Pertinence relative à un terme en une position du texte

Pour mesurer la pertinence relative à une requête composée d'un seul terme t en une position x dans un document d , nous cherchons l'occurrence de ce terme la plus proche et attribuons une valeur avec une fonction décroissante selon la distance de cette occurrence. Plus précisément, nous modélisons la proximité pour une position x du document d à un terme $t \in T$ par la fonction $\mu_t^d : \mathbb{Z} \rightarrow [0, 1]$ définie par

$$\mu_t^d(x) = \max_{i \in d^{-1}(t)} \left(\max\left(\frac{k - |x - i|}{k}, 0\right) \right).$$

Notons que cette fonction est bien définie sur \mathbb{Z} , car l'influence d'un terme s'étend de part et d'autre de ses occurrences et peut donc « déborder » avant le début du document (position 0) ou après sa fin (position $l - 1$).

La figure 3 montre un exemple de document de longueur $l = 12$ où apparaissent les trois termes A , B , et C . Les valeurs des fonctions μ_A , μ_B , et μ_C y sont données pour les positions entre -2 et 13 . La notation ϵ sert à indiquer les positions hors du domaine de définition de la fonction d .

4.2 Pertinence relative à une requête en une position du texte

Ensuite, nous pouvons définir récursivement une pertinence relative à une requête pour chaque position du texte. Notre modèle de requête est celui du modèle booléen classique, c'est-à-dire avec des nœuds qui portent des opérateurs AND et OR, et des feuilles qui portent des termes. Nous ne considérons pas d'« opérateur » NEAR étant donné son inconsistance avec les opérateurs AND et OR, et ce sont les fonctions μ_t que nous avons définies dans la section précédente, et qui attribuées aux feuilles d'une requête capturent la proximité aux termes de la requête. Nous définissons la fonction attribuée à un nœud portant un opérateur OR par

$$\mu_{q \text{ OR } q'} = \max(\mu_q, \mu_{q'}).$$

De même, pour un opérateur AND, on pose

$$\mu_{q \text{ AND } q'} = \min(\mu_q, \mu_{q'}).$$

Ces formules sont les mêmes que celles de la section 2, toutefois ici elles s'appliquent à des fonctions $\mathbb{Z} \rightarrow [0, 1]$ et non à des nombres.

Les deux dernières lignes de la figure 3 montrent les valeurs des fonctions $\mu_{A \text{ AND } B}^d$ et $\mu_{(A \text{ AND } B) \text{ OR } C}^d$ pour les différentes positions dans le document exemple. La figure 4 montre le tracé de cette dernière fonction.

La fonction μ_q^d obtenue à la racine de l'arbre de la requête q par l'application remontante de ces formules représente pour chaque position dans le document la pertinence à la requête. Nous qualifierons cette pertinence de *locale* dans la mesure où elle dépend de la position dans le document. Cette pertinence locale prend

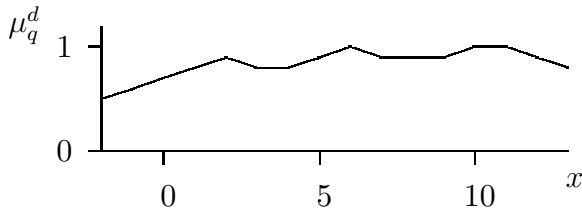


Figure 4 – Représentation floue de la pertinence relative à la requête $\mu_q^d(x)$ avec $q = (A \text{ AND } B) \text{ OR } C$

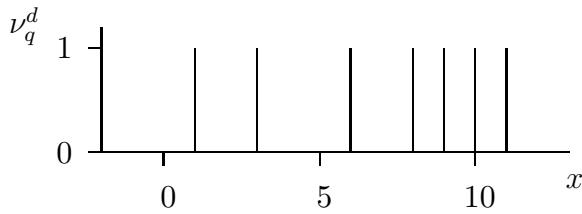


Figure 5 – Représentation de ν_q^d avec $q = \{A, B, C\}$

en compte la *proximité* aux termes de la requête. Cette pertinence est d'autant plus élevée que tous les termes requis par des AND sont proches (et suffisamment proches pour ne pas donner une pertinence nulle) et augmentée par le plus proche des termes demandés par un OR.

Enfin, notons que cette pertinence locale est un ensemble flou dont les éléments sont les positions dans le texte du document d , et la fonction μ_q^d indique leur degré d'appartenance à cet ensemble flou. Nous parlerons de l'*ensemble (flou) des positions dans le document d*.

4.3 Détermination du score d'un document

En recherche d'information, une des premières mesures de similarité (c'est-à-dire la pertinence système) entre un document et une requête composée d'un ensemble de termes est le *niveau de coordination* (*coordination level* en anglais) qui compte le nombre d'occurrences de

termes de la requête dans un document. On peut noter que cette méthode de calcul de score est un cas particulier du modèle vectoriel. Cette mesure peut s'obtenir en calculant le cardinal de l'ensemble des positions où apparaît un terme de la requête

$$c(q, d) = |\cup_{t \in q} d^{-1}(t)|.$$

Comme les ensembles intervenant dans cette réunion sont disjoints deux à deux, ceci s'exprime aussi

$$c(q, d) = \sum_{t \in q} |d^{-1}(t)|,$$

ou encore

$$c(q, d) = \sum_{x \in \mathbb{Z}} \nu_q^d(x),$$

avec $\nu_q^d(x) = 1$ si et seulement si $d(x) \in q$. Remarquons bien qu'ici la requête traitée est un ensemble de termes (par exemple $\{A, B, C\}$) et que le calcul de pertinence par le niveau de coordination ne s'applique pas à des requêtes booléennes. La figure 5 montre une représentation de ν_q^d tandis que la figure 4 montre μ_q^d la représentation floue du document de l'exemple de la figure 3 selon notre modèle de proximité.

Dans notre modèle, par analogie, nous calculons le score d'un document d pour la requête q avec

$$s(q, d) = \sum_{x \in \mathbb{Z}} \mu_q^d(x).$$

Notons que ceci représente, au sens des ensembles flous, le cardinal de l'ensemble des positions dans le document d . Ici, μ_q^d représente la proximité à la requête pour chaque position, alors que ν_q^d représente une proximité binaire à un ensemble de termes.

Nous obtenons ainsi un score appartenant à \mathbb{R}^+ qui permet de classer les documents par ordre décroissant. De plus, notre méthode prend en compte la proximité entre les termes de la requête, ce que nous pensons utile pour ramener les documents pertinents en premier et éliminer le bruit en tête de la liste des documents retournés par un système basé sur notre modèle.

5 Conclusion

Nous avons rappelé le modèle booléen de la recherche d'information et son extension classique basé sur la théorie des sous-ensembles flous. Ensuite, à partir de notre hypothèse : *les documents ayant des occurrences des termes de la requête proches doivent être classés en premier*, nous avons défini la notion de proximité floue puis nous avons présenté notre modèle de recherche d'information basé sur la proximité et utilisant des requêtes booléennes. Nous n'avons pas besoin d'introduire d'opérateur NEAR pour prendre en compte la proximité, ce qui évite les problèmes d'inconsistance que pose cet opérateur avec les opérateurs AND et OR. Par ailleurs le paramètre k permet de régler la portée de l'influence des occurrences des termes. Une valeur de l'ordre de 5 permet de demander une proximité de l'ordre de l'expression, une valeur de 15 à 30 la situe au niveau de la phrase et une valeur de l'ordre de 100 la porte au niveau du paragraphe. La prochaine étape de notre travail est de réaliser une implantation de ce modèle afin de le tester sur des collections de test comme celles de la conférence TREC (cf. <http://trec.nist.gov/>) afin d'étudier l'influence du paramètre k et de comparer les performances de notre modèle avec celles des modèles classiques de recherche documentaire et celles des modèles où la proximité est prise en compte avec des intervalles.

Références

- [1] Charles L. A. Clarke, Gordon V. Cormack, and Elizabeth A. Tudhope. Relevance ranking for one to three term queries. *Information Processing and Management*, 36(2) :291–311, 2000.
- [2] D. Hawking and P. Thistlewaite. Proximity operators - so near and yet so far. In D. K. Harman, editor, *TREC-4 proceedings*. NIST, 1995.
- [3] E. Michael Keen. Some aspects of proximity searching in text retrieval systems. *Journal of Information Science*, 18 :89–98, 1992.
- [4] Patrick C. Mitchell. A note about the proximity operators in information retrieval. In *Proceedings of the 1973 meeting on Programming languages and information retrieval*, pages 177–180. ACM Press, 1973.
- [5] Sadaaki Miyamoto. *Fuzzy Sets in information retrieval and cluster analysis*. Kluwer Academic Publishers, 1990.
- [6] Y. Rasolofo and J. Savoy. Term proximity scoring for keyword-based retrieval systems. In *proceedings of the 25th European Conference on Information Retrieval, ECIR 2003 proceedings*, number 2633 in LNCS, pages 207–218. Springer, 2003.
- [7] Gerard Salton and Michael J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.