Un modèle de recherche d'information basé sur les graphes et les similarités structurelles pour l'amélioration du processus de recherche d'information

Yaël Champclaux

▶ **To cite this version:**

Yaël Champclaux. Un modèle de recherche d'information basé sur les graphes et les similarités structurelles pour l'amélioration du processus de recherche d'information. Informatique [cs]. Université Paul Sabatier - Toulouse III, 2009. Français. NNT : . tel-00446372

HAL Id: tel-00446372

https://theses.hal.science/tel-00446372v1

Submitted on 12 Jan 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par *l'Université Toulouse III - Paul Sabatier*
Discipline : *Informatique*

Présentée et soutenue par Yaël Champclaux le 4 décembre 2009

Titre : Un modèle de recherche d'information basé sur les graphes et les similarités structurelles pour l'amélioration du processus de recherche d'information

---

Jury :

Dkaki Taoufiq, MCF, Encadrant.
Flory Andre, Professeur, Examinateur.
Gaussier Eric, Professeur, Rapporteur.
Mélançon Guy, Professeur, Rapporteur.
Mothe Josiane, Professeur, Directeur de thèse.
Sedes Florence, Professeur, Président du jury.

`

Ecole doctorale : *MITT Mathématiques Informatique Télécommunications de Toulouse*
Unité de recherche : *IRIT – UMR 5505*

TITRE: Un modèle de recherche d'information basé sur les graphes et les similarités structurelles pour l'amélioration du processus de recherche d'information

DIRECTEUR DE THESE: Josiane Mothe

ENCADRANT DE THESE: Taoufiq Dkaki

LIEU ET DATE DE SOUTENANCE: Université Paul Sabatier, 118 route de Narbonne 31062 Toulouse cedex 9, Grand auditorium de l'IRIT, le 4 décembre 2009 à 10h30.

RESUME:

Cette thèse d'informatique s'inscrit dans le domaine de la recherche d'information (RI). Elle a pour objet la création d'un modèle de recherche utilisant les graphes pour en exploiter la structure pour la détection de similarités entre les documents textuels d'une collection donnée et une requête utilisateur en vue d'améliorer le processus de recherche d'information. Ces similarités sont dites « structurelles » et nous montrons qu'elles apportent un gain d'information bénéfique par rapport aux seules similarités directes. Le rapport de thèse est structuré en cinq chapitres. Le premier chapitre présente un état de l'art sur la comparaison et les notions connexes que sont la distance et la similarité. Le deuxième chapitre présente les concepts clés de la RI, notamment l'indexation des documents, leur comparaison, et l'évaluation des classements retournés. Le troisième chapitre est consacré à la théorie des graphes et introduit les notations et notions liées à la représentation par graphe. Le quatrième chapitre présente pas à pas la construction de notre modèle pour la RI, puis, le cinquième chapitre décrit son application dans différents cas de figure, ainsi que son évaluation sur différentes collections et sa comparaison à d'autres approches.

MOTS-CLES: Recherche d'information, graphes, comparaison, similarités structurelles, graphes bipartites, modèle de recherche d'information.

DISCIPLINE: Informatique

LABORATOIRE: Institut de Recherche en Informatique de Toulouse, UMR 5505 – CNRS.

Remerciements

Je tiens tout d'abord à remercier Josiane Mothe pour avoir dirigé ma thèse, pour m'avoir fait une place au sein de l'équipe SIG-EVI, et pour m'avoir permis de rencontrer mon encadrant direct Tao. C'est à lui que s'adresse maintenant mes remerciements : il a été à mes cotés durant ces cinq années, m'a fait profiter de sa compétence autant que de son humanité et de sa gentillesse.

Je remercie Eric Gaussier et Guy Melançon qui ont accepté d'être mes rapporteurs, André Flory et Florence Sedes qui ont accepté de faire partie de mon jury.

Parallèlement à l'écriture de ma thèse j'ai eu l'occasion de travailler à l'université du Mirail en tant qu'enseignant et en tant que technicien, ce qui m'a permis de découvrir des gens aussi différents qu'intéressants que je voudrais également associer à ma thèse : Claude Chrisment, Bernard Dousset, Gilles Hubert, Maryse Salles, Eloïse Loubier, llhème Ghalamallah, Guillaume Cabanac, Ronan Tournier pour l'équipe de l'IRIT ; Bertrand Jouve, Sophie Corre, Edyta Bellouni, Ghislain Delrieux, Romain Boulet, Alain Couvreur, Christine Calvet, Pierre Ratinaud pour la maison de la recherche ; Jean-Pierre, Jean-Christophe et Daniel pour l'équipe du CRIE (ex-PRIM), Bernard Coulette, Pierre-Jean Charrel, Ollivier Haemmerlé, Caroline Thierry, Françoise Adreit, Jean-Christophe Sakdavong, Nathalie Hernandez, Sophie Ebersold, Cathy Comparot-Poussier, Eric Jacoboni, Brahim Hamid … pour l'équipe du Mirail. Je suis très heureux d'avoir croisé sur ma route des personnes qui m'ont donné envie de faire ce métier.

De plus j'ai eu la chance de partager mon bureau avec d'autres thésards, Quoc-Dinh Truong et Han Nhi Tran : j'ai une pensée pour vous et les années passées ensembles.

Sans partager le même bureau, j'ai eu la chance de rencontrer Adil Anwar, Issam Kabbaj et Younes Lakhrissi dont la bonne humeur, la chaleur et l'écoute m'ont apporté énormément. Je vous remercie tous trois pour ces moments partagés dont je me souviendrai avec bonheur. J'ai également une pensée pour ceux d'entre nous qui se sont lancé dans l'écriture d'une thèse : Hannas, Rhama, Akim, Samba, Romain, Eric, Adel … Je vous souhaite du courage, on peut finir un jour !

Mes amis Thomas, Julien, Julien (l'autre), Sylvie, Donat, Jamie, Mustapha, Marie & Jérôme, Abdelkrim, Pauline, Jean-Luc, Jean-Louis, Christian, les Fred(s), les autres Julien(s) .... Merci pour la richesse que vous apportez à ma vie.

J'adresse un merci tout particulier à mon colocataire, Matthieu Desjammes, pour ses conseils, son aide généreuse aussi bien technique que théorique, pardon pour les jours et les nuits que tu as dû me consacrer.

Je dédie cette thèse à ma famille, mes parents, ma sœur, et ma petite nièce Jade.

Table des matières

Index des figures

Index des tableaux

Introduction Générale

Cette thèse d'informatique se situe dans le domaine de la **recherche d'information (RI)**. Elle s'intitule « modèle de recherche d'information basé sur les graphes et les similarités structurelles pour l'amélioration du processus de recherche d'information ». Elle a pour objet la création d'un **modèle de recherche** utilisant les **graphes** et exploitant la **structure** sous-jacente pour détecter un certain type de **similarités** entre les **documents** textuels d'une collection donnée et une **requête** utilisateur en vue d'améliorer le résultat du processus de RI.

Contexte de la recherche :

La RI concerne les méthodes et mécanismes qui permettent la création et l'utilisation d'une **base d'information**. Une base d'information est un **système documentaire** permettant d'exploiter une collection de documents. La gestion concerne principalement le stockage des documents, ainsi que leur recherche et leur présentation en vue d'une utilisation (consultation par exemple). Un **Système de recherche d'information** (SRI) est un ensemble logiciel permettant d'effectuer l'ensemble des tâches nécessaires à la RI. Un SRI possède trois fonctions fondamentales qui définissent le **modèle de recherche** : **représenter** le contenu des documents, **représenter** le besoin de l'utilisateur et **comparer** ces deux représentations. La **représentation** des documents et de la requête dans le système se fait à l'issue d'une phase appelée **indexation** qui consiste à choisir les termes représentatifs des documents et à les ajouter à un index qui à chaque terme associe le document dans lequel il se trouve avec éventuellement des informations additionnelles comme la fréquence d'apparition du terme dans le document. Le modèle doit mettre en correspondance les représentations des documents et la représentation du besoin de l'utilisateur exprimé sous la forme d'une requête afin de retourner à celui-ci les documents en rapport avec sa requête. Généralement, cela se fait à l'aide d'un **calcul de similarité**. L'opération de **comparaison des représentations** est fondamentale en RI. Elle constitue le cœur du modèle de recherche. Les modèles de recherche s'appuient sur des théories mathématiques qui offrent des opérations pour comparer les représentations des documents de la collection et la représentation de la requête de l'utilisateur. Généralement, le calcul de similarité qui est effectué exploite les termes communs aux documents comparés pour évaluer leur ressemblance. Par exemple, dans le modèle booléen les documents sont indexés de façon binaire et le SRI retourne ceux qui répondent à l'expression logique qui représente la requête. Pour le modèle vectoriel les documents et la requête sont représentés comme des vecteurs dans l'espace des termes indexés, le SRI retourne les documents qui sont « à proximité » de la requête dans cet espace.

Les **similarités** exploitées dans les deux modèles précités sont dites « **de surface** » c'est-à-dire qu'elles exploitent les **attributs** des objets comparés, ici les termes communs. La contribution principale de cette thèse est d'introduire dans le domaine de la RI un modèle de recherche exploitant un autre type de **similarités** appelées « **structurelles** », c'est-à-dire caractéristiques du **système de relations** entre les objets comparés.

Nous proposons ci-après un cas d'utilisation où notre méthode pourrait être utile : un utilisateur souhaite retrouver un article qu'il se souvient avoir lu, mais dont il ignore l'auteur, le journal, la date ou tout autre élément qui pourrait lui permettre de le retrouver à l'aide d'une base de données. Il doit rechercher par rapport au souvenir qu'il a du contenu. Dans le cas présent il s'agit de trouver un document qui parle « d'un prestidigitateur ayant coupé une femme en deux sur la scène d'un théâtre » mais le document en question emploie en réalité les termes suivants : « un magicien a découpé une jeune fille du public lors de son spectacle ». A cause de la grande différence entre l'expression de l'utilisateur et le contenu réel du document, un système à recherche exacte ne

peut pas retrouver le document ; un système exploitant les relations entre les documents a une chance de retrouver ce document du fait que « magicien » est similaire à « prestidigitateur », « femme » à « fille », …

L'idée de départ tient en deux points :

- Des termes apparaissant dans un même document ont un certain degré de similarité,
- Des documents sont similaires s'ils sont composés de termes similaires.

Par exemple, la similarité que nous associons au couple « prestidigitateur » « magicien » est due à leur probable apparition commune dans un même document. Notre idée va un peu plus loin que la simple cooccurrence de deux mots. Le couple « scène » « spectacle » peut être associé à une valeur de similarité du fait que l'un peut apparaître dans des documents traitant de théâtre, que l'autre peut apparaître dans des documents traitant de culture et que « théâtre » et « culture » peuvent apparaître dans un même document. Ainsi, « théâtre » et « culture » vont par leur présence commune dans un même document être considéré comme ressemblants et ils vont rendre ressemblant les documents qui les contiennent séparément. Il est possible par la relation qui lie les termes et les documents de propager une certaine ressemblance. Cette propagation de la ressemblance nous permet de construire une mesure de similarité dite « structurelle ». Cette similarité s'articule autour des similarités directes entre documents et se construit par propagation de proche en proche.

Problématique :

Le but de ce travail est d'introduire en RI une nouvelle façon de comparer les documents aux requêtes. Cette façon de comparer est inspirée par des travaux en psychologie cognitive qui basent la comparaison de deux entités sur les attributs de ces entités ainsi que sur les relations entretenues par ces entités avec les autres entités de leurs entourages alors que traditionnellement en RI, c'est la comparaison basée sur les termes indexant les documents qui est utilisée pour la comparaison de ceux-ci avec la requête. Avec ce type de comparaison, les différents SRI retournent une liste de documents contenant des termes de la requête.

Nous proposons une méthode qui permet de reconnaître comme éventuellement pertinent un document n'ayant aucun des termes de la requête. Ce travail apporte une méthode de comparaison de documents avec la requête basée sur les propriétés structurelles du réseau de documents étudiés.

Cette méthode de comparaison constitue le cœur de notre modèle de RI. Ce modèle se base sur la théorie des graphes pour représenter les données : les documents et la requête sont représentés par un graphe composé de nœuds *document* et de nœuds *terme* reliés par un arc si le terme apparaît dans le document. Une fois les données représentées par un graphe, notre méthode consiste à propager des similarités « initiales » entre documents et des similarités « initiales » entre termes par les liens document-terme dans le réseau que constitue le graphe. Cette propagation est réalisée par un algorithme itératif. Quand la propagation prend fin, nous disposons d'un score de similarité associé à chaque couple de nœuds du graphe.

Par cette méthode nous espérons d'une part générer de l'information relationnelle non disponible au départ et d'autre part exploiter cette information pour améliorer la qualité des retours du SRI qui implémente le modèle de RI proposé.

Cette proposition tourne autour de plusieurs axes qui sont autant de contributions :

- Adaptation puis application à la RI d'un algorithme de comparaison d'objets :
    - Prise en compte de la pondération,
    - Proposition d'une normalisation.

- Etude de l'algorithme :
    - Choix d'une initialisation,
    - Preuve de la convergence,
    - Etude de la complexité,
    - Etude des propriétés de tri,
    - Etude des propriétés de propagation.
- Test de l'algorithme en situation réelle :
    - Etude des paramètres et de la vitesse de convergence sur des collections de test,
    - Comparaison à d'autres méthodes sur les mêmes collections.

Cette thèse est composée de cinq chapitres, les trois premiers chapitres abordent les aspects théoriques liés à notre modèle et les deux derniers traitent de notre modèle et des expérimentations effectuées.

Dans le premier chapitre nous avons souhaité répondre à la question : comment un objet peut être vue comme similaire à une autre ? Pour introduire la notion de similarité, nous décrirons dans un premier temps, les modèles de similarité en psychologie qui ont servi de base à nos travaux. En effet l'acte de comparer est chez l'humain, quasi permanent. Ainsi nous nous pencherons sur les processus mis en jeu lors de l'activité de comparaison dans le cadre d'activités cognitives telles que la résolution de problème, le raisonnement analogique, le raisonnement à partir de cas et l'activité de catégorisation. La finalité de notre étude étant la proposition d'un modèle de RI et de la mesure de similarité associée, il nous a paru opportun d'étudier la comparaison d'objets dans le cadre d'activité de catégorisation ou de classification ; l'objet de notre étude pouvant être vu comme un cas particulier de l'activité de catégorisation où la catégorie recherchée est celle des documents similaires à la requête. Ainsi le premier chapitre est composé de trois parties : la première traite des modèles de similarité en psychologie, la seconde aborde la catégorisation et le raisonnement analogique sous l'angle des sciences cognitives et la dernière traite des activités de catégorisation et classification en RI.

Dans le second chapitre, nous introduirons les concepts de base de la RI au travers de la description du processus de RI. Ce processus consiste en l'acquisition des données que constituent les documents et la requête. Ces données textuelles subissent une série de traitements afin de construire un index. Une fois l'index construit, les représentations des données issues de l'index peuvent être comparées afin de déterminer la liste des documents qui sont pertinents pour la requête. La méthode de représentation et plus encore la méthode de comparaison des représentations sont au cœur des modèles de RI. C'est pourquoi après avoir décrit l'indexation des données, nous décrirons les principaux modèles de RI. Enfin nous présenterons les collections et méthodes d'évaluation de SRI que nous avons retenues pour évaluer notre modèle.

Dans le troisième chapitre, nous introduirons les définitions et notations usuelles concernant les graphes et leurs caractéristiques. Notre modèle base sa méthode de représentation des données sur l'usage des graphes. Nous décrirons différents types de graphes et caractéristiques associées en insistant sur les graphes bipartites qui sont au centre de notre méthode. Nous présenterons les différents critères qui nous permettrons de caractériser les graphes représentant les collections documentaires étudiées au chapitre 5. Puis nous présenterons l'usage qu'il est fait des graphes en RI, d'une part dans la recherche basée sur le contenu et d'autre part dans le cadre spécifique du Web où la structure est exploitée à des fins de recherche.

Dans le quatrième chapitre, nous décrirons pas à pas l'élaboration de notre modèle de recherche, en commençant par décrire l'algorithme à l'origine de notre modèle, puis la méthode utilisée pour adapter puis appliquer cet algorithme à la RI. Nous intègrerons à notre modèle la prise en compte de la pondération des termes dans les documents puis nous proposerons une

normalisation ainsi qu'une écriture matricielle de nos formules. Ensuite nous nous pencherons sur le fonctionnement de l'algorithme : l'initialisation, les conditions d'arrêt. Puis nous réaliserons la preuve de la convergence de l'algorithme. Enfin, nous illustrerons les principes de la méthode sur des cas d'école.

Le cinquième chapitre est consacré aux expérimentations : nous décrirons et commenterons les différentes évaluations utilisées pour analyser le comportement de notre algorithme en situation réelle. Nous évaluerons d'abord notre modèle sur de petites collections en étudiant successivement la vitesse de convergence, l'influence des paramètres de notre algorithme, la comparaison avec d'autres modèles et l'adaptation de l'algorithme au filtrage et au ré-ordonnancement de documents.

Enfin, nous conclurons et proposerons différentes pistes de recherche.

Chapitre 1 : Similarité

La notion de comparaison mériterait un chapitre entier dans un livre sur la pensée humaine, là où certains réduiraient ce thème à une sous-partie du domaine de la prise de décision. En fait, la comparaison intervient dans tout ce que nous faisons. De plus, les recherches montrent que le simple fait de comparer deux choses peut produire d'important changement de notre savoir. Earl Miner dans [Miner, 1987] constate à propos de l'acte de comparaison: « Il est manifestement impossible de comparer ce qui est identique. Des différences doivent exister ou alors nous identifions plus que nous comparons. De la même façon, si les différences sont trop grandes, la comparaison devient infaisable, les résultats logiques ou pratiques ne satisfont pas. ». Ainsi, il semble que comparaison et similarité (ou sa notion duelle différence) soient intimement liés.

Comme nous le verrons, la comparaison intervient également en RI. La RI, apparue au début des années cinquante, est un domaine de recherche née de la rencontre entre la science de l'information et l'informatique.

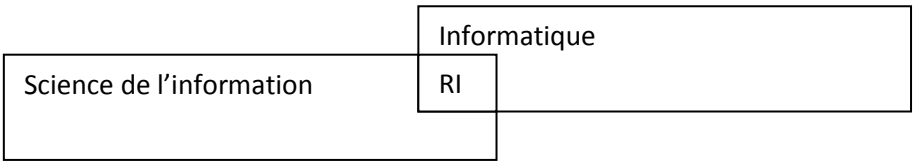


Figure 1 : la RI au croisement des sciences de l'information et de l'informatique

La science de l'information est elle-même au carrefour de nombreuses disciplines comme le montre la figure suivante :

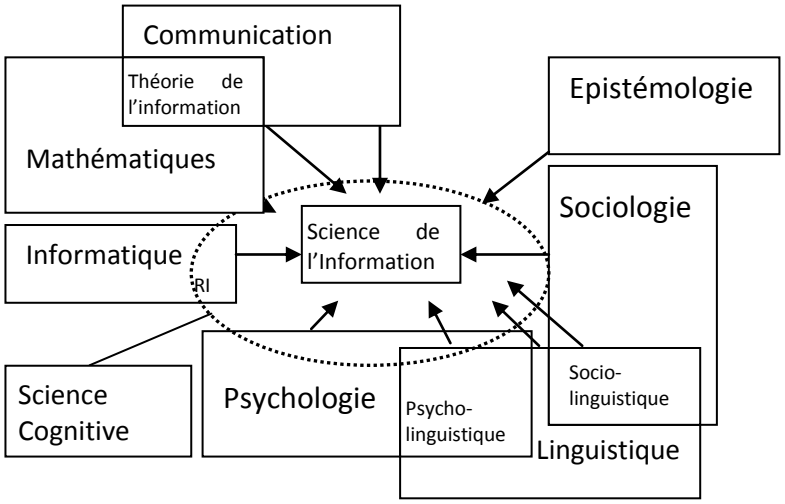


Figure 2 : disciplines influençant la science de l'information [Ingwersen, 1991]

Le modèle de recherche constitue le noyau des SRI dont la finalité est de fournir la liste des documents correspondant à la requête d'un utilisateur donné. Un SRI est un système automatique manipulant de l'information au service d'un utilisateur. Comme tout système ayant pour but d'analyser ou d'organiser automatiquement un ensemble de données ou de connaissances, le SRI doit comparer et pour cela utiliser un opérateur de similarité dont le but est d'établir les ressemblances ou les relations qui existent entre les informations manipulées. La principale contribution de notre modèle - 4ème chapitre - est de proposer une méthode de comparaison

document/requête qui ne se limite pas au seul contenu, et qui exploite les relations entretenues par la requête avec les documents disponibles. La particularité de notre travail se situe au niveau de la manière de comparer des documents et trouve sa source dans les recherches en RI d'une part et en psychologie d'autre part. La méthode utilisée pour comparer les documents et la requête est basée sur la comparaison des relations entretenues à l'instar des mécanismes utilisés notamment dans le raisonnement analogique.

Une des premières fonctions de la comparaison est d'accéder à la similarité de deux objets. Le philosophe Quine dans [Quine, 1969] observe que « la similarité est fondamentale pour l'apprentissage, le savoir, la pensée, (…) Une prédiction raisonnable dépend de la similitude des circonstances et de notre tendance à attendre que des causes similaires aient des effets similaires ».

Cette notion de similarité fait l'objet de nombreuses recherches dans différents domaines tels que l'analyse de données, le raisonnement à partir de cas, la reconnaissance des formes, la résolution de problèmes, l'apprentissage, le transfert, ... Il est difficile d'appréhender l'ensemble des travaux effectués autour de ce thème car ils se distinguent par les buts poursuivis (soit analyser des données, soit reconnaître des formes, …), les langages de représentation utilisés pour décrire les données et la similarité entre ces données ainsi que par les approches théoriques sous-jacentes (approche spatiale, approche relationnelle). Ce chapitre vise à présenter quelques-uns des liens qui existent entre ces différents domaines et à montrer la richesse de la notion de similarité. Il nous a paru pertinent de mettre en rapport l'activité de comparaison mentale telle qu'elle est étudiée au travers des sciences cognitives et une activité de comparaison automatique telle qu'effectuée par un SRI. Le thème de la similarité sera donc abordé selon trois angles.

La première partie intitulée similarité et regroupement traite, après un rappel sur la notion de distance, des modèles d'estimation de la similarité en psychologie. La seconde partie intitulée similarité en sciences cognitives aborde la catégorisation ainsi que le raisonnement analogique. Une dernière partie traite de l'utilisation de la similarité en RI en particulier pour la catégorisation, la classification et l'analyse de données. Enfin nous concluons par les aspects fondamentaux des notions de comparaison et de similarité.

1. Similarité et regroupement

L'homme est sans cesse confronté à des situations nouvelles. Pour gérer ces situations, simplifier les problèmes rencontrés, faciliter le raisonnement et organiser ses connaissances, il procède souvent par comparaison de la nouvelle situation avec les situations précédentes. Cette comparaison consiste à mettre dans le même espace, par exemple la mémoire, les situations et à voir en quoi elles se ressemblent et du même coup en quoi elles diffèrent. Ainsi, les situations, concepts ou objets de raisonnement jusque-là inconnus deviennent comparables selon certaines dimensions, ce qui a pour effet de réduire l'incertitude liée à ces objets ou situations problématiques. Par exemple, il est possible de dire qu'une voiture est d'un rouge plus brillant qu'une autre et ainsi comparer sur l'échelle de la brillance deux voitures. Pour faire cela, il faut reconnaître l'identité des objets à comparer, ici les deux sont des voitures, puis il faut positionner les deux objets (voitures) sur une dimension donnée (la brillance) et les comparer en termes de distance.

La comparaison est un mécanisme fondamental du raisonnement, le positionnement dans un espace choisi et la distance sont des outils pour réaliser cette comparaison. C'est pourquoi nous commencerons par un rappel sur la notion de distance du point de vue mathématique. Puis, dans un second temps, nous présenterons les différents modèles de similarité en psychologie cognitive afin de voir quels mécanismes sont pris en compte dans les modèles qui formalisent l'activité de catégorisation dans laquelle la similarité joue un rôle central.

1.1 La distance mathématique

Une *distance* sur un ensemble E est une application $d: E \times E \Rightarrow \mathbb{R}^+$ vérifiant les propriétés[1] suivantes :

- Symétrie : $\forall x, y \in E, d(x,y) = d(y,x)$
- Séparation : $\forall x, y \in E, d(x,y) = 0 \Leftrightarrow x = y$
- Inégalité triangulaire : $\forall x, y, z \in E, d(x,z) \leq d(x,y) + d(y,z)$

Si l'espace est orienté, alors la *distance algébrique* entre deux points *a* et *b* est définie soit par le réel non nul positif représentant la distance précédemment définie si le vecteur $\overrightarrow{ab}$ va dans le même sens que celui de l'espace, soit par le réel négatif sinon. La *distance algébrique* n'est pas une distance, vu qu'elle est *non-symétrique* : $d(a,b) = -d(b,a)$.

La distance dans les espaces vectoriels :

Dans un espace vectoriel normé $(E, \|.\|)$, une distance *d* est définie à partir de la norme en posant: $\forall (x,y) \in E \times E, d(x,y) = \|y - x\|$

En particulier, dans $\mathbb{R}^n$, il existe plusieurs manières de définir la distance entre deux points :

Soient deux points de *E*, (x_1, x_2, ... ,x_n) et (y_1, y_2, ... ,y_n), les différentes distances sont :

- Distance de Manhattan (*1-distance)*: $\sum_{i=1}^{n} |x_i - y_i|$ (1)
- Distance euclidienne (*2-distance)*: $\sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$ (2)
- Distance de Minkowski (*p-distance)*: $\sqrt[p]{\sum_{i=1}^{n} |x_i - y_i|^p}$ (3)
- Distance de Chebyshev (∞-*distance)*: $\lim_{p \to \infty} \sqrt[p]{\sum_{i=1}^{n} |x_i - y_i|^p} = \sup_i(|x_i - y_i|)$ (4)

La distance de Manhattan est la plus adaptée pour comparer des attributs psychologiquement et/ou physiologiquement séparables par exemple : le volume et la fréquence d'un son [Atteneave, 1950] ou la taille et la forme d'un objet [Thibaut, 1997].

La distance euclidienne est le meilleur modèle pour la comparaison de deux attributs liés, par exemple : la saturation et la luminosité d'une couleur [Gardenfors, 2000].

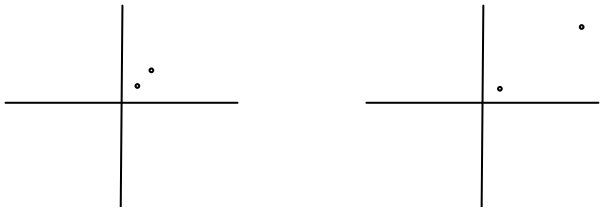
S'il n'y a aucun présupposé sur le type de distance à mettre en évidence, la distance de Minkowski constitue un choix raisonnable car les résultats qu'elle produit constituent une bonne modélisation du jugement humain [Nosofsky, 1984]. Pour choisir la valeur de *p,* Kruskal propose de calculer les représentations pour plusieurs valeurs de p et de choisir celle qui minimise le stress [Borg et al. 2005].

1.2 Les modèles de similarité du point de vue psychologique

Il existe principalement quatre modèles de similarité : les modèles géométriques, les modèles basés sur les caractéristiques, les modèles à alignement structurel et les modèles basés sur la notion de distance transformationnelle. Ces modèles fournissent des informations théoriques sur la similarité et décrivent comment celle-ci peut être mesurée de manière empirique.

[1] La propriété de minimalité à savoir : $\forall x,y \in E, d(x,y) \geq d(x,x)=0$ découle de manière naturelle de la séparation et du fait que les distances sont définies sur R^+

Le point de vue spatial consiste à modéliser les concepts individuels comme des points dans un espace multidimensionnel et à considérer que la ressemblance entre deux objets est inversement proportionnelle à la distance qui les sépare.



Similaire Non similaire

Figure 3 : principe de la distance mentale [Shepard, 1962]

Pour présenter cette approche nous reprenons les propos de Goldstone qui, dans [Goldstone, 1991] déclare que : les modèles géométriques qui comptent parmi les modèles les plus influents [Togerson, 1965][Caroll et al., 1974] et sont illustrés par les modèles MDS (*Multidimensional scaling)* [Nosofsky, 1992]. Un modèle MDS prend en entrée soit des jugements de similarité ou de non-similarité, soit des matrices de confusion (tableau où chaque entité est comparée à chacune des autres entités), soit des probabilités que des données puissent être groupées ensemble, soit n'importe quelle autre mesure de similarité subjective entre paires d'entités d'un ensemble. La sortie est un modèle géométrique de la similarité entre entités, avec chaque entité représentée comme un point dans un espace à n dimensions. La similarité entre deux entités est alors inversement proportionnelle à la distance qui les sépare. La distance d(i,j) entre deux entités est calculée par la distance de Minkowski. La métrique euclidienne p=2 est une bonne métrique quand les jugements de similarité reposent sur des notions qui se recouvrent, la métrique p=1 est une bonne métrique quand les entités à comparer sont clairement séparables selon des dimensions distinctes [Garner, 1974].

Par exemple, l'expérience décrite dans [Goldstone et al., 2005] montre que s'il est demandé à des personnes de donner une valeur de similarité aux couples (Russie, Cuba), (Russie, Jamaïque), (Jamaïque, Cuba) sur une échelle de 1 à 10, on obtient : Similarité (Russie, Cuba)=7, Similarité (Russie, Jamaïque)=1, Similarité (Jamaïque, Cuba)=8. Ainsi il est possible de mettre en correspondance des pays selon diverses dimensions (soit le climat ou la position géographique, soit l'affiliation politique,…).

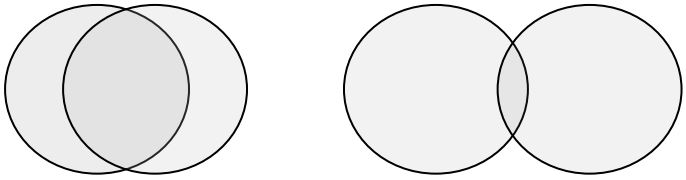
Les algorithmes MDS [Nosofsky, 1992] positionnent les entités dans un espace où celles qui sont similaires sont proches. Il se trouve que l'application de tels algorithmes peut permettre d'interpréter les dimensions sous-jacentes d'un ensemble d'entités à comparer. Les algorithmes MDS peuvent être utilisés pour générer une représentation compressée qui exprime les similarités relationnelles d'un ensemble d'entités. Cette compression facilite l'encodage, la mémorisation et le traitement des données. La représentation numérique sous forme de coordonnées dans un espace peut être utilisée pour tout type de stimulus (mots, sons, images) et une fois construite, elle permet de prédire les jugements de similarité, la performance de la mémoire ou la vitesse d'apprentissage. Les modèles MDS ont été appliqués avec succès pour exprimer des structures cognitives dans les positions classiques du jeu d'échec [Horgan et al., 1989] et les scénarios de vol aérien [Schvaneveldt, 1985].

Les modèles MDS respectent la notion de distance au sens mathématique (minimalité, symétrie, inégalité triangulaire). Ils ont cependant été critiqués car dans la pratique, les règles de distance ne sont souvent pas respectées. Par exemple la clause de minimalité est violée si tous les objets identiques ne sont pas identiquement similaires ; la clause de symétrie est violée si un objet est jugé ressemblant à un autre, de façon plus importante que l'inverse. Par exemple « Corée du Nord » est jugé plus ressemblante à « Chine » que « Chine » l'est pour « Corée du Nord » [Tversky, 1977]. La clause de l'inégalité triangulaire peut également être violée. Il est néanmoins possible d'améliorer les

modèles géométriques [Novorsky, 1991] en prenant en compte la similarité entre entités ainsi que des préjugés - savoirs, fréquence, saillance - sur certaines entités.

1.2.1 Les modèles basés sur les attributs

L'approche basée sur les caractéristiques consiste à représenter deux objets à comparer par deux ensembles de caractéristiques, la partie commune de ces deux ensembles contient les traits communs, la partie disjointe de ces deux ensembles contient les caractéristiques différentes.



Similaire Non similaire

Figure 4 : intersection d'ensembles de caractéristiques [Tversky, 1977]

L'approche basée sur les attributs a été développée pour répondre aux limitations de l'approche géométrique, en effet les évaluations subjectives de la similarité ne satisfont pas les hypothèses des modèles géométriques de la similarité. Notamment, comme le précise Goldstone dans [Goldstone, 1991] la limitation stricte du nombre de proches voisins qu'un objet peut avoir [Tversky et al., 1986], la difficulté à décrire des objets définis par un grand nombre de dimensions [Krumhansl, 1978] et le fait de ne pas accroître la similarité entre objets quand une nouvelle dimension est introduite [Tversky et al., 1982].

Dans le modèle de contraste, la similarité entre deux entités est déterminée par une combinaison linéaire des mesures des attributs communs et distincts de chaque entité :

Soient *A* et *B* deux entités, $A \cap B$ représente les caractéristiques que *A* et *B* ont en commun, *A-B* représente les caractéristiques que *A* possède mais pas *B*, de même *B-A* représente les caractéristiques que *B* possède mais pas *A*. La similarité entre *A* et *B* est donnée par la formule :

$$S(A,B) = \theta f(A \cap B) - \alpha f(A - B) - \beta f(B - A) \qquad (5)$$

Où f est une fonction additive. θ, α et β sont des coefficients attribués aux parties conjointes et disjointes.

Des expériences portant sur la comparaison de stimuli picturaux montrent que dans le cadre des jugements de similarité entre entités ayant un grand nombre de dimensions distinctives (par exemple deux photos d'une personne à deux âges très différents), une importance plus grande est donnée aux parties communes. De même, dans le cadre de jugements de dissimilarité entre entités ayant un grand nombre de dimensions communes (par exemple deux photos d'une personne avec et sans lunettes), une importance plus grande est accordée aux parties différentes [Gati et al., 1984].

Le modèle de contraste permet de traiter des caractéristiques concrètes comme abstraites. Il permet également de prédire les similarités asymétriques (exemple de la Chine et de la Corée du Nord). De plus les jugements de similarité et de dissimilarité ne sont pas « miroir ». Par exemple une expérience décrite dans [Tversky, 1977] montre que 67 % des personnes interrogées considèrent l'Allemagne de l'Est et l'Allemagne de l'Ouest comme plus similaires entre elles que Ceylan et Népal. 70 % des personnes considèrent dans le même temps que l'Allemagne de l'Est et l'Allemagne de l'Ouest comme plus différents l'un de l'autre que Ceylan et Népal.

Sjoberg [Sjoberg, 1972] a proposé un autre modèle basé sur les caractéristiques où la similarité dépend du ratio entre les propriétés communes et distinctes des entités comparées. Cette mesure

peut être mise en rapport avec la mesure de [Jaccard, 1901]. Il suffit de voir *f* comme la fonction qui retourne le cardinal :

$$S(A,B) = \frac{f(A \cap B)}{f(A \cup B)} \qquad (6)$$

Eisler et Ekman [Eisler et al., 1959] considèrent que la similarité de deux entités A et B est proportionnelle à $f(A \cap B)/(f(A) + f(B))$.

Il est possible de généraliser le modèle de contraste à partir de la formule de Tversky:

$$S(A,B) = \frac{f(A \cap B)}{f(A \cup B) - \alpha f(A - B) - \beta f(B - A)} \qquad (7)$$

Où α et $\beta \geq 0$.

Les prémisses fondamentales de ce modèle sont que les entités peuvent être décrites en termes de caractéristiques constituantes. Les analyses basées sur les caractéristiques ont été appliquées en perception du langage [Jakobson et al., 1963], physiologie de la perception[Hubel et al., 1968], contenu sémantique [Katz et al., 1963] et en catégorisation [Medin et al., 1978].

1.2.2 Les modèles à alignement structurel

Les modèles basés sur les caractéristiques comme les modèles géométriques ne sont pas performants pour comparer des entités très structurées [Hummel, 2000][Hummel, 2001]. De plus, les modèles basés sur les caractéristiques font l'hypothèse que les parties communes et les différences sont indépendantes. Or ce n'est pas le cas : trouver des différences entre deux entités requiert de trouver d'abord les caractéristiques communes pour pouvoir les différencier ensuite. Par exemple, différencier une voiture et une moto par leur nombre de roues nécessite dans un premier temps de constater que toutes deux possèdent des roues. La différence du nombre de roues entre voiture et moto est une différence alignable, par opposition aux différences non alignables, qui reflètent des propriétés que l'une des deux entités possède et pas l'autre. Par exemple une voiture a des portières, une moto non. A partir de leurs travaux sur l'analogie, différents auteurs dont [Gentner, 1983][Holyok & Tagard, 1989] ont proposé des modèles à alignement structurel et ont montré que lors de la comparaison de deux entités, les caractéristiques communes influencent plus la similarité si elles sont dans des parties placées en correspondance et les parties seront placées en correspondance si elles partagent de nombreuses caractéristiques communes et si elles sont consistantes avec d'autres correspondances émergentes[Goldstone, 1994][Markman et al., 1993]. Pour être mises en correspondance, deux représentations structurées doivent être structurellement consistantes, c'est-à-dire conforme à la loi de la mise en correspondance 1 à 1 : un élément de la première entité ne doit correspondre qu'à un élément de la seconde, et respectant certaines contraintes de connectivité par exemple que les relations entre éléments qui correspondent doivent également correspondre.

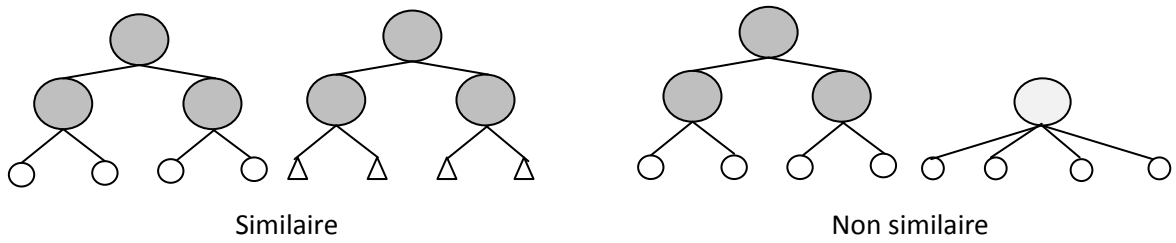


Figure 5 : modèle de similarité basé sur un alignement structurel [Gentner, 1983]

Cette approche permet de comparer des représentations complexes et des structures relationnelles hiérarchiques. L'idée de l'alignement structurel est que, pour comparer deux entités

structurées, il faut prendre en compte non seulement les correspondances entre éléments mais aussi les ressemblances dans les connexions qui lient ces éléments. L'alignement structurel a été appliqué dans la comparaison de phrases [Bassok et al., 1997], dans la comparaison d'habitude de consommation [Zang et al., 1998] et dans l'étude du domaine légal [Simon et al., 2002]. Des recherches ont montré que pour la catégorisation basée sur l'induction, les modèles à alignement structurel sont plus performants que les modèles à caractéristiques [Lassaline, 1996].

1.2.3 Les modèles basés sur une distance transformationnelle

Spécifier la structure d'un objet nécessite d'une part de déterminer ses caractéristiques mais aussi de déterminer comment ces caractéristiques sont liées. Par exemple, spécifier la structure d'une phrase requiert la spécification des relations entre les mots. Les représentations structurées peuvent exprimer une réelle complexité là où une liste de caractéristiques ou de valeurs selon des dimensions ne le peuvent pas [Bierbeman, 1985]. Pour répondre aux problèmes liés à la simplicité des représentations (points dans un espace ou ensemble) des modèles géométriques et des modèles basés sur les caractéristiques, [Hanh et al., 1997][Hanh et al., 1998] ont proposé une approche où la similarité entre deux entités est inversement proportionnelle au nombre d'opérations nécessaires pour transformer une entité en l'autre. L'hypothèse est que toute représentation mentale peut être transformée en une autre représentation avec un nombre fini d'étapes de transformation. La notion centrale de ce modèle est la distorsion représentationnelle qui offre une base théorique aux jugements de similarité.

La similarité entre deux entités est fonction de la complexité requise pour transformer une représentation en une autre. Plus cette transformation est simple, plus les entités sont similaires. Par exemple, « 1 2 3 4 5 6 7 8 » est plus proche de « 2 3 4 5 6 7 8 9 » que de « 4 7 10 13 16 19 22 25 » car dans le premier cas il suffit d'une opération (soustraire 1 à chaque élément) pour retrouver la chaîne de chiffres d'origine, alors que dans le deuxième cas plusieurs opérations sont nécessaires : soustraire 1 puis diviser par 3.

Pour déterminer la complexité d'une représentation ou d'une transformation entre représentations, la théorie de la complexité de Kolmogorov est utilisée [Li et al., 1997]. Cette théorie définit la complexité $K(x)$ d'une représentation de x comme la longueur du plus court programme permettant de générer cette représentation. De même la complexité $K(y|x)$ d'une transformation de x en y est le plus court programme permettant de transformer la représentation x en représentation y. Cette mesure de similarité a été appliquée avec succès dans le cadre de l'apprentissage automatique [Bennet et al., 1998][Li et al., 2001].

Cette approche permet de traiter avec des représentations structurées qui posent problème aux approches classiques. Par exemple, les structures arborescentes comme utilisées dans les réseaux sémantiques peuvent être transformées par opération sur les arbres ou bien les phrases peuvent être transformées par des opérations linguistiques (morphologiques, syntaxiques ou sémantiques). Elle permet d'évaluer la similarité indépendamment de la représentation et ainsi d'aborder les processus basés sur la similarité sous l'angle de l'analyse rationnelle. Néanmoins, les auteurs reconnaissent la nécessité de prendre en compte la nature des représentations mentales pertinentes. L'ensemble des transformations ou instructions peut être utilisé pour modifier les représentations ainsi que la capacité des systèmes cognitifs à prendre en compte la découverte de transformations simples entre représentations.

1.2.4 Conclusion sur les approches de la similarité

Les deux approches classiques que sont l'approche spatiale et l'approche basée sur les caractéristiques ont été appliquées avec succès dans le domaine de la modélisation cognitive, voir [Ortony, 1979][Osherson, 1990] pour le modèle de contrastes, [Rips, 1965][Shepard, 1987] pour les

modèles spatiaux. Les modèles MDS présentent l'avantage de représenter par des dimensions les traits caractéristiques d'un ensemble de données. Navarro dans [Navarro et al., 2004] propose de combiner les deux approches en utilisant les statistiques pour déterminer si une source de variation est exprimable en termes de caractéristiques ou en termes de dimensions. Ces deux approches partagent le fait de n'utiliser que des représentations non structurées. Cette limite a permis l'apparition d'approches plus modernes : l'approche basée sur un alignement structurel et l'approche basée sur une distance de transformation qui ont introduit de nouvelles façons de comparer. L'approche structurelle montre l'intérêt de la prise en compte de la structure des entités dans l'activité de comparaison, rendant plus complexe la vision simpliste qui consiste à ramener le problème à un espace mathématique où les objets sont comparables. L'approche transformationnelle introduit également une idée majeure : la notion de transformation. Ce point de vue se base sur l'idée que la similarité entre deux entités est déterminée par une relation transformationnelle. L'approche transformationnelle a été appliquée presque exclusivement à des stimuli perceptuels, tandis que l'approche structurelle a été appliquée à des stimuli conceptuels (histoires, proverbes, théories scientifiques). Il ressort que le point de vue structurel est utile pour comprendre la perception [Marr et al., 1978][Bieberman, 1987].

Ces modèles ont connu différents succès dans la prédiction de jugement de similarité. Les tests entre ces approches pour la comparaison d'entités continuent d'être un sujet de recherche. Il est intéressant d'étudier la similarité dans d'autres domaines comme la catégorisation où certains modèles sont complètement basés sur la notion de similarité et où d'autres en revanche considèrent que la similarité ne suffit pas et qu'il faut également prendre en compte un savoir abstrait, théorique [Rips et al., 1993]. L'étude de l'analogie présente également un intérêt car elle met en jeu des mécanismes d'appariement et une prise en compte de la structure des concepts.

2. Similarité en science cognitive

« This sense of sameness is the very keel and backbone of our thinking » [James, 1890]

La capacité d'accéder au concept de similarité est à la fois la quille (qui empêche de chavirer) et la colonne vertébrale de notre pensée.

Comme il est aussi possible de lire dans [Goldstone,1991] : la similarité joue un rôle fondamental dans les théories de la cognition. De nombreuses activités cognitives utilisent la notion de similarité au travers de la comparaison d'entités. Ainsi, la capacité des gens à résoudre des problèmes dépend de la ressemblance de ces problèmes avec ceux précédemment rencontrés et résolus [Ross, 1987][Holyoack et al., 1987][Novick, 1988]. Dans le même ordre d'idées, la catégorisation dépend de la ressemblance des objets à catégoriser avec une abstraction, un prototype, ou un objet précédemment catégorisé [Rosch, 1975][Medin et al., 1978]. La recherche dans la mémoire dépend de la ressemblance entre les indices de recherche et les souvenirs stockés en mémoire [Hintzmann, 1986]. Le raisonnement déductif est basé sur le principe de comparaison de faits à des prémisses. Les faits similaires aux prémisses permettent de déduire de nouvelles connaissances. Le raisonnement inductif est basé sur l'idée que si un évènement est similaire à un évènement déjà apparu, alors son exemple vient enrichir les connaissances [Richard, 1995]. Les théories sur le transfert montrent que des aptitudes nouvelles seront plus facilement apprises si elles sont proches de choses précédemment apprises [Singley et al., 1989][Bracke, 1998]. Nous allons maintenant aborder les activités de la catégorisation et de raisonnement analogique qui présentent des similitudes avec la méthode de comparaison utilisée au cœur de notre modèle.

2.1 Similarité et catégorisation

La catégorisation permet à l'individu d'organiser et de réduire la complexité de son environnement, en le découpant et en le regroupant en objets qu'il attribue à différentes catégories.

Si la catégorisation n'existait pas, l'environnement serait perçu comme perpétuellement nouveau. Sans catégorisation des situations, aucune anticipation ou prévision ne serait possible.

Des travaux du domaine de la cognition comparée montrent que la capacité à former des catégories n'est pas le propre de l'homme. Herrnstein et Loveland dans [Herrnstein et al., 1964] ont démontré que des pigeons peuvent apprendre à catégoriser des photographies en deux classes (celle qui représente des hommes, et celle qui n'en représente pas) et appliquer cette règle catégorielle pour trier des photographies qu'ils n'ont jamais vues auparavant. D'autres travaux ont par la suite confirmé cette capacité des animaux à organiser objets ou évènements en catégories, et à y répondre de manière adaptée notamment chez le singe [D'Amato et al., 1988].

Intuitivement, similarité et catégorisation semblent étroitement liées. De nombreuses études se sont concentrées sur les sortes de relations de similarité entre un nouvel objet à catégoriser et une catégorie. Les modèles à base de prototypes considèrent que les représentants constituent un résumé des caractéristiques les plus typiques de la catégorie à laquelle les objets appartiennent, et que les nouveaux objets sont classés sur la base de leur ressemblance à ces prototypes [Reed, 1972][Hampton, 1995]. L'idée sous-jacente est que les catégories sont structurées par des effets prototypiques, déterminant des espaces catégoriels hétérogènes, caractérisés par des cas centraux typiques et des limites non tranchées [Rosch, 1976]. Ainsi, la catégorie se définit en référence à un prototype, soit le meilleur représentant de la catégorie. Les autres objets de la catégorie se repèrent sur un gradient de typicalité, selon leur plus ou moins grande distance ou similitude avec le prototype.

Les modèles à exemplaires (ou de l'exemplaire) considèrent que les individus retiennent des exemples spécifiques d'une catégorie et classifient les nouveaux objets sur la base de leurs ressemblances avec les exemples retenus [Medin et al., 1978][Kruschke, 1992]. Ces modèles se basent sur l'hypothèse que seul le stockage des informations spécifiques peut rendre compte à la fois des effets contextuels et de la formation d'une catégorie abstraite au moment de la récupération de l'information. De plus, un rôle important est attribué à la familiarité des différentes sources d'information, familiarité qui augmente graduellement durant les processus de traitement, jusqu'à atteindre un seuil suffisant d'activation pour permettre la décision catégorielle prise par rapport à l'exemplaire le plus semblable trouvé en mémoire [Chemlal et al., 2006].

Les modèles à prototypes comme à exemplaires considèrent qu'une mesure de proximité est calculable pour chaque paire d'objets. La plupart des recherches expérimentales évaluant ces modèles adoptent une procédure d'apprentissage, utilisant souvent des stimuli variant sur quelques dimensions. Par exemple, Goldstone [Goldstone, 1995] apprend aux sujets à discriminer des carrés, variant suivant leur taille et leur brillance. Sloutsky et ses collègues [Sloutsky et al., 2001] utilisent des visages schématiques, dont la taille du nez et des oreilles varie, pour une tâche d'induction. Par contre, quand le matériel devient complexe, comme c'est le cas dans la catégorisation naturelle, il est difficile de faire des prédictions spécifiques en se basant uniquement sur la théorie de l'exemplaire [Chemlal et al., 2006].

Il existe plusieurs modèles mixtes de la catégorisation [Komatsu, 1992] qui prennent en compte des informations liées à l'exemplaire, et la possibilité de mémorisation d'une information abstraite, qui aura elle-même un rôle dans la catégorisation. Ils sont de deux types : soit ils associent les contraintes des modèles à l'exemplaire et des modèles à un prototype [Knapp et al., 1984][Malt, 1989], soit ils associent les contraintes des modèles à exemplaire et celles des modèles classiques à base de règles [Allen et al., 1991][Smith et al., 1998].

2.2 Similarité et analogie

Le procédé central du raisonnement analogique est la mise en correspondance analogique. Le principe est qu'une situation familière dite situation de base ou situation source, est mise en

correspondance avec une situation moins familière : la situation cible [Gentner, 1983][Hall, 1989]. Les connaissances initiales relatives au domaine source sont plus nombreuses que celles relatives au domaine cible. L'objectif est d'exploiter l'appariement effectué entre les éléments des domaines source et cible de manière à transférer un certain nombre de connaissances du premier vers le deuxième. La situation familière doit permettre d'avoir un point de vue sur la situation nouvelle et de faire des déductions à son sujet.

La figure suivante extraite de [Falkenhainer, 1989] montre l'analogie entre deux situations, l'une étant le siège d'un flux de liquide, l'autre d'un flux thermique.

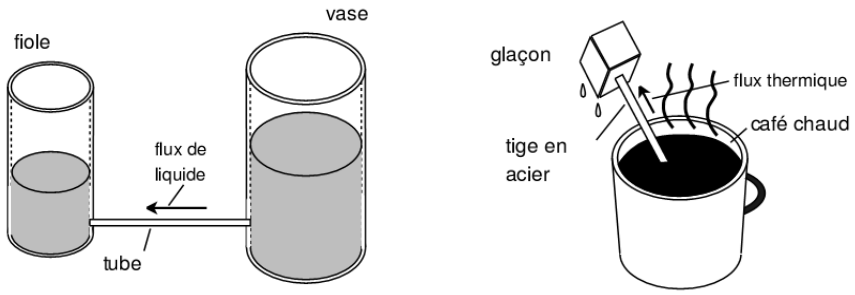


Figure 6 : analogie de Falkenhainer

Le domaine source est représenté par un vase, une fiole remplie d'eau et un tube creux qui relie les deux contenants. Le domaine cible est représenté par une tasse de café chaud, un glaçon et une tige d'acier les reliant. De plus, des connaissances concernant la propriété des objets et leurs relations sont à priori disponibles : la pression de l'eau dans le vase est supérieure à la pression de l'eau dans la fiole et cette différence engendre un flux de liquide dans le tube. Le café a une température supérieure au glaçon, et la tige de métal est conductrice de chaleur. Le raisonnement par analogie consiste à appareiller le vase et le café, la fiole et le glaçon, la pression et la température et à imaginer qu'il existe un flux thermique du café vers le glaçon.

Il existe d'autres analogies connues comme celle de Rutherford qui consiste à proposer un modèle planétaire de l'atome en faisant une analogie entre le domaine spatial et le domaine atomique en appareillant soleil et noyau, planète et électron, force gravitationnelle et force électromagnétique.

La mise en correspondance analogique requiert la découverte d'une structure commune entre deux situations, l'alignement de ces situations, puis l'inférence de propriétés de la base vers la source. L'individu en train de raisonner par analogie doit être en mesure d'évaluer la mise en correspondance et ses inférences. Deux procédés peuvent alors apparaître : la re-représentation d'une ou des deux analogies pour améliorer la correspondance et l'abstraction de la structure commune aux deux analogies.

D'un point de vue fondamental, la similarité est comme l'analogie car elles nécessitent toutes deux un alignement [Gentner, 1983]. La différence est que pour l'analogie, seuls les prédicats relationnels sont partagés, alors que pour la similarité, les prédicats relationnels et les attributs des objets sont partagés.

La figure suivante montre la distinction entre similarité et analogie en positionnant ces deux concepts dans un espace dépendant du degré de similarité des attributs des objets comparés et du degré de similarité de leurs relations.

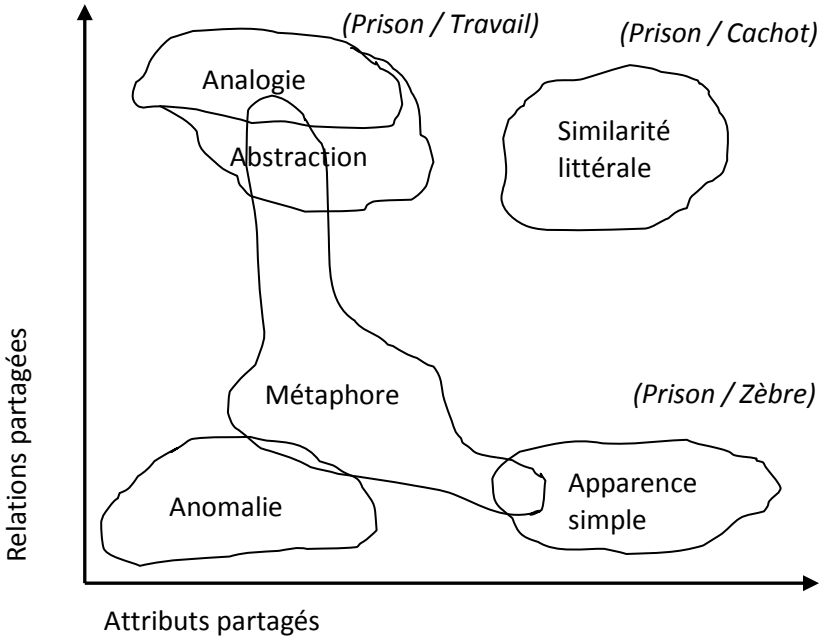


Figure 7 : espace de similarité [Gentner et al., 1997]

L'analogie correspond aux cas dans lesquels les comparaisons ont un haut degré de similarité relationnelle et un faible degré de similarité d'attributs. Plus la similarité entre attributs est grande plus la comparaison tend à une similarité littérale. Les mises en correspondance de type apparences simples partagent seulement des attributs (par exemple « planète » et « balle »). Le coin en bas à gauche de la figure montre une comparaison anormale qui ne partage ni relation ni attribut. Enfin, on peut voir que la métaphore va de la comparaison relationnelle (« la gare est une ruche ») à la comparaison d'attributs (« la lune est comme une pièce d'argent »).

Depuis les années soixante, les travaux sur l'analogie ont donné lieu à plusieurs modèles générateurs d'analogies. French dans [French, 2002] propose de les classer en trois catégories : les modèles symboliques, les modèles connexionnistes et les modèles hybrides.

2.2.1 Les modèles symboliques

Les modèles symboliques utilisent les paradigmes propres à l'intelligence artificielle tels que la logique et la planification.

Le premier modèle symbolique fut Argus [Reitman, 1965] dont le programme résout des analogies proportionnelles en se basant sur un réseau de concepts, des interactions entre ce réseau et le problème à résoudre. Puis est apparu ANALOGY [Evans, 1968] dont le but est de traiter des analogies du type « A est pour B ce que C est pour ? ». Les objets traités sont des figures géométriques. La principale caractéristique d'ANALOGY est de générer une description haut niveau construite par programme, à partir de la description bas niveau de figures géométriques. Cette application ne traite que des analogies sur un domaine donné et pas entre deux domaines différents.

Le modèle JCM [Becker, 1969] tente d'introduire des notions liées à l'apprentissage telles que la mémoire de travail et la mémoire à long terme dans lesquels sont stockées les représentations entre lesquelles des relations ordonnables existent. Le modèle ZORBA-1 [Kling, 1971] se base sur la logique formelle pour générer des preuves de théorème automatiquement, en choisissant des problèmes sources résolus, la preuve est appliquée à un problème cible.

D'autres modèles symboliques ont vu le jour parmi lesquels celui de Carbonel [Carbonell, 1983] appliquant la théorie des moyens et des fins (Means-ends analysis) à la recherche d'analogies, MEDIATOR, le premier programme de raisonnement à partir de cas (Case-Based Reasonning), Prodigy/Analogy [Veloso et al., 1993] combinant l'approche dérivationnelle de Carbonell et le raisonnement à partir de cas.

Le modèle le plus important concernant les modèles symboliques est le modèle SME (*Structure Mapping Engine*) qui est l'application de la théorie de la mise en correspondance structurelle développée par Gentner [Gentner, 1983]. Le principe est que les bonnes analogies sont réalisées sur la base des relations entre les entités plutôt que sur leurs attributs, et les analogies entre systèmes de relations sont préférées aux analogies entre relations individuelles.

De nombreux modèles ont utilisé le SME : le système MAC/FAC [Forbus et al., 1995] (*Many Are Called / Few Are Chosen*) dont l'idée est que la recherche en mémoire d'une situation analogue est basée sur un processus à deux étapes. Dans un premier temps, toutes les sources qui présentent des similarités apparentes (attributs) avec la cible sont recherchées. Dans un deuxième temps, seules les sources qui ont une bonne similarité littérale (relations) sont conservées. Ce système a été testé avec des histoires, des fables, des pièces de Shakespeare et des descriptions de phénomènes physiques. MAGI [Ferguson, 1994] détecte les régularités dans une situation en mettant en correspondance une scène avec elle-même. MAGI a été testé sur des exemples visuels et sur des représentations à la fois perceptuelles et visuelles. IAM [Keane et al., 1988] met en correspondance des parties du domaine source et cible et construit graduellement une interprétation basée sur ces sélections. I-SME (*Incremental SME*) [Forbus et al., 1994] basée sur la même architecture que IAM utilise conjointement des traitements séries et parallèles.

Un modèle SEQL [Kuehne, 2001] est basé sur l'idée que les catégories reposent sur des représentations structurées et sont formées au cours d'une série d'abstractions au fil des comparaisons réalisées. Ce système prend des représentations de figures géométriques en entrée.

Tous ces modèles partagent l'idée que l'analogie repose sur les relations entre les entités plutôt que sur leurs attributs.

2.2.2 Les modèles connexionnistes

L'avantage de l'approche connexionniste est que les représentations associées ont une mesure de similarité intégrée qui traite facilement le problème de « similaire mais pas identique ». ACME [Holyoak et al., 1989] propose une architecture où l'analogie résulte de l'activation des nœuds du réseau. Trois contraintes doivent être satisfaites simultanément : la similarité structurelle, la similarité sémantique, et l'importance pragmatique.

A partir des représentations de la source et de la cible, un réseau local satisfaisant les contraintes est construit à l'intérieur duquel les nœuds *hypothèse* correspondent à toutes les hypothèses possibles d'appariement de la source vers la cible. Des liens inhibiteurs et initiateurs implémentent les contraintes. De cette façon les hypothèses contradictoires entrent en compétition et ne sont pas actives en même temps alors que les nœuds consistants se supportent mutuellement. Cette méthode donne toutes les mises en correspondance possibles et trouve la meilleure. ARCS [Thagard et al, 1990] est un modèle de recherche couplé avec ACME dans lequel la mise en correspondance est dominée par les similarités structurelles et la recherche est dominée par les similarités sémantiques. Le modèle REMIND (*Retrieval from Episodic Memory through INferencing and Disambiguation*) [Lange et al., 1993] est un modèle qui intègre compréhension du langage et recherche en mémoire par un mécanisme de propagation d'activation. L'idée est que des inférences faites sur les caractéristiques des indices provoquent une recherche en mémoire d'épisodes qui partage les mêmes inférences. Le modèle LISA [Hummel et al. 1997], à l'inverse d'ACME, ne nécessite pas que tous les attributs de la source soient connectés à tous les attributs de la cible pour effectuer l'appariement. Pour associer les structures pertinentes il se base sur une activation sélective et des

contraintes dynamiques. Seules les structures de nœuds oscillant de façon synchrone sont retenues [Shastri et al., 1993]. STAR-I et STAR-II sont des modèles connexionnistes distribués basés sur la notion de produit tensoriel [Smolensky, 1990]. Jani et Levine [Jani et al., 2000] ont développé un modèle connexionniste basé sur la notion de résonance adaptative [Carpenter et al., 1986]. Ce système dispose d'un mécanisme d'association basé sur des triades synaptiques.

2.2.3 Les modèles hybrides

COPYCAT [Hofstadter, 1984][Mitchell, 1993] résout des analogies de la forme « ABC est comme ABD, alors UVW est comme ? » et fournit des réponses possibles du type UVD, UVX, … L'architecture de COPYCAT regroupe un réseau sémantique (simulant la mémoire à long terme) définissant les concepts, une mémoire de travail, une petite mémoire procédurale, et des agents qui construisent, examinent et éventuellement détruisent les structures dans la mémoire de travail et sont en interaction continuelle avec le réseau sémantique. Le système peu à peu devient un ensemble de structures consistantes qui déterminera la mise en correspondance entre la source et la cible.

TABLETOP [French, 1995], LETTER-SPIRIT [McGraw, 1995], and METACAT [Marshall et al., 1997] sont décrits dans [Hofstadter, 1995]. Ces modèles ont plusieurs points communs : ils génèrent une analogie par l'interaction d'agents selon un procédé descendant dans la mémoire à long terme et un procédé montant dans la mémoire de travail ; ils simulent un parallélisme et ils sont de nature stochastique (aléatoire et dépendante du temps). Ces modèles permettent que des mises en correspondance partielles influencent la construction des futures représentations. AMBR [Kokinov, 1988] est basé sur le principe du modèle DUAL [Kokinov, 1994]. Ce modèle est composé d'une multitude de micro-agents qui encodent chacun une partie du savoir déclaratif ou procédural et possèdent une partie neuronale représentant la pertinence de ce savoir dans le contexte courant. La mise en correspondance est issue du comportement collectif des agents ; il en résulte une analogie.

D'autres modèles combinent approches symboliques et mécanismes connexionnistes comme par exemple ASTRA [Eskeridge, 1994] et ABR-Conposit [Barnden, 1994]. ASTRA implémente l'idée d'un raisonnement analogique continu où on prend en compte l'importance des différentes étapes nécessaires à la création d'analogies plutôt que de les traiter indépendamment. ABR-Conposit implémente une mise en correspondance Mémoire de Travail-Mémoire de Travail, crée et modifie les représentations en Mémoire de Travail, avec l'idée de combler le fossé symbolique de l'approche connexionniste.

2.3 Discussion sur la notion de similarité

Différents auteurs proposent différentes approches :

Marcotorchino dans [Marcotorchino, 1991] propose une distinction entre similarités non-informées et similarités informées. Dans le cas des similarités non-informées, la comparaison se fait sur une base uniquement locale où l'on ne prend en compte que les informations qui sont explicitement présentes alors que dans le second cas des informations d'ordre statistique ou symbolique sont utilisées.

« Eléments » et « relations » sont distingués pour la première fois dans [Premack, 1983]. Un « élément » est ce qui est perçu comme un tout et une « relation » est une structure constituée d'au moins deux éléments. Une relation peut aussi être perçue comme un tout et servir d'argument dans une relation plus abstraite. Par exemple, un texte peut être considéré en tant qu'élément ou en tant que structure de relations entre des sections, des paragraphes,… Les similarités recherchées sont alors situées au niveau des éléments ou au niveau des relations.

Halford dans [Halford, 1992] propose quatre niveaux d'abstraction pour définir les similarités en s'appuyant sur la distinction élément/relation. Au premier niveau, la similarité est identifiée au niveau des éléments et seules les caractéristiques communes des objets (les attributs) sont retenues. Au second niveau, la similarité est identifiée au niveau des relations entre deux éléments (similarité

de relations, de premier ordre). Au troisième niveau, la similarité est identifiée au niveau des relations entre relations (relations de 2ème ordre). Au quatrième niveau, la similarité est identifiée au niveau des relations entre des systèmes de relations. Le terme de « similarité de surface » est utilisé pour nommer les similarités au premier niveau, et le terme de « similarités structurelles » regroupe les similarités des trois niveaux suivants. Les similarités structurelles exploitées dans notre approche sont des similarités structurelles de 1er niveau.

Chez Gentner [Gentner, 1983][Gentner et al., 1989] les similarités de surface sont celles qui sont identifiées sur la base des objets et de leurs propriétés descriptives. Les similarités dites de structure sont celles qui sont identifiées sur la base des relations entre deux objets (relation de bas niveau), entre un objet et une relation, ou entre plusieurs relations (relations de haut niveau). Gentner considère aussi différentes combinaisons de ces similarités : « la simple apparence », qui regroupe uniquement des similarités au niveau des propriétés descriptives partagées par les objets (attributs d'objets); « la similarité littérale », qui regroupe des propriétés descriptives et des relations objet-objet; « la métaphore » qui peut partager des attributs et/ou des relations ; « l'analogie », qui s'appuie sur des similarités de relations (premier et deuxième ordre), en considérant peu les similarités au niveau des attributs ; « l'abstraction » qui s'appuie sur l'appariement de relation d'ordre supérieur et pas sur les similarités au niveau des attributs. Dans les cas de l'analogie et de l'abstraction, c'est le rôle que jouent les objets dans la structure relationnelle qui détermine leur appariement. Le type de prédicat importe pour évaluer la similarité, et pas seulement le nombre d'éléments communs [Thorndike, 1913] ou le rapport entre le nombre de similarités et le nombre de différences [Tversky, 1977].

Holyoak différencie similarité de surface et de structure dans le cadre de la résolution de problème et montre que la similarité de surface n'a pas d'influence pour la résolution de la tâche, alors que la similarité de structure si. Gentner ne partage pas cette opinion et considère que les similarités de surface sont utiles pour la détection d'analogie et donc favorables à l'apprentissage et à la résolution de problèmes. Holyoak et Thagard [Holyoak et al., 1989] considèrent que la mise en appariement d'une source et d'une cible doit être basé sur des contraintes syntaxiques, sémantiques et pragmatiques. Les contraintes syntaxiques, qui font référence aux rôles grammaticaux des mots, sont identiques à celles proposées par Gentner. Cependant, dans le cadre d'une correspondance analogique Thagard pense qu'il faut autoriser l'appariement de relations nommées différemment. Les contraintes sémantiques, qui correspondent aux relations entre concept, sont de quatre types : identité, synonymie, hyperonymie (taxonomie de relations) et méronymie (taxonomie de composants). Enfin, les contraintes pragmatiques expriment les éléments importants pour atteindre le but de l'analogie et correspondent en pratique à une pondération des descripteurs. L'approche de Holyoak et Thagard [Holyoak et al., 1995] se base sur l'idée que les contraintes agissent simultanément les unes sur les autres et qu'elles participent toutes au résultat de la comparaison [Medin *et al.,*1993]. Les similarités sont examinées à différents niveaux d'abstraction, lorsqu'un niveau élevé de structuration est atteint, il met en évidence les relations causales pertinentes qui rapprochent les données initiales du but du problème [Bracke, 1998].

Plusieurs auteurs expriment un lien entre similarité de surface et similarité de structure : Medin dans [Medin et al., 1989] et Rips dans [Rips, 1989] considèrent que les similarités de surface (similarités perceptuelles facilement accessibles) s'avèrent très pertinentes parce qu'elles sont fréquemment contraintes et parfois générées par les propriétés plus profondes des concepts qui constituent des similarités de profondeur. La capacité à tirer profit d'indices de surface pour trouver des similarités plus profondes est appellée « *psychological essentialism* ». De même Gentner considère que les similarités de surfaces servent les similarités structurelles notamment dans la comparaison d'histoires similaires où les similarités de surface aident à la découverte de similarités structurelles.

Vosniadou dans [Vosniadou, 1989] précise la position de Medin et Ortony dans le cas où des similarités sont recherchées entre deux domaines différents. Les concepts sont alors différents, les similarités sont situées au niveau des relations et la comparaison est de l'ordre de l'analogie. Les similarités de surface, s'il y en a, sont souvent inutiles voire nuisible. L'auteur insiste sur ce qu'elle

appelle les similarités saillantes – *salient* -, qui peuvent être perceptuelles, conceptuelles ou relationnelles et donc à la fois de surface et de structure.

3. Similarité et recherche d'information

La RI poursuit différents buts : la recherche ad hoc a pour but de restituer les documents d'une collection donnée qui sont pertinents par rapport à une requête, la catégorisation de documents cherche à classer les documents dans différentes catégories définies préalablement. La classification de documents consiste à regrouper des documents dans des classes distinctes en fonction de leurs propriétés intrinsèques. Dans toutes ces activités de RI, la similarité joue un rôle central. La recherche est abordée dans le chapitre suivant, cette partie traite de la catégorisation et de la classification

3.1 Catégorisation de documents

Le problème de la catégorisation consiste à classer des documents en fonction des sujets dont ils traitent ou des utilisateurs qu'ils sont susceptibles d'intéresser. Les documents qui ne traitent d'aucun des sujets retenus ne sont associés à aucune catégorie. Les termes « routage » (*routing*) et « filtrage » (filtering) sont deux synonymes employés quand le système de catégorisation compare un grand nombre de profils archivés à des documents individuels [Croft, 1995]. Le système est traditionnellement entraîné sur un ensemble de documents sensés être similaires aux documents à classer dans la pratique. La difficulté est de ne pas retenir trop de caractéristiques pour classer les documents afin d'éviter un problème de sur-apprentissage. Face à ce problème, deux méthodes sont utilisées [Schutze et al., 1995] : la re-paramétrisation qui consiste à remplacer l'espace des caractéristiques par un espace de dimension réduite dans lequel les caractéristiques les plus importantes sont retenues et la sélection de caractéristiques les plus à même de différencier les documents pertinents des non pertinents. Une fois l'espace des caractéristiques réduit, il existe plusieurs méthodes d'apprentissage. Schutze dans [Schutze et al., 1995] compare les méthodes de régression logistique, d'analyse linéaire discriminante et les réseaux de neurones à la méthode de *relevance feedback* de Rocchio [Rocchio, 1971] utilisée sur l'ensemble d'entraînement pour générer la catégorie « requête ». Il montre que les trois méthodes obtiennent des résultats 10-15 % supérieurs à la méthode de Rocchio sur la tâche de routage de TREC2 et TREC3. Dumais dans [Dumais et al., 1998] compare cinq méthodes d'apprentissage sur la collection *Reuters* : la méthode relevance feedback de Rocchio, les arbres de décision, les réseaux bayésiens, la classification naïve bayésienne et les machines à vecteurs de support (*SVM)*. Il montre un avantage pour les SVM en termes de qualité de classification et en termes de temps de calcul nécessaire pour obtenir cette classification. La méthode des arbres de décision, plus lente que les SVM, arrive néanmoins en seconde position, suivie par les réseaux bayésiens et la classification naïve bayésienne. La méthode relevance feedback obtient les moins bons résultats.

3.2 Regroupement de documents

Alors qu'en catégorisation les objets sont associés en fonction de leur similarité à des représentants de catégorie, la classification consiste à organiser un ensemble d'objets en classes homogènes.

Ainsi en RI, le principal intérêt de la classification de documents est de faire apparaître dans une collection la structure en thèmes, sous-thèmes,… Les méthodes de classification produisent des structures soit hiérarchiques [1988, Willet], soit non-hiérarchiques. Les structures hiérarchiques sont utilisées quand une analyse des données est nécessaire ; les structures non-hiérarchiques sont utilisées quand le nombre de données est très important car elles offrent des traitements plus rapides.

3.2.1 Méthodes hiérarchiques

Dans les méthodes hiérarchiques, les documents les plus similaires sont regroupés dans des groupes aux plus bas niveaux, tandis que les documents moins similaires sont regroupés dans des groupes aux plus hauts niveaux. Le principe est de regrouper les individus les plus proches afin de former une nouvelle classe, ou bien d'ajouter des individus à une classe déjà existante qui leurs est proche. Ces méthodes nécessitent la définition d'une fonction de similarité (ou de distance) entre groupes. Pour cela, le Cosinus – cf. chapitre 2 section 2.2- est largement utilisé mais la distance entre classes [Korfhage, 1997] peut être aussi :

- la plus petite distance qui sépare un document de la première classe d'un document de la seconde. Cela correspond à la distance qui sépare les deux documents les plus proches.
- la distance qui sépare les deux documents appartenant respectivement à chacune des classes les plus éloignés.
- la moyenne de toutes les distances qui séparent le document de la première catégorie du document de la seconde.
- la distance entre les centroïdes de ces groupes. Le centroïde est le vecteur moyen de tous les éléments dans le groupe.
- la distance entre les medoïdes de ces groupes. Le medoïde est l'élément le plus au centre du groupe.

Des mesures de dissimilarités peuvent être employées quand il s'agit de maximiser la distance entre les groupes. Une mesure de dissimilarité est une mesure dont la valeur est proportionnelle à l'éloignement entre deux documents dans l'espace des documents.

3.2.2 Méthodes non hiérarchiques

Dans les méthodes non-hiérarchiques, les groupes sont au même niveau. Ces méthodes sont généralement itératives : elles consistent à agréger à chaque itération les classes d'objets qui préservent au plus une certaine compacité au sens de la moyenne des distances à l'intérieur des classes ou de l'inertie intra-classe.

Plusieurs méthodes sont employées parmi lesquelles :

- La méthode des *centres mobiles* : consiste à déterminer un nombre de classes *k* et une métrique *d*. On choisit au hasard *k* points appelés « centres » pour représenter le centre des *k* classes. Les individus sont associés à une classe dont ils sont proches du centre selon une métrique *d*. Les centres de gravité des classes ainsi obtenues constituent les nouveaux centres qui fournissent une nouvelle partition.
- La méthode des *nuées dynamiques* : consiste, à partir d'une partition initiale, à améliorer itérativement la partition de l'espace en minimisant la variance et en maximisant l'écart entre les classes.
- La méthode *k-means* : consiste à partitionner les données en *k* groupes distincts. Pour cela, des *prototypes* sont positionnés dans les régions de l'espace les plus peuplées. Chaque individu est alors affecté au prototype le plus proche. Les prototypes sont positionnés par une procédure itérative qui les amène progressivement dans leur position finale stable.

L'une des difficultés de ces méthodes concernent le choix du nombre de classes, il peut être fixé arbitrairement ou automatiquement [Liu et al., 2002].

Certaines méthodes s'adaptent à la classification comme à la catégorisation : Eric Gaussier propose dans [Gaussier et al., 2002] un modèle génératif hiérarchique adapté à la classification

automatique de documents comme à la catégorisation de nouveaux documents dans un hiérarchie existante. Dans ce modèle les documents sont modélisés comme des couples de termes adjacents plutôt que comme des vecteurs de termes pondérés. Les données sont structurées par un modèle probabiliste hiérarchique générique.

Les résultats d'une classification hiérarchique comme non-hiérarchique peuvent être exploités en RI : si un document est pertinent par rapport à une requête, alors les documents du même groupe ont plus de chance d'être également pertinents pour cette requête. Le nombre de groupes est inférieur au nombre de documents, les réponses du système peuvent être regroupées, plutôt qu'être mises dans une liste individuellement. L'avantage de cette présentation de résultats est que l'utilisateur peut avoir une idée globale des résultats.

La modèle de recherche d'information présenté au chapitre 4 peut être utilisée pour catégoriser ou classer des documents : le résultat du calcul de similarité permet d'identifier les documents similaires ainsi que les termes similaires. La similarité calculée est caractéristique des ressemblances structurelles qui lient les documents comme les termes. Les résultats de ce calcul peuvent être exploités pour former des catégories. En gardant les composantes connexes du graphe bipartite représentant les documents composés de termes, il est possible de générer des catégories dont les éléments n'appartiennent qu'à un groupe (*hard clustering*). De même, il est possible de classer des documents sur la base des similarités entre les termes qui les composent, si les termes choisis pour représenter les classes appartiennent à une composante connexe du graphe représentant les données, alors les éléments classés appartiendront à plusieurs classes (*soft clustering*). Bien qu'il existe des ponts entre ces activités, les expérimentations du chapitre 5 ne portent pas sur les activités de catégorisation et de classification, mais sur la recherche ad hoc.

Conclusion du chapitre Similarité

La notion de similarité est complexe et est utilisée dans des domaines variés. Cette notion est centrale à cette thèse, nous retiendrons principalement la distinction entre la similarité de surface et la similarité structurelle. Les domaines de RI exploitant la notion de similarité utilisent généralement des mesures de distance basées sur les attributs des objets comparés. Par exemple, pour obtenir des groupes de documents similaires, les documents sont positionnés dans un espace dont les dimensions (les attributs) permettent le regroupement, selon des critères ou des métriques donnés. Se sont les mesures de distance au sens géométrique qui permettent la composition des groupes. Nous avons souhaité exploiter un autre type de mesures de similarité pour comparer des documents et plus spécifiquement des documents à une requête. Ce type de mesures est inspiré des travaux de Gentner [Gentner, 1997] dans lesquels la comparaison de deux objets se base sur les attributs communs ainsi que sur les relations entretenues avec les autres objets du voisinage.

Le but recherché en RI ad hoc est de trouver dans un ensemble de document le sous groupe de ceux qui sont similaires à la requête. Les modèles de RI permettent la représentation et la comparaison des représentations des documents et des requêtes. Ils utilisent des mesures de distances basées sur les attributs (les termes composant les documents) comme les méthodes de classification ou de catégorisation de documents textuels.

Nous avons souhaité introduire la notion de similarité structurelle en RI. Cette similarité s'appuie sur les attributs pour définir une similarité initiale entre documents et entre termes. La similarité initiale est ensuite propagée par les relations (liens document-terme) d'un document à l'autre par un processus itératif qui converge. Une fois la convergence atteinte, les documents et les termes peuvent être ordonnés selon cette mesure qui caractérise leur ressemblance structurelle. Nous pensons que la similarité structurelle basée sur les attributs et les relations peut constituer un apport par rapport à l'usage d'une mesure basée sur les seuls attributs des objets comparés.

De plus comme nous le verrons dans les chapitres 4 et 5, en plus d'emprunter l'idée d'utiliser les similarités structurelles au cœur de notre modèle de recherche pour comparer les documents et la requête, nous empruntons également l'idée de la méthode MAC/FAC en réalisant un premier tri basé sur la similarité d'attributs, avant d'exploiter les similarités relationnelles pour élire les meilleures entités parmi celles retenues en première instance.

Chapitre 2 : Recherche d'information

Au début des années soixante, quelques années après l'invention de l'ordinateur, la RI est apparue comme une réponse au besoin de gérer l'explosion de la quantité d'informations. C'est la science de la recherche de l'information dans des documents (dans les documents eux-mêmes, dans les méta-données qui décrivent les documents ou encore, c'est le cas de cette étude, dans les relations qu'entretiennent les documents entre eux) qu'ils soient dans une base de données, dans une base documentaire ou sur le Web. Très tôt, le monde de la recherche s'est intéressé aux SRI qui sont outils dans lesquels sont mis en œuvre des techniques et des mécanismes assurant la gestion automatique des informations documentaires, afin de répondre à un besoin d'information croissant.

Un SRI a pour fonction de permettre à l'utilisateur d'accéder à des documents qui contribuent à combler son besoin d'information, exprimé sous forme de requête, qui motive sa recherche. Ainsi le système peut être vu par l'utilisateur comme un instrument de prédiction de la pertinence des documents d'un corpus par rapport à sa requête.

Pour évaluer l'adéquation entre un ensemble de documents et une requête, le SRI doit posséder d'une part une représentation interne des documents disponibles et de la requête utilisateur et d'autre part d'une méthode de comparaison afin de déterminer leur degré de correspondance. Les représentations internes ainsi que la manière de les comparer définissent le modèle de recherche. La représentation interne d'un document est généralement constituée d'un ensemble de termes index associés à des poids, c'est-à-dire à une valeur reflétant l'importance du terme dans le document dans lequel il apparaît. Cet ensemble est construit lors de l'opération d'indexation. Un modèle théorique doit donner une interprétation précise du poids d'un terme de l'index, il doit aussi prendre en compte, à défaut de les générer, les relations possibles entre les termes d'indexation. Finalement, un modèle doit déterminer la ressemblance entre un document et une requête à partir de leurs représentations respectives. Il est intéressant d'étudier les modèles de recherche car, d'une part cela permet de découvrir et comprendre les besoins qui leur ont donné naissance et de connaître les solutions apportées et d'autre part, la connaissance des modèles théoriques, de leurs propriétés et de leurs limites permet d'envisager d'éventuelles améliorations.

Dans l'histoire de la RI, le modèle booléen fut le premier proposé. L'adjectif « booléen » fait référence à l'usage de l'algèbre de Boole. Un SRI booléen retourne les documents qui contiennent un ou plusieurs termes de la requête. Ainsi un document est soit pertinent soit non pertinent par rapport à une requête donnée. Pour pallier cette absence de nuances dans la pertinence d'un document par rapport à une requête, une extension de ce modèle nommée « booléen étendu » a été proposée permettant de prendre en compte la pondération des termes. Le modèle vectoriel est le premier à intégrer un élément fondamental : la capacité d'ordonner les documents restitués selon un critère de pertinence. Le modèle probabiliste, apparu consécutivement au modèle vectoriel, permet de quantifier l'incertitude dans la représentation des informations ainsi que l'imprécision dans l'expression des besoins. Les différents modèles peuvent être classés en fonction des théories mathématiques sur lesquelles ils se basent. Ainsi il existe trois principales familles de modèles :

- Les modèles ensemblistes
- Les modèles algébriques
- Les modèles probabilistes

Ces modèles classiques donnent naissance à des modèles étendus, dont l'intérêt est de pallier certaines déficiences (dans la représentation ou dans la comparaison) des modèles dont ils s'inspirent. Notre modèle se base sur la théorie des graphes et peut donc être classé dans la famille des modèles algébriques. Le père des modèles algébriques est le modèle vectoriel ; celui-ci exploite les similarités directes (les mots communs) pour comparer deux documents, notre modèle apporte

l'usage des similarités indirectes. Dans la suite, nous présenterons le modèle booléen, pour son caractère historique et pour le fait qu'il représente les modèles à recherche exacte, le modèle vectoriel qui est de la même famille que le nôtre et le modèle probabiliste car dans le chapitre 5 nous comparons notre mesure à la mesure Okapi qui est une mesure de l'approche probabiliste qui fait référence en RI. A la suite de quoi nous positionnerons notre approche par rapport à ces modèles.

L'objectif de cette partie est de familiariser le lecteur d'une part avec le vocabulaire de la RI et d'autre part avec les méthodes de traitement et usages utilisés dans les SRI.

Van Rijsbergen dans [Rijsbergen, 1979] définit la recherche d'information comme suit :
« L'utilisateur exprime son besoin d'information sous la forme d'une requête en vue d'obtenir de l'information. La RI consiste à restituer les documents qui peuvent être pertinents par rapport au besoin d'information exprimé dans la requête. Il est probable que ce procédé soit réitéré puisque la requête demeure un moyen imparfait d'expression du besoin d'information et que les documents restitués à un moment donné permettent d'améliorer la requête utilisée pour la prochaine itération ».

Plusieurs éléments ressortent de cette définition, le premier concerne la façon d'exprimer une requête, le second concerne les fonctions des SRI qui vont permettre de restituer des documents pertinents par rapport à la requête d'un utilisateur. Le troisième point concerne le cycle de RI, c'est-à-dire la suite d'interaction entre l'utilisateur et le système. Ce cycle est principalement composé de l'analyse des résultats présentés et de la formulation d'une nouvelle requête afin de préciser la recherche.

Dans ce chapitre nous insisterons sur le deuxième point qui est au centre de notre travail, à savoir les mécanismes qui permettent à un système automatique de comparer une collection de documents textuels à la requête d'un utilisateur. Ces mécanismes sont composés d'une part des traitements effectués sur les données d'entrée (documents et requête) en vue d'obtenir les représentations de ces données et d'autre part des méthodes utilisées pour comparer les représentations obtenues.

Dans la première partie est décrit le processus de RI, les objets manipulés dans ce processus, ainsi que les différents traitements pouvant être effectués sur les données textuelles.

Dans la seconde partie nous présenterons les différents modèles pour la RI afin de permettre au lecteur de positionner notre modèle parmi les modèles existants. C'est dans cette partie qu'est abordée la comparaison des représentations.

Puis dans une troisième partie nous présenterons les critères d'évaluation des modèles de recherche qui sont utilisés pour évaluer notre modèle dans le dernier chapitre.

Enfin, la quatrième partie présente les campagnes d'évaluations ainsi que les collections que nous utiliserons.

1. Les concepts de base de la recherche d'information

Historiquement, la gestion des documents concernait surtout des spécialistes : bibliothécaires, documentalistes ou conservateurs. Ceux-ci doivent d'une part stocker les documents et en assurer la pérennité, et d'autre part en rendre l'accès possible. Avec l'explosion de la quantité d'informations mises à disposition du grand public et des entreprises, notamment au travers d'Internet, l'automatisation du stockage et de la consultation de l'information est devenue un besoin. Le développement d'outils et de méthodes pour gérer ces quantités d'informations est bien plus qu'un besoin, une nécessité. Ce sont les SRI qui assurent le stockage et permettent la consultation d'informations. Du coté du système, le processus de RI est composé de plusieurs fonctions :

— L'indexation des documents et des requêtes.

— La mise en correspondance requête-documents avec un ordonnancement des documents quand le modèle le permet.
— La restitution des documents reconnus pertinents par rapport à la requête.

Du point de vue utilisateur, le processus de recherche est composé de deux fonctions principales :

— L'interrogation du système par une requête
— L'analyse des documents restitués par le système et une éventuelle reformulation de requête.

Dans un premier temps, nous décrirons le processus de RI, les objets manipulés ainsi que les différents traitements opérés par le SRI sur ces objets pour répondre à ce pourquoi ils ont été créés.

1.1 Le processus de recherche d'information

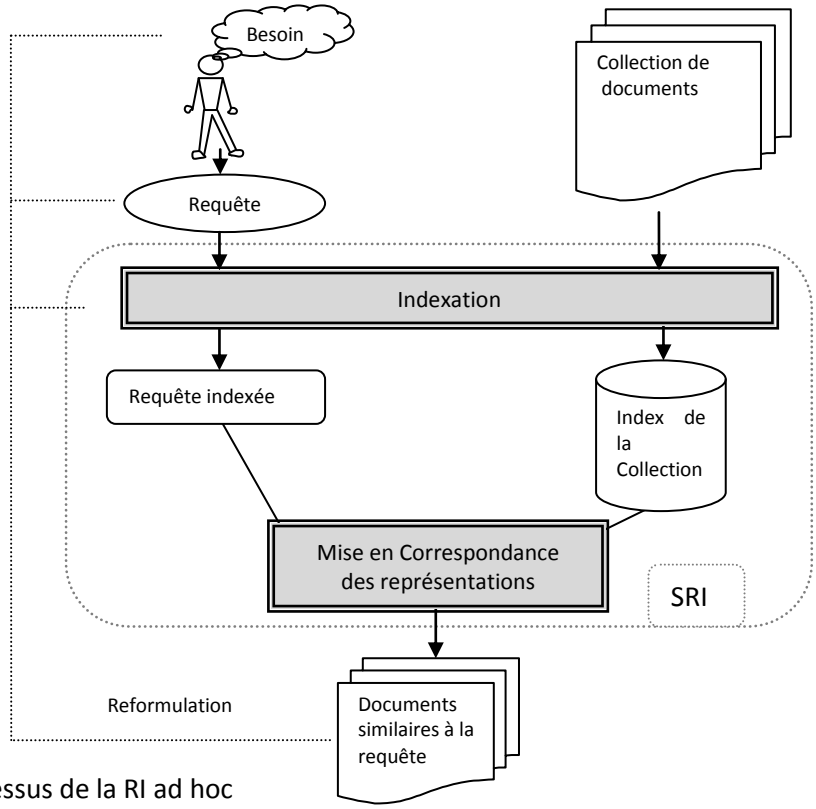


Figure 8 : le processus de la RI ad hoc

La figure 8 représente le processus de RI : un utilisateur formule son besoin d'information sous la forme d'une requête, celle-ci est alors indexée par le système. Dans le même temps ou préalablement, la collection de documents est également indexée. Grâce à l'index (collection et requête indexées), le système est en mesure de construire les représentations puis de mettre en correspondance la représentation de la requête avec les représentations des documents de la collection. Puis il retourne une liste de documents considérés par le SRI comme pertinents par rapport à la requête utilisateur.

Notre étude ne traite pas des éventuelles étapes consécutives à la présentation de résultats telles que la reformulation de la requête. Toutefois, concernant ce dernier point, nos travaux pourront servir de base à des recherches ultérieures.

Dans la suite, nous présenterons les différents éléments de la figure 8. Nous définirons les notions de requête et de document, puis nous détaillerons les étapes de l'indexation. Nous aborderons ensuite la mise en correspondance de la requête avec les documents, les mécanismes de reformulation de requête et le reclassement de documents. Nous détaillerons les modèles de recherche les plus importants et nous présenterons en quoi les choix que nous adoptons diffèrent

concernant la mise en correspondance des représentations dans notre modèle. Nous présenterons les critères d'évaluation des SRI les plus utilisés. Enfin nous présenterons les campagnes et les collections d'évaluation utilisées au chapitre 5.

1.2 Les données d'entrée

1.2.1 Les requêtes

La requête est créée par l'utilisateur, c'est elle qui initie le processus de recherche. Elle traduit un besoin d'information, c'est-à-dire une nécessité ressentie de combler une déficience constatée en information, une lacune ou un défaut. C'est une situation problématique qui amène l'utilisateur à formuler une requête [Schutz et al., 1973].

La requête doit contenir les concepts clés du besoin et les relations entre ces concepts. Elle est issue d'une analyse conceptuelle du besoin d'information qui est effectuée dans l'esprit de l'utilisateur de façon plus ou moins précise. En effet, l'utilisateur fait face à un « problème de vocabulaire » quand il tente de traduire son besoin d'information en une requête. Les mots sont souvent polysémiques et les concepts peuvent être décrits par un grand nombre de mots. [Furnas et al., 198] a montré que deux individus utilisent le même mot pour décrire un concept seulement 10 à 20 % du temps. Les travaux de [Kuhlthau et al., 1990] montrent que le besoin d'information de l'utilisateur devient plus clair et plus précis au cours du processus de recherche. De façon similaire, les sentiments d'incertitude, de frustration et de confusion présents au début de la recherche décroissent au fur et à mesure que la recherche progresse. D'autres travaux approfondissent la notion de besoin d'information [Dervin, 1983][Green, 1990][Kuhlthau, 1993] du point de vue cognitif. Une fois formulée la requête peut avoir la forme d'une expression en langue naturelle, ou encore d'une liste de concepts avec éventuellement un degré d'importance associé, ou encore une formule logique de concepts coordonnés par des opérateurs logiques.

Une fois la requête exprimée, il est nécessaire de lui donner une forme utilisable par un SRI pour entamer le processus de recherche.

1.2.2 Les documents

En se penchant sur l'histoire sémantique du terme *document*, on peut voir que différents sens lui ont été associés au fil du temps, « Document » vient du latin *Documentum* qui signifie leçon, exemple, modèle et secondairement preuve, document, texte. Ce terme partage la même racine que doctrine ou docteur; ces termes sont de la famille du verbe « docere » qui signifie instruire, enseigner [Vignaux, 2003]. Le sens principal du terme « document » est enseignement, jusqu'au 18ème siècle où il devient « preuve ». Sa définition change au 19ème : « Chose qui enseigne ou renseigne ; titre, preuve. Un document précieux. Les documents font défaut pour établir ce point d'histoire » [Littré]. Une définition est adoptée pour trois langues (Allemand, Anglais, Français) en 1935 : « Document : Toute base de connaissance, fixée matériellement, susceptible d'être utilisée pour consultation, étude ou preuve. Exemples: manuscrits, imprimés, représentations graphiques ou figurées, objets de collections, etc. »

Les idées générales que l'on peut extraire de ces définitions sont qu'un document est une trace d'activité humaine, trace laissée dans l'objectif d'être interprétée par des personnes souvent différentes du ou des personnes à l'origine de cette même trace. En conséquence, on peut voir le document comme une chose porteuse de sens pour un auditoire donné. Son contenu s'exprime en une forme interprétable pour quelqu'un. Avec l'avènement des ordinateurs, le document quitte son support matériel natif (le papier pour l'écrit et les bandes analogiques pour les documents audiovisuels) et devient numérique. Il est alors stockable sous la forme d'une représentation binaire

dans les mémoires des ordinateurs. Le document peut être directement pensé et créé sous forme numérique ou bien numérisé à partir de son support original.

Dans l'optique de cette thèse nous entendrons par « document » document textuel numérique. Les analyses effectuées sont faites sur des collections de documents fréquemment utilisées par les chercheurs du domaine. De manière analogue à la requête, les documents des collections étudiées doivent être « indexés » pour être traité par un SRI.

1.3 L'indexation

La représentation des données textuelles a été le champ de nombreuses études [Sparck Jones, 1974][Salton, 1983][Salton1986][Lewis, 1992]. Afin de réduire la complexité des documents et les rendre plus faciles à manipuler, le document doit être transformé.

Dans un SRI, dont l'objectif final est de retourner une liste de documents pertinents par rapport à une requête utilisateur, il est nécessaire de pouvoir rechercher les documents de la collection dont le contenu ressemble ou correspond au contenu de la requête. La recherche implique une méthode de tri et la comparaison de contenu implique une analyse à défaut de pouvoir directement comparer les concepts véhiculés dans le document à ceux présents dans la requête.

Les mots « représentants » ces concepts sont comparés. Les mots qui sont des unités linguistiques porteuses de sens constituent les unités les plus souvent utilisées dans les systèmes actuels. Pour avoir un système de recherche de qualité, il est important que son index reflète au mieux le contenu de la collection originale.

Indexer un document c'est élire ses termes représentatifs afin de générer la liste des termes d'indexation et ajouter à l'index de la collection, pour chacun de ces termes, la liste des références de chaque document le contenant. Par référence on entend identifiant, c'est-à-dire un moyen de retrouver de façon non ambigüe des documents ou un document ou une partie de document où le terme apparaît.

L'indexation des documents est une étape primordiale car elle détermine de quelle manière les connaissances contenues dans les documents fournis sont représentées. Elle a lieu à chaque ajout d'un document dans l'ensemble des documents étudiés.

Qu'elle soit manuelle, semi-automatique ou automatique, l'indexation doit répondre à deux principaux problèmes, le choix des termes représentatifs de chaque document et l'évaluation de leur pouvoir de représentation.

L'indexation automatique implique une analyse automatique du contenu de chaque document de la collection. Cette analyse comprend plusieurs étapes, le but étant d'extraire les termes représentatifs du contenu et d'évaluer leur pouvoir de représentation du contenu ainsi que leur pouvoir de caractérisation du document dans lequel ils apparaissent. Concernant le choix des termes, plusieurs possibilités existent : choisir des groupes de mots ou des mots seuls, retenir des mots ayant certaines propriétés. Bien que l'idée de choisir un groupe de mots comme représentant de concept semble bonne, l'expérience a montré que l'utilisation de représentations complexes améliorait de façon marginale le processus de recherche [Croft, 1995]. De nombreuses expérimentations ont utilisé des mots seuls extraits des documents et de la requête pour la représentation du contenu, et ont montré de bons résultats [Baeza et al., 1992].

Parfois, les termes interdépendants sont sensibles au contexte et ne sont pas forcément de bons représentants dans un contexte différent [Lesk, 1969]. De plus, les méthodes d'analyse syntaxiques même sophistiquées ne permettent pas de produire de bons représentants [Fagan, 1987][Fagan, 1988]. D'autres travaux ont tenté d'améliorer les résultats de Fagan comme ceux de [Lewis, 1990][Smeaton, 1992] et ont montré que le problème réside dans la désambiguïsation des expressions complexes. Par exemple, l'expression « amélioration du processus d'indexation des documents textuels en recherche d'information » est difficile à découper en unités porteuses de sens ainsi un système automatique devra associer « amélioration du processus », « amélioration d'indexation », « amélioration en recherche d'information », … Le risque de telles approches est de

générer de mauvaises associations. Pour ces raisons, c'est une indexation basée sur des mots seuls que nous retiendrons. Une fois choisi ce type d'indexation, il est nécessaire de différencier les termes retenus en fonction de leur valeur présumée de représentant de contenu. Cette différenciation a lieu lors de l'étape de pondération qui est une des étapes finales du traitement que subit le texte indexé.

Voici la suite des opérations traditionnellement effectuées sur les données textuelles lors de l'indexation :

1.3.1 L'analyse lexicale

L'analyse lexicale est l'étape qui permet de transformer un document textuel en un ensemble de termes (« lexème » est parfois employé). Pendant cette phase, la ponctuation, la casse, et la mise en page sont supprimées.

1.3.2 La sélection

Afin de ne garder que les termes importants, plusieurs techniques peuvent être mises en œuvre parmi celles-ci, on utilise souvent un anti-dictionnaire (stoplist) qui permet de ne pas conserver les mots vides de sens c'est-à-dire ne reflétant pas le contenu informationnel des documents. c'est une liste de mots vides qui contient généralement les articles, pronoms, prépositions, les mots outils, ainsi que les mots athématiques c'est-à-dire présents dans le document pour l'introduire ou le présenter mais n'ayant pas de réels rapports avec le sujet traité.

Le traitement lié à un anti-dictionnaire est très simple. Quand un mot est rencontré dans un texte à indexer, s'il apparaît dans l'anti-dictionnaire, il n'est pas considéré comme un index.

La suppression des mots vides doit être contrôlée car elle influence la qualité de la recherche. Il est évident que le rôle d'un mot dans un document dépend du contexte dans lequel il est employé, et qu'il peut avoir un pouvoir d'information différent dans un autre contexte. Ainsi, un anti-dictionnaire devrait être dépendant de la collection c'est-à-dire constitué en fonction de la collection et mis à jour avec de nouveaux documents. Néanmoins, en pratique on utilise souvent un anti-dictionnaire clef en main, par exemple l'anti-dictionnaire du système SMART [Salton, 1988].

Dans nos expériences, nous utiliserons l'anti-dictionnaire du système SMART, et dans certains cas nous enrichirons notre anti-dictionnaire d'autres termes ayant certaines fréquences d'apparition dans les documents, par exemple les termes n'apparaissant que dans un document.

1.3.3 L'utilisation de radicaux

Les variantes d'un mot peuvent être morphologiques [Frakes, 1992] ou sémantiques [Paice, 1996]. Les variantes morphologiques des mots ont la plupart du temps un sens très proche. Par exemple, il peut être utile de retrouver des documents contenant les mots « transmission », « transmis », « transmet », « transmettra », « transmetteur » à partir d'une requête comportant le mot « transmettre ». Pour cela il est possible d'éliminer les différences non significatives et de garder la partie commune. Sur l'exemple, les mots ont la même racine (le lemme) et une terminaison différente (la désinence).

Il est intéressant de représenter plusieurs variantes d'un mot sous une forme unique appelée racine ou radical. Dans la littérature, une différence morphologique entre racine et radical est faite : la racine est la forme abstraite servant de base de représentation à tous les radicaux qui en sont les manifestations. En effet le radical d'un mot est une simple réduction du nombre de lettres de ce mot, et celui-ci peut différer (avoir plus ou moins de lettres) que la racine morphologique correcte. Par exemple, le mot « computation » peut être représenté par plusieurs radicaux « computa », « comput », « compu », sa racine linguistiquement correcte étant « compute ».

Les algorithmes qui permettent de ramener un mot à un radical sont appelés des algorithmes de radicalisation et les algorithmes qui permettent de ramener un mot à un radical particulier qu'est sa racine des *Lemmatisers*. Les algorithmes de radicalisation peuvent être linguistiques, parmi eux citons [Porter, 1980], ils peuvent être automatiques quand ils se basent sur des méthodes statistiques comme par exemple les n-grammes [Adamson, 1974] ou être un mélange des deux comme [Krovetz, 1993][Paice, 1996]. Ils peuvent également se baser sur des lexiques afin de valider ou d'invalider une tentative de transformation d'un mot en radical [Savoy, 1993].

Les algorithmes de suppression des affixes (préfixes et suffixes) sont les plus fréquents [Frakes, 1996]. Nous utiliserons principalement l'algorithme de Porter. L'idée de l'algorithme de Porter est de ramener un mot de langue anglaise à un radical en supprimant sa terminaison. Pour cela il applique successivement plusieurs règles de transformation visant à supprimer le pluriel, les participes passés puis les différentes dérivations telles que « able », « ness », « tly ». Cet algorithme a fait ses preuves [Andrews, 1971][Dawson, 1974] et a été appliqué à d'autres langues, notamment au français et à l'italien [Wechsler et al., 1997] et à l'allemand [Kraaij et al., 1996].

Plusieurs études [Hull, 1996][Jacquemin, 1999] démontrent qu'il est difficile d'évaluer et de comparer les différents algorithmes de radicalisation pour les besoins de la RI. Les expériences réalisées utilisent généralement un seul algorithme de radicalisation, ce qui ne permet pas de mettre en lumière l'influence de tel ou tel algorithme. Néanmoins, l'idée résultante est que l'usage de la radicalisation est bénéfique pour la RI [Lennon, 1981][Frakes et al., 1992][Koskenniemi et al., 1996][Goldsmith et al., 1999].

1.3.4 La pondération

La pondération d'un terme d'indexation est l'association de valeurs numériques à ce terme de manière à représenter son pouvoir de discrimination pour chaque document de la collection. Cette caractérisation est liée au pouvoir informatif du terme pour le document donné. Pour approfondir la notion de pondération en RI, le lecteur pourra se référé à [Salton, 1987] où Salton décrit et compare différentes approches de la pondération.

Informer au sens commun c'est « mettre au courant de quelque chose, donner connaissance d'un fait » [Guiraud, 1967]. Quand un SRI retourne un document à l'utilisateur, celui-ci en tire de l'information. Cette information a un sens pour un utilisateur donné, il se peut que la même information sémantique puisse prendre une importance plus ou moins grande, susciter un intérêt plus ou moins vif selon les individus et les circonstances. L'information a donc un sens pour un utilisateur donné qui lui attribue une certaine valeur. Ensuite l'information a une certaine force : en effet l'énoncé d'un évènement probable informe peu voire pas (« demain il fera jour »), par contre l'annonce d'un évènement improbable suscite plus d'intérêt (« demain il fera beau »). Autrement dit, plus un phénomène est probable moins il est informant. Ainsi, à une information sont attachés un sens et une probabilité qui permet de quantifier, de mesurer son contenu d'informations.

Dans la langue naturelle chaque signe (lettre, phonème, mot, catégorie grammaticale,…) revient avec une fréquence stable, donc prévisible [Guiraud, 1973]. Plusieurs distributions aléatoires rencontrées dans le langage sont de la même forme que celles analysées par les théoriciens de l'information. L'existence de telles distributions a été relevée en 1916 par le sténographe français J.B. Estoup qui a montré que si les mots d'un texte sont rangés par ordre de fréquence décroissante la fréquence du second terme apparaissant dans cette liste est la moitié du premier, la fréquence du $3^{ème}$ est le tiers de celle du premier, … Cette constatation a donnée naissance vingt ans plus tard à la célèbre loi de distribution de Zipf [Zipf, 1949] qui dit que la fréquence d'un mot est inversement proportionnelle à son rang dans la liste des termes classés par fréquence décroissante ou encore que

le produit de la fréquence de n'importe quel mot par son rang est constant. Cette loi est écrite sous la forme :

$$rang \times fréquence = constante \qquad (1)$$

Mandelbrot a montré dans [Mandelbrot, 1965] que la formule de Zipf devait être aménagée en :

$$(rang + b)^a \times fréquence = constante \qquad (2)$$

b correspond à l'aplatissement du sommet de la courbe rang×fréquence qui devient négligeable quand le rang croît. *b* indique qu'un terme peu fréquent est plus important qu'un terme très fréquent.
a est un indice légèrement supérieur à 1.

La relation entre fréquence et rang permet de choisir les termes représentatifs. Luhn dans [Lhun, 1955] a montré que la fréquence d'apparition d'un terme dans un texte en langue naturelle est caractéristique de son pouvoir de représentation du contenu de ce texte. Le pouvoir de représentation d'un terme est parfois nommé l'informativité du terme. Cette notion fait référence à la quantité de sens qu'un mot porte. Un terme très fréquent dans la collection (fréquence absolue) est peu discriminant car il est restitué dans de nombreux documents et inversement un terme très peu fréquent dans un texte a peu d'influence sur le processus de recherche car il n'est pas représentatif du contenu sémantique de ce texte.

Le rapport entre rang×fréquence et importance du terme est représenté par les courbes suivantes :

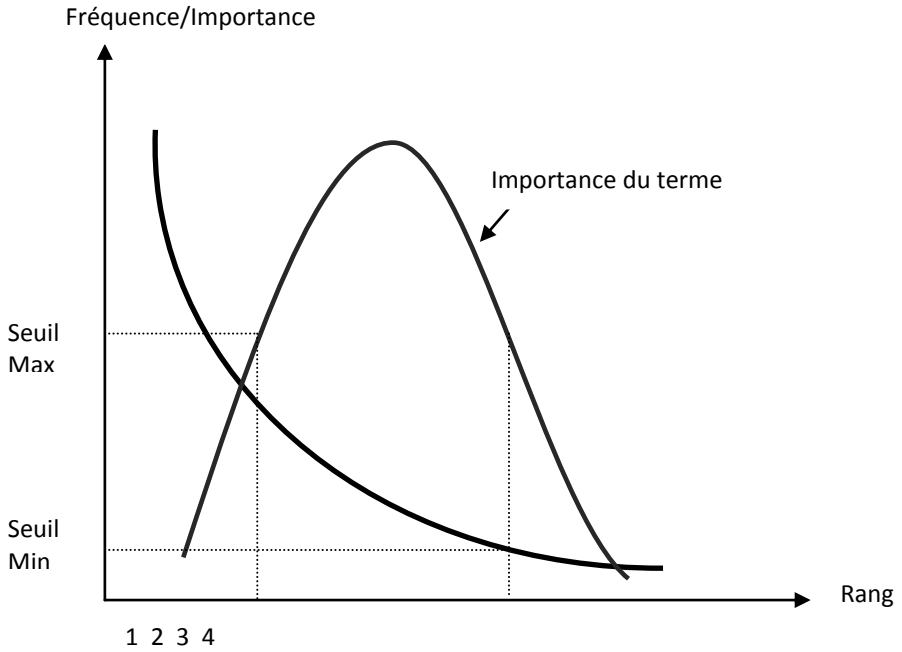


Figure 9 : rapport entre la fréquence d'un terme et son importance

Les termes représentatifs vont être triés en définissant un seuil maximum en dessous duquel les termes à rang×fréquence élevé ne sont pas conservés et un seuil minimum au dessus duquel les termes à rang×fréquence faible ne sont pas conservés non plus.

La fréquence relative d'un terme dans un document est représentative du pouvoir de représentation du terme pour le document, dans le même temps, la fréquence absolue d'un terme

dans la collection est caractéristique du pouvoir de discrimination du terme pour les documents. Il est donc important de prendre en compte la fréquence relative et la fréquence absolue d'un terme lors de sa pondération. La pondération c'est l'association d'une valeur appelée poids à un terme.

Pour associer un poids à un terme on peut procéder de différentes manières :

- *0 ou 1* : exprime la présence (1) ou l'absence (0) d'un terme dans le document.
- *tf* : *term-frequency* est la fréquence du terme dans le document c'est-à-dire le nombre d'occurrences d'un terme dans le document.
- *idf* : *Inverse of Document Frequency* est la fréquence absolue inverse. C'est un facteur qui varie inversement proportionnel au nombre *n* de documents où un terme apparaît dans une collection de *N* documents.

La fréquence absolue inverse est égale à [Salton et al., 1987] :

$$idf = log\ (N/n) \qquad (3)$$

Avec *N* le nombre total de documents dans la collection et *n* le nombre de documents où le terme apparaît.
Le poids d'un terme *j* dans le document *i* s'écrit alors généralement [Sparck Jones, 1972] :

$$poids_i(j) = tf_{ij} \times idf_j \qquad (4)$$

Où tf_{ij} est la fréquence d'apparition du terme *j* dans le document *i* et idf_j est la fréquence absolue inverse du terme *j* dans la collection.

Ainsi le poids d'un terme augmente si celui-ci est fréquent dans le document et décroît si celui-ci est fréquent dans la collection.

La formule *tf×idf* fournit une bonne représentation du poids pour les corpus dont les documents sont de taille homogène c'est-à-dire composés de documents de tailles similaires. Dans le cas de corpus non homogènes, il peut être intéressant de procéder à une normalisation du poids. Pour cela une normalisation est incorporée à la formule du poids [Salton, 1987]:

$$poids_i(j) = tf_{ij} \times idf_j \,/ \sum_{k=1}^{t} tf_{ik}.idf_k \qquad (5)$$

$$\text{Ou } poids_i(j) = tf_{ij} * idf_j \,/ \sqrt{\sum_{k=1}^{t} (tf_{ik}.idf_k)^2} \qquad (6)$$

Avec $\sum_k tf_{ik}.idf_k$ la somme des poids des termes du document *j* et *t* le nombre de termes dans le document.

Il existe d'autres types de normalisation développés dans [Salton et al., 1987].

Pour nos tests, nous utiliserons principalement *tf×idf*, à l'exception des tests de notre algorithme en tant que réordonnanceur – cf. 5eme chapitre - où nous discuterons de l'influence de différentes pondérations. Pour les expérimentations qui portent sur de petites collections, nous retiendrons tous les termes d'indexations (pas de seuil). Quant aux expérimentations sur de grandes collections, nous procèderons à une sélection en ne conservant pas dans l'index les termes n'apparaissant que dans un document (termes ayant leur fréquence dans le document égal à leur fréquence dans la collection), par souci de réduire la taille des données traitées.

1.3.5 Illustration des étapes d'indexation

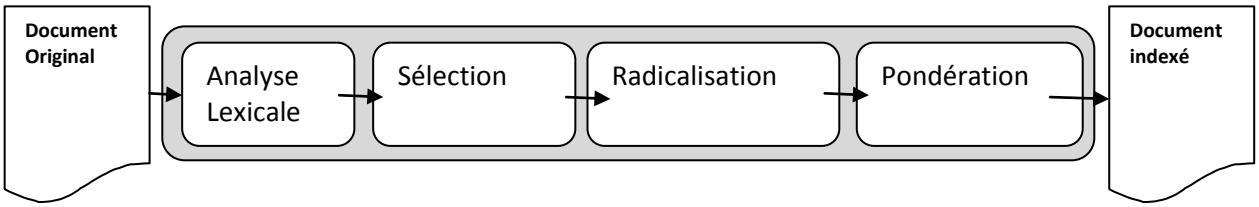


Figure 10 : suite des traitements effectués lors de l'indexation

Pour donner un exemple d'indexation de document telle que nous la pratiquons dans les expérimentations au chapitre 5, on va indexer la première phrase du 1er document du corpus CISI, ayant pour titre « 18 Editions of the Dewey Decimal Classifications » et pour auteur « Comaromi, J.P. ».

— Document original :

The present study is a history of the DEWEY Decimal Classification. The first edition of the DDC was published in 1876, the eighteenth edition in 1971, and future editions will continue to appear as needed.

— Après analyse lexicale :

the present study is a history of the dewey decimal classification the first edition of the ddc was published in 1876 the eighteenth edition in 1971 and future editions will continue to appear as needed

— Après suppression des mots vides :

present study history dewey decimal classification edition ddc published 1876 eighteenth edition 1971 future editions continue needed

— Après radicalisation avec l'algorithme Porter :

present studi histori dewey decim classif edit ddc publish 1876 eighteenth edit 1971 futur edit continu need

1.3.6 Le résultat de l'indexation : l'index

Le résultat d'une indexation donne un ensemble de termes et leurs pondérations pour chaque document comme suit :

$$d_j \rightarrow \{\ldots(t_i, a_{ij})\ldots\}$$

Avec t le terme d'indice i dans le vocabulaire et a_{ij} son poids dans le document d_j.

Avec cette structure, il est facile de trouver les termes inclus dans un document. Cependant, étant donné une requête contenant quelques termes, il est plus intéressant de retrouver les documents correspondant à chacun de ces termes. Pour cela un fichier inversé est construit avec la structure suivante :

$t_i \rightarrow \{\ldots(d_j, a_{ij})\ldots\}$

L'entrée de l'index correspondant au document de l'exemple 1.4.5 avec une pondération tf×idf est :

$d_1 \rightarrow$ {(edit , 0.090) ; (dewey, 0.25); (decim, 0.125); (classif, 0.019); (present, 0.003); (studi, 0.002); (histori, 0.039); (publish, 0.008); (ddc, 0.4); (eighteenth, 1.0); (futur, 0.010); (continu, 0.014); (need, 0.022)}

1.4 L'appariement document / requête

La comparaison entre le document et la requête revient à calculer un score représentatif de la ressemblance entre le document et la requête.

Ce score de similarité entre le document et la requête est donné par une fonction nommée Retrieval Status Value. Elle est notée RSV(*d,q*), où *d* est un document et *q* est une requête.

Traditionnellement le système de recherche retourne à l'utilisateur une liste de documents classés par RSV.

Cette fonction est fondamentale pour la RI car c'est elle qui détermine comment comparer la requête aux documents indexés. Le processus d'indexation et la fonction d'appariement constituent les deux éléments essentiels du modèle de recherche. Avant de présenter les différents modèles de la RI, il est nécessaire d'introduire un dernier concept celui de reformulation de requête.

1.5 Mécanismes de reformulation de requête

La qualité d'un SRI dépend de sa capacité à retrouver des documents pertinents pour l'utilisateur. Elle est donc liée à l'indexation (au choix des termes par le système) et au choix des termes que l'utilisateur a fait pour formuler sa requête. C'est pour résoudre les problèmes liés à un mauvais choix de termes et pour pallier les lacunes de l'indexation qu'ont été introduits (pour la première fois dans [Rijsbergen, 1979]) les mécanismes de reformulation de requête.

La reformulation consiste à réajuster les poids des termes de la requête ou à rajouter des termes reliés à ceux de la requête initiale. Elle peut être manuelle (avec intervention de l'utilisateur) ou automatique. Parmi les méthodes de reformulation de requête, nous pouvons citer la réinjection de pertinence ou relevance feedback. Elle consiste à exploiter l'information que le jugement de pertinence de l'utilisateur fournit sur les documents initialement restitués par le système en lien avec sa requête. La reformulation peut se faire sans expansion de la requête par simple repondération des termes (en maximisant le poids des termes apparaissant dans des documents jugés pertinent et en minimisant le poids des termes présents dans des documents reconnus non pertinent par l'utilisateur) ou par ajouts des termes à la requête originale. L'ajout d'un terme peut se faire de plusieurs manières : soit par une interaction entre l'utilisateur et le système soit de façon automatique, on parle alors de réinjection de pertinence aveugle. Le système considère les premiers documents comme pertinents. Salton et Buckley proposent dans [Salton et al., 1990] une comparaison de différentes méthodes de réinjection de pertinence, il en ressort que la méthode Ide Dec-Hi obtient les meilleurs résultats en termes de précision devant la méthode de Rocchio [Rocchio, 1971] et devant les approches de réinjection de pertinence probabilistes. Selberg propose dans [Selberg, 1997] une description des méthodes de réinjection de pertinence dans les approches classiques de RI (booléenne, vectorielle et probabiliste).

1.6 Le reclassement en recherche d'information

C'est un domaine en plein essor[2] qui concerne aussi bien la RI que l'apprentissage automatique. L'apprentissage automatique est un outil pratique qui permet de régler les paramètres de manière automatique, de combiner plusieurs sources d'informations.

En RI, l'apprentissage automatique a donné lieu à la tâche d'apprentissage à ordonner – *learning to rank*.

L'objectif est de concevoir et d'appliquer des méthodes pour apprendre automatiquement une fonction de classement à partir de peu de données d'entraînement, de sorte à ce que l'on puisse trier des documents sur la base de leur degré de pertinence ou de ressemblance basée sur les caractéristiques – les attributs.

Il existe trois familles de modèle à apprentissage : les modèles génératifs, les modèles discriminatifs et les modèles hybrides.

Les modèles génératifs font l'hypothèse que les documents peuvent être générés par un modèle de language [Croft et al., 2003]. Un modèle de langage est un ensemble de propriétés et de contraintes sur des séquences de mots obtenues à partir d'exemples qui permet de déterminer la probabilité qu'une phrase quelconque puisse être générée par le modèle. Les données d'entraînement sont directement utilisées pour estimer les paramètres du modèle et enrichir le modèle du document. La probabilité qu'un document appartienne à la classe des documents pertinents pour la requête est estimée par la probabilité conditionnelle qu'un document pris au hasard appartienne à l'ensemble des pertinents. Les modèles discriminatifs cherchent d'abord à maximiser la qualité de la classification puis, dans un second temps, une fonction de coût va réaliser l'adaptation du modèle de classification final. Il existe plusieurs modèles discriminatifs comme l'approche de l'entropie maximum [Berger et al., 1996] et les SVM appliquées à la RI [Burges, 1998]. Ces méthodes ont toutes deux été appliquées avec succès à la classification de documents [Ogilvie et al., 2003]. Les approches hybrides utilisent une combinaison des approches génératives et discriminatives, par exemple en faisant un entraînement discriminatif avec une approche générative pour la modélisation et une approche discriminante pour l'entraînement ou encore en combinant les prédictions. Les modèles discriminatifs se montrent plus performants si l'on dispose de nombreux exemples d'apprentissage, dans le cas contraire ce sont les modèles génératifs qui dominent. Croft, dans [Croft, 2007], envisage que l'attention portée aux caractéristiques linguistiques doublée de la prise en compte de données structurées pourrait être une piste de recherche importante. [He, 2008] propose un état de l'art complet de ces approches.

2. Les modèles de recherche

L'indexation choisit les termes pour représenter le contenu d'un document ou d'une requête, le modèle permet de donner une interprétation des termes choisis pour représenter le contenu d'un document. Étant donné un ensemble de termes pondérés issus de l'indexation, le modèle remplit deux fonctions :

- La première est de créer une représentation interne pour un document ou pour une requête basée sur ces termes,
- La seconde est de définir une méthode de comparaison entre une représentation de document et une représentation de requête afin de déterminer leur degré de correspondance (ou similarité).

[2] Cette tâche a été l'objet d'un workshop à SIGIR 2007et 2008 ainsi qu'à ICML 2007 et 2008.

Le modèle joue un rôle central dans la RI. C'est lui qui détermine le comportement clé d'un SRI. De nombreux modèles existent. Dans la suite nous présenterons d'abord le modèle booléen qui est historiquement un des premiers modèles étudiés et qui a servi de point de départ aux recherches du domaine puis le modèle vectoriel (approche algébrique) qui sert de base à notre modèle (approche basée sur les graphes) et enfin, le modèle probabiliste qui, bien qu'étant une approche différente de la notre permettra justement la comparaison avec notre approche.

Pour chacune des approches décrites, les deux points importants seront définis : la représentation et la comparaison. Concernant la représentation interne des documents et de la requête, les principaux modèles utilisent une représentation par mots-clés, et c'est dans la comparaison des représentations que chaque approche a sa propre manière de faire.

2.1 Le modèle booléen

Le modèle booléen est le premier modèle de la RI. Il est basé sur la théorie des ensembles. Un document est représenté par l'ensemble des termes qui le composent.

Le modèle booléen peut être expliqué en considérant une requête formée d'un terme comme une définition non ambigüe d'un ensemble de documents.

Ainsi la requête `retrieval` définit simplement l'ensemble de tous les documents indexés avec le terme `retrieval`. Les requêtes peuvent être composées de plusieurs termes reliés entre eux par des opérateurs de la logique booléenne. Georges Boole a défini trois opérateurs de base : le produit logique AND, la somme logique OR, la différence logique NOT.

Une requête combinant deux termes reliés par un AND retrouvera un ensemble de documents inférieur ou égal à l'ensemble des documents restitués par chacun des termes pris séparément. Par exemple la requête `information` AND `retrieval` retrouvera les documents qui ont été indexés avec les deux termes ; le résultat est donc l'intersection des deux ensembles.

Une requête combinant deux termes reliés par un OR retrouvera un ensemble supérieur ou égal à l'ensemble des documents restitués par chacun des termes pris séparément. Par exemple la requête `graph` OR `model` retrouvera les documents indexés avec `graph` ou avec `model` (ou les deux), le résultat est l'union des deux ensembles.

La figure suivante montre différents ensembles restitués (parties de disques grisés) pour différentes requêtes :

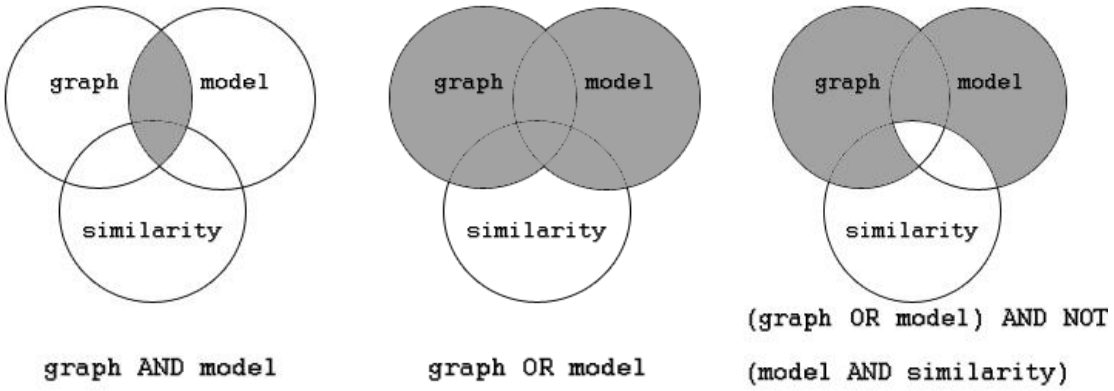


Figure 11 : requêtes booléennes sous forme de diagramme de Venn

Un document est représenté par une liste de termes, par exemple $d = t^1, t^2, \ldots t^n$. Une requête est représentée par une expression logique quelconque de termes utilisant les opérateurs *and*, *or* et *not*.

La correspondance RSV(*d*, *q*) entre une requête *q* et un document *d* est déterminée de la façon suivante:

RSV(d, t_i) = 1 si $t_i \in$ d; 0 sinon. (7)

RSV(d, q_1 AND q_2) = 1 si RSV(d, q_1) = 1 ET RSV(d, q_2) = 1; 0 sinon. (8)

RSV(d, q_1 OR q_2) = 1 si RSV(d, q_1) = 1 OU RSV(d, q_2) = 1; 0 sinon. (9)

RSV(d, NOT q_1) = 1 si RSV(d, q_1) = 0; 0 sinon. (10)

Ce modèle présente de nombreux avantages : il est tout d'abord facile à implémenter et est tout à fait fonctionnel [Frakes et al., 1992]. Il permet aux utilisateurs d'exprimer des contraintes structurelles et conceptuelles [Marcus, 1991]. Les utilisateurs trouvent que l'utilisation de synonymes (grâce à la clause OR) et de groupes de mots (grâce à la clause ET) sont utiles pour la formulation de la requête [Cooper, 1988]. L'approche booléenne possède un grand pouvoir d'expressivité : elle est tout à fait adaptée aux requêtes qui appellent une sélection exhaustive et non ambigüe. Enfin l'approche booléenne peut être tout à fait utile dans la fin du processus de recherche en raison de la clarté et de l'exactitude avec laquelle les concepts sont représentés.

Néanmoins, ce modèle a reçu diverses critiques : la première est qu'il est difficile pour un utilisateur non expert de formuler des requêtes adéquates à l'aide d'expressions booléennes [Fox et al., 1988][Belkin et al., 1992]. Par exemple l'expression « A et B » laisse penser, dans le langage courant, que ce que représentent A et B est quelque chose de plus que A seul. Or, en logique booléenne le ET logique (AND) entre deux ensembles représente leur intersection c'est-à-dire une partie commune. Inversement le OU logique (OR) exprime la somme des ensembles alors que le OU de la langue suggère plutôt un choix : l'un ou l'autre. En plus d'éviter la confusion entre le ET de la langue et le OR logique lors du passage de la requête en langage naturel à la requête booléenne, un utilisateur qui souhaite formuler des requêtes complexes devra se familiariser avec le concept de priorité et l'utilisation de parenthèses et de parenthèses imbriquées.

Le fonctionnement des opérateurs AND et OR pose des problèmes : le AND logique ne fait aucune différence entre deux cas pourtant distincts : si aucun terme ne satisfait la requête ou si tous sauf un satisfont la requête l'opérateur ne retrouve pas de document (*Null Output problem)*. Symétriquement, un usage trop fréquent de OR ramène trop de documents (*Overload Output problem)*.

Le problème majeur de l'approche booléenne est que les documents qui répondent à la requête sont retournés dans un ordre quelconque, et sont tous identiquement similaires à la requête.

2.2 Le modèle vectoriel

Le modèle vectoriel est un modèle algébrique où l'on représente les documents et les requêtes par des vecteurs dans un espace multidimensionnel dont les dimensions sont les termes issus de l'indexation [Salton, 1983]. Comme on l'a vu précédemment, la création de l'index implique le parcours de la collection, la recherche des termes pertinents, le traitement lexical des termes retenus et enfin l'analyse statistique de la distribution de ces termes dans les documents et dans la collection pour leur attribuer un poids. Ainsi, les documents et la requête sont représentés comme des vecteurs dans le repère des termes. La comparaison de la requête au document est effectuée en comparant leurs vecteurs respectifs. On ramène ainsi une proximité sémantique à une mesure de distance géométrique.

Soit R l'espace vectoriel défini par l'ensemble des termes: <t_1, t_2, …, t_n>

Un document *d* et une requête *q* peuvent être représentés par des vecteurs de poids comme suit:

$d \rightarrow <w_{d1}, \quad w_{d2}, \ldots, w_{dn}>$

$q \rightarrow <w_{q1}, \quad w_{q2}, \ldots, w_{qn}>$

w_{di} et w_{qi} correspondent aux poids du terme t_i dans le document d_i et dans la requête *q* et *n* correspond au nombre de termes de l'espace.

Étant donnés ces deux vecteurs, leur degré de correspondance est déterminé par leur similarité. Plusieurs approches peuvent être utilisées pour déterminer la similarité :

	Notation vectorielle	Notation ensembliste	
— Produit scalaire :	$Sim_0(d, q) = \sum_i w_{di} \times w_{qi}$	$\lvert d \cap q \rvert$	(11)
— Cosinus :	$Sim_1(d, q) = \frac{\sum_i w_{di} \times w_{qi}}{\sqrt{\sum_i w_{di}^2 \times \sum_i w_{qi}^2}}$	$\frac{\lvert d \cap q \rvert}{\sqrt{\lvert d \rvert} \times \sqrt{\lvert q \rvert}}$	(12)
— Coefficient de Dice :	$Sim_2(d, q) = \frac{\sum_i w_{di} \times w_{qi}}{\frac{1}{2}\left(\left(\sum_i w_{di}^2\right) + \left(\sum_i w_{qi}^2\right)\right)}$	$\frac{2 \times \lvert d \cap q \rvert}{\lvert d \rvert + \lvert q \rvert}$	(13)
— Mesure de Jaccard :	$Sim_3(d, q) = \frac{\sum_i w_{di} \times w_{qi}}{\sum_i w_{di}^2 + \sum_i w_{qi}^2 - \sum_i w_{di} \times w_{qi}}$	$\frac{\lvert d \cap q \rvert}{\lvert d \cup q \rvert}$	(14)
— Mesure de recouvrement :	$Sim_4(d, q) = \frac{\sum_i w_{di} \times w_{qi}}{min\left(\sum_i w_{di}, \sum_i w_{qi}\right)}$	$\frac{\lvert d \cap q \rvert}{\min(\lvert d \rvert, \lvert q \rvert)}$	(15)

Les documents ayant les plus hauts degrés de correspondance sont retournés en réponse à la requête.

Voici un exemple qui illustre un des intérêts de l'approche vectorielle, à savoir ramener un problème complexe de comparaison de documents à un problème de comparaison de mesures de similarité ou de distances.

Soit l'espace des termes rencontrés pendant l'indexation : {*model, graph, similarity*}
Soit le document d_1 représenté par le vecteur : {(*model*, 1), (*graph*, 2)}
Soit le document d_2 représenté par le vecteur : {(*model*, 2), (graph, 1), (*similarity*, 2)}
Soit la requête q représentée par : {(*graph*, 1), (*similarity*, 2)}

On représente les vecteurs d_1, d_2 et q dans le repère des termes :

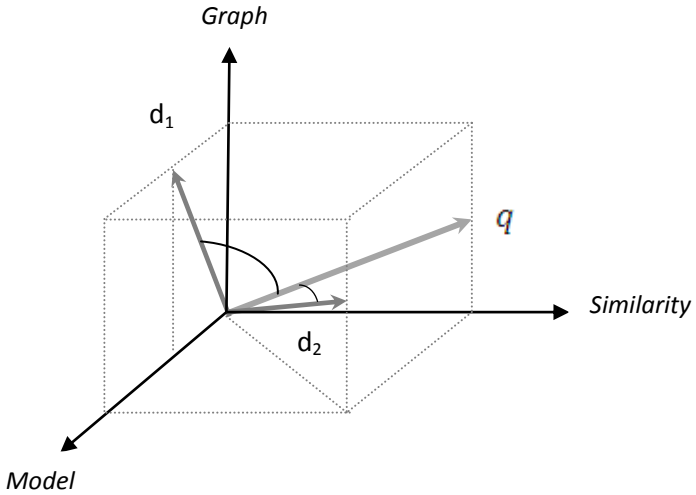


Figure 12 : vecteurs documents et vecteur requête dans l'espace des termes

En RI, dans les modèles à espace vectoriel, il est d'usage de représenter les documents, la requête et les termes d'indexation dans une matrice dont chaque ligne représente un document (la requête étant considérée comme un document) et chaque colonne représente un terme de l'index.

Ainsi les documents et la requête de l'index peuvent être représentés comme suit :

	Graph	Model	Similarity
d1	1	2	0
d2	2	2	2
Q	1	0	2

Ce type de représentation dit matricielle est très utile dès lors que l'on quitte l'espace à trois dimensions.

D'une manière générale, les résultats de recherche tendent à prouver que les systèmes de recherche vectoriels sont plus performants en terme de précision que les systèmes de recherche booléens [Turtle et al., 91].

2.3 Le modèle probabiliste

Plusieurs approches [Maron et al., 1960][Rijsbergen, 1979][Robertson et al. 1982][Bookstein, 1983][Fuhr, 1989] ont tenté de définir la pondération de façon plus formelle s'appuyant souvent sur la théorie des probabilités.

La notion de probabilité d'apparition d'un évènement, par exemple la probabilité de pertinence P(*R*) est formalisée au travers du concept d'expérimentation qui est le procédé par lequel l'observation est faite. L'ensemble des valeurs que peut prendre un fait constitue l'espace de départ. Pour P(*R*) l'espace de départ est {pertinent, non-pertinent}. Le modèle probabiliste considère que les termes d'indexation sont indépendants c'est-à-dire que leur probabilité d'apparition est la même avec ou sans la présence des autres termes. Sous cette hypothèse, on cherche à estimer la probabilité qu'un document soit pertinent par rapport à une requête.

PERT et *NPERT* représentent respectivement la pertinence et la non-pertinence (ou de façon équivalente, l'ensemble de documents pertinents et l'ensemble de documents non pertinents).

Le modèle probabiliste tente d'estimer la probabilité P(*PERT*|*D*) (resp. P(*NPERT*/D)) qu'un document *d* appartienne à la classe des documents pertinents (resp. non pertinents). Autrement dit, on observe la pertinence ou la non pertinence sachant le document *D*. Seules la présence et l'absence de termes dans les documents et dans les requêtes sont considérées comme des caractéristiques observables. Autrement dit, les termes ne sont pas pondérés, mais prennent seulement les valeurs 0 (absent) ou 1 (présent).

On suppose que l'on a une requête fixe. On tente de déterminer les caractéristiques de *R* et *NR* pour cette requête donnée.

La correspondance RSV(*d, q*) entre une requête *q* et un document *d* est déterminée de la façon suivante :

$$\mathrm{RSV}(d,q) = \mathrm{O(D)} = \frac{\mathrm{P}(PERT/\mathrm{D,Q})}{\mathrm{P}(NPERT/\mathrm{D,Q})} \qquad \text{le ratio de Odds} \qquad (16)$$

Plus cette proportion est élevée pour un document, plus ce document est pertinent pour la requête. Cependant, les deux probabilités nécessaires ne sont pas directement calculables. En utilisant les règles de Bayes suivantes :

$$\mathrm{P}(PERT|\mathrm{D,Q}) = \frac{\mathrm{P(D,Q}|PERT) * \mathrm{P}(PERT)}{\mathrm{P(D,Q)}} \qquad (17)$$

$$\mathrm{P}(NPERT|\mathrm{D},\mathrm{Q}) = \frac{\mathrm{P}(\mathrm{D},\mathrm{Q}|NPERT) * \mathrm{P}(NPERT)}{\mathrm{P}(\mathrm{D},\mathrm{Q})} \quad (18)$$

$$\mathrm{O}(\mathrm{D}) = \frac{\mathrm{P}(PERT|\mathrm{D},\mathrm{Q})}{\mathrm{P}(NPERT|\mathrm{D},\mathrm{Q})} = \frac{(\mathrm{P}(\mathrm{D},\mathrm{Q}|\,PERT) * \mathrm{P}(PERT))}{(\mathrm{P}(\mathrm{D},\mathrm{Q}|NPERT) * \mathrm{P}(NPERT))} = \mathrm{Const} \times \frac{\mathrm{P}(\mathrm{D}|PERT,Q)}{\mathrm{P}(\mathrm{D}|NPERT,Q)} \quad (19)$$

Où

P(*PERT*) est la probabilité qu'un document choisi au hasard soit pertinent. P(*PERT*) est une constante dépendante de la requête.

P(D|*PERT*,Q) est la probabilité d'observer *D* sachant que l'on observe la pertinence en présence de Q.

P(D,Q) est la probabilité conjointe du couple D, Q.

O(D) permet alors d'ordonner les documents en fonction de leur estimation de pertinence.

Maron suggère dans [Maron et al., 1960], que P(*PERT*) pourrait être défini par les statistiques sur l'usage du document. C'est-à-dire par le quotient du nombre d'utilisations du document courant par le nombre total d'utilisations. Cette idée est l'origine du classement par popularité, très à la mode sur internet [Joachims et al., 2005].

Selon Jacques Savoy [Savoy, 1994], le modèle de recherche probabiliste est plus efficace que le modèle de recherche booléen, mais moins performant que le modèle de recherche vectoriel. Pour plus d'information, il existe des états de l'art comme [Crestani et al. 1998], ou la présentation des expériences probabilistes en RI de « l'école de Londres » [Sparck Jones , 2000].

Dans la partie qui décrit notre modèle, nous comparerons notre méthode d'ordonnancement à la méthode Okapi qui est la plus connue des méthodes probabilistes. L'approche Okapi aussi connue sous le nom de pondération (ou poids) BM25, a été crée avec le souhait de construire un modèle probabiliste prenant en compte la fréquence des termes ainsi que la taille des documents [Sparck Jones et al., 2000]. Une façon possible pour donner un score à un document *d* c'est de calculer la fréquence absolue inverse des termes qui apparaissent dans la requête *q* – cf. formule (3). Il est possible d'améliorer cette formule en factorisant la fréquence des termes et la taille du document :

$$\mathrm{RSV}(d,q) = \sum_{t_i \in q} \left[\log \frac{N}{n_i}\right] \cdot \frac{(k_1+1)tf_{id}}{k_1\left((1-b)+b\times\left(\frac{dl}{avdl}\right)\right)+tf_{id}} \quad (20)$$

Si la requête et longue alors une pondération similaire peut être ajoutée pour les termes de la requête :

$$\mathrm{RSV}(d,q) = \sum_{t_i \in q} \left[\log \frac{N}{n_i}\right] \cdot \frac{(k_1+1)tf_{id}}{k_1\left((1-b)+b\times\left(\frac{dl}{avdl}\right)\right)+tf_{id}} \cdot \frac{(k_3+1)tf_{iq}}{(k_3+tf_{iq})} \quad (21)$$

Enfin si des jugements de pertinence sont disponibles :

Soient *V* l'ensemble des documents restitués par le système. |*VR*| le nombre de documents reconnus pertinents, |*VR*$_i$| le nombre de documents pertinents contenant le terme t_i.

$$\mathrm{RSV}(d,q) = \sum_{t_i \in q} \left[\log \frac{\frac{(|VRi|+0{,}5)}{|VR|-|VRi|+0{,}5}}{\frac{n-|VRi|+0{,}5}{(N-n-|VR|+|VRi|+0{,}5)}} \cdot \frac{(k_1+1)tf_{id}}{k_1\left((1-b)+b\times\left(\frac{dl}{avdl}\right)\right)+tf_{id}} \cdot \frac{(k_3+1)tf_{iq}}{(k_3+tf_{iq})}\right] \quad (22)$$

Avec *dl* la taille du document, *avdl* la taille moyenne des documents,

k_1 est un paramètre qui permet d'atténuer l'influence de la fréquence des termes dans le document,

k_3 est un paramètre qui permet de graduer la fréquence des termes dans la requête. Prendre k_1= k_3=0 revient à ne pas prendre en compte la fréquence des termes. *b* est le paramètre correspondant à la normalisation par la taille du document. b =1 correspond à une normalisation

complète, b=0 correspond à une absence de normalisation. $0 \leq b \leq 1$ exprime une normalisation intermédiaire ou graduée par rapport à la taille du document.

Cette mesure a été l'objet de nombreuses expérimentations et fait partie des mesures classiques utilisées aujourd'hui en RI [Sparck Jones et al., 2000].

2.4 Modèles et similarité

Les modèles booléens, vectoriels et probabilistes représentent les modèles classiques de RI. En effet, ils utilisent un procédé en trois étapes (décrites précédemment) qui consiste à normaliser les données textuelles en entrée, puis représenter ces données dans une structure adéquate, et enfin mettre en correspondance les représentations des documents avec la représentation de la requête Notre modèle procède comme les modèles classiques, par opposition aux modèles basés sur les concepts où les relations sémantiques des mots sont prises en compte.

L'approche booléenne est une approche dite à similarité exacte : les documents ne contenant pas les termes d'indexation ne sont pas retournés. De plus, les documents retournés sont identiquement similaires à la requête (ils ne se différencient pas dans leur ressemblance à la requête).

Le modèle vectoriel par son approche algébrique introduit la possibilité de pondérer les termes d'indexation et d'ordonner les documents en ramenant la comparaison de deux documents à une comparaison de vecteurs dans l'espace des termes. Ce qui permet de graduer la similarité entre un document et une requête sur la base des termes qu'ils ont en commun.

Le modèle probabiliste définit le coefficient de similarité entre un document et la requête comme la probabilité que le document soit pertinent connaissant la requête.

Ces approches font l'hypothèse de l'indépendance entre les termes d'indexations.

Notre approche présente quelques similitudes avec l'approche vectorielle. En effet les données de départ peuvent être représentées dans une matrice document-terme. Tout en restant très proche de l'approche vectorielle notre méthode s'en dissocie quand notre algorithme propage les similarités par les liens entre documents et termes – cf. chapitre 4 et 5.

Nous faisons les hypothèses suivantes :

- Les termes de la requête sont de bons représentant du besoin de l'utilisateur, les documents contenant ces termes sont pertinents pour la requête (Ce qui va dans le sens des trois approches classiques)
- Les termes apparaissant dans les documents pertinents mais n'apparaissant pas dans la requête bénéficient de la ressemblance directe avec la requête des termes avec lesquels ils co-occurrent, ceci constitue une hypothèse nouvelle par rapport aux modèles classiques.
- Les documents qui contiennent les termes de documents pertinents n'apparaissant pas dans la requête sont considérés comme potentiellement pertinents. Ceci est l'hypothèse qui permet d'accepter l'idée de la propagation des similarités entre documents sur la base de la propagation des similarités entre termes et réciproquement.

Dans le chapitre 4 qui décrit notre approche, nous illustrerons le comportement de notre algorithme en le comparant à la mesure Cosinus du modèle vectoriel, cela afin de mettre en rapport une mesure qui représente la similarité directe (la mesure Cosinus) à une mesure qui représente la similarité structurelle (qui est une similarité basée sur les liens directs et indirects). Dans le chapitre 5 qui traite des expérimentations sur des collections de test, nous comparerons également notre

méthode à la mesure Cosinus et aussi à la mesure Okapi car elle est réputée efficace dans le domaine.

3. L'évaluation des systèmes de recherche d'information

Depuis la naissance du domaine de la RI, l'évaluation des modèles et méthodes proposés a toujours été un centre d'intérêt important. Pour cela des mesures de qualité des systèmes de recherche ainsi que des ensembles de données ont été développés afin de tester ces systèmes sur une base commune. La qualité d'un système doit être mesurée en comparant les réponses du système avec les réponses que l'utilisateur espère. Plus les réponses du système correspondent à celles que l'utilisateur espère, meilleur est le système.

Pour réaliser une telle évaluation, une expérimentation qui utilise les éléments suivants doit être établie:

- Un ensemble de documents.
- Un ensemble de requêtes.
- La liste de documents pertinents pour chaque requête.
- Des mesures et des critères quantifiables.

3.1 Les notions de bases

Consécutivement au traitement d'une requête par le système, les documents de la collection forment deux partitions selon deux caractéristiques :

- Les documents restitués et les documents non restitués.
- Les documents pertinents et les documents non pertinents.

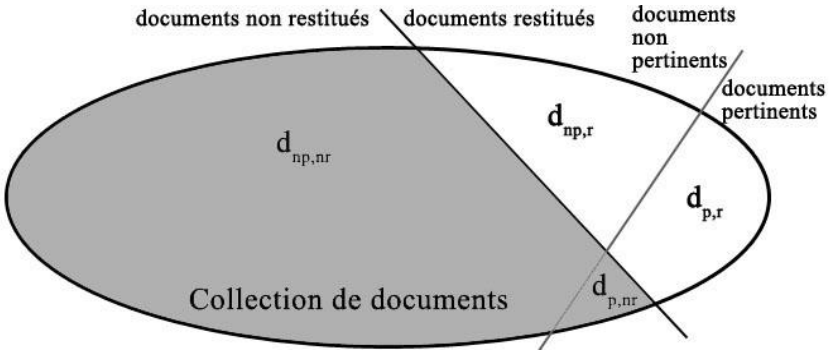


Figure 13 : représentation des partitions de la collection lors d'une interrogation

$d_{np,nr}$ les documents non pertinents non restitués
$d_{np,r}$ les documents non pertinents restitués (bruit)
$d_{p,r}$ les documents pertinents restitués
$d_{p,nr}$ les documents pertinents non restitués (silence)

Pour mesurer les performances qualitatives des SRI, on procède à la comparaison de combinaisons des ensembles de documents pertinents, de documents non pertinents, de documents restitués et de documents non restitués sur l'ensemble des requêtes. Il existe à cet effet de nombreuses mesures, chacune mettant en évidence telle ou telle propriété du système. De nombreuses mesures existent, nous avons retenu les mesures suivantes :

- La précision à *n* documents restitués,
- La mesure F,
- La précision exacte,
- La précision interpolée,

— La précision moyenne.

3.2 La précision et le rappel

Le rappel mesure la proportion de documents pertinents restitués parmi tous les documents pertinents disponibles. Si le rappel vaut 1 c'est que les documents pertinents disponibles ont tous été restitués par le système, inversement si le rappel vaut 0 c'est qu'aucun document pertinent n'a été restitué. Cette mesure permet aussi de déterminer le silence, c'est-à-dire la proportion de documents pertinents non trouvés.

La précision mesure la proportion de documents pertinents restitués parmi tous les documents restitués. Elle mesure la capacité du système à trouver exclusivement des documents pertinents. La précision vaut 1 quand tous les documents restitués sont pertinents. Elle vaut 0 si aucun des documents restitués n'est pertinent. Cette mesure détermine également le bruit, c'est-à-dire la proportion de documents non pertinents restitués par le système.

$$\text{Précision}^{3} = \frac{dp,r}{dp,r+dnp,r} \qquad \text{Rappel} = \frac{dp,r}{dp,r+dp,nr}$$

La valeur de ces taux est influencée par le processus d'indexation. En effet, plus l'indexation est exhaustive, plus le taux de rappel est potentiellement important : toutes les informations susceptibles d'être pertinentes peuvent être restituées, mais certaines ne seront pas pertinentes pour l'utilisateur. Symétriquement, plus l'indexation est spécialisée, plus le taux de précision est élevé, mais cela induit le risque d'avoir un taux de rappel faible : les informations restituées seront pertinentes, mais d'autres informations pertinentes ne seront pas restituées.

Un système qui aurait 100 % à la fois pour la précision et pour le rappel signifie qu'il a trouvé tous les documents pertinents et rien que les documents pertinents. En pratique, cette situation n'arrive pas. Le plus souvent, il est possible obtenir un taux de précision et de rappel aux alentours de 30 % [Jian-Yun Nie, 2004], ce qui implique que le problème auquel répond la RI est non trivial.

Les deux métriques ne sont pas indépendantes: quand l'une augmente, l'autre diminue. Il est facile d'avoir 100% de rappel: il suffit de donner toute les documents disponibles comme réponse à chaque requête. Cependant, la précision dans ce cas serait très basse. De même, il est possible d'augmenter la précision en donnant très peu de documents en réponse, mais le rappel en souffrira. Il faut donc utiliser les deux métriques conjointement.

Le but d'un SRI est d'améliorer la précision sans trop sacrifier le rappel et vice-versa.

Il est intéressant pour étudier la qualité de l'ordonnancement[4] des scores de similarités obtenus par chaque document avec la requête, de regarder la précision P_n ou le rappel R_n du sous ensemble des documents constitués des *n* premiers du retour. Ces deux mesures se notent respectivement P@n et R@n.

Ainsi, il est utile d'examiner la précision à 10 documents restitués si l'on s'intéresse à la capacité du système de restituer des documents pertinents en tête de liste (ce qui est une préoccupation traditionnelle des utilisateurs des moteurs de recherche). La précision à 5, 10, 30, … documents restitués présente néanmoins des limites : par exemple si une requête donnée a seulement 8

[3] En référence à la figure 13

[4] On entend par ordonnancement, le fait qu'un système ordonne la liste des documents qu'il retourne dans l'ordre décroissant des scores de similarité avec la requête. C'est-à-dire l'ordre de pertinence décroissante, pertinence selon le système. En conséquence on ne peut parler de précision à *n* que dans le cas où le système renvoie les documents triés.

documents pertinents, et que le SRI restitue bien ces 8 documents en tête de liste, le SRI aura une précision à 10 documents restitués égale à 0,8, ce qui n'illustre pas que tous les documents pertinents disponibles ont été trouvés. De plus, dans cet exemple, une précision à 10 documents restitué égale à 0,8 ne permet pas de déterminer où se situent les deux documents non pertinents parmi les dix restitués. Pour pallier ce défaut, soit on regarde la précision à différents *n*, soit on utilise la R-précision.

3.3 La précision exacte ou R-précision

La R-précision est la précision à *n* quand *n* est égal au nombre total de documents pertinents. Cette mesure est plus réaliste pour l'étude de l'ordonnancement en tête de liste, mais pour l'obtenir, il est nécessaire de connaître au préalable le nombre de documents pertinents disponibles dans le corpus pour une requête donnée.

Une R-précision de 1.0 signifie une précision et un rappel optimaux.

3.4 La mesure F

Examiner la succession des précisions à n ou des rappels à *n* permet d'évaluer un ordonnancement dans son ensemble et ainsi de déterminer telle ou telle propriété de l'algorithme étudié. Il peut être intéressant d'avoir une valeur synthétisant ces deux mesures. On voudrait pouvoir maximiser la précision et le rappel, mais comme on l'a vu ces deux mesures évoluent souvent de façon opposée. La précision est globalement décroissante au fur et à mesure que le SRI restitue des documents, alors que le rappel est globalement croissant. On peut choisir la mesure F comme valeur synthétique exploitant la précision et le rappel. Elle est calculée comme suit :

F = (2 × Rappel × Précision) / (Rappel + Précision)

Pour évaluer l'ordonnancement, il est possible de calculer la mesure F_n à chaque rang *n :*
F@n=2 × R@n × P@n / (R@n+P@n).

Nous avons exploité cette propriété dans notre travail pour trouver expérimentalement à quel rang *n* la meilleure mesure F est obtenue et chercher s'il est possible d'intégrer un seuil à notre mesure de similarité de manière à obtenir automatiquement une bonne « coupe » sur une liste de documents restitués par notre méthode.

3.5 La courbe précision-rappel

Une autre manière de représenter le couple précision-rappel en fonction de *n* (nombre de documents restitués) est de représenter pour chaque *n*, le nuage des points des couples précision-rappel. Ainsi, en reliant les points par un segment, la courbe précision-rappel a l'allure suivante :

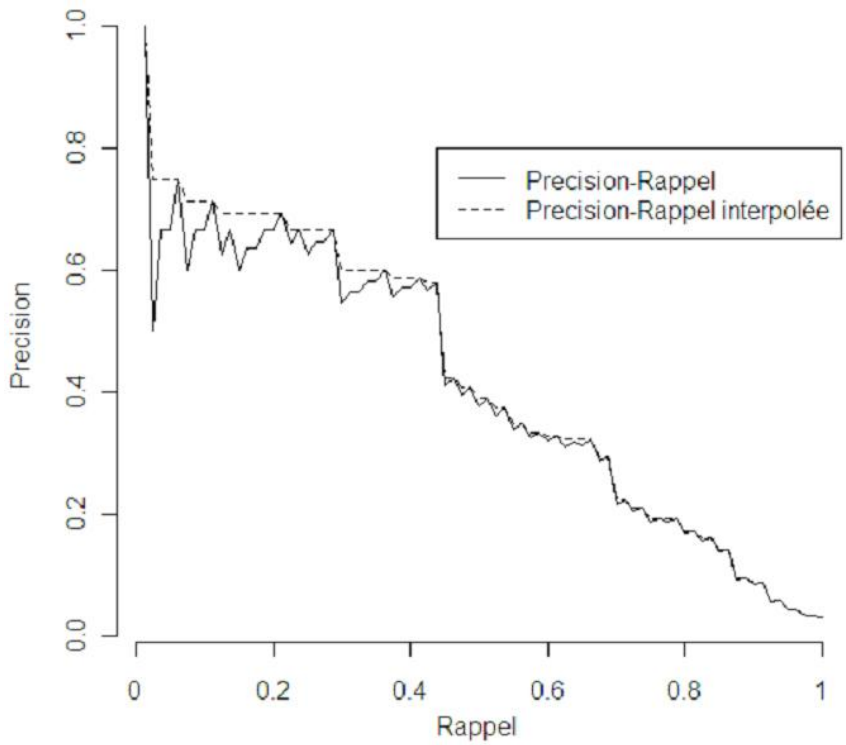


Figure 14 : courbe précision-rappel pour la requête 157 du corpus Cranfield avec la méthode SimRank

Cette courbe globalement décroissante est dentelée, la précision décroit au fur et à mesure que des documents non pertinents sont restitués, le rappel ne varie pas, puis un document pertinent est restitué, alors la précision et le rappel augmentent, puis continuent de croître tant que les documents restitués sont pertinents et décroissent sinon. Plus cette courbe décroît tardivement, meilleur est l'algorithme à l'origine de l'ordonnancement étudié.

Les corpus d'étude sont la plupart du temps fournis avec un ensemble de requêtes, et il peut être intéressant pour comparer deux SRI de disposer de mesures moyennes reflétant le comportement général d'une méthode pour l'ensemble des requêtes d'un corpus donné. A partir de courbes précision-rappel de requêtes différentes il n'est pas possible de calculer une courbe moyenne, la série des valeurs de rappel étant à priori distinctes d'une courbe à l'autre. Nous avons utilisé la technique de [Yang, 1999] qui permet de transformer la courbe précision-rappel en une courbe interpolée (la courbe en pointillé sur la figure 14) continue entre 0 et 1, à partir de laquelle il est possible d'extraire onze valeurs de précision pour onze valeurs de rappel fixées : le rappel varie de 0 à 1 par pas de 0,1. Le rappel varie à chaque fois qu'un document pertinent est restitué et ne varie plus tant que les documents restitués sont non pertinents. A un rappel donné, il est donc possible d'avoir plusieurs précisions. Pour réaliser la courbe en escalier, on associe la précision maximale à un rappel donné parmi les couples rappel-précision pour ce rappel.

Une fois cette opération effectuée sur chaque courbe précision-rappel de chaque requête du corpus étudié, une courbe précision-rappel est obtenue en faisant la moyenne des précisions pour chacune des onze valeurs de rappel fixées.

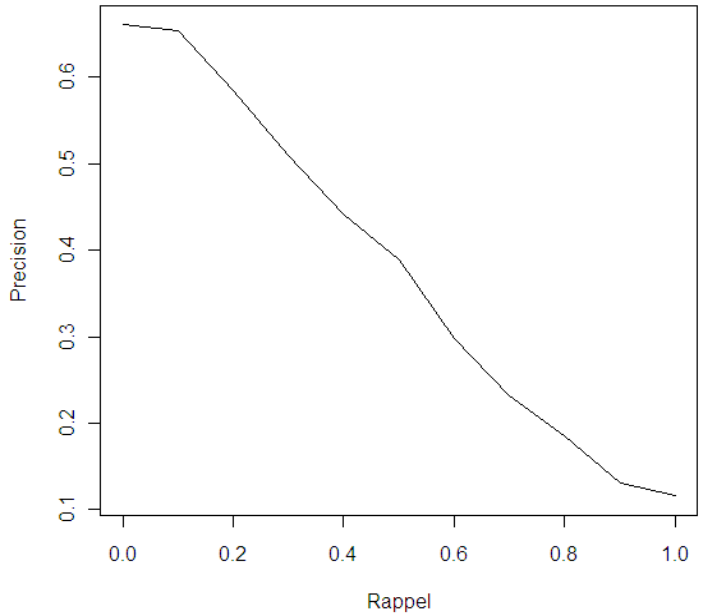


Figure 15 : moyenne des courbes de précisions à 11 points de rappel obtenue pour l'ensemble des requêtes Cranfield avec la méthode SimRank

L'aspect en dents-de-scie a disparu, l'analyse du comportement du SRI est facilitée. La moyenne des précisions à 11 points de rappel mesure la qualité d'un algorithme d'ordonnancement sur une collection de test.

3.6 La précision moyenne

La précision moyenne est une mesure de performance globale. La précision moyenne est une moyenne de précision sur un ensemble de points de rappel.

$$\text{MAP} = \frac{1}{n}\sum_{i=1}^{N} p(i) * R(i) \qquad (23)$$

Avec
R(i) = 1 si le i^{ème} document restitué est pertinent
R(i) = 0 si le i^{ème} document restitué est non pertinent
p(i) la précision à *i* documents restitués.
n le nombre de documents pertinents restitués.
N le nombre total de documents.

La MAP est moyenne des précisions à *i* pour chaque *i* tels que le i^{ème} document est pertinent. Comme la courbe précision à onze points de rappel, la précision moyenne décrit la performance globale d'un système. Elle synthétise en une valeur la qualité d'un système. Elle présente l'avantage par rapport à la courbe précision à 11 points de rappel d'être calculée sans interpolation.

3.7 Les autres critères d'évaluation

L'étude de [Baccini et al. 2010] montre que sur l'ensemble des mesures de RI existantes, un nombre restreints (5 ou 6) de ces mesures permet d'avoir une vue claire des propriétés d'un système

donné. Les mesures que nous avons retenues correspondent à celle identifiées dans [Baccini et al. 2010].

Il ne faut pas perdre de vue que d'autres critères peuvent être mis en avant pour évaluer un système de recherche par exemple le temps de réponse, l'effort fourni par l'utilisateur, la présentation du résultat, éventuellement la taille de l'index, … De nombreuses recherches portent sur la notion d'utilité et font référence au point de vue utilisateur. Il y a d'autres critères d'évaluation comme le ratio de couverture qui fait référence aux documents restitués connus et reconnus pertinents par l'utilisateur, le ratio de nouveauté qui fait référence aux documents restitués reconnus pertinents par l'utilisateur et qui lui étaient inconnus jusque là, l'effort de rappel qui correspond au rapport entre le nombre de documents que l'utilisateur souhaitait retrouver et le nombre de documents lus pour les trouver, ...

4. Les campagnes et collections de test

Depuis l'apparition de l'expression « recherche d'information » dans le mémoire de fin d'étude de Calvin Mooers en 1950, le monde de la RI n'a cessé de développer des outils et méthodes de recherche et, en parallèle des campagnes d'évaluation pour tester les méthodes et outils développés.

Les objectifs des campagnes d'évaluation sont les suivants : encourager la RI sur de grandes collections fermées, développer la communication entre l'industrie, l'université et l'état en mettant en place un forum ouvert pour faciliter les échanges d'idées sur la recherche, augmenter la vitesse de transfert de la technologie du laboratoire de recherche aux enseignes commerciales, rendre disponible et accessible des techniques d'évaluation appropriées pour les industriels et les académiciens. Les principes de bases de l'évaluation en RI ont été mis en place depuis les années soixante pour juger de l'efficacité des systèmes et ainsi faire évoluer la performance des mêmes systèmes, technologiquement mais également par rapport aux attentes des utilisateurs.

Dans nos travaux, nous utiliserons trois corpus différents : le corpus Cranfield, le corpus CISI, et le corpus TREC ad hoc 1998.

4.1 Le projet Cranfield

Cyril Cleverdon [Cleverdon, 1962], libraire du « Cranfield College Of Aeronautics », est à l'origine de deux contributions majeures : Cranfield 1 et Cranfield 2.

Le projet Cranfield 1 (1958-1962), vise à tester l'efficacité de différentes façons d'indexer et de rechercher des documents. Le but est de comparer quatre méthodes d'indexation, sur une base commune constituée d'un ensemble d'articles scientifiques et de rapports (1800 documents au total). Les expérimentateurs ont formé un ensemble de 1200 requêtes en demandant aux auteurs de documents faisant référence dans le domaine (les documents sources) de formuler le besoin d'information qui a motivé l'écriture de l'article. Les systèmes étaient alors évalués par rapport à leur capacité à retrouver les documents sources. Retrouver les documents source correspond à maximiser le rappel. La façon de créer les requêtes ainsi que la méthode des documents sources ont été critiqués, néanmoins cette expérimentation a posé les bases de la construction de collections de test et de la création de critères d'évaluation.

Une deuxième expérimentation a été proposée : Cranfield 2 a permis quelques avancées importantes. Le corpus de test est plus petit que son prédécesseur : 1400 documents et 225 requêtes. Les documents pertinents sont simplement ceux que les experts ont jugés comme tel. C'est dans cette expérimentation qu'est apparu pour la première fois l'usage des courbes de précision-rappel lors de l'évaluation.

4.2 Le corpus CISI

Le corpus CISI est comme le corpus Cranfield un corpus de petite taille : 1460 documents et 120 requêtes. Dans notre étude, nous ne traiterons que les 76 requêtes pour lesquelles il existe des documents pertinents.

4.3 TREC Text REtrieval Conference

La campagne TREC est une série d'évaluations annuelles des méthodes et outils pour la RI. TREC est un projet international initié au tout début des années quatre-vingt dix par le NIST[5] dans le but de proposer des moyens homogènes d'évaluation de systèmes documentaires sur des collections de documents conséquentes. Il est aujourd'hui co-sponsorisé par le NIST, l'ITL[6] et l'IARPA[7] (ex-DARPA[8]).

Les conditions de participation sont les suivantes : le NIST diffuse courant décembre[9] un appel à participation qui explique dans les grandes lignes les objectifs et le déroulement du projet pour l'année à venir. Les demandes de participation doivent être déposées en janvier, aussi bien pour les anciens participants que pour les nouveaux. Les demandes d'intégration à TREC sont étudiées par un comité de programme qui se prononce en février. La participation à la conférence annuelle elle-même est soumise à l'envoi au NIST de résultats.

Les pistes explorées évoluent au fil des années, elles reflètent les intérêts du moment : en 1995, les deux tâches principales étaient celle de routage et celle de recherche ad hoc ; les tâches secondaires étaient la recherche multilingue, le filtrage, la fusion de bases de données, la tâche interactive et la tâche de « confusion » (recherche d'erreur). En 2008, les tâches proposées ont été : la tâche Blog, la tâche Légale, la tâche Recherche d'Information Chimique, la tâche Entreprise, la tâche de Réinjection de pertinence, la tâche Entité, la tâche Web et la tâche « million de requêtes ». Certaines tâches ne sont plus au goût du jour, il en est ainsi pour la tâche Nouveauté, la tâche Question/Réponse, la tâche Spam, …

Parmi les tâches proposées à TREC, celle qui nous concerne tout particulièrement est la tâche ad hoc car nous proposons un modèle de RI ad hoc basé sur les graphes – chapitre 4. L'objectif de cette tâche est d'évaluer les performances des systèmes de recherche d'informations recherchant la réponse à des requêtes dans un fond de documents textuels statiques. Cette tâche se rapproche de la façon dont les chercheurs utilisent les bibliothèques où la collection est connue mais les questions susceptibles d'être posées ne le sont pas. La piste ad hoc est le premier mécanisme sur lequel se sont construits les corpus de documents de TREC. La tâche ad hoc de TREC, surtout lorsque la requête est courte (2 ou 3 mots), ressemble beaucoup à la recherche d'informations sur le Web. Mais les documents ne sont précisément pas des pages Web.

Le corpus utilisé dans nos expérimentations est celui de TREC-7 1998. Il comprend trois sources : le *Financial Times* 1992, 1993 et 1994, *Foreign Broadcast Information Service 1996*, et *LA Times* 1989 et 1990. Les requêtes utilisées dans le chapitre 5 sont celles numérotées de 351 à 400.

---

[5] National Institute of Standard and Technology
[6] Information Technology Laboratory Retrieval group (of the Information Acess Division)
[7] Intelligence Advanced Research Projects Activity
[8] Defense Advanced Research Projects Agency
[9] Informations issue de http://trec.nist.gov/faq.html

4.4 Autres campagnes d'évaluation

D'autres campagnes d'évaluation ont vu le jour :

- Les campagnes NTCIR(NII-NACSIS Test Collection for IR Systems). Apparus en 1999, Les ateliers d'évaluation du NTCIR (Research Center for Information Resources) sont conçus dans le but d'améliorer tous les domaines de l'accès à l'information y compris la recherche d'information, la production de résumés, l'extraction terminologique, etc. La collection test utilisée comprend des textes publiés en 1998 et 1999, en chinois traditionnel, en coréen, en japonais et en anglais.
- Les campagnes CLEF (Cross-Language Evaluation Forum) : Projet européen d'évaluation des SRI qu'ils soient monolingue ou multilingue de langue européenne. Ce projet a vu le jour en l'an 2000. CLEF propose des tâches principales (les tâches monolingue, bilingue, multilingue et de recherche dans un domaine spécifique) et des tâches additionnelles dont le but est d'identifier les nouveaux besoins et les nouvelles exigences afin d'acquérir de nouvelles méthodes pour l'évaluation des SRI monolingues ou multilingues.
- Les campagnes FIRE (Forum for Information Retrieval Evaluation). Apparues en 2008 ; ces campagnes ont pour but de fournir des collections de grande échelle pour l'évaluation de la recherche d'information en langues indiennes (Bengali, Hindi, Marathi, Tamil).

Les campagnes d'évaluation sont un élément incontournable de la RI, elles fournissent des outils d'évaluation des systèmes, elles permettent la comparaison de systèmes et elles définissent des cadres d'études utiles aux chercheurs.

Conclusion du chapitre Recherche d'information

Nous avons présenté dans ce chapitre les principales notions et concepts de la recherche d'information. Nous avons développé les principales étapes d'un processus de recherche d'information que sont la représentation ou indexation de l'information et la comparaison de l'information et du besoin en information.

Les corpus Cranfield et CISI ont été retenus pour notre étude. Comme nous le verrons dans les chapitres suivants, la complexité de notre algorithme impose l'utilisation de corpus restreints (moins de 5000 documents), mais dans un souci de réalisme, nous avons également réfléchi à l'usage de notre algorithme sur un corpus de grande taille : celui de la tâche TREC ad hoc 1998.

Pour nos tests nous comparerons notre mesure, basée sur une similarité indirecte, à la mesure Cosinus (qui représente la similarité directe) d'une part, et à la mesure Okapi BM25 d'autre part qui est une référence dans le domaine de la RI.

Pour évaluer et comparer notre méthode, nous nous baserons sur les différentes mesures de RI que nous avons présentées : la MAP, la mesure F moyenne, La meilleure mesure F, la R-précision, la courbe de précision à 11 points de rappel et les précisions à 5, 10, 30, 100 documents retrouvés que nous noterons p@5, p@10, p@30, p@100.

Le travail développé dans cette thèse concerne principalement deux aspects de la RI : les fonctions de tri et les structures de données. Le but des fonctions de tri est d'estimer la pertinence des documents par rapport à une requête donnée de manière à ce que le système puisse ordonner les documents en fonction de leur degré de pertinence. Le principal problème de l'aspect structure de données est de trouver une forme qui permette à la fois de stocker des données et d'utiliser une fonction de tri efficiente.

Le modèle que nous présenterons dans la quatrième partie est un modèle algébrique utilisant la représentation matricielle du modèle vectoriel mais surtout la structure de graphe bipartite qui répond à nos attentes de représentation. C'est par exploitation de cette structure que notre approche extrait de l'information en vue d'améliorer les résultats en termes de précision. Notre modèle, pour exploiter les relations entre documents, entre documents et termes, et entre termes se base sur une représentation sous forme de graphe, graphe associé à la matrice document-terme. Dans la mesure où notre approche exploite la représentation du corpus sous forme de graphe, nous allons présenter dans la partie suivante, les définitions et propriétés nécessaires à la compréhension de notre modèle.

Chapitre 3 : Graphe

La théorie des graphes débute avec les travaux d'Euler [Euler, 1736]. L'histoire veut que Léonard Euler en visite dans la ville de Königsberg en Prusse orientale (aujourd'hui elle se nomme Kaliningrad et appartient à la fédération de Russie) ait tenté de répondre à un problème : Est-il possible de trouver un circuit qui emprunte une seule fois chacun des sept ponts disposés comme suit ?

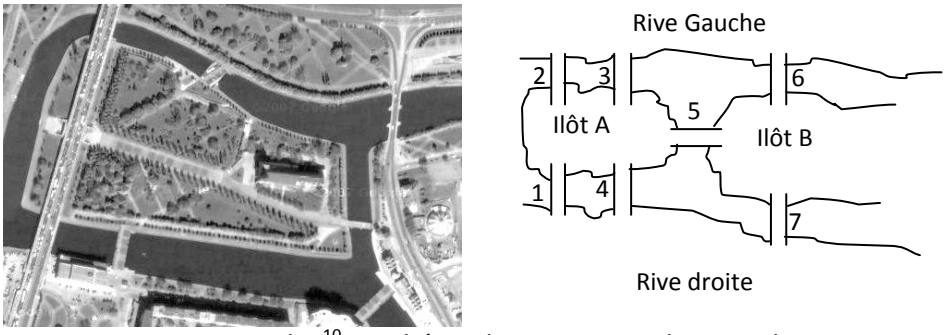


Figure 16 : plan[10] et schéma des sept ponts de Königsberg

Euler choisit, pour étudier ce problème, de le représenter sous forme de graphe, c'est-à-dire de diagramme où il relie aux quatre lieux possibles les sept ponts de la façon suivante :

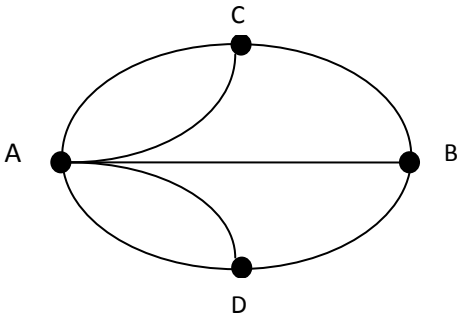


Figure 17 : représentation en graphe des sept ponts de Königsberg

Cette modélisation permet de traduire le problème initial en problème autour des propriétés du graphe : « peut-on circuler sur le graphe à partir d'un des points en empruntant une fois et une seule chaque lien ». Euler démontra que ce problème n'a pas de solution.

De la même façon, de nombreuses méthodes, propriétés, procédures ont été imaginées ou trouvées à partir d'un dessin ou d'un schéma. C'est un des principes fondateurs de la théorie des graphes. En effet, un graphe permet de représenter la structure et les connexions d'un ensemble complexe en exprimant les relations entre ses éléments. Les graphes permettent de modéliser une grande variété de problèmes en se ramenant à l'étude de sommets et de liens.

[10] L'image est un photomontage issu d'une vue satellite de Kaliningrad issue de Google Map dans lequel nous avons ajouté 3 ponts qui ne sont plus présents de nos jours.

Après les travaux d'Euler, Kirchhoff développa, au milieu du 19ème siècle, la théorie des arbres pour l'appliquer à l'analyse de circuits électriques. La théorie des graphes a également été utilisée en chimie pour modéliser les molécules en considérant que les atomes sont des sommets et les liens entre atomes sont des arêtes.

A partir de 1950, la théorie des graphes a connu un développement intense en devenant une branche à part entière des mathématiques grâce aux travaux de chercheurs tels que König, Menger, Cayley, Kuhn, Ford, Fulkerson, Roy et Erdös. Son essor en France est dû aux travaux de Claude Berge qui initia le passage à l'aire moderne de cette théorie en rassemblant les travaux épars dans la littérature dans son ouvrage « Théorie des graphes et applications » [Berge, 1958]. Comme nous le verrons, la théorie des graphes présente des liens évidents avec l'algèbre linéaire, la topologie, la théorie des nombres et les statistiques.

De nombreux domaines ont aujourd'hui recours aux graphes dans le but de traiter des problèmes rencontrés dans le monde réel. Parmi ces domaines, on trouve notamment ceux traitant des réseaux. Les réseaux peuvent être de diverses natures:

- Biologiques (chaînes de protéines, réseaux de gènes, topologie du cerveau,…),
- Technologiques (réseaux routiers, réseaux de télécommunication, Internet,…),
- Sociaux (réseaux d'affiliation, réseaux d'échange internationaux,…),
- De mots (réseaux de cooccurrence, réseaux sémantiques,…),
- …

Il s'avère que les différents domaines dans lesquels les réseaux apparaissent ont souvent un besoin commun d'analyser le réseau soit dans le but de le mesurer soit dans le but de comprendre les différents phénomènes qui se déroulent en son sein. Ces phénomènes peuvent être de différentes natures : diffusion ou routage d'information, diffusion de virus, résistance aux pannes, gestion des congestions, étude des comportements sociaux ou biologiques, … Ainsi les mathématiciens ont d'abord commencé à répondre à des problèmes classiques tels que les problèmes de flots [Ahuja et al., 1993], de connectivité, de couplage [Lovász, 1986], de parcours eulérien (qui traverse chaque arête) [Fleischner, 1990], de parcours hamiltonien (qui traverse chaque sommet) [Lawler et al., 1987], de coloration[Jensen et al., 1995] , …

La rencontre de la théorie des graphes et de l'informatique a permis le développement d'algorithmes associés pour le traitement des problèmes rencontrés. Ces algorithmes sont aujourd'hui appliqués dans des domaines nombreux et variés : ainsi les algorithmes de plus court chemin trouvent des applications en géo localisation ou dans l'agencement des tâches lors de la réalisation d'un projet, les algorithmes de parcours hamiltonien trouvent des applications commerciales type gestion de carnet de route, les algorithmes de coloration s'appliquent en cartographie, les algorithmes de codage type Huffman exploitant les arbres sont aujourd'hui appliqués dans les domaines où la compression intervient (donnée binaire avec le zip, image avec le jpeg, son avec le mp3,…).

En plus des domaines où les réseaux apparaissent, la théorie des graphes trouve des applications en intelligence artificielle, théorie des jeux, théorie de la décision, science du langage, représentation des connaissances ainsi qu'en RI.

Notre modèle de RI – cf. chapitre 4 - utilise un graphe bipartite pour représenter les collections étudiées. Dans ce graphe un nœud *document* est relié à un nœud *terme* si le terme apparaît dans le document. Les liens entre nœuds documents et nœuds termes sont valués par le poids du terme dans le document.

Dans ce chapitre, la première partie permet d'introduire les définitions (sommets, nœuds, liens, arcs, …), et les notions (chemin, distance, voisinage…). Elle permet aussi de présenter différents types de graphes (graphe simple, graphe bipartite). Dans une seconde partie nous introduirons différents critères et méthodes utilisés couramment pour caractériser les graphes (diamètre, coefficient de regroupement, densité…).

Enfin, dans la troisième partie, nous présenterons un état de l'art sur l'usage qu'il est fait des graphes en RI, dans le cadre général de la recherche d'information sur le contenu et dans le cadre plus spécifique du Web, en insistant sur l'aspect de l'exploitation de l'information contenue dans la structure d'hyperliens.

1. Definitions et notations

1.1 Graphe et sous-graphe

Un graphe *G(V,E)* est défini par la donnée d'un ensemble fini *V* dit *ensemble de sommets* qu'on note aussi *V(G)*, et d'un sous ensemble *E* du produit cartésien $V \times V = \{(x, y)/x \in V, y \in V\}$. *E* est appelé *ensemble des arêtes* aussi noté *E(G)*.

Un *sommet* peut-être également nommé *nœud* et une *arête* peut également être nommée *arc* ou *lien*.

Si une arête *e* relie deux sommets *x* et *y*, on dit que l'arête est incidente aux sommets *x* et *y*. Les sommets *x* et *y* sont alors *adjacents*, ou *incidents* à *e*. Un graphe est dit *complet* si toute paire de sommets de G est une arête.

Un *cycle* une suite d'arêtes consécutives (*chaîne*) dont les deux sommets extrémités sont identiques. Si la chaîne est *élémentaire*, c'est-à-dire qu'elle ne passe pas deux fois par un même sommet, alors on parle de *cycle élémentaire*.

Le *nombre de sommets* de G est appelé l'*ordre* de G est $|V(G)|$ et est également noté $|G|$.

Le sous-graphe d'un graphe *G(V,E)* est obtenu en enlevant un ou plusieurs sommets de *G* ainsi que toutes les arêtes incidentes aux nœuds supprimés.

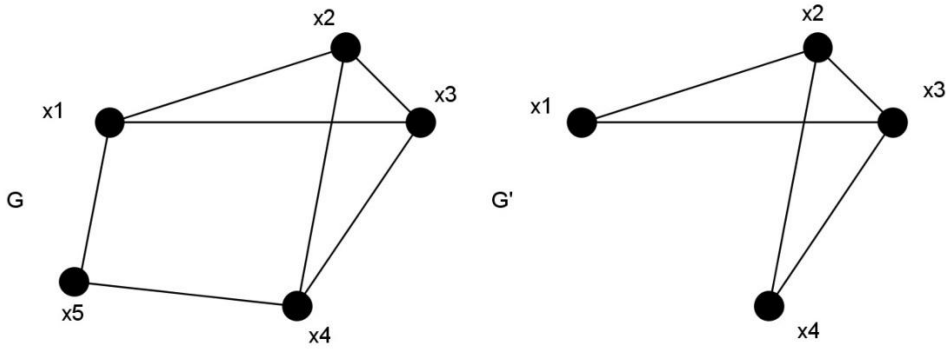
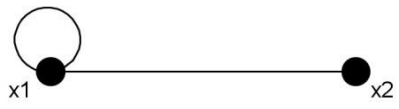


Figure 18 : sous-graphe G' d'un graphe G

G' est le sous graphe engendré par $\{x_1, x_2, x_3, x_4\}$.

1.1 Graphe avec boucle

Il est possible d'enrichir la définition précédente pour prendre en compte l'idée de boucle sur un sommet, c'est-à-dire le fait que l'arc relie le sommet à lui-même :



G(V=$\{x_1, x_2\}$; E=$\{(x_1, x_2)$;$(x_1,x_1)\}$)

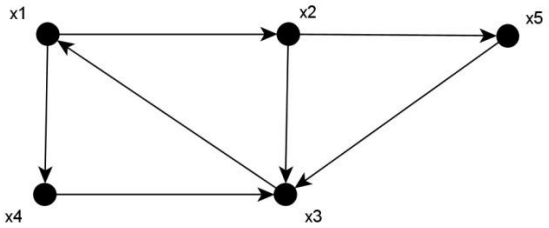
Figure 19 : graphe *G* formé de 2 sommets et deux arcs dont une boucle

L'arc de G (x_1, x_1) est appelé une *boucle*.
Remarque : Il existe d'autres définitions de graphe qui ne tiennent pas compte de la notion de boucle.

1.2 Graphe orienté

Un graphe peut être enrichi par des informations topologiques additionnelles. En particulier, si l'on fait une distinction entre (*x,y*) et (*y,x*) pour *x* et *y* dans *V*. Si le sommet *x* a un lien vers le sommet *y* et que cela n'implique pas que *y* en ait un vers *x*, alors le graphe est dit *orienté*. Les graphes orientés sont parfois appelés des *digraphes*.

L'orientation d'une arête se traduit par une flèche indiquant un sens de parcours comme sur l'exemple suivant :

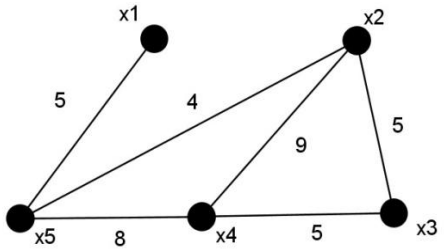


G(V={x_1, x_2,x_3,x_4,x_5} ; E={(x_1, x_2);(x_1,x_3);(x_2,x_5);(x_5,x_4);(x_4,x_1);(x_2,x_4);(x_3,x_4)})
Figure 20 : graphe orienté

Notre méthode – cf. chapitre 4 - est inspirée d'une méthode de comparaison de nœuds prévue à l'origine pour la comparaison de nœuds dans un graphe orienté.

1.3 Graphe valué

Si on associe une valeur *w(x,y)* (par exemple un poids) à chaque arête *(x,y)* de E, alors le graphe est dit *valué* et est noté *G(V,E,w).*



G(V={x_1,x_2,x_3,x_4,x_5} ;E={(x_1,x_5);(x_2,x_3);(x_2,x_4);(x_2,x_5);(x_3,x_4);(x_4,x_5)},w={w(x_1,x_5)=4,w(x_2,x_3)=5,w(x_2,x_4)=9, w(x_2,x_5)=4,w(x_3,x_4)=5,w(x_4,x_5)=8})

Figure 21 : graphe valué

Notre méthode – cf. chapitre 4 - utilise un graphe où les arcs entre nœuds termes et nœuds documents sont valués par les poids des termes apparaissant dans les documents.

1.4 Connexité

Un graphe *G(V,E)* est connexe s'il existe pour chaque paire de sommet une chaîne (une suite d'arcs) reliant chacun des deux sommets. Un graphe non connexe se décompose en composantes connexes :

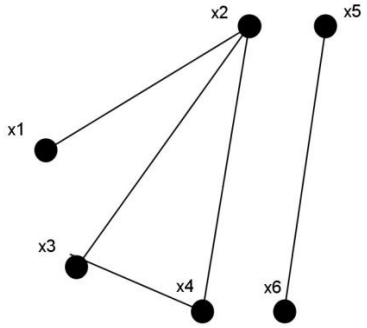


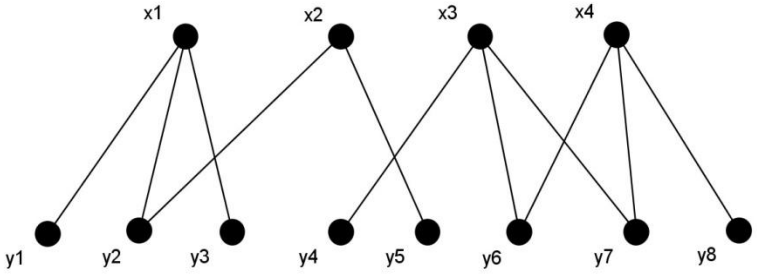
Figure 22 : graphe non connexe

G(V={x_1, x_2, x_3, x_4, x_5, x_6},E={(x_1,x_2) ,(x_2,x_3) ,(x_3,x_4) ,(x_4,x_2) ,(x_5,x_6)}) est non connexe,{x_1, x_2, x_3, x_4} et {x_5, x_6} sont les deux composantes connexes de *G*.

Notre méthode – cf. chapitre 4 - permet de trouver dans le réseau document-terme que constitue la collection la composante connexe du réseau comprenant le nœud représentant la requête. Les documents n'appartenant pas à cette composante sont considérés comme non similaires à la requête.

1.5 Graphe bipartite

Si l'ensemble des nœuds V est séparé en deux sous-ensembles ⊤ et ⊥, avec $V = \top \cup \perp$ et ⊤∩⊥= ∅, tel qu'il n'y a pas de liens entre les nœuds d'un même ensemble alors le graphe est dit *bipartite* et est noté $G(\top, \perp, E)$.

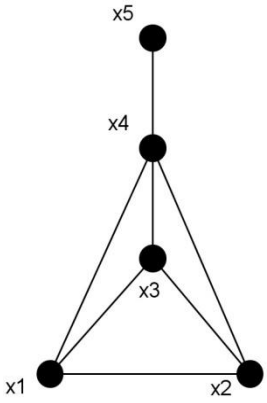


G(⊤*={x_1, x_2,x_3,x_4};*⊥*={y_1,y_2,y_3,y_4,y_5,y_6,y_7,y_8}; E={(x_1, y_1) ;(x_1,y_2) ;(x_1,y_3) ;(x_2,y_2) ;(x_2,y_5) ;(x_3,y_4) ;(x_3,y_6) ;(x_3,y_7) ;(x_4,y_6) ;(x_4,y_7) ;(x_4,y_8)})*

Figure 23: graphe bipartite

1.6 Matrice d'adjacence

On peut représenter un graphe G par une *matrice d'adjacences*. Une matrice d'adjacence est une matrice carrée d'ordre *n (n=|V|)*, dont les lignes et les colonnes représentent les sommets du graphe. Une valeur non nulle dans une case (*i,j*) de la matrice indique que le sommet *i* est adjacent au sommet *j*.



$$M = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Figure 24 : graphe G et sa matrice d'adjacences

La matrice de l'exemple ci-dessus représente le graphe *G*, elle a plusieurs caractéristiques : les valeurs de sa diagonale sont nulles (une valeur non nulle indiquerait qu'il existe une boucle), elle est symétrique $M_{ij}=M_{ji}$.

2. Caractéristiques des graphes

Les deux données de départ qui permettent d'analyser un graphe sont :

- Le nombre de nœuds |V(G)| que l'on notera *n,*
- Le nombre de liens |E(G)| que l'on notera *m.*

2.1 Densité d'un graphe

La densité d'un graphe simple sans boucle est définie par :

$$\delta = \frac{2m}{n\times(n-1)} \qquad (1)$$

C'est le nombre de liens existants divisé par le nombre de liens possibles. La densité représente la probabilité que deux nœuds distincts pris au hasard soient liés.

Si la densité est égale à 1, alors tous les nœuds du graphe sont reliés ; il s'agit d'un *graphe complet*.

2.2 Voisinage : degré, degré moyen, coefficient de regroupement

Le *voisinage* d'un nœud *x*, *N(x)* est l'ensemble des nœuds auxquels il est relié.

$$N(x) = \{y \in V, (x,y) \in E\} \qquad (2)$$

Le *degré* d'un nœud *x* correspond au nombre de ses voisins :

$$d^{\circ}(x) = |N(x)| \qquad (3)$$

Dans le cas d'un graphe orienté, les arêtes entrantes et les arêtes sortantes d'un nœud sont différenciées. Le degré d'un nœud est alors la somme du nombre d'arêtes entrantes en *x* et du nombre d'arêtes sortantes de *x*. Ces deux nombres sont respectivement appelés le *demi-degré entrant de x* et le *demi-degré sortant de x*.

Un nœud *y* est un *successeur* du noeud *x* s'il existe une arête ayant son extrémité initiale en *x* et son extrémité terminale en *y*. L'ensemble des successeurs de *x* dans *G* se note : $R_{+}G(x)$. De même,

on dit que *y* est un *prédécesseur* de *x* s'il existe une arête de la forme (*y, x*). L'ensemble des prédécesseurs de *x* dans *G* se note : $R\_G(x)$

L'union des successeurs et des prédécesseurs d'un nœud *x* forme le voisinage de *x* :

$$N(x) = R_+G(x) \cup R\_G(x) \quad (4)$$

Le *degré moyen* d'un graphe *G* est la moyenne des degrés de tous les nœuds du graphe :

$$d°_{moy}(G) = \frac{1}{n}\sum_{x\in V} d°(x) \quad (5)$$

Le degré moyen correspond à la moyenne du nombre de voisins des nœuds.
Le degré moyen est en rapport avec la taille du graphe et sa densité, ainsi on a :

$$d°_{moy}(G) = \delta(n-1) \quad (6)$$

(6) découle de (5) et de (1)

La distribution des degrés d'un graphe est la proportion p_k pour chaque entier *k* de nœuds ayant *k* pour degré.

$$p_k = \frac{1}{n} \times |\{x \in V, d°(x) = k\}| \quad (7)$$

On cherche à calculer combien de nœuds sont de degré 1, 2, … jusqu'à *n*. Cette donnée permet d'étudier la répartition des degrés.

Le coefficient de regroupement *(clustering*) d'un nœud *x* parfois nommé *cliquicité* correspond au nombre de voisins de *x* divisé par le nombre possible de liens :

$$cc(x) = \frac{|N(x)\times N(x)\cap E|}{\frac{1}{2}(d°(x)\times(d°(x)-1))} \quad (8)$$

Cette valeur correspond à la densité du sous-graphe induit par N(*x)*. C'est la probabilité que deux voisins de *x* pris au hasard soient reliés.

Le coefficient de regroupement d'un graphe *G(V,E)* correspond à la moyenne des coefficients de regroupement des nœuds de *G* :

$$cc(G) = \frac{1}{n}\sum_{x\in V} cc(x) \quad (9)$$

Il existe une autre définition [Bornholdt et al., 2003] qui consiste à calculer la probabilité que deux nœuds soient reliés sachant qu'ils ont un voisin commun.

$$cc_{bis}(x) = \frac{3\times Nombre\ de\ triangles\ dans\ G}{Nombre\ de\ triplets\ connexes\ dans\ G} \quad (10)$$

Le nombre de triplets connexes de *G* correspond au nombre de sommets connectés à une paire non ordonnée de sommets. Les deux formules utilisent l'information du nombre de liens dans le voisinage et celle du nombre total de liens possibles, la première formule calcule la moyenne de ces rapports, la deuxième formule calcule le rapport de ces moyennes [Iamnitchi et al., 2004].

2.3 Chemins et distances : distance moyenne, diamètre

Un *chemin* entre deux nœuds *x* et *y* d'un graphe *G* est une suite d'arcs qui connectent ces deux nœuds. La *longueur du chemin* est le nombre d'arcs de cette suite. La longueur d'un plus court chemin entre *x* et *y* est appelée *distance*.

La distance entre deux nœuds *x* et *y*, notée *d(x,y)* correspond à la longueur (nombre d'arcs) de la plus petite chaîne liant *x* à *y*.

La *distance moyenne* d'un graphe *G (V,E)* est la moyenne des distances pour toutes les paires de nœuds de *G*:

$$\frac{2}{n\times(n-1)}\sum_{\{x,y\}\subset V} d(x,y) \qquad (11)$$

Le *diamètre* d'un graphe est la distance maximale, c'est-à-dire la distance entre les deux nœuds les plus éloignés :

$$\max_{(x,y)\in V^2} d(x,y) \qquad (12)$$

Le diamètre donne une information importante qui aide à caractériser le graphe étudié en indiquant la distance (le nombre de sauts) entre les deux nœuds les plus éloignés. Si cette valeur est faible (infèrieure à 8), alors le graphe étudié peut être un graphe « petit monde » car la distance qui sépare deux sommets quelconque est faible.

2.4 Petits mondes

La notion de petits mondes a été introduite par Milgram dans [Milgram et al., 1969]. L'expèrience a consisté à demander à cinquante personnes de la ville d'Omaha dans le Nebraska de faire suivre une lettre jusqu'à un agent de change, vivant à une adresse fournie, dans la ville de Sharon dans le Massachusetts. Les participants pouvaient seulement passer les lettres, de main à main, à des connaissances personnelles qu'ils pensaient être capables d'atteindre l'objectif, directement ou via leurs connaissances. Milgram décrit le fait que les lettres arrivées à destination étaient passées par six personnes en moyenne et en conclut que deux citoyens américains pris au hasard sont à une « distance » de six poignées de main l'un de l'autre. Bien que seulement 5 % des lettres soient arrivées à destination, cette expèrience est considérée comme le point de départ de l'étude des petits mondes.

Les petits mondes sont caractérisés par une distance courte entre les individus du réseau et par le fait que deux individus qui possèdent un voisin commun ont une forte probabilité d'être adjacents.

Un graphe dont le diamètre serait faible et dont le coefficient de reqroupement est grand est un petit monde.

Il est aujourd'hui admis que bon nombre de réseaux du monde réel sont des réseaux petits monde, parmi eux on trouve :

- Réseaux de réaction chimique [Alon et al., 1999],
- Réseaux neuronaux [Watts et al., 1998],
- Réseaux biologiques [Bornholdt, 2004]
- Réseaux de chaîne alimentaire [Montoya et al., 2002],
- Réseaux de collaboration scientifique [Van Raan, 1990],
- Réseaux de citations des papiers scientifiques [Seglen, 1992],
- Réseaux de collaboration des acteurs [Watts et al., 1998],
- Réseaux sémantiques ou lexicaux [Steyvers et al., 2005][Gaume et al..2006],
- Réseaux sociaux [Bearman et al., 2004][Boulet, 2009],
- …

Comme nous le verrons dans le cinquième chapitre, les collections documentaires étudiées sous la forme de réseaux de documents présentent des analogies avec les petits mondes.

2.5 Graphe aléatoire

Les méthodes d'analyse de réseaux en mathématiques se basent sur l'idée de récupérer les attributs de base d'un graphe à étudier souvent issu du monde réel et d'utiliser ces attributs comme paramètres d'un graphe aléatoire afin d'étudier les similitudes et les différences entre le graphe d'origine et le graphe aléatoire calculé.

Un graphe aléatoire d'Erdös-Rényi de paramètres $n \in \mathbb{N}$ et $p \in [0,1]$, noté *G(n,p)* est un graphe à *n* sommets où chaque arête existe avec la probabilité *p*.[Erdös et al., 1959]. Pour modéliser des graphes de grande échelle dont la distribution des degrés suit une loi de puissance, d'autres modèles sont utilisés [Aiello et al., 2000][Kumar et al., 2000]. D'autres modèles permettent de modéliser des réseaux type petits mondes [Watts et al., 1998] ou des réseaux d'interaction sous forme de graphe bipartite [Guillaume et al., 2004].

2.6 Le cas particulier des graphes bipartites

Un graphe bipartite présente deux particularités par rapport à un graphe classique :

Ses sommets appartiennent à deux ensembles distincts et il n'existe pas de lien d'un sommet d'un ensemble vers un sommet du même ensemble. Cette différence présente un réel intérêt dans la modélisation de nombreux cas pratiques. En effet les graphes bipartites sont utilisés dans différents domaines par exemple les réseaux pair à pair où des nœuds *individu* sont reliés aux nœuds des fichiers offerts ou demandés, les graphes d'occurrences de mots où les mots sont reliés aux phrases dans lesquels ils apparaissent. Les notions utilisées dans l'analyse des graphes classiques sont portables au cas bipartite comme le nombre de sommets et d'arêtes, ou la densité du graphe. D'autres en revanche doivent être adaptées comme par exemple la densité locale d'un nœud du fait que dans le cas bipartite, un nœud donné ne possède que liens des indirects vers les nœuds de même type.

L'exemple suivant introduit un genre de problème particulier modélisable par un graphe bipartite.
Les travaux dont il est question aux chapitres 4 et 5 traitent de graphes similaires - bien que plus grands en termes de nombre de sommets et de nombre de liens -, utilisés dans un autre cadre, celui de la RI.

Soit

$G(\top$ ={A,B,C,D} ;$\perp$={o_1,o_2,o_3,o_4,o_5,o_6,o_7,o_8};E={(A,o_1);(A,o_3);(B,o_2);(B,o_4);(B,o_5);(C,o_4);(C,o_6);(C,o_6) ;(C,o_7);(D,o_6);(D,o_7);(D,o_8)})

Un graphe bipartite modélisant des personnes (A, B, C, D) et des objets (o_1,o_2,…,o_8).

Un lien entre une personne et un objet indique que la personne achète l'objet ou réciproquement que l'objet est acheté par la personne.

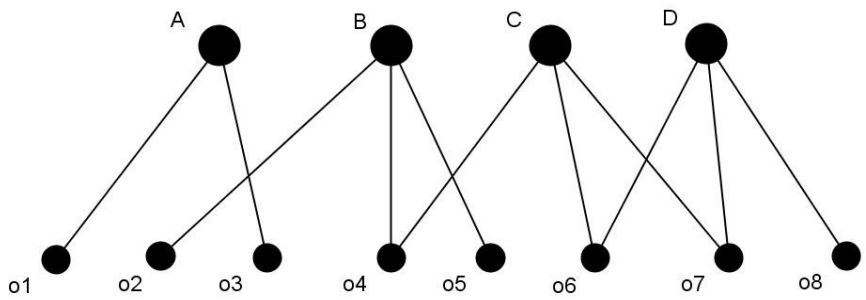


Figure 25 : représentation du graphe bipartite *G*

Pour étudier un tel graphe, on peut utiliser la *projection*. C'est le fait de ne considérer qu'une des deux classes de nœuds et à relier les nœuds considérés entre eux s'ils ont un voisin commun dans le graphe bipartite.

Ainsi, le graphe précédent a deux projetés : le T-projeté et le ⊥-projeté.

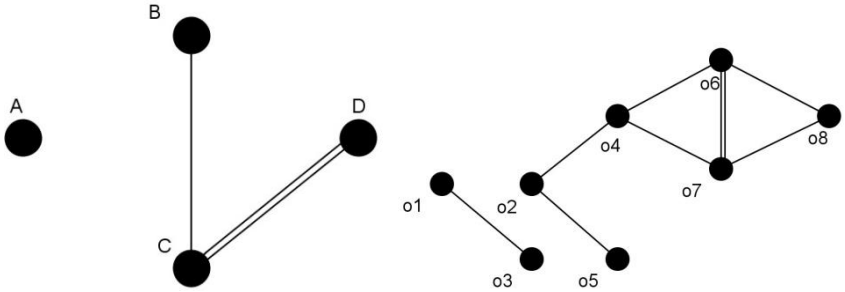


Figure 26 : T-projeté & ⊥-projeté

Le T-projeté permet d'étudier les personnes, ici celles achetant les mêmes objets sont liées. Le ⊥-projeté permet d'étudier les objets, ici ceux étant achetés par les mêmes personnes sont liés. Le choix de telle ou telle projection est souvent motivé par le point de vue sur le problème. On peut rechercher les caractéristiques vues précédemment de chacun des graphes projetés : répartition des degrés, coefficients de regroupement,…. La distance entre nœuds dans un graphe projeté correspond dans le bipartite d'origine à la *distance-2-voisin* du nœud, c'est-à-dire au nombre d'arcs séparant deux nœuds de même type.

Néanmoins, une grande quantité d'information est perdue lors de la projection [Latapy et al., 2008]. Il peut être intéressant d'utiliser des notions adaptées au bipartisme ; Matthieu Latapy dans [Latapy, 2007] propose de définir la redondance d'un nœud comme la proportion des paires de voisins du nœud à avoir un autre voisin en commun dans le bipartite. La redondance d'un nœud est définie comme suit :

$$rc(v) = \frac{|\{\{x,y\} \subseteq N(v), \exists v' \neq v, (v',x) \in E \text{ et } (v',y) \in E\}|}{\frac{|N(v)|(|N(v)|-1)}{2}} \qquad (13)$$

Ce coefficient représente la fraction des nœuds au voisinage d'un nœud *v* donné ayant des liens avec un autre nœud que *v*. Il est possible de dériver de cette définition le coefficient de redondance des nœuds de type 1 *rc(*⊤*),* des nœuds de type 2 *rc(*⊥*),* du graphe *rc(G)* et procéder à l'étude de la distribution et des corrélations.

Les graphes réels constituent un champ d'étude [Mélançon , 2006][Chrisment et al., 2006] [Latapy et al., 2008][Dousset, 2009] que les critères développés précédemment permettent de caractériser. Nous allons maintenant regarder de quelle manière les graphes sont utilisés en RI en se focalisant sur les aspects recherche sur le contenu et recherche exploitant les liens structurels.

3. Graphe et recherche d'information

Tout d'abord, nous verrons, au travers des réseaux sémantiques comment est utilisé le pouvoir représentatif des graphes dans le cadre de la représentation du contenu des documents et de la recherche sur le contenu.

Ensuite nous verrons les approches de RI basés sur des réseaux (réseaux de neurones et réseaux bayésiens) qui appartiennent également aux approches de recherche sur le contenu. Puis nous présenterons un modèle de recherche basé sur les graphes bipartites.

Enfin nous étudierons l'exploitation de la structure d'hyperliens qui est faite dans le cadre de la recherche d'information sur le Web.

3.1 Représentation des contenus des documents, recherche sur le contenu

3.1.1 Réseaux sémantiques

Les modèles sémantiques ont été introduits en RI pour tenir compte du contenu sémantique des documents et des requêtes, en intégrant des connaissances sur le sens des termes d'indexation et sur leur liaison sémantique. Il s'agit de réseaux dont les nœuds représentent des concepts, des objets ou des faits et les arcs représentent les relations entre ces concepts. Le but des réseaux sémantiques est de fournir une représentation souple des connaissances.

Quillian présente les réseaux sémantiques comme un mécanisme d'associations général pour encoder le sens des mots [Quillian, 1968]. Dans son modèle de la « mémoire humaine » fondé sur un réseau sémantique de mots construits à partir d'expériences en psycholinguistique où des individus sont soumis à des tâches terminologiques portant sur le sens des mots et où la mesure physique de leurs temps de réponse permet de définir des « distances sémantiques ». Ces recherches sont à la base de l'école sémantique lexicale qui a produit le réseau sémantique *Wordnet*[11] [Miller et al., 1990], très utilisé aujourd'hui en traitement de la langue.

Les réseaux sémantiques sont utilisés pour élaborer des représentations sémantiques dans les systèmes de traitement automatique du langage et les systèmes de recherche d'informations [Cornuejols, 2002]. Ils ont été appliqués avec succès dans le domaine de l'indexation de documents [Gaines et al., 1994], de l'éducation [Gaines et al., 1995] et dans le domaine de la recherche d'informations [Kheirbek, 1995][Martin, 1996]. Les réseaux sémantiques constituent le cœur des ontologies, il existe des SRI à base d'ontologies [Decker et al., 1999][Martin et al., 1999][Vallet, 2005] ainsi que des métamoteurs de recherche (Kartoo[12], MapStan[13]) qui présentent leurs résultats sous forme de réseaux sémantiques, calculés à partir des liens sémantiques entre les pages Web. Par exemple, Kartoo présente ses résultats sous forme de graphe où les nœuds sont les pages Web et les liens entre ces pages sont annotés d'un mot qui les lie sémantiquement.

Dans les modèles basés sur les réseaux sémantiques [Chen et al., 1990], les dépendances entre les termes sont codées via des liens typés, logiques et déterministes. La recherche des éléments pertinents pour une requête utilise des méthodes d'inférence logique.

Salton [Salton et al., 1983] distingue deux sortes de liens : ceux des réseaux logiques où les liens ont une valeur booléenne, ceux des graphes conceptuels où les liens sont flexibles.

[11] http://wordnet.princeton.edu/
[12] http://kartoo.com
[13] http://search.mapstan.net

3.1.1.1 Réseaux logiques

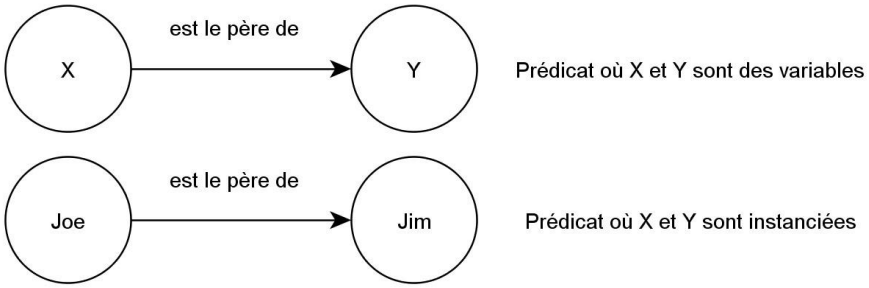


Figure 27 : exemple de réseau logique

Dans un réseau logique, un graphe est interprété comme un énoncé logique des propriétés des différentes entités. Les primitives de base d'un réseau logique sont les prédicats; des propositions peuvent être combinées en utilisant des opérateurs logiques. De nombreuses études ont été menées pour tenter de définir une logique pour la RI. Le premier à avoir tenté de définir un modèle logique pour la RI est Cooper [Cooper, 1971]. Depuis, plusieurs autres modèles ont été définis [Crestani et al. 2001][Chevallet, 2004]. Dans ces modèles, des hypothèses nécessaires à la modélisation de la pertinence sont posées : la RI est un processus formalisable (les requêtes et les documents peuvent être formalisés), il existe une relation de pertinence entre la requête et le document s'il existe une chaîne de déduction incertaine commençant au document et se terminant à la requête. Alors le calcul de pertinence consiste à prouver que la requête implique le document. Plusieurs formalismes peuvent être utilisés pour définir la notion d'implication : la logique de 1er ordre (les documents et la requête sont indexés par des termes, la requête est une formule appartenant au langage), la logique modale (les documents sont représentés par des mondes et les requêtes sont des formules qui peuvent être vérifiées dans les mondes).

Les modèles logiques permettent de mieux capter la nature inférentielle des activités de recherche d'information, et offrent un traitement de l'incertitude adéquate. Les modèles logiques ont été appliqués avec succès dans la RI appliquée au domaine médical [Chiaramella et al.1990], les résultats montrent un rappel et une précision élevés, résultats confirmés dans [Xu, 1990].

3.1.1.2 Graphes conceptuels

Un graphe conceptuel est un graphe orienté, fini et bipartite [Sowa, 1984][Sowa, 2000]. C'est un graphe dont les deux classes de sommets sont étiquetées respectivement par des noms de « concepts » et des noms de « relations conceptuelles » entre ces concepts.

Dans un graphe conceptuel un concept est formé de deux éléments : un type et un référent. Par exemple *President : B. H. Obama*, ici *Président* est le type et *Obama* le référent. Les concepts dans le modèle proposé par Sowa forment une hiérarchie sur laquelle on peut définir une relation d'ordre partiel <.

Par exemple : *B.H. Obama* < *Président* < *Homme*. Cela traduit le fait que *B.H. Obama* est un *Président*, et que le *Président* est un *Homme*.

Les relations conceptuelles définissent les liens et spécifient les rapports qui existent entre les concepts du graphe :

Figure 28 : la relation conceptuelle

De nombreux types de relations entre concepts ont été définis [Sowa, 1984] [Nogier, 1991]:

- AGT : agent (entité intervenant de façon active)
- PAT : patient (entité intervenant de façon passive)
- OBJ : objet (entité affectée)
- LOC : lieu
- DEST : destination (aboutissement qui peut être de nature spatiale ou abstraite)
- ORIG : origine (provenance spatiale ou abstraite)
- APP : appartenance
- POSS : possession
- …

Les quatre opérations fondamentales qui permettent de construire un nouveau graphe sont :

- La recopie d'un graphe,
- La restriction d'un concept (remplacer ce concept par une de ses spécialisations),
- La jointure de graphes (lier deux graphes par un concept commun),
- La simplification d'un graphe (éliminer des arêtes).

Bien que les relations conceptuelles soient uniquement binaires et que l'héritage multiple soit difficile à traiter, les graphes conceptuels constituent un système de représentation souple qui permet de se rapprocher de l'efficacité descriptive du langage naturel en prenant en compte de multiples contraintes linguistiques et philosophiques : les notions de dénotation, de substituts nominaux, d'appartenance à un ensemble, d'abstraction, de généralisation, de spécialisation, … Un autre intérêt de ce modèle vient du fait que des raisonnements peuvent être effectués sur les connaissances représentées. Ces raisonnements peuvent être vus comme des opérations de graphes. Les graphes conceptuels ont été utilisés avec succès en RI notamment dans le cadre du traitement du langage naturel (génération, analyse du discours, génération du texte), de l'ingénierie de connaissances (acquisition de connaissances, modèle sémantique de données, extraction de connaissances), de la génération automatique du texte et de l'indexation et la recherche d'images.

.

3.1.2 Réseaux bayésiens

Un réseau bayésien est un graphe de dépendance acyclique et orienté. Dans ce graphe, les nœuds représentent des variables de proposition ou des constantes et les arcs représentent les relations de dépendance entre les propositions. Si la proposition représentée par le nœud *q* dépend de la proposition représentée par le nœud *p* alors il existe un arc du nœud *p* vers le nœud *q*.

Le nœud *q* est associé à une table des probabilités conditionnelles. Cette table spécifie pour toutes les valeurs possibles de *p* et de *q* la force de l'implication calculée par la probabilité conditionnelle P (*q*|*p*). Si le nœud *q* a d'autres parents, la matrice du nœud *q* spécifie les dépendances pour tous les parents.

A partir d'un ensemble de probabilités préalables pour les racines des graphes, il est possible de calculer la probabilité ou degré de croyances [Grossman et al., 98] associé à tous les nœuds restants [Pearl, 88].

Les réseaux inférentiels bayésiens sont utilisés en RI [Turtle et al., 1990][Turtle et al., 1991] pour représenter les dépendances entre termes d'une part et entre terme et document d'autre part. Ils reprennent des éléments du modèle probabiliste et en particulier le calcul de probabilité conditionnelle. Les réseaux inférentiels permettent d'utiliser diverses sources d'évidence pour calculer les probabilités. Ils disposent de deux principaux mécanismes :

- La propagation descendante des probabilités,
- La révision ascendante de ces probabilités.

Le principal problème dans les réseaux inférentiels consiste à initialiser les dépendances entre éléments et à définir leurs probabilités.

Le modèle décrit dans [Savoy et al., 1991] a pour but de déterminer l'importance des termes dans l'expression du besoin d'information à partir des termes de la requête et de la dépendance de ces termes avec d'autres termes d'indexation. Les termes sont pondérés par leurs degrés de croyance. Plus la croyance d'un terme est élevée, plus grande est sa pertinence dans la représentation du besoin d'information.

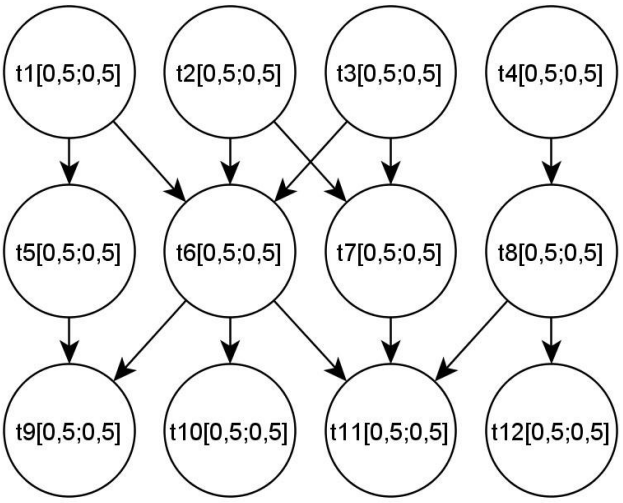


Figure 29 : réseau inférentiel de termes

Le réseau inférentiel de ce modèle comprend un ensemble de nœuds représentant des termes ou des concepts, chaque nœud a une valeur : soit pertinent, soit non pertinent par rapport à une requête utilisateur. A chaque nœud est associé un vecteur [t pertinent, t non pertinent] initialisé à [0.5, 0.5]. Etant donné le réseau ci-dessus, le but est de calculer les vecteurs de croyance associés aux nœuds afin de déterminer l'importance des termes dans l'expression du besoin d'information. Pour calculer la croyance d'un nœud *x*, l'évidence *e+* donnée par le sous arbre qui a *x* pour racine et l'évidence rattachée au reste du réseau *e-* sont prises en compte. Les termes de la requête ont un vecteur croyance tel que *t pertinent* > *t non pertinent*. Le degré de similarité de la requête est fonction de la fréquence d'apparition des termes dans le document et dans la collection et de la croyance associée à ces termes. Le poids d'un document correspond à la somme des poids de ces mots. Le poids d'un mot est fonction de sa fréquence d'apparition dans le document et de sa fréquence d'apparition dans la collection.

Ce modèle réutilise l'approche probabiliste classique en ajoutant la prise en compte de la dépendance entre les termes d'indexation.

Le modèle de Croft [Turtle et al., 1991] utilise également cette méthode. Il permet, par rapport au précédent, de calculer la probabilité associée au nœud sachant qu'un certain document a été observé ; c'est-à-dire la probabilité qu'un concept ou terme soit pertinent sachant qu'un certain document a été restitué. Ce modèle utilise une méthode basée sur une structure inférentielle préétablie, divisée en 6 parties comme suit :

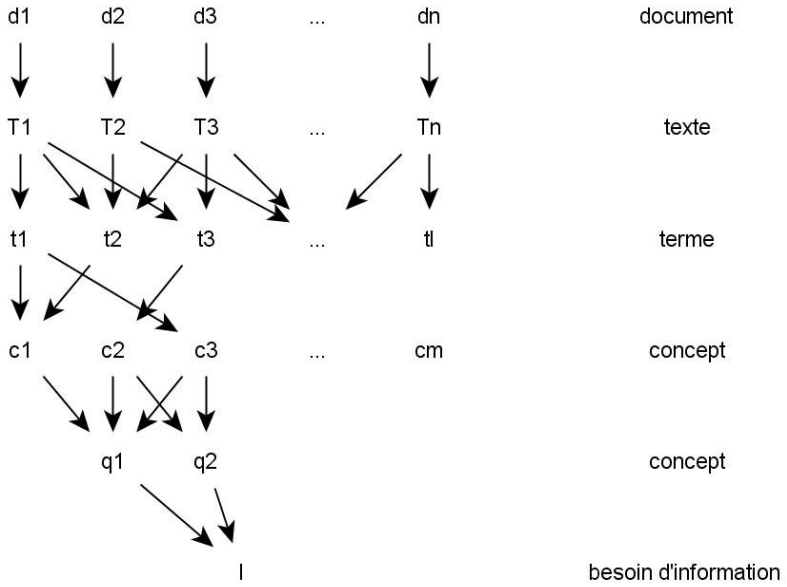


Figure 30 : exemple de réseau bayésien

La contrainte d'une telle structure est que les nœuds peuvent être connectés à des nœuds directement adjacents, mais la connexion entre nœuds d'une même couche n'est pas autorisée.

La figure comporte deux réseaux : le réseau documents (construit une fois pour toutes) et le réseau requête qui est différent à chaque requête et peut évoluer en cours d'utilisation.

Le réseau de documents représente un savoir statique sur la collection de documents. Il peut être modifié afin de caractériser un nouvel état de la collection lorsque des documents sont ajoutés ou retirés. Il comporte trois types de nœuds interconnectés :

- Les nœuds *documents* représentent les documents de la collection,
- Les nœuds *textes* correspondent à la représentation des documents,
- Les nœuds *représentation* d'un concept correspondent aux concepts décrivant un document.

Un document peut être caractérisé par plusieurs nœuds, les nœuds peuvent être partagés par plusieurs documents.

L'idée de séparation entre textes et documents réside dans le fait qu'un document peut être décrit par plusieurs textes ; de même un terme peut décrire plusieurs concepts. Notons que sur la figure précédente, un document est considéré comme décrit par un seul texte.

La probabilité initiale d'observer un document est donnée par $P(d_i)=1/N$. Chaque nœud *texte* contient la spécification de sa dépendance avec son père ($P(t_i/d_i)=1$) et chaque nœud *concept* contient la spécification des probabilités conditionnelles des dépendances avec ses parents.

Le réseau requête explique le besoin d'information de l'utilisateur au système. Il possède une seule feuille correspondant au fait qu'une information recherchée est rencontrée ; il possède plusieurs racines correspondant aux concepts d'interrogation.

Un nœud *requête* contient une spécification de sa dépendance avec les nœuds *concept* du réseau des documents.

Sur l'exemple de la figure ci-dessus c_3 dépend seulement de t_1, donc les connexions entre eux sont associées avec les probabilités $P(c_3|t_1)$, $P(\neg c_3|t_1)$, $P(c_3|\neg t_1)$ et $P(\neg c_3|\neg t_1)$. Comme c_1 dépend de t_1 et t_2 les probabilités de c_1 et de $\neg c_1$ doivent être conditionnées par les combinaisons de t_1 et t_2 (c'est-à-dire (t_1, t_2), $(t_1, \neg t_2)$, $(\neg t_1, t_2)$ et $(\neg t_1, \neg t_2)$).

Le principe de RI est le suivant : pour une requête donnée, on crée le réseau requête et on le rattache au réseau document. On peut alors calculer la probabilité qu'une information pertinente soit restituée sachant qu'un document donné a été observé ; cela pour chaque document de la base. Enfin, on peut ordonner les documents en fonction de la probabilité trouvée.

Les réseaux inférentiels permettent une recherche qui prend en compte la dépendance des termes entre eux ainsi que la dépendance avec les documents. Diverses sources d'évidence peuvent être combinées : les termes extraits automatiquement, les phrases, les paragraphes ou des descripteurs. Plusieurs langages d'interrogation sont possibles : langage naturel ou booléen. Les réseaux inférentiels se sont montrés performants dans le contexte de la RI notamment dans [Callan et al., 1995] où le système INQUERY obtient de bons résultats en termes de précision moyenne sur la collection ad hoc TREC-2.

Néanmoins, l'usage d'un tel réseau est peu pratique avec un grand nombre de *nœuds requête*. En effet, le nombre de probabilités à calculer pour un nœud croît exponentiellement avec son nombre de parents. Pour cette raison, chaque couche du réseau nécessite des approximations [Metzler et al., 2004].

Ces modèles offrent également des mécanismes de rétroaction de pertinence qui consistent à ajouter de nouveaux termes représentatifs (*r*) comme parents d'un nœud d'interrogation (*q*) et à ré-estimer le poids relatif des contributions des différents parents.

3.1.3 Réseaux de neurones`

La théorie connexionniste et son outil les réseaux de neurones artificiels sont inspirés des éléments connus du fonctionnement des neurones naturels (biologiques).

La connaissance est mémorisée au travers de cellules et de connexions inhibitrices ou excitatrices plus ou moins fortes entre ces cellules. La restitution d'une connaissance est induite par un stimulus externe qui active certaines cellules. Un processus de propagation d'activation à travers le réseau de cellules permet d'obtenir la réponse (état de certaines cellules) du réseau au stimulus initial.

Un réseau de neurones est composé de nœuds et de liens. A chaque nœud sont associées des entrées et sorties valuées. A chaque lien est associé un poids traduisant le degré d'interconnexion des nœuds qu'il relie. Le fonctionnement du réseau est basé sur la propagation des signaux d'activation depuis les entrées jusqu'aux sorties. Enfin, la connaissance peut évoluer par apprentissage. L'apprentissage consiste, à partir d'exemples ou de prototypes fournis au réseau, à modifier le réseau de telle sorte que, pour ces exemples, il réponde correctement. L'apprentissage est basé sur une modification des poids des connexions. Le pouvoir de généralisation du réseau de neurones lui permet alors de répondre même dans des cas non appris.

Des modèles de RI exploitant l'approche connexionniste ont été proposés pour prendre en compte la dépendance entre termes ainsi que pour introduire un aspect dynamique des représentations des documents.

L'application des principes de l'approche connexionniste au domaine de la RI est la suivante :

- Le stimulus externe correspond à la requête de l'utilisateur,
- Le problème à résoudre par le réseau est la recherche des documents susceptibles d'être pertinents par rapport à la requête,
- La réponse correspond aux documents jugés pertinents par le réseau,
- L'apprentissage doit permettre une optimisation du processus de recherche.

Les SRI utilisant l'approche connexionniste peuvent être divisés en deux catégories en fonction du type d'architecture de réseaux connexionnistes qu'ils utilisent. Certains modèles [Mozer, 1984] [Lin, 1989] sont basés sur des réseaux de type cartes auto organisatrices de Kohonen [Kaski et al., 1998]. Ils réalisent la classification des documents d'une collection par rapport à leur similarité. Cependant, ils permettent difficilement de prendre en compte l'évolution dans le temps de la collection de documents. D'autres modèles sont basés sur des réseaux à couches, généralement une couche mots-clés et une couche documents [Kwok, 1989][Belew, 1989]. Cette approche a été abordée par notre équipe [Mothe, 1994].

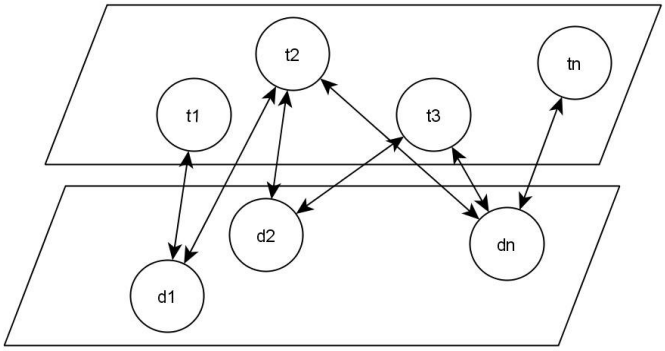


Figure 31 : réseaux de neurones à deux couches

Les modèles à couches assurent une représentation simple des techniques classiques de recherche d'information : recherche à partir de termes d'interrogation, technique de relevance feedback,...

Ils comprennent une couche de termes qui sera activée par une requête initiale d'un utilisateur et une couche document. Les deux couches sont liées par des connexions pondérées. La requête active la couche des termes et cette activation est propagée vers la couche des documents. L'activation finale des documents donne la réponse du moteur de recherche.

Cette approche a la capacité de modéliser des concepts par association d'éléments : termes, documents, liens termes-documents, liens termes-termes... Ce qui assure la modélisation des dépendances entre les termes d'indexation. Enfin, la capacité d'apprentissage des modèles connexionnistes peut également être mise à profit pour améliorer la représentation des documents, son évolution dans le temps, l'indexation n'étant plus alors uniquement basée sur une analyse statistique du contenu du document. L'approche neuronale, bien qu'ayant des résultats encourageants par exemple dans [Gallant et al., 1992] où elle obtient de meilleurs résultats que SMART sur la collection Cisi, reste inefficace en RI [Scholtes, 1994]. Concernant des tâches spécifiques comme la réinjection de pertinence, l'approche neuronale associée à une approche probabiliste permet par exemple d'améliorer les résultats en termes de précision sur la collection CACM [Crestani, 1995].

Le modèle de réseau de neurones à deux couches peut être considéré comme un graphe bipartite car composé de nœuds *documents* et de nœuds *termes*. Il ressemble donc au modèle que nous proposons au chapitre 4, néanmoins il existe des différences notables.

Dans le cas des réseaux à deux couches :

- le graphe est complet : les nœuds termes sont tous reliés aux nœuds documents,
- les pondérations des arcs et les seuils à l'intérieur d'un neurone varient au cours de la phase nécessaire d'apprentissage où les liens document-terme des documents pertinents vont être renforcés tandis que les liens document-terme des documents non pertinents vont être atténués,
- la requête est prise en compte via l'activation des neurones termes correspondants.

Dans le cas d'un graphe bipartite tel que nous l'utilisons dans notre approche pour représenter la collection de documents :

- le graphe ne varie pas. Un lien entre un nœud *document* et un nœud *terme* indique que le terme apparaît dans le document ; ce lien est pondéré,
- le nombre de liens et les poids associés à ces liens ne varient pas pendant le calcul des similarités.
- le calcul utilisé ne constitue pas un apprentissage, c'est une propagation de similarités initiales entre nœuds dans le réseau jusqu'à convergence du processus,
- la requête est incluse dans le modèle ; elle est considérée comme un document.

3.1.4 Graphes bipartites et modèles de recherche d'information

Le modèle GVC [Truong et al., 2008] considère les documents et les termes d'indexation comme les nœuds d'un graphe. A partir de la matrice document-terme représentant les documents et la requête dans l'espace des termes, un graphe bipartite orienté est construit. Ses nœuds sont de deux types : les nœuds *document/requête* et les nœuds *terme*. Un arc relie un nœud document *i* à un nœud terme *j* si le terme *j* indexe le document *i*. Le problème du calcul du score de pertinence entre un document et la requête est ramené à un problème de comparaison du nœud document au nœud requête.

Ce modèle s'inspire de [Blondel et al., 2004] pour la comparaison de graphes, en proposant en plus une prise en compte d'héritage [Wasserman et al., 1994] des propriétés de *hub* et *autorité*[Kleinberg, 1998]. L'idée est qu'une page *p* citée par une page *autorité* voit non seulement sa qualité de *hub* renforcée mais aussi sa qualité d'*autorité*.

La méthode de comparaison consiste à initialiser la matrice des similarités entre les nœuds du graphe par les valeurs de similarité initiales de ces nœuds (par exemple en utilisant le Cosinus), puis à mettre à jour par propagation ces valeurs de similarité après la réalisation d'une fermeture transitive du graphe de départ.

Soit G la matrice d'adjacences de taille $(n+m)\times(n+m)$ du graphe bipartite ayant *n* sommet documents (requête comprise) et *m* sommets termes, pour calculer la ressemblance initiale entre deux documents ou entre deux termes, la fonction cosinus est utilisée : le score de similarité initial entre un nœud *i* et un nœud *j* est alors obtenu par :

$$\begin{cases} S_0(i,j) = \frac{\sum_{k=1}^{n+m} \mathrm{G(i,k)}\times\mathrm{G(j,k)}}{\sqrt{\sum_{k=1\to n+m} G(i,k)\times G(i,k)}\times\sqrt{\sum_{k=1\to n+m} G(j,k)\times G(j,k)}} \\ S_0(i,i) = 1 \end{cases} \qquad (17)$$

Dans la mesure où il n'y a pas de liens entre deux nœuds de même type, la similarité entre deux nœuds de types différents est forcément 0.

S_0 peut alors s'écrire $\begin{bmatrix} S_T & 0 \\ 0 & S_D \end{bmatrix}$ où S_T est la matrice des valeurs de similarité entre termes et S_D la matrice des valeurs de similarité entre documents.

La matrice d'adjacence G s'écrit $\begin{bmatrix} 0 & 0 \\ W & 0 \end{bmatrix}$ où W est la matrice document-terme.

On calcule la fermeture transitive de G :

$$\mathrm{G} \leftarrow \mathrm{G} + \sum_{\mathrm{n}=2}^{\infty} \mathrm{f(n)g}\left(\frac{\mathrm{G^n}}{\|\mathrm{G^n}\|}\right) \qquad (18)$$

Avec g une fonction monotone de [0,1] dans [0,1] qui permet de généraliser la prise en compte nombre de chemin possible de longueur *n* entre deux sommets du graphe. $G^n(i,j)$ représente le nombre de chemin de longueur *n* entre *i* et *j*.

$f(n)= \alpha^n$ où α est une constante positive inférieure à 1. *f(n)* permet la prise en compte de l'influence de la longueur du chemin entre deux nœuds.

Les valeurs de S_k (la matrice de similarité à l'itération k) sont mises à jour à l'aide de la formule suivante :

$$s_{D_{k+1}} = \frac{WW^T S_{D_k} WW^T}{\sqrt{\left\|W^T W S_{T_{2k}} W^T W\right\|^2 + \left\|WW^T S_{D_{2k}} WW^T\right\|^2}} \qquad (19)$$

On ne prend en considération que les similarités entre documents pour évaluer la ressemblance avec la requête. On peut alors ordonner les valeurs de la ligne requête de la matrice S_k qui correspondent à la ressemblance de la requête avec les autres documents du graphe. Cette méthode

a été appliquée avec succès à la RI et montre une performance supérieure par rapport à celles des méthodes classiques comme la mesure Cosinus ou même Okapi [Truong et al., 2008], en effet la précision moyenne obtenue sur les corpus CISI et Cranfield améliore de plus de 50 % la précision obtenue avec Okapi.

3.2 Prise en compte des liens hypertextes

3.2.1 Tri indépendant de la requête : le *Page Rank* de Google

Sergey Brin et Lawrence Page ont initié le moteur de recherche Google en 1998 [Brin et al., 1998]. Ce moteur utilise une approche particulière pour trier les pages internet restituées consécutivement à une requête utilisateur. L'ensemble des pages internet que le moteur est capable d'indexer est considéré comme un graphe orienté. Les nœuds de ce graphe sont les pages internet, un lien entre un nœud *i* et un nœud *j* reflète le fait que la page *i* contient un hyperlien vers la page *j*. Google exploite la structure des hyperliens entre pages pour déterminer la qualité d'une page donnée.

La méthode consiste à considérer d'une part les pages dont le contenu correspond aux termes de la requête, et d'autre part de choisir parmi ces pages celles issues de sites reconnus. La mesure appelée *PageRank* reflète la qualité d'une page selon les critères précédents.

Soit une page *p*, ayant les pages $p_1,p_2,\ldots,p_n$ qui pointent vers elle, c'est-à-dire que les pages p_i citent *p*. Soit $N_{out}(p)$ le nombre de pages sortantes de la page *p*, c'est-à-dire le nombre des pages que la page *p* cite.

Le *Page Rank* de la page *p* est calculé comme suit :

$$PR(p) = (1-d) + d\sum_{1<i<n} PR(p_i)/N_{out}(p_i) \qquad (20)$$

d est un facteur d'amortissement généralement initialisé à 0.85 [Brin et al., 1998].

Le *Page Rank* peut être calculé en utilisant un simple algorithme itératif, et correspond au vecteur propre principal de la matrice normalisée des liens du Web. La formule présentée est évidemment une formule simplifiée de la formule appliquée dans Google en 2009 qui prend en compte de nombreux autres critères pour évaluer la qualité d'une page, l'âge de la page, de quelle façon elle est référencée, sur quel domaine, ses scores précédents, sa structure, la taille des caractères, si des thèmes communs existent sur les pages voisines afin d'augmenter ou pas la pertinence de la page évaluée sur ce thème[14] …

Utiliser l'information des liens entre les pages pour mesurer l'importance des sites permet de trier les résultats d'une recherche par mots clés. Les pages sont indexées indépendamment du contenu de la requête, l'analyse de liens est calculée une fois pour toutes et peut être utilisée pour trier les pages en fonction de requêtes spécifiques.

[14] Ces informations sont issues de la communauté des référenceurs par exemple WebRankInfo pour la France, en effet Google garde précieusement les détails de sa méthode, mais a communiqué sur le fait que le PageRank n'est qu'un des critères utilisées http://www.google.com/corporate/tenthings.html.

3.2.2 Tri dépendant de la requête

3.2.2.1 Le système WebQuery

Jeromy Carriere et Rick Kazman ont proposé un modèle [Carriere et al., 1997] pour combiner l'analyse des liens et la requête utilisateur. La méthode est la suivante : un ensemble de départ de pages correspondant à la requête est construit à partir du résultat produit par un moteur de recherche (les 200 premières pages restituées sont retenues), puis cet ensemble est enrichi par son voisinage. Le voisinage est constitué des pages « pointés par » ou « pointant sur » les pages de l'ensemble de départ. A partir de l'ensemble résultat, on construit le graphe de voisinage – le sous-graphe engendré par l'ensemble des voisins. Dans ce graphe, chaque page est un nœud et il existe un lien entre une page *p* et une page *q* si elles sont reliées par un hyperlien.

Pour chaque nœud du graphe de voisinage la connectivité est définie comme le nombre total de nœuds entrants et sortants. Les pages peuvent alors être classées en fonction de leur connectivité. Un défaut de cette approche est de considérer chaque lien comme ayant un apport égal pour déterminer la connectivité d'un nœud.

3.2.2.2 L'algorithme Hits

Kleinberg dans [Kleinberg, 1998] a développé une méthode d'analyse permettant d'exploiter la structure en hyperliens du Web considéré comme un graphe orienté, afin de proposer aux usagers une recherche offrant des pages à la fois pertinentes et de qualité. Comme dans le modèle précédent, l'idée est de partir d'un ensemble de départ de pages retrouvées par rapport aux termes de la requête, en l'occurrence, les 200 premières restituées par Altavista [Gibson et al., 1998]. Cet ensemble de pages de départ est ensuite enrichi avec les pages ayant un lien avec les pages de cet ensemble de départ jusqu'à un seuil afin d'éviter le surnombre. Les pages d'un même domaine qui n'ont un intérêt que du point de vue de la navigation sont supprimées. A chaque nœud du graphe ainsi obtenu est attribué un score *Hub* et un score *Autorité*.

Une page ayant une forte *autorité* doit avoir un contenu pertinent et une page ayant un fort score *hub* doit contenir des liens vers des pages ayant un haut score d'*autorité*. L'intuition suivie est qu'une page ayant de nombreux liens vers d'autres pages est un bon *distributeur* (*hub*), une page beaucoup pointée a une bonne *autorité*. Une page pointant sur de nombreuses pages *autorité* sera un meilleur *distributeur*, et de façon similaire, une page pointée par de nombreuses pages qui sont de bons *distributeurs* sera une meilleure page *autorité* que si elle était pointée par de moins bonnes pages *hub*.

Soit *G(V,E)* le graphe à étudier, h_p et a_p les scores *hub* et *autorité* d'une page *p* de *G*. L'algorithme *Hits* consiste à initialiser les scores h_p et a_p de chaque page à 1, puis à mettre à jour ces scores récursivement comme suit :

$$a(p) \coloneqq \sum_{q \to p} h(q) \qquad (21)$$
$$h(p) \coloneqq \sum_{p \to q} a(q) \qquad (22)$$

Avec $q \to p$ indique que la page *q* contient un hyperlien vers la page *p* et de même, $p \to q$ indique que la page *p* contient un hyperlien vers la page *q*.

Ces opérations sont effectuées sur toutes les pages, ce procédé est répété en normalisant les scores à chaque étape. Dans [Kleinberg, 1998] il est démontré que ces scores convergent vers des valeurs stables.

Il est possible qu'une page ayant un contenu pertinent par rapport à une requête échappe à cet algorithme du fait qu'elle soit trop peu pointée ou qu'elle n'appartienne pas au graphe de voisinage.

Un autre problème survient quand le graphe de voisinage contient des pages traitant de sujets différents de celui de la requête. Dans ce cas les hauts scores *hub* et *autorité* peuvent concerner des pages non pertinentes. Ce problème est appelé *topic drift*. Bharat montre dans [Bharat et al., 1998] que l'usage de liens pondérés associés à l'analyse de contenu, soit du document directement, soit des balises hypertextes améliore la qualité des résultats.

3.2.2.3 Le modèle de Vincent Blondel

L'algorithme hits peut consister à comparer chaque nœud du graphe étudié aux deux nœuds du graphe suivant :

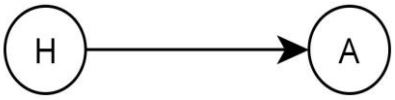


Figure 32 : graphe Hub→Autorité.

Vincent Blondel dans [Blondel et al., 2004] propose une généralisation de l'algorithme *hits* à tout type de graphes.

Par exemple soit le graphe G suivant :

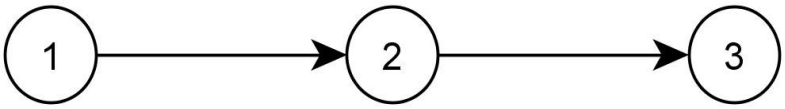


Figure 33 : graphe modèle 1→2→3

Il est possible de comparer un graphe à analyser à ce graphe modèle, en associant à chaque nœud du graphe à analyser trois scores correspondant à leurs ressemblances respectives avec les nœuds 1, 2 et 3 du graphe modèle.

Ainsi, à chaque nœud *j* de *G* trois scores x_{i1}, x_{i2}, x_{i3} sont associés et initialisés à une valeur positive puis mis à jour par les relations suivantes :

$$\begin{cases} x_{i1} \leftarrow & \sum_{j:(i,j)\in E} x_{i2} \\ x_{i2} \leftarrow \sum_{j:(j,i)\in E} x_{i1} & + \sum_{j:(i,j)\in E} x_{i3} \\ x_{i3} \leftarrow & \sum_{j:(j,i)\in E} x_{i2} \end{cases} \quad (23)$$

Pour généraliser cette approche de comparaison à tout type de graphe, chaque nœud d'un graphe à analyser $G_A(V_A,E_A)$ ayant n_A nœuds est associé à autant de scores que le graphe modèle $G_B(V_B,E_B)$ a de nœuds (n_B). Ainsi, chaque score x_{ij} ($0<i<n_B$; $0<j<n_A$) est initialisé avec une valeur positive puis mis à jour par l'équation suivante :

$$x_{ij} \longleftarrow \sum_{r:(r,i)\in E_B, s:(s,j)\in E_A} x_{rs} + \sum_{r:(i,r)\in E_B, s:(j,s)\in E_B} x_{rs} \quad (24)$$

Soit S_k la matrice des scores à l'instant *k,* l'équation précédente peut s'exprimer de façon matricielle :

$$S_{k+1} = BS_kA^T + B^TS_kA \quad (25)$$

Où *A* est la matrice d'adjacences de G_A et *B* est la matrice d'adjacences de G_B.

Pour assurer la convergence, plus précisément pour que la suite S_k admette alors deux valeurs d'adhérence, une pour les *k* pairs, l'autre pour les *k* impairs, une normalisation est appliquée comme suit :

$$S_{k+1} = \frac{BS_kA^T + B^TS_kA}{\left\|BS_kA^T + B^TS_kA\right\|} \qquad (26)$$

Cette méthode a été appliquée avec succès à l'extraction de synonymes [Blondel et al., 2004]. La méthode consiste à construire un graphe à partir d'un dictionnaire, chaque nœud du graphe est un mot et il existe un lien entre un nœud *i* et un nœud *j* si *j* apparaît dans la définition de *i*. Un mot *w* est choisi à partir duquel son graphe de voisinage G_w est construit. G_w est un sous-graphe de *G* dont les nœuds pointent sur ou sont pointés par *w*. Puis on compare chaque nœud de G_w avec le nœud 2 du graphe modèle 1→2→3 - cf. figure 33 -. Les nœuds ayant un haut score de ressemblance au nœud 2 sont de bons candidats pour la synonymie avec le mot *w*.

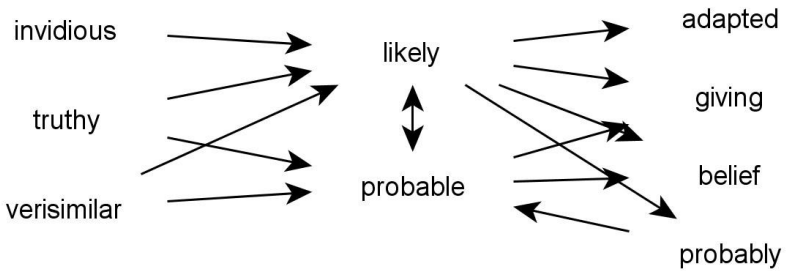


Figure 34 : graphe de voisinage de *likely* [Blondel et al., 2004]

La figure précédente représente le graphe de voisinage du mot *likely*, *probable* est un bon synonyme.

Conclusion du chapitre graphe

Dans la première partie nous avons introduit le vocabulaire ainsi que les notions et notations nécessaires à la compréhension des graphes utilisés dans le chapitre 5. En effet, nous considérons une collection de documents comme un graphe bipartite valué dans lequel chaque document est représenté par un nœud, chaque terme est également représenté par un nœud, et il existe un lien entre un nœud *document* et un nœud *terme* si le terme apparaît dans le document. Une valeur numérique est associée à un lien entre un nœud document et un nœud terme. Cette valeur correspond au poids du terme dans le document.

Dans la seconde partie nous avons introduit les critères d'analyse des graphes. Les collections étudiées au chapitre 5 sont des collections traditionnelles, les graphes qui les représentent sont des graphes de terrain, c'est-à-dire des graphes réels dans le sens où ils sont associés à des données du monde réel. Nous caractériserons les graphes particuliers que constituent les collections de documents que nous étudions. Il peut être utile de profiter de la représentation de nos données textuelles sous forme de graphe pour mettre en rapport d'éventuelles propriétés de notre système avec des caractéristiques spécifiques du graphe étudié.

Dans la partie concernant l'usage des graphes en RI, nous avons d'abord introduit les réseaux sémantiques qui sont à la base de nombreux outils performants pour la RI (thésaurus, ontologies,...) et de modèles de recherche (basés sur les concepts). Le but des modèles de RI basés sur les concepts est de combler le fossé entre l'objectif de la RI et la méthode qui la réalise : les techniques de RI classiques – cf. chapitre 2 - traitent essentiellement le signifiant, mais peu le signifié. L'objectif de la RI est de retrouver des documents pertinents par rapport à une requête, c'est-à-dire dont le contenu est similaire à celui de la requête. En pratique, les SRI recherchent les documents partageant les mêmes mots que la requête. Les modèles classiques supposent qu'il y a correspondance entre les mots et les sens. Cette supposition est fausse car un mot peut avoir plusieurs sens. Les modèles basés sur les concepts sont utilisés pour pallier cette déficience. Les modèles de RI basés sur les réseaux permettent la prise en compte de la dépendance entre les termes d'indexation qui fait défaut dans les approches classiques algébriques ou probabilistes. Le modèle que nous présentons au chapitre 5 peut être qualifié de modèle classique au sens où il fait l'hypothèse de la correspondance entre mot et sens : il utilise les termes d'indexation comme source de la pertinence des documents, sans apport d'une connaissance extérieure type thésaurus. Néanmoins, l'exploitation des relations document-terme qui est faite dans notre approche rejoint l'idée des modèles basés sur les concepts selon laquelle le sens tout entier n'est pas contenu dans le mot, il est partagé. La similarité entre deux documents est une moyenne des similarités entre les termes auxquels ils sont reliés. Notre approche, basée sur les relations structurelles, corrige également l'hypothèse de l'indépendance entre termes en considérant que la ressemblance entre termes est renforcée quand des termes apparaissent dans des documents similaires. Les approches de RI basées sur le contenu ont permis de nombreux développements. Notre approche n'est pas une approche basée sur le contenu au sens ou elle n'exploite pas de lien sémantique entre les termes d'indexation, par contre elle exploite un lien structurel. Nous souhaitons, par nos travaux, introduire cette notion de lien structurel, qui selon nous a un sens, si on suppose que la relation qui lie un terme à un document en a un.

Les travaux de recherche présentés dans le cadre de la recherche d'information sur le Web ont mis en évidence l'intérêt de l'exploitation de la structure, notamment pour améliorer une recherche basée sur le contenu.

Dans le chapitre suivant nous présenterons notre modèle de RI basé sur les graphes qui comme [Truong et al., 2008] s'inspire de Kleinberg et de Blondel, et exploite la structure du graphe

document-terme au travers d'un graphe bipartite simple dans lequel les similarités sont propagées dans le but d'identifier les documents similaires à la requête.

Chapitre 4 : Modèle théorique et mesure de similarité structurelle

Dans ce chapitre nous présenterons d'abord les travaux de [Jeh et al., 2002] qui sont à la base de nos travaux. Ces travaux introduisent une approche générale pour la comparaison d'objets d'un domaine représenté par un graphe. Une première méthode concerne la comparaison de nœuds au sein d'un graphe simple orienté ; une seconde méthode concerne la comparaison de nœuds au sein d'un graphe bipartite. C'est cette seconde méthode que nous avons adapté puis appliquée au domaine de la RI.

L'objectif en RI est de fournir à un utilisateur une liste de documents pertinents par rapport à sa requête. Cela requiert d'être en mesure de comparer les représentations des documents avec la représentation de la requête - cf. chapitre 2.

Notre souhait est de proposer une mesure de comparaison entre documents et requêtes basée sur les similarités structurelles et non pas sur les seules similarités de surface comme c'est le cas pour la mesure Cosinus, le coefficient de Dice ou la mesure de Jaccard – cf. chapitre 2, section 2.2. Nous pensons que l'information apportée par la structure relationnelle présente un intérêt pour la RI et mérite d'être étudier.

L'adaptation d'une méthode générale à un domaine particulier nécessite la prise en compte des spécificités du domaine. Pour cela, nous proposerons plusieurs méthodes parmi lesquelles une version basique, puis une version avec prise en compte de la pondération des mots et aussi une version pondérée et normalisée. De plus, nous discuterons de problèmes liés à l'utilisation d'un algorithme itératif de propagation au sein d'un réseau (ici propagation des similarités inter-nœuds). Ces problèmes concernent la convergence de l'algorithme, son initialisation, sa condition d'arrêt ainsi que sa complexité.

En prévision d'un usage de notre méthode dans le cadre de grandes bases documentaires, pour répondre aux problèmes liés à la complexité tels que l'espace et le temps de calcul, nous proposerons plusieurs manières d'utiliser notre méthode sur des sous-ensembles de collection. Nous évoquerons l'implémentation de notre modèle avant une seconde partie, expérimentale celle-là, où nous testerons notre algorithme. Les tests consisteront à appliquer notre algorithme sur des exemples simples afin de vérifier si son fonctionnement correspond à nos attentes : notre méthode permet-elle de bien trier des documents ?, est-elle sensible à la pondération ?, peut on constater l'effet de propagation des similarités ?, … Le chapitre 5 quant à lui présentera l'évaluation sur des collections internationales.

1. L'origine de notre modèle

Dans [Jeh et al., 2002] les auteurs proposent une méthode générale pour mesurer la similarité structurelle entre objets d'un domaine impliquant une relation d'objet à objet. Dans un second temps, ils proposent une seconde version de cette méthode adaptée aux domaines bipartites.

1.1 Le modèle SimRank basé sur un graphe orienté

Dans cette approche, les objets et leurs relations sont modélisés par un graphe orienté *G(V,E)* dont les nœuds *V* représentent les objets du domaine étudié et dont les arcs *E* représentent les

relations entre ces objets. L'hypothèse de départ est que « des objets sont similaires s'ils sont reliés par des objets similaires ». Le but de cette approche est de déterminer les ressemblances entre nœuds du graphe en leur attribuant un score de similarité appelé *SimRank* et définit dans ce qui suit.

Soit *I(v)* l'ensemble des prédécesseurs d'un nœud *v, |I(v)|* est le cardinal de l'ensemble des prédécesseurs.

Le Score SimRank *s(a,b)* entre un objet *a* et un objet *b* et est défini par :

$s(a,b) = 1$ si a = b et sinon : (1)

$$s(a,b) = \frac{C}{|I(a)||I(b)|}\sum_{i=1}^{|I(a)|}\sum_{j=1}^{|I(b)|} s(I_i(a), I_j(b)) \quad (2)$$

Où *C* est une constante[15] entre 0 et 1 ;

Ces équations sont appliquées à chaque paire de nœuds du graphe de départ ; il en résulte, pour un graphe de taille *n,* un ensemble de n^2 équations.

L'équation (2) nous dit que pour calculer *s(a,b)* il faut itérer un calcul sur toutes les paires de prédécesseurs *(Ii(a), Ij(b))* et sommer la similarité *S(Ii(a), Ij(b))* de ces paires. La normalisation est obtenue en divisant la somme obtenue par le nombre de paires de prédécesseurs *|I(a)||I(b)|*.

La similarité entre *a* et *b* se confond avec la similarité moyenne entre les prédécesseurs de *a* et les prédécesseurs de *b*.

Exemple :

Soit le graphe *G* suivant :
G(V={Univ, Prof A, Prof B, Elève A, Elève B} ;
E={(Univ, Prof A) ; (Univ, Prof B) ; (Prof A, Elève A) ; (Elève A, Univ) ; (Prof B, Elève B) ; (Elève B, Prof B)})

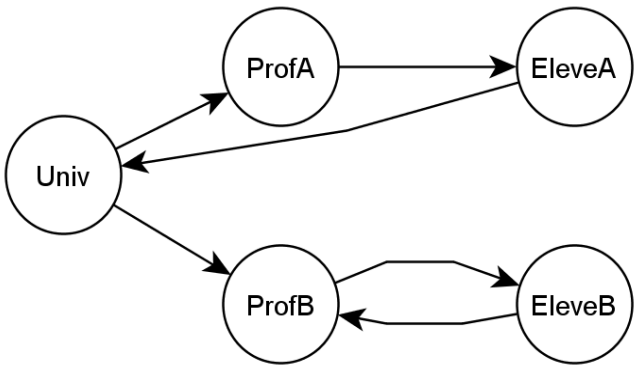


Figure 35 : graphe *G* Université-Professeur-Elève [Jeh et al., 2002]

Les nœuds représentent des pages Web et les arcs représentent les liens hypertextes entre ces pages. *Univ* est la page d'accueil de l'université, Prof A et Prof B sont les pages de deux professeurs, *Elève A* et *Elève B* sont les pages de deux étudiants.

[15] Les auteurs définissent C comme un facteur d'amoindrissement, et précisent que sa valeur dépend de l'utilisation qui en est faite, et du domaine où ces formules sont appliquées.

Le calcul de similarité ci-dessus nous permet d'attribuer des scores aux couples de nœuds du graphe *G*. On peut représenter le résultat par un graphe G² de couples de nœuds de G et de leurs scores (scores obtenus avec la constante C fixée à 0.8).

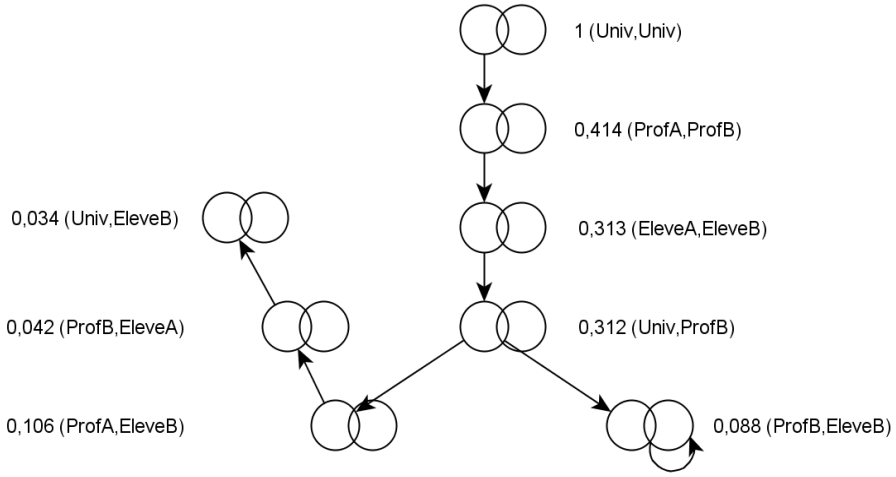


Figure 36 : graphe G^2 Université-Professeur-Elève [Jeh et al., 2002]

Dans ce graphe, chaque nœud représente une paire de nœuds de *G*. Un nœud (*a*, *b*) de *G²* pointe vers un nœud (*c*, *d*) si dans G *a* pointe vers *b* et *c* pointe vers *d*.

Un nœud de type [*a*,*a*] a un score de similarité égal à 1, ce qui signifie qu'un objet est complètement similaire à lui-même - cf. (1).

Les deux scores consécutifs au score du nœud source *(Univ,Univ)* sont obtenus par les couples *(Prof A, Prof B)* et *(Elève A, Elève B)*. Ce qui signifie que les pages les plus similaires à la page de l'université sont celles des professeurs suivies par celles des élèves.

Ce calcul général de similarité permet d'ordonner des objets en prenant en compte leur similarité structurelle dans le contexte où ils interviennent. Au fil des itérations, la similarité se propage de nœuds en nœuds, depuis les nœuds sources de G² vers leurs successeurs. Par exemple le nœud (*Univ, Univ*) ressemble totalement à lui-même et « donne » une partie de sa ressemblance à lui-même au nœud (*ProfA, ProfB*) (du fait que *ProfA* et *ProfB* sont reliés à *Univ)* qui donne une partie de sa ressemblance au nœud (EleveA, EleveB), et ainsi de suite. Cette propagation correspond à une propagation de paire en paire dans G.

1.2 SimRank générique bipartite

Après avoir défini et implémenté le modèle Basic SimRank, Glen Jeh et Jennifer Widom ont décidé de l'étendre aux domaines comportant deux types d'objets. Une structure appropriée pour représenter un tel domaine est un graphe bipartite. On peut alors calculer deux types de scores de similarité :

- Un score de similarité s_1 pour les nœuds de type 1. Deux objets de type 1 seront considérés comme similaires s'ils pointent vers des nœuds de type 2 similaires,
- Un score de similarité s_2 pour les nœuds de type 2. Deux objets de type 2 seront similaires s'ils sont pointés par des objets de type 1 similaires.

Ces notions peuvent être formalisées par deux fonctions s_1 et s_2:

Considérons un graphe *G(V ;E)*, *O(v)* est l'ensemble des successeurs d'un nœud *v* et *I(v)* l'ensemble de ces prédécesseurs.

|O(v)| est le cardinal de l'ensemble des successeurs et *|I(v)|* est le cardinal de l'ensemble des prédécesseurs.

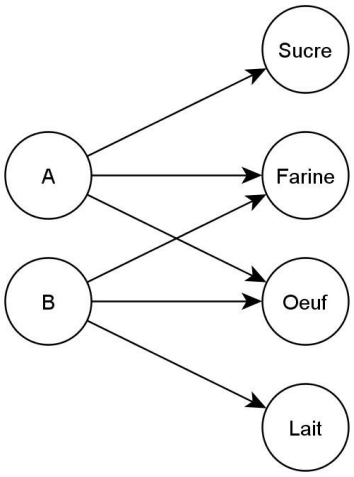
Le Score SimRank s_i *(a,b)* entre deux objets *a* et *b* de type *i* est défini par :

$$s_1(a,b) = \frac{C_1}{|O(a)||O(b)|}\sum_{i=1}^{|O(a)|}\sum_{j=1}^{|O(b)|} s_2(O_i(a), O_j(b)) \qquad (3)$$

$$s_2(a,b) = \frac{C_2}{|I(a)||I(b)|}\sum_{i=1}^{|I(a)|}\sum_{j=1}^{|I(b)|} s_1(I_i(a), I_j(b)) \qquad (4)$$

Exemple :

Soit le graphe G suivant :



G(={A,B}; ={Sucre, Farine, Œufs, Lait};E={(A, Sucre);(A, Farine) ;(A, Œufs) ;(B, Farine); (B, Œufs); (B; Lait)}).

Figure 37 : graphe *G* Personnes-Aliments [Jeh et al., 2002]

Les nœuds ronds représentent des personnes et des aliments. Un arc entre une personne et un aliment représente le fait que la personne achète l'aliment ou réciproquement que l'aliment est acheté par la personne. Dans cet exemple il apparaît clairement que *Farine* et *Œufs* se ressemblent du fait qu'ils sont tous deux achetés par les deux personnes. Qu'en est-il de la ressemblance entre *Sucre* et *Lait* ?

Comme dans la version précédente, on peut représenter le résultat de l'application du calcul des similarités aux nœuds de *G* par un graphe G^2 de couples de nœuds.

Les scores ont été obtenus avec les constantes C1 et C2 fixées à 0.8.

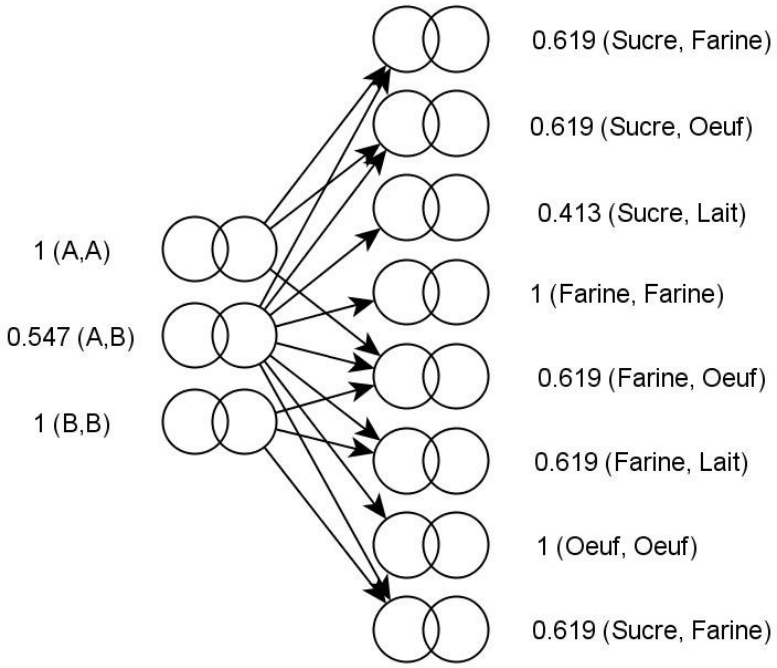


Figure 38 : graphe G^2 Personnes-Aliments [Jeh et al., 2002]

Sucre et *Lait* ont obtenu un score de similarité non nul. En effet ils sont achetés par deux personnes différentes, mais qui ont une certaine similarité due au fait que ces deux derniers ont acheté de la *Farine* et des *Œufs*. Cela illustre le phénomène de propagation de la similarité.

Le score du couple (*Farine, Œufs*) est identique à celui de (*Sucre, Œufs*). Or, *Farine* et *Œufs* sont achetés par les deux personnes contre un seul achat de la personne *A* pour *Sucre*.

Les auteurs invitent les acteurs de différents domaines de recherche à appliquer cette méthode dans des domaines où la relation bipartite intervient. Il nous a semblé intéressant d'appliquer cet algorithme à la RI. En effet la propagation des ressemblances amène à considérer comme proche *Sucre* et *Lait* qui ont pour seul point commun d'être achetés par des personnes achetant également de la *Farine* et des *Œufs.* Cette propagation de proche en proche de la similarité et cette capacité à ordonner des objets en fonction de leurs relations nous ont parus pouvoir s'appliquer à la RI.

2. Notre modèle de recherche d'information basé sur les graphes bipartites

Les documents sont considérés comme un ensemble de termes. L'application de la méthode décrite auparavant consiste d'une part à représenter ces données sous forme d'un graphe bipartite dans lequel les nœuds de type 1 sont des documents et les nœuds de type 2 sont des termes et d'autre part à définir la relation structurelle qui les lie : la contenance c'est-à-dire le fait qu'un document contienne des termes et réciproquement que des termes soient contenus dans un document.

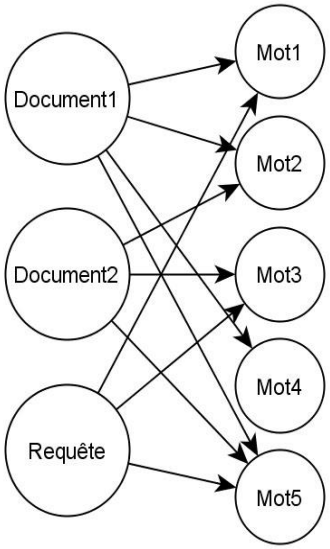
Un nœud *document* est relié par un arc à un nœud *terme* si le terme indexe le document.
La requête est intégrée à ce graphe comme un nœud *document* supplémentaire.

Exemple :

Soit un corpus de recherche composé de deux documents constitués de termes représentatifs :
$document_1 : \{mot_1, mot_2, mot_4, mot_5\}$

document₂ : {mot₂, mot₃, mot₅}
Et soit une requête de recherche :
Requête : {mot₁, mot₃, mot₅}

Le corpus et la requête sont représentés par le graphe G suivant :



G(={document₁, document₂, requête}; ={mot₁, mot₂, mot₄, mot₄, mot₅};E={(document₁, mot₁);(document₁, mot₄) ;(document₁, mot₅) ; (document₂, mot₂);(document₂, mot₃) ;(document₂, mot₅)})
Figure 39 : graphe bipartite document-terme

La méthode proposée permet de trier tous les couples d'objets. L'objectif dans le cadre de la RI est de trier les documents en fonction de leur similarité par rapport à la requête. Ainsi, seule une sous-partie des résultats nous intéresse : la liste des couples document-requête triés par le score SimRank de chaque couple. Nous espérons que l'ordre créé par la similarité structurelle permettra d'obtenir de bons résultats sur les mesures retenues lors de l'évaluation de notre méthode sur des collections de test.

2.1 SimRank pour la recherche d'information

L'application des formules (3) et (4) au domaine de la RI nous amène à remplacer les objets de type 1 par des documents, les objets de type 2 par des termes.

La similarité $S_d(d_i, d_j)$ entre deux documents d_i et d_j est définie comme suit :

$$S_d(d_i, d_j) = \begin{cases} 1 \; si \; d_i = d_j \\ \frac{C_1}{|T_{d_i}||T_{d_j}|} \sum_{t_k \in T_{d_i}} \sum_{t_l \in T_{d_j}} S_t\left(t_k(d_i), t_l(d_j)\right) \; si \; d_i \neq d_j \end{cases} \quad (5)$$

T_{d_i} est l'ensemble des termes du document d_i.
$|T_{d_i}|$ est le nombre de termes appartenant au document d_i.
$t_k(d_i)$ est le $k^{ème}$ terme du document d_i (le $i^{ème}$ document de la collection).
C_1 est une constante de propagation.

La similarité $S_t(t_i, t_j)$ entre deux termes t_i et t_j est définie comme suit :

$$S_t(t_i,t_j)=\begin{cases}1\ si\ t_i=t_j\\ \frac{C_2}{|D_{t_i}||D_{t_j}|}\sum_{d_k\in D_{t_i}}\sum_{d_l\in D_{t_j}}S_d\left(d_k(t_i),d_l(t_j)\right)\ si\ t_i\neq t_j\end{cases} \quad (6)$$

D_{t_i} est l'ensemble des documents contenant le terme t_i.

$|D_{t_i}|$ est le nombre de documents contenant le terme t_i.

$d_i(t_j)$ est le i^{eme} document contenant le terme t_j (le j^{eme} terme du vocabulaire).

C_2 est une constante de propagation.

On peut remarquer que si on prend C_1=1 et C_2=0 alors on retrouve la formule du Cosinus :

en effet, C_2=0 implique que $S_t(t_i,t_j)$ vaut 1 si $t_i=t_j$ et 0 sinon ; ce qui induit que la similarité entre deux documents est égale à :

$$S_d(d_i,d_j)=\frac{\sum_{t\in T_{d_i}\cap T_{d_j}}S_t(t,t)}{|T_{d_i}||T_{d_j}|}=\frac{\left|T_{d_i}\cap T_{d_j}\right|}{|T_{d_i}||T_{d_j}|}$$

2.2 Extension du modèle au cas des graphes bipartites pondérés

Les formules (2) et (3) correspondent à un graphe bipartite non orienté, non pondéré, la transposition que nous proposons dans le domaine document-terme au travers des formules (5) et (6) s'applique à un corpus représenté par un graphe bipartite document-terme dont les arcs représentent l'appartenance des termes aux documents. Un terme appartient ou n'appartient pas à un document ; de la même façon un document contient un terme ou ne le contient pas. Or, comme nous l'avons vu au chapitre 2, dans le domaine de la RI les meilleurs résultats sont obtenus lorsque les documents sont représentés sous la forme d'une liste de termes pondérés. Une telle description se traduit par un graphe bipartite pondéré dans lequel les arcs entre documents et termes sont pondérés par le poids des termes apparaissant dans les documents. C'est pourquoi il est nécessaire de généraliser les formules (5) et (6) au cas des graphes bipartites pondérés.

Soit un corpus décrit par *T* et *D* :

$T=(t_j)_{j=1..m}$ est l'ensemble des termes du corpus avec *m* le nombre de termes du vocabulaire .

$D=(d_i)_{i=1..n}$ est l'ensemble des documents du corpus avec *n* le nombre de documents de la collection.

Avec $d_i=(w_{i1},w_{i2},\ldots,w_{ij},\ldots w_{im})$ où w_{ij} est le poids du terme *j* dans le document *i*.

Afin de prendre en compte les poids des termes d'indexation ce qui correspond à donner aux arcs document-terme un poids qui jusque-là valait 1, nous proposons les formules suivantes qui généralisent les formules (5) et (6) au cas des graphes bipartites pondérés.

$$S_d(d_i,d_j)=\begin{cases}1\ si\ i=j\\ \frac{C_1}{|d_i||d_j|}\sum_{t_k\in T_{d_i}}\sum_{t_l\in T_{d_j}}w_{ik}\times w_{jl}\times S_t(t_k,t_l)\ si\ i\neq j\end{cases} \quad (7)$$

$$S_t(t_i,t_j)=\begin{cases}1\ si\ i=j\\ \frac{C_2}{|t_i||t_j|}\sum_{d_k\in D_{t_i}}\sum_{d_l\in D_{t_j}}w_{ki}\times w_{lj}\times S_d(d_k,d_l)\ si\ i\neq j\end{cases} \quad (8)$$

A priori, le choix de la normalisation par le cardinal apparaît comme peu naturel du fait que l'on ne prend plus en compte la simple présence ou absence des termes dans les documents. On peut s'interroger sur l'impact du choix de la normalisation à adopter. Notre étude est une étude basée sur la relation structurelle, même si nos données ont au départ un sens spatial – cf. matrice document-terme. Cette étude étant un point de départ à d'éventuelles autres recherches, nous avons choisi

pour nos expérimentations de normaliser sur la base de la norme 1 à la fois pour les documents et pour les termes.

2.2.1 SimRank normalisé avec la norme 1

$$S_d(d_i, d_j) = \begin{cases} 1 \; si \; i = j \\ \frac{C_1}{(\sum_{k=1}^{m} |w_{ik}|) \times (\sum_{l=1}^{m} |w_{jl}|)} \sum_{t_k \in T_{d_i}} \sum_{t_l \in T_{d_j}} w_{ik} \times w_{jl} \times S_t(t_k, t_l) \; si \; i \neq j \end{cases} \quad (9)$$

$$S_t(t_i, t_j) = \begin{cases} 1 \; si \; i = j \\ \frac{C_2}{(\sum_{k=1}^{n} |w_{ki}|) \times (\sum_{l=1}^{n} |w_{lj}|)} \sum_{d_k \in D_{t_i}} \sum_{d_l \in D_{t_j}} w_{ki} \times w_{lj} \times S_d(d_k, d_l) \; si \; i \neq j \end{cases} \quad (10)$$

En vue d'une étude ultérieure dont il sera question dans la partie dédiée aux perspectives, nous proposons une autre manière de normaliser. En effet nous souhaitons comparer notre méthode de tri principalement à la méthode Cosinus représentante à nos yeux de la similarité vectorielle directe. Notre méthode exploite les similarités directes et indirectes pour trier des documents par rapport à une requête.

Un des objets de cette thèse est de comparer la similarité structurelle (directe et indirecte) à la similarité vectorielle (directe uniquement). Nous envisageons une étude ultérieure qui consisterait à déterminer la similarité indirecte pure au sens où elle serait séparée de la similarité directe ce qui n'est pas le cas dans cette étude-ci. Pour ce faire, nous envisageons une opération de type « soustraction » entre la similarité directe fournie par la mesure Cosinus et la similarité structurelle fournie par la mesure SimRank. Pour éventuellement réaliser une telle opération les deux mesures doivent être comparables, alors dans cette éventualité nous prévoyons une formule utilisant la norme euclidienne comme le Cosinus.

2.2.2 SimRank normalisé avec la norme euclidienne

Ce sont les formules (9) et (10) qui seront appliquées et testés car elles constituent la première brique de notre modèle basés sur les similarités structurelles. Néanmoins, nous avons proposé de normaliser avec la norme 2 à l'instar de la mesure Cosinus qui est le produit scalaire normalisé par la norme 2.

En remplaçant le produit scalaire par le SimRank on obtient :

$$S_d(d_i, d_j) = \begin{cases} 1 \; si \; i = j \\ \frac{C_1}{\sqrt{\sum_{k=1}^{m} w_{ik}^2} \times \sqrt{\sum_{l=1}^{m} w_{jl}^2}} \sum_{t_k \in T_{d_i}} \sum_{t_l \in T_{d_j}} w_{ik} \times w_{jl} \times S_t(t_k, t_l) \; si \; i \neq j \end{cases} \quad (11)$$

$$S_t(t_i, t_j) = \begin{cases} 1 \; si \; i = j \\ \frac{C_2}{\sqrt{\sum_{k=1}^{n} w_{ki}^2} \times \sqrt{\sum_{l=1}^{n} w_{lj}^2}} \sum_{d_k \in D_{t_i}} \sum_{d_l \in D_{t_j}} w_{ki} \times w_{lj} \times S_d(d_k, d_l) \; si \; i \neq j \end{cases} \quad (12)$$

Avec la norme euclidienne notre formule se rapproche de la formule du Cosinus classique entre deux documents. En effet le Cosinus peut être vu comme un cas particulier de notre formule avec $S_d(d_i, d_j) = 1$ si *i=j* et $S_d(d_i, d_j) = 0$ sinon. L'usage de cette formule sera évoqué dans les perspectives de nos travaux.

2.3 Traduction du modèle sous forme matricielle dans le cas de la norme-1

En RI, le corpus et les collections étudiés sont de plus en plus conséquents, les opérations de traitements sont parfois lourdes – cf. chapitre 2 -, les collections comptent de nombreuses requêtes,… L'automatisation des traitements est une nécessité, ainsi l'utilisation des calculs matriciels constitue un gain de temps car ils permettent de traiter globalement les données d'entrées traditionnellement représentées par une matrice document-terme.

Pour le traitement automatique de ce type d'entrée, nous proposons notre formule sous une forme matricielle.

Soient

W la matrice document-terme représentant le corpus et la requête, w_{ij} est le poids du terme *j* dans le document *i*. *W'* est la transposée de *W*.

T_q la matrice de similarité terme-terme à l'itération *q*, T' est la transposée de T.

D_q la matrice similarité document-document à l'itération *q*.

n le nombre de termes.

m le nombre de documents.

Les formules (9) et (10) deviennent :

$$D_q = \frac{C_1}{\sum_{l=1}^{n} W(i,l).\sum_{l=1}^{n} W(j,l)} \times W T_q' W' \qquad (13)^{16}$$

$$T_{q+1} = \frac{C_2}{\sum_{l=1}^{m} W(l,i).\sum_{l=1}^{m} W(l,j)} \times W' D_q W \qquad (14)^{17}$$

D est la matrice de similarité entre documents. Une case *ij* contient $S_d(d_i, d_j)$ qui représente combien le document *i* ressemble au document *j*.

De même *T* est la matrice de similarité entre termes. Une case *ij* contient $S_t(t_i, t_j)$ qui représente combien le terme *i* ressemble au terme *j*.

Une itération de cet algorithme consiste à effectuer (13) puis (14). Un algorithme itératif pose deux types de problèmes : l'initialisation et la convergence.

Dans la pratique, pour réaliser l'implémentation de (13) et (14), nous avons choisi de commencer par donner une valeur initiale à la matrice *D* en posant D_0 = I avec I est la matrice unité. Ce qui correspond à exploiter l'information triviale qui consiste à considérer que « un document est similaire à lui-même et n'est pas similaire aux autres ». Puis nous calculons T_0 en appliquant la formule (14). La formule 18 met à jour le score SimRank de chaque couple de termes dans *T*.

La question qu'il importe de vérifier est celle relative à la convergence des suites T_k et D_k. Si elles convergent, alors on considère que leur limite est le SimRank.

2.4 Convergence des similarités entre documents et entre termes

Pour démontrer la convergence, nous démontrons dans un premier temps la croissance des suites D_q et T_q. Puis nous montrerons qu'elles sont bornées.

---

[16] A chaque itération, chaque élément de la diagonale est remis à 1

[17] idem

2.4.1 Croissance des suites des valeurs de similarité

Dans cette partie nous allons montrer que les suites D_q et T_q sont croissantes :

D_q (respectivement T_q) est croissante si elle vérifie *D_{q+1}-D_q >= 0 (respectivement T_{q+1}-T_q >= 0)*. Cela traduit que pour tout *i,j, $D_{q+1}(i,j) \geq D_q(i,j)$ (respectivement $T_{q+1}(i,j) \geq T_q(i,j)$)*.

Preuve :
Supposons que *$T_{q+1} - T_q \geq 0$*.

Montrons que *$D_{q+1} - D_q \geq 0$*. Il suffit de montrer que $D_{q+1}(i,j) - D_q(i,j) \geq 0$ pour tout i,j tel que $i \neq j$[18].

$$D_{q+1}(i,j) - D_q(i,j) = \frac{C_1}{(\sum_{k=1}^{m}|w_{ik}|) \times (\sum_{l=1}^{m}|w_{jl}|)} \sum_{k=1}^{m}\sum_{l=1}^{m} w_{ik} \times w_{jl} \times \mathrm{T_{q+1}(k,l)} - \frac{C_1}{(\sum_{k=1}^{m}|w_{ik}|) \times (\sum_{l=1}^{m}|w_{jl}|)} \sum_{k=1}^{m}\sum_{l=1}^{m} w_{ik} \times w_{jl} \times \mathrm{T_q(k,l)}$$

Soit

$$D_{q+1}(i,j) - D_q(i,j) = \frac{C_1}{(\sum_{k=1}^{m}|w_{ik}|) \times (\sum_{l=1}^{m}|w_{jl}|)} \sum_{k=1}^{m}\sum_{l=1}^{m} w_{ik} \times w_{jl} \times (\mathrm{T_{q+1}(k,l)} - \mathrm{T_q(k,l)})$$

Par hypothèse on a $\mathrm{T_{q+1}} - \mathrm{T_q} \geq 0$, de plus C_1 ainsi que les poids w_{ij} sont positifs.
On a donc $\frac{C_1}{(\sum_{k=1}^{m}|w_{ik}|)\times(\sum_{l=1}^{m}|w_{jl}|)} \sum_{k=1}^{m}\sum_{l=1}^{m} w_{ik} \times w_{jl} \geq 0$

Il s'en suit que $\mathrm{T_{q+1}} \geq \mathrm{T_q} \Longrightarrow \mathrm{D_{q+1}} \geq \mathrm{D_q}$

De manière analogue on montre que $\mathrm{D_q} \geq \mathrm{D_{q-1}} \Longrightarrow \mathrm{T_{q+1}} \geq \mathrm{T_q}$
De plus, de manière évidente *$D_1 > D_0$* en effet *$D_0 = I$* (matrice identité) et *D_1* est positif ou nul avec les éléments de la diagonale égaux à un.
On en déduit que les suites *D_q* et *T_q* sont croissantes.

2.4.2 Borne des suites des valeurs de similarité

Dans cette partie nous allons montrer que les suites D et T sont bornées.
Preuve :
Supposons que $0 \leq T(i,j) \leq 1$

Montrons que $D(i,j) \leq 1$

$$D(i,j) = \frac{C_1}{(\sum_{k=1}^{m}|w_{ik}|)\times(\sum_{l=1}^{m}|w_{jl}|)} \sum_{k=1}^{m}\sum_{l=1}^{m} w_{ik} \times w_{jl} \times \mathrm{T(k,l)}$$

[18] Dans le cas ou i=j, D(i,j) et T(i,j) valent 1 car on est sur la diagonale de D et T respectivement.

Or $T(i,j) \leq 1$ alors $D(i,j) \leq \frac{C_1}{(\sum_{k=1}^{m}|w_{ik}|)\times(\sum_{l=1}^{m}|w_{jl}|)} \sum_{k=1}^{m}\sum_{l=1}^{m} w_{ik} \times w_{jl}$

Et $(\sum_{k=1}^{m}|w_{ik}|) \times \left(\sum_{l=1}^{m}\left|w_{jl}\right|\right) = \sum_{k=1}^{m}\sum_{l=1}^{m} w_{ik} \times w_{jl}$
Donc $D(i,j) \leq C_1$
Comme $0 \leq C_1 \leq 1$
Alors $D(i,j) \leq 1$
De manière analogue on démontre $T(i,j) \leq 1$ en supposant que $0 \leq D(i,j) \leq 1$
Les suites D_q et T_q sont croissantes et bornées ; elles convergent.

2.5 Complexité de l'algorithme

Le calcul de (13) consiste en deux multiplications matricielles : la première entre *W* (de taille *d×t*, avec *d* le nombre de documents de la collection et *t* le nombre de termes du vocabulaire) et *T* (de taille t^2) et la seconde entre le résultat de la multiplication précédente et W^T (de taille *t×d)*.

Le calcul de (14) consiste également en deux multiplications matricielles : la première entre W^T (de taille *t×d)* et *D* (de taille d^2) et la seconde entre le résultat de la multiplication précédente et *W* (de taille *d×t*).

La complexité de chaque itération est majorée par σ(max(*d,t*)^3).

2.6 Méthode d'utilisation

Notre application nous permet de traiter des corpus de taille réduite. En effet les multiplications matricielles nécessaires au calcul des similarités entre documents et requête constituent un calcul complexe.

La complexité élevée rend difficile le traitement SimRank sur des corpus de taille élevée (supérieurs à 10 000 documents). Il n'est possible d'utiliser le SimRank sur de tels corpus qu'en travaillant sur une partition du corpus, ou en complément d'une autre méthode de telle manière à ne l'appliquer que sur une sous-partie des données de départ.

Concernant la partition de corpus, plusieurs méthodes peuvent être envisagées : découper le corpus en sous-parties de taille fixée judicieusement, ou alors partitionner en prenant en compte certaines propriétés spécifiques des documents comme la taille par exemple,…

Concernant l'usage du SimRank sur une sous-partie de corpus, plusieurs utilisations sont envisageables, parmi lesquelles, le filtrage de documents arrivés tête de liste après l'usage d'une autre méthode de tri, le tri de documents répondants à des critères spécifiques, ou pourquoi pas le repêchage de certains documents mal classés.

Dans les expérimentations du 5ème chapitre nous avons envisagé l'usage de notre méthode à la suite d'un premier tri. Le tri des données de départ est réalisé par une des méthodes : Okapi et Cosinus.

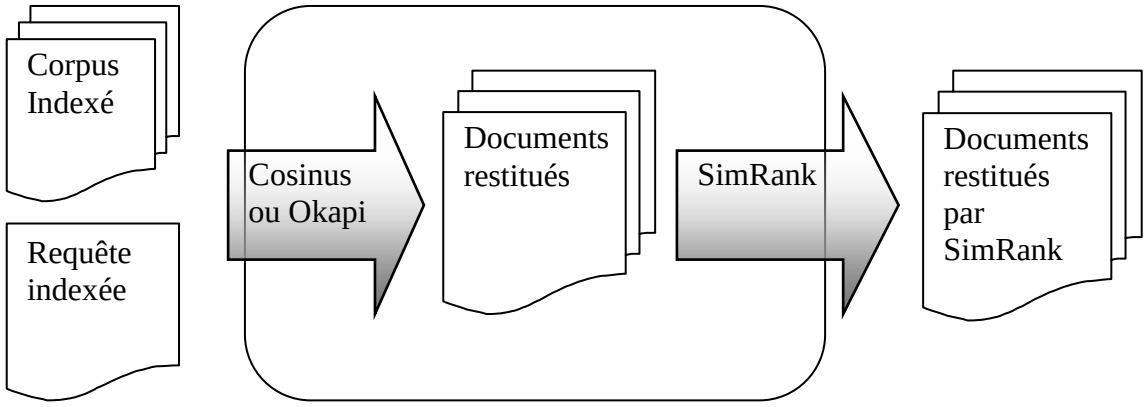


Figure 40 : notre méthode en deux phases (filtrage Cosinus ou Okapi puis tri SimRank)

3. Illustration des principes de la méthode

Afin de vérifier si notre algorithme possède bien les différentes propriétés attendues en termes de tri, de sensibilité à la pondération, et de propagation, nous réalisons une série d'expériences. Sur la base de cas d'école nous allons examiner quelle valeur de similarité est attribuée à des documents comparés en fonction des termes qu'ils ont en commun, du poids attribué à ces termes et des relations entretenues avec les autres documents.

Dans la suite nous notons $s(d_i, d_j)$ le SimRank entre deux documents d_i et d_j. Le SimRank entre d_i et d_j est calculé selon (13) et (14) correspondant au SimRank norme 1 des formules (9) et (10).

C_1, C_2 sont fixées à 0.8. Le nombre d'itérations est fixé à 20. Nous noterons $cos(d_i, d_j)$ le Cosinus entre les documents car nous avons souhaité étudier la similarité structurelle (directe et indirecte) avec la similarité vectorielle (directe uniquement) comme témoin.

3.1 Influence du nombre de termes communs sur la similarité entre deux documents

L'objectif de cette partie est d'apprécier l'évolution du SimRank en fonction du nombre de termes communs lors de la comparaison de deux graphes.

On va comparer successivement un document témoin composé de 10 termes à onze documents également composés de 10 termes. Le nombre de termes que ces derniers ont en commun avec le document témoin varie de 0 à 10.

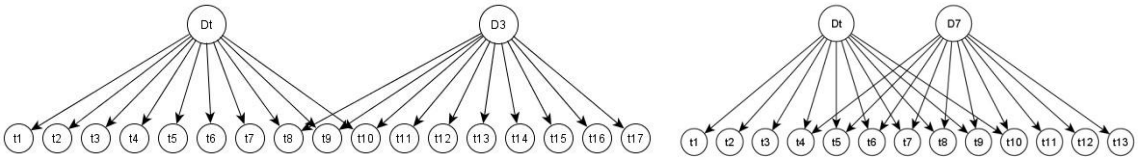


Figure 41 : graphe représentant la comparaison du document témoin avec un document ayant 30 % des termes en commun avec lui (à gauche), et avec un document ayant 70 % des termes en commun avec lui (à droite)

La figure 41 illustre deux des onze comparaisons dont les résultats sont présentés dans la figure 42.

Nous allons faire maintenant constater l'évolution des scores SimRank et Cosinus entre le document témoin et les autres documents en fonction du nombre de termes qu'ils ont en commun :

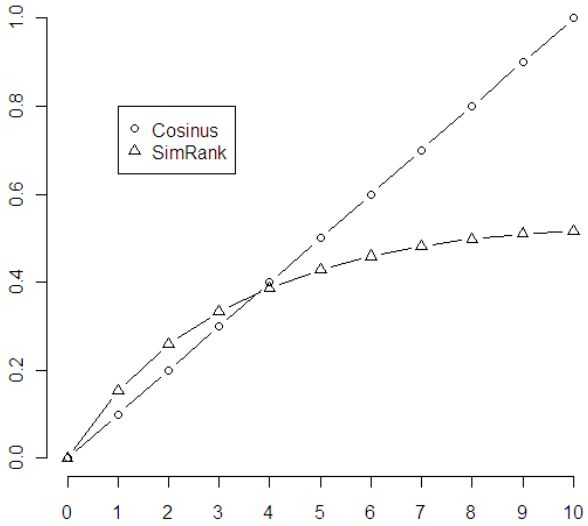


Figure 42 : évolution du score SimRank et du score Cosinus entre deux documents en fonction du nombre de termes communs

La figure 42 indique l'évolution du score SimRank et du score Cosinus entre deux documents. Le premier constat est que les deux mesures ont un score nul quand les documents ne partagent aucun terme. Le second constat est que le Cosinus suit une droite x=y alors que le score SimRank suit une courbe qui croît plus vite que la courbe du Cosinus jusqu'à ce que le nombre de termes commun dépasse 5 termes en communs, à partir de quoi elle continue de croître mais plus lentement. On peut remarquer que le score SimRank obtenu quand les deux documents partagent tous leurs termes, ce qui correspond à la similarité entre le document et lui-même, n'est pas égal à 1. Enfin, on constate que plus des documents ont de termes en communs plus leurs scores SimRank et Cosinus sont élevés.

L'influence du nombre de termes non communs sur l'évolution du score est maintenant examinée. Pour cela un document témoin composé de 5 termes est successivement comparé à dix documents possédant également ces 5 termes mais dont la taille (en nombre de termes) varie. Ainsi, on va constater l'influence sur les scores SimRank et Cosinus de l'augmentation de la proportion de termes non communs. La taille des documents varie de 10 termes à 100 termes :

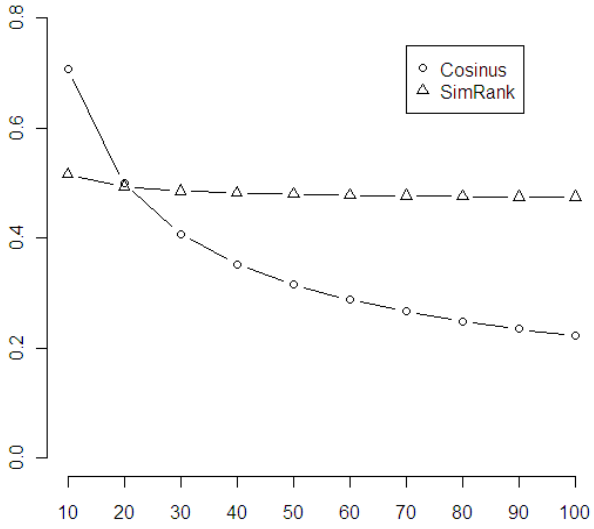


Figure 43 : évolution du score SimRank et du score Cosinus entre deux documents en fonction du nombre de termes non-communs

La figure 43 montre l'évolution des scores SimRank et Cosinus entre deux documents partageant 5 termes, quand la taille d'un des deux documents augmente.

On constate que la courbe des scores Cosinus décroît continuellement de 0,7 quand les deux documents ont 5 termes en commun et 5 termes différents à 0,2 quand les documents ont toujours 5 termes en commun et 95 termes différents. La courbe des Scores SimRank décroît avec une pente proche de l'horizontale de 0,516 à 0,475. Cela semble indiquer que la hauteur du score SimRank lors de la comparaison de deux documents dépend plus de la proportion de termes communs que de la proportion de termes non communs. Ce qui est une différence notable avec la mesure Cosinus où plus la proportion de termes non communs augmente plus le score Cosinus est faible.

3.2 Influence de la pondération sur la similarité entre deux documents

L'objectif de cette partie est de constater l'influence de la pondération des termes sur la similarité entre deux documents. Pour cela, nous allons utiliser deux documents de 5 termes qui possèdent un terme en commun. Nous allons faire varier le poids de ce terme de 1 à 10 dans l'un des deux documents et regarder l'influence produite sur les scores SimRank et Cosinus :

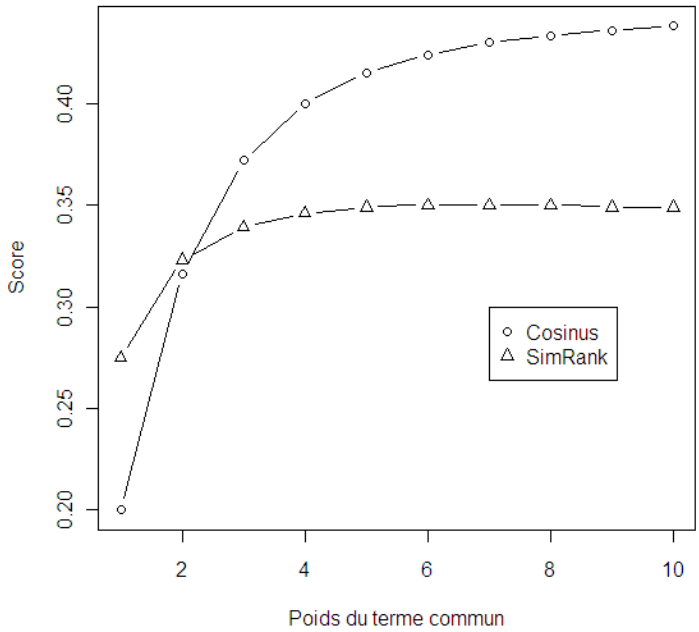


Figure 44 : évolution du score SimRank et du score Cosinus entre deux documents en fonction du poids du terme commun

La figure 44 présente l'évolution des scores SimRank et Cosinus en fonction du poids du terme qu'ils ont en commun. Le premier constat est que la courbe des scores SimRank n'est pas continuellement croissante. En effet elle croît quand le poids du terme commun évolue jusqu'à 5 où elle décroît légèrement. Il semble que quand le poids attribué à un terme partagé dépasse le nombre total des poids des termes dans le document, la pondération n'ait plus d'influence sur le score SimRank. Le score Cosinus croît continuellement, mais cette croissance est de moins en moins importante au fur et à mesure que le poids du terme commun augmente.

Nous allons maintenant reprendre cet exemple, utiliser deux documents de 5 termes et faire varier le poids d'un terme présent dans un seul des deux documents et constater l'évolution des scores SimRank et Cosinus lorsque le poids du terme présent dans un seul des deux documents varie de 1 à 10.

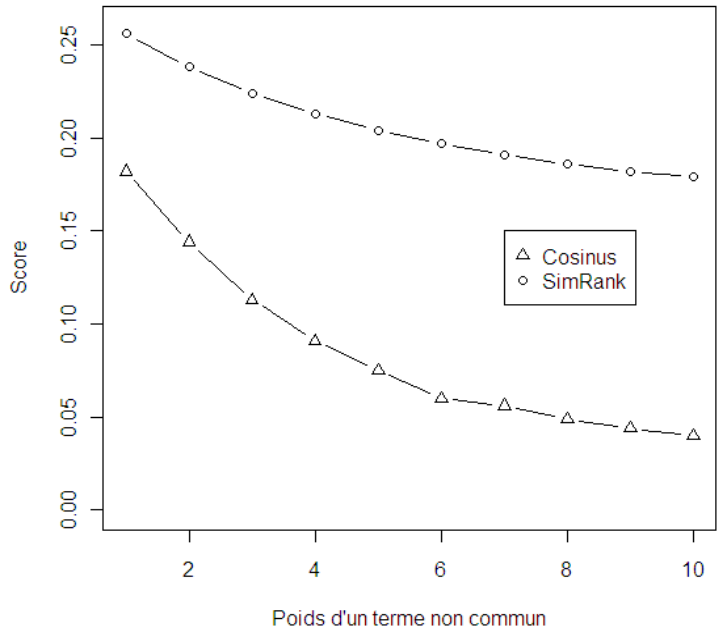


Figure 45 : évolution du score SimRank et du score Cosinus entre deux documents en fonction du poids d'un terme appartenant à un seul des deux documents

La figure 45 présente l'évolution des scores SimRank et Cosinus en fonction du poids d'un terme qu'ils n'ont pas en commun. Les deux courbes décroissent continuellement, la variation est moins importante pour la courbe des scores SimRank qui passe de 0,26 quand les termes non communs sont pondérés à 1 à 0,18 quand un des termes non communs est pondéré à 10. La courbe du score Cosinus passe de 0,18 à 0,04. La pente de la courbe des scores Cosinus est plus importante quand le poids du terme non commun est inférieur à 5, à partir de cette valeur elle décroît plus lentement. En conclusion, la pondération d'un terme commun augmente la similarité entre deux documents et la pondération d'un terme non commun diminue la similarité entre deux documents.

3.3 Illustration des propriétés de propagation des similarités du SimRank

L'objectif de cette partie est d'illustrer une propriété de notre algorithme : la propagation des similarités.

Ainsi, nous allons constituer un ensemble de cinq documents numérotés de d_1 à d_5 possédant chacun deux termes. Les cinq documents partagent deux à deux un seul terme. Cet exemple est illustré par la figure 46 :

	t_1	t_2	t_3	t_4	t_5	t_6
d_1	1	1	0	0	0	0
d_2	0	1	1	0	0	0
d_3	0	0	1	1	0	0
d_4	0	0	0	1	1	0
d_5	0	0	0	0	1	1

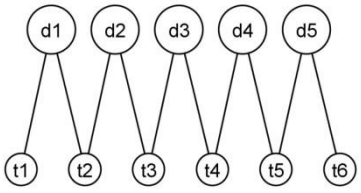


Figure 46 : matrice et graphe représentant cinq documents liés deux à deux par un terme commun

Nous allons calculer les scores SimRank et les scores Cosinus de chaque couple formé par d_1 et un autre document. Les scores sont :

$s(d_1,d_1)= 0,789$ $\cos(d_1,d_1)= 1$
$s(d_1,d_2) = 0,447$ $\cos(d_1,d_2) = 0,5$
$s(d_1,d_3) = 0,184$ $\cos(d_1,d_3) = 0$
$s(d_1,d_4) = 0,076$ $\cos(d_1,d_4) = 0$
$s(d_1,d_5) = 0,038$ $\cos(d_1,d_5) = 0$

On a $s(d_1,d_1) > s(d_1,d_2) > s(d_1,d_3) > s(d_1,d_4) > s(d_1,d_5)$ et $\cos(d_1,d_1) > \cos(d_1,d_2) > \cos(d_1,d_3) = \cos(d_1,d_4) = \cos(d_1,d_5) = 0$.

Si le document d_1 est considéré comme une requête et les documents d_2, d_3, d_4 et d_5 comme des documents susceptibles d'être pertinents pour la requête d_1, le seul score Cosinus non nul est celui de d_1 avec d_2. Cela signifie que le seul document pertinent par rapport à d_1 au sens du Cosinus est d_2. Le score SimRank attribué à chacun des cinq documents indique qu'ils sont tous pertinents par rapport à d_1 à des degrés plus ou moins importants. La mesure Cosinus ne permet pas de faire une différence entre d_3, d_4 et d_5 alors que la mesure SimRank le permet.

Considérons maintenant le document d_2 comme une requête et les documents d_1, d_3, d_4 et d_5 comme susceptibles d'être pertinents pour cette requête. Les scores SimRank et Cosinus des couples de documents formés avec d_2 sont :

$s(d_2,d_2)= 0,702$ $\cos(d_2,d_2)= 1$
$s(d_2,d_1) = 0,447$ $\cos(d_2,d_1) = 0,5$
$s(d_2,d_3) = 0,390$ $\cos(d_2,d_3) = 0,5$
$s(d_2,d_4) = 0,155$ $\cos(d_2,d_4) = 0$
$s(d_2,d_5) = 0,076$ $\cos(d_2,d_5) = 0$

Dans ce cas de figure, la mesure Cosinus considère les documents d_1 et d_3 comme identiquement pertinents pour d_2 et considère d_4 et d_5 comme non pertinents. La mesure SimRank offre des résultats différents : d_1 est considéré comme le plus pertinent par rapport à d_2, devant d_3, lui-même devant d_4, lui-même devant d_5.

Ces résultats amènent un commentaire au sujet de la différence de score SimRank entre les couple (d_2, d_1) et (d_2, d_3) (couples ayant un score Cosinus identique). Le fait que d_1 obtienne un score SimRank avec d_2 plus élevé que le score SimRank de d_2 avec d_3 est dû au fait que d_3 est lié à d_4, lui-même lié à d_5 et que tous deux (d_4 et d_5) ne partagent pas de termes avec d_2. En fait d_3 fait profiter d_4 de sa ressemblance directe avec d_2 mais, en retour, amoindrit sa ressemblance avec d_2. C'est le lien à des documents non directement pertinents qui fait que le SimRank différencie d_1 et d_3 dans leur similarité avec d_2. Cela se reproduit pour d_5 : d_5 reçoit de la ressemblance avec d_2 par l'intermédiaire de d_4. Cela a pour effet de faire augmenter la similarité entre d_5 et d_2, au détriment de d_4 dont la ressemblance à d_2 diminue par rapport au cas où d_4 n'aurait pas de lien vers d_5.

Cet exemple illustre la propagation des similarités : le graphe représenté à la figure 46 fournit les informations suivantes : d_2 ressemble à d_1 et à d_3, d_3 ressemble à d_2 et d_4, d_4 ressemble à d_3 et d_5.
Le Cosinus peut être utilisé pour quantifier cette ressemblance : ici le Cosinus de chaque couple ayant un terme en commun est égal à 0,5 ce qui indique que les documents comparés ont la moitié de leurs attributs en commun.

En traitant cet exemple dans un cadre classique de RI – cf. chapitre 2 -, où l'on compare un ensemble de documents à une requête, un seul des documents de l'exemple sera considéré comme

une requête et le Cosinus ne sera calculé que par rapport à lui. Sur cet exemple, le Cosinus trouvera alors seulement deux documents similaires dans le meilleur des cas. Par exemple si la requête est d_2 d_3 ou d_4, alors elle est liée par un terme à deux autres documents par opposition à d_1 et d_5 qui n'ont qu'un seul voisin.

Le Cosinus pourrait déterminer la ressemblance des documents retrouvés avec les documents restants, par exemple si d_3 est la requête, le Cosinus trouve d_2 et d_4 comme pertinents. En utilisant à nouveau le Cosinus, on trouve que d_2 ressemble à d_1, que d_4 ressemble à d_5 et ainsi on peut considérer que d_1 et d_5 ressemblent un peu à d_3. C'est précisément pour effectuer ce genre d'analyse que nous pensons que notre algorithme peut être efficace pour la RI. Sur l'exemple de la figure 46, le SimRank permet de dire, toujours dans le cas où d_3 est la requête : d_3 ressemble à d_2 et d_4, d_2 ressemble aussi à d_1, d_4 ressemble aussi à d_5 alors d_1 et d_5 ressemblent tous deux à d_3.

L'analyse structurelle réalisée par le SimRank nous permet donc de quantifier les relations indirectes qui lient les documents entre eux.

3.4 Comparaison d'une mesure structurelle à une mesure de similarité basée sur les attributs

Il apparaît que le SimRank retrouve les documents liés à la requête, que le lien soit direct ou indirect. Pour réaliser cela, l'algorithme propage les similarités entre les documents. Cette propagation est réalisée par un algorithme convergeant.

On peut se demander par rapport aux exemples précédents si le SimRank est une sorte de Cosinus global, en ce sens où il peut être équivalent à calculer le Cosinus de chaque document avec une requête, puis calculer le Cosinus de chaque document retrouvé avec les documents non retrouvés, puis calculer le Cosinus des documents retrouvés à l'étape précédente avec ceux non retrouvés, jusqu'à ce qu'aucun document ne soit retrouvé. Si tel est le cas, alors notre méthode retrouve les documents directement pertinents d'abord, à la suite desquels elle retrouve les documents directement ressemblant à eux (parmi les documents non encore retrouvés), à la suite desquels elle retrouve ceux directement ressemblants à eux,… Cette idée peut être infirmée par l'exemple suivant qui montre que le SimRank peut attribuer un score plus élevé à un document n'ayant aucun terme commun avec la requête qu'à un document ayant des termes communs avec la requête.

Soit un corpus composé de trois documents et une requête.
La requête *R* est composée de cinq termes : t_1, t_2, t_3, t_4, t_5.
d_1 est composé de cinq termes : t_1, t_2, t_3 qu'il partage avec la requête, avec en plus t_6 et t_7.
d_2 est composé de quatre termes : t_4 et t_5 qu'il partage avec la requête, avec en plus t_8 et t_9
d_3 est composé de trois termes : t_6 et t_7 qu'il partage avec d_1 et t_8 qu'il partage avec d_2.
t_1, t_2 et t_3 ont un poids de 2, les autres termes ont un poids de 1.

Cet exemple est illustré par la figure suivante :

	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9
d_1	1	1	1	0	0	1	1	0	0
d_2	0	0	0	1	1	0	0	1	1
d_3	0	0	0	0	0	1	1	1	0
R	2	2	2	1	1	0	0	0	0

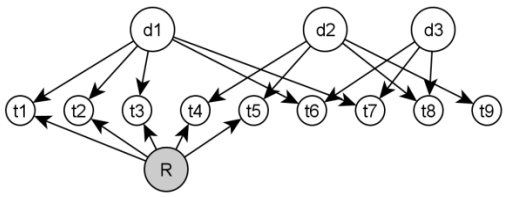


Figure 47 : matrice et graphe représentant trois documents et une requête

Les scores SimRank et les scores Cosinus que chaque document obtient avec la requête fournissent les résultats suivants :

$s(R,d_1)= 0,421$ $\cos(R,d_1)= 0,717$
$s(R,d_2) = 0,325$ $\cos(R,d_2) = 0,267$
$s(R,d_3) = 0,247$ $\cos(R,d_3) = 0$

Ainsi $s(R,d_1)> s(R, d_2) > s(R,d_3)$ et $\cos(R,d_1) > \cos(R,d_2) > \cos(R,d_3)$.

Sur cet exemple, le SimRank et le Cosinus ordonnent tous deux les documents de la même manière, d_1 est le plus pertinent devant d_2, lui-même devant d_3.

d_3 n'ayant aucun terme commun avec la requête obtient un score Cosinus de 0. Le score SimRank de d_3 est de 0,27, il est dû à la ressemblance directe de d_3 avec d_1 et d_2, et à la ressemblance directe de d_1 et d_2 avec *R*.

Lorsque le poids du terme t_9 augmente jusqu'à 10 les résultats sont :

$s(R,d_1)= 0,409$ $\cos(R,d_1)= 0,717$
$s(R,d_2) = 0,221$ $\cos(R,d_2) = 0,05$
$s(R,d_3) = 0,225$ $\cos(R,d_3) = 0$

Avec $s(R,d_1)> s(R, d_3) > s(R,d_2)$ et $\cos(R,d_1) > \cos(R,d_2) > \cos(R,d_3)$.

L'ordre de pertinence des documents par rapport à la requête n'est plus identique pour les deux méthodes :

Le Cosinus conserve le même ordre que précédemment quand le poids de t_9 valait 1.

Le SimRank considère maintenant d_2 comme plus pertinent par rapport à R que d_3, malgré le fait que d_3 ait deux termes directement en commun avec *R*. Cet exemple illustre que le SimRank et le Cosinus n'ordonnent pas forcément les documents de la même manière.

Conclusion du chapitre Modèle théorique

Dans cette partie nous avons décrit l'adaptation d'une méthode de comparaison d'objets basée sur les graphes à la RI. Cette adaptation s'est traduite par la définition d'une nouvelle fonction de similarité tenant compte de la structure du graphe induit par la relation entre documents et termes d'indexation. Un document est vu conceptuellement comme le nœud d'un graphe bipartite auquel sont connectés les nœuds représentant les termes qui l'indexent. La similarité entre documents est calculée comme la moyenne des similarités des termes qui les composent. Réciproquement, la similarité entre termes est calculée comme la moyenne des similarités entre les documents qui les contiennent.

A partir de cette définition récursive, nous avons défini deux formules : l'une définissant la similarité entre documents, l'autre définissant la similarité entre termes.

Nous avons montré que ces formules convergent. Ceci nous a permis de définir une mesure de similarité structurelle inter-documents et inter-termes : le SimRank. Ce dernier est défini comme la limite des suites des valeurs de similarité entre documents et entre termes. Nous avons intégré la prise en compte de la pondération des termes dans les documents, ce qui se traduit conceptuellement par une pondération des arcs du graphe représentant les documents et les termes. Nous avons également écrit notre algorithme de calcul des similarités sous forme matricielle et étudié sa complexité.

Nous avons illustré les propriétés au travers d'exemples simples. Le SimRank ordonne des documents en fonction de leur similarité structurelle. Lorsque deux documents sont comparés à l'aide du SimRank, on constate que plus ils ont de termes en commun, plus leurs scores sont élevés. Cela va dans le sens de la philosophie vectorielle qui se base principalement sur l'idée que la comparaison de vecteurs dans un espace vectoriel dépend de la proportion de dimensions communes entre les vecteurs comparés. De la même manière, plus les documents comparés ont des termes dissemblables, plus leurs scores sont faibles. Notre méthode ne va donc pas à l'encontre de ces préceptes, ce qui est positif pour l'usage souhaité à savoir le tri de documents par rapport à une requête.

Les exemples suggèrent également que le SimRank est sensible à la pondération, plus les termes communs (respectivement non-communs) à deux documents sont pondérés, plus ils améliorent (respectivement diminuent) le score de ressemblance entre ces deux documents. Ce qui est positif pour l'usage souhaité.

Une propriété intéressante qui se dégage de notre algorithme est la notion de propagation des similarités qui permet qu'un document puisse être considéré comme ressemblant à un autre sans pour autant partager de termes avec celui-ci. Un deuxième effet de la propagation des similarités est que plus de documents sont considérés pertinents par rapport à une requête donnée qu'avec une méthode de type Cosinus. En effet le SimRank retrouve comme le Cosinus les documents ayant des termes en commun avec la requête, auxquels il ajoute ceux ayant des liens avec eux. Si de tels documents n'existent pas alors le SimRank et le Cosinus retournent le même nombre de documents. D'un point de vue graphe, le SimRank retrouve tous les documents de la composante connexe du sous-graphe contenant la requête. De plus, il est capable d'ordonner ces documents par rapport à leur ressemblance (structurelle) avec la requête.

Ces différentes propriétés nous permettent d'envisager d'utiliser notre algorithme pour répondre au problème de la RI, à savoir trier des documents en fonction de leur ressemblance à une requête. Reste à définir en situation réelle le comportement de notre algorithme.

Notre algorithme SimRank utilise cinq paramètres :

- La valeur attribuée au score de ressemblance entre deux objets identiques,
- La valeur attribuée au score de ressemblance entre deux objets différents,
- Le nombre d'itérations,
- Le coefficient modulant la ressemblance entre termes dans le calcul de la ressemblance entre documents (noté C_1),
- Le coefficient modulant la ressemblance entre documents dans le calcul de la ressemblance entre termes (noté C_2).

Comme lors des calculs dans les exemples précédents nous avons dû faire un choix au niveau du paramétrage de l'algorithme.

Concernant la valeur de similarité attribuée entre deux objets (documents ou termes) identiques, il nous a semblé logique de choisir 1, la similarité maximale sur une échelle de 0 à 1. Cette valeur à son importance car elle correspond à la similarité attribuée au couple formé par un objet et lui-même. C'est cette valeur qui à chaque itération permet de propager les similarités jusqu'à convergence de l'algorithme.

Concernant la similarité entre deux objets différents, nous avons, dans le doute choisi de l'initialiser à 0. Ce qui traduit l'idée d'une différence maximale. Ce paramètre concerne principalement l'initialisation de notre algorithme, car quand le calcul est lancé, la similarité entre deux objets différents est égale à la moyenne des similarités des objets auxquels ils sont reliés et peut donc être non nulle.

Concernant les constante C1 et C2, nous avons choisi de les initialiser à 0,9 de manière à accorder, pour une première étude, autant d'importance à la relation d'un document vers un terme qu'à celle d'un terme vers un document.

Nous aurons à déterminer expérimentalement le nombre d'itérations nécessaires pour atteindre la convergence. En effet, bien que la convergence soit prouvée, il est nécessaire de vérifier expérimentalement la vitesse de convergence de notre algorithme sur des données réelles, de manière à déterminer si notre algorithme, qui semble utilisable pour le tri de documents par rapport à une requête est ou n'est pas viable en pratique.

Nous effectuerons également des tests sur les constantes C1 et C2 pour examiner leur influence sur des mesures données (précision à 10 documents restitués, mesure F et MAP) afin de déterminer si nos choix de départ sont valables. Enfin nous procéderons à l'évaluation des résultats obtenus par notre méthode sur les collections Cisi, Cranfield et TREC. Nous comparerons ces résultats à deux méthodes : Cosinus et Okapi.

Chapitre 5 : Expérimentations

L'étude de notre algorithme au chapitre précédent montre que celui-ci permet de comparer des documents sur la base de leurs relations. La comparaison que nous utilisons présente plusieurs propriétés :

- Le score de similarité attribué à un couple de documents augmente quand le nombre de termes communs entre les deux documents augmente,
- Ce score est sensible au poids des termes indexant les documents : une pondération élevée d'un terme commun à deux documents accentue leur ressemblance et de la même manière une pondération élevée de termes non communs aux documents atténue leur ressemblance,
- Le calcul du score de similarité exploite les relations entretenues entre les documents (deux documents entretiennent une relation s'ils ont des termes en commun directement ou s'ils ont des termes communs avec un ou plusieurs documents possédant des termes communs aux deux documents). Cela a pour effet d'identifier comme similaires des documents possédant peu ou ne possédant pas de termes communs avec le document comparé.

Les deux premières propriétés vont dans le sens de la philosophie des modèles vectoriels.

La mesure Cosinus, qui est la mesure utilisée traditionnellement pour déterminer la similarité d'un couple de documents, traduit la proportion de termes communs aux documents comparés. Plus les termes communs sont nombreux et plus le poids de ces termes est élevé, plus le score Cosinus du couple dans l'espace des termes d'indexation est élevé ; il en est de même pour notre mesure.

La troisième propriété traduit l'idée que des documents structurellement liés peuvent être similaires. C'est cette propriété qui constitue à notre sens le principal avantage de notre méthode par rapport à une approche vectorielle uniquement basée sur la similarité entre attributs de surface. Nous pensons que l'information apportée par les similarités structurelles présente un intérêt indéniable pour la RI.

Après avoir illustré notre algorithme sur des exemples simples, nous allons l'appliquer aux requêtes des collections de référence. Comme nous l'avons vu dans le chapitre 2, ces collections fournissent un ensemble de requête et un ensemble de documents qui permettent aux concepteurs de SRI d'évaluer leurs systèmes sur une base commune. De plus il existe différentes mesures telles que la MAP, la précision, le rappel ou la mesure F qui permettent d'évaluer la liste des documents pertinents retrouvés par telle ou telle méthode. Nous utiliserons ces mesures pour évaluer les retours de notre méthode et nous comparerons les résultats obtenus aux résultats obtenus par les systèmes basés sur Cosinus et Okapi sur les mêmes données.

Avant de comparer notre méthode au Cosinus et à Okapi, il est nécessaire d'étudier le comportement de notre algorithme sur ces nouvelles données afin éventuellement d'adapter les différents paramètres. Il faut également s'intéresser au nombre d'itérations nécessaires pour obtenir la convergence dans la pratique. En effet, le calcul du score de similarité structurelle entre deux documents - cf. chapitre 4 - consiste en l'application d'un algorithme itératif de propagation des similarités par les relations document-terme entretenues par les documents à comparer. Le score de similarité entre deux documents est obtenu après convergence de l'algorithme. Le nombre d'itérations nécessaires pour atteindre la convergence est fonction des données traitées. C'est pourquoi nous commencerons par une étude de la vitesse de convergence de notre algorithme sur les collections Cisi et Cranfield. Ensuite, nous chercherons à déterminer l'influence des valeurs des paramètres C1 et C2 des formules (13) et (14) sur les performances de notre système. Enfin nous comparerons les résultats obtenus par notre méthode à ceux obtenus par les méthodes basées sur Cosinus et Okapi.

1. Evaluation sur de petites collections

La complexité de notre algorithme nous permet actuellement d'envisager un usage sur des collections ayant moins de 10 000 documents. Cette limite est due au dépassement de la mémoire d'exécution (2Go) de l'application qui implémente notre algorithme. Les collections Cranfield et CISI correspondent à cette contrainte.

Cisi est une collection composée de 1460 documents et 112 requêtes. Dans les expérimentations suivantes nous ne tiendrons compte que des 76 requêtes ayant des documents pertinents.

La collection Cranfield est composée de 1400 documents et 225 requêtes.

	Cisi	Cranfield
Nombre de documents	1460	1400
Nombre de termes index	5649	4381
Nombre de requêtes	112	225
Nombre moyen de termes par requête	27,7	9
Nombre de documents pertinents par requête	27,8 41 (76 documents)	8

Tableau 1 : caractéristiques des collections CISI et Cranfield

Le tableau 1 présente différentes caractéristiques des collections étudiées. On peut remarquer quelques similitudes comme le nombre de documents et le nombre de termes index ainsi que quelques différences comme le fait que les requêtes ont une taille supérieure pour CISI et un nombre moyen de documents pertinents par requête également plus élevé pour CISI que pour Cranfield.

Comme nous l'avons vu dans le chapitre précédent, les données étudiées sont représentées par un graphe bipartite. Il est intéressant d'examiner les caractéristiques des graphes représentant ces données.

	Cisi	Cranfield
Nombre de liens	65920	74872
Degré moyen des documents	45.1	53.5
Degré moyen des termes	11.6	17.0
Degré moyen du graphe monopartite associé[19]	18.5	25.9
Coefficient de regroupement des documents	0.00798	0,0122
Coefficient de regroupement des termes	0,00794	0,0121
Coefficient de regroupement du graphe monopartite associé	0,00260	0,0044
Densité	0.00799	0.0122
Diamètre	6	6

Tableau 2 : caractéristiques des graphes représentant les collections CISI et Cranfield

[19] Le graphe monopartite associé au graphe bipartite représentant une collection est le même graphe mais ses nœuds ne sont pas différenciés

Le tableau 2 indique que le graphe représentant la collection Cranfield est plus lié que le graphe représentant la collection CISI. D'une part il est plus dense et d'autre part le degré moyen des documents (le nombre moyen de termes par document) et le degré moyen des termes (le nombre moyen de documents où apparaît le terme) sont plus élevés pour Cranfield que pour CISI. Cela est visible également en examinant les degrés du graphe monopartite associé respectivement à chacune des collections.

La densité globale des deux graphes est faible. Chaque document est connecté à environ 50 termes (degré moyen des documents) et chaque terme est connecté en moyenne à 15 documents. Le diamètre des deux graphes représentant les collections est de 6. Cela nous conforte quant à l'idée d'utiliser notre algorithme de propagation dans ce genre de réseau, en effet le nombre d'itérations nécessaires pour atteindre tous les documents à retrouver ne devrait pas excéder le diamètre. Nous pensons que notre algorithme doit atteindre les documents liés à la requête avant de converger. Dans le pire des cas, la requête concerne indirectement le document le plus éloigné d'elle (à distance égale au diamètre), notre algorithme doit donc propager les similarités indirectes jusqu'à lui (à chaque itération il se rapproche d'un nœud sur le chemin qui en compte sept au maximum) et certainement continuer quelques itérations pour enfin atteindre la convergence et attribuer des scores SimRank tels que définis par les formules (13) et (14) du chapitre 4.

1.1 Etude de la vitesse de convergence

Le SimRank d'un couple de documents est défini comme la moyenne des SimRank des couples de termes apparaissant dans les documents. Les scores SimRank entre documents et les scores SimRank entre termes constituent des suites convergentes – cf. chapitre 4. Le score final retenu est obtenu après convergence de ces suites. Pour étudier la convergence de l'algorithme, nous avons réalisé 20 itérations pour chacune des requêtes des deux collections étudiées. Nous avons retenu un certain nombre de critères permettant l'étude de la convergence. Ainsi, nous regardons à chaque itération les indicateurs suivants :

- La moyenne des écarts en valeur absolue de chaque score document/requête courant avec le score document/requête à l'itération suivante (delta moyen).
- La moyenne des valeurs absolues des écarts du score du document convergeant le moins vite pour une requête donnée avec ce score à l'itération suivante (delta max).
- Le score maximum obtenu par un document (score du 1er document).
- Le nombre de différences dans l'ordre des scores à une itération donnée et aà l'itération suivante (différences). Ce nombre est calculé de la façon suivante : à l'issue d'une itération la liste des documents retournés est examinée, le rang des documents retrouvés est mémorisé. A l'itération suivante le compteur des documents qui ont changé de rang est incrémenté.

1.1.1 Etude des écarts absolus des scores de similarité document/requête en fonction des itérations

Les graphiques suivants contiennent deux courbes : delta moyen et delta max en fonction des itérations.

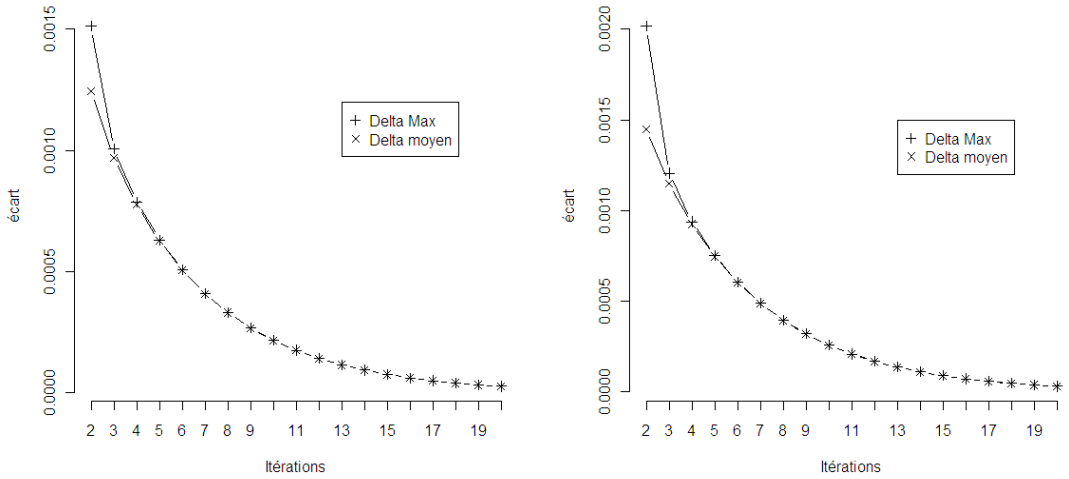


Figure 48 : moyenne des écarts absolus entre les scores en fonction des itérations pour les 76 requêtes de CISI (à gauche) et pour les 225 requêtes de Cranfield (à droite)

Delta moyen représente la moyenne des écarts entre un score document/requête donné et ce même score à l'itération suivante, cela pour tous les scores documents/requêtes.

Delta max représente la moyenne des écarts entre le score du document qui a eu la plus grande variation de score et ce même score à l'itération suivante. La courbe associée représente l'évolution de l'écart entre le score qui varie le plus par rapport au score initial et lui-même à l'itération suivante.

Nous aurions pu étudier l'évolution des écarts relatifs. L'écart relatif entre le score obtenu à une itération donnée et à l'itération suivante est calculé en divisant la valeur de l'écart par le score, ce qui permet de ramener l'écart à la proportion du score. Nous avons montré que les suites des scores convergent – cf. chapitre 4 -, c'est-à-dire que les scores tendent vers le score limite. Quand ce score limite est proche d'être atteint, la suite des écarts relatifs et la suite des écarts absolus se comportent de la même manière. Si l'une converge alors l'autre aussi.

Ces graphiques montrent que notre algorithme converge rapidement. Le fait que la courbe des variations du score document/requête qui varie le plus soit confondue avec la courbe des variations moyennes nous laisse penser que les scores convergent de façon homogène, bien qu'il puisse y avoir des exceptions.

1.1.2 Etude du score qui varie le plus et du rapport avec sa variation en fonction des itérations

Nous nous intéressons dans cette section à l'étudie de la convergence des scores SimRank d'un point de vue numérique. Le rapport Delta max / Score max permet de ramener l'écart du document dont les scores varient le plus à la hauteur du score qui varie le plus ce qui nous permet de vérifier si plus de 10 itérations sont nécessaires pour atteindre la convergence comme semble l'indiquer la figure 48.

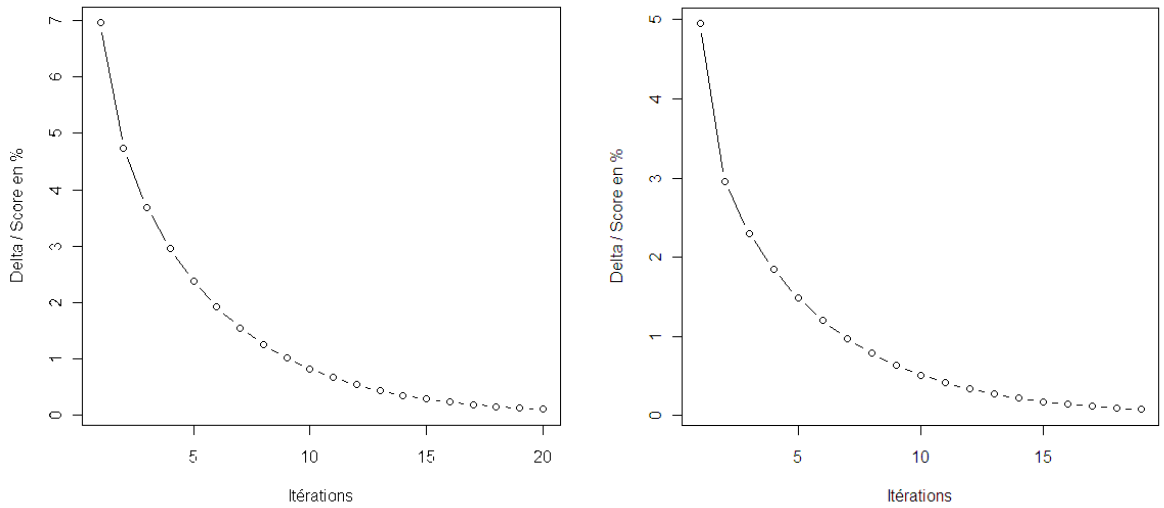


Figure 49 : évolution du rapport Delta max / Score max en fonction des itérations pour CISI (à gauche) et pour Cranfield (à droite)

La figure 49 montre également que notre algorithme converge, l'écart entre un score et lui-même à l'itération suivante est inférieur à 1% de ce score dès la 10ème itération pour CISI et à la 7ème pour Cranfield.

L'évolution moyenne des scores du 1er document nous permet de proposer un critère d'arrêt basé sur l'évolution du score – voir figure 50.

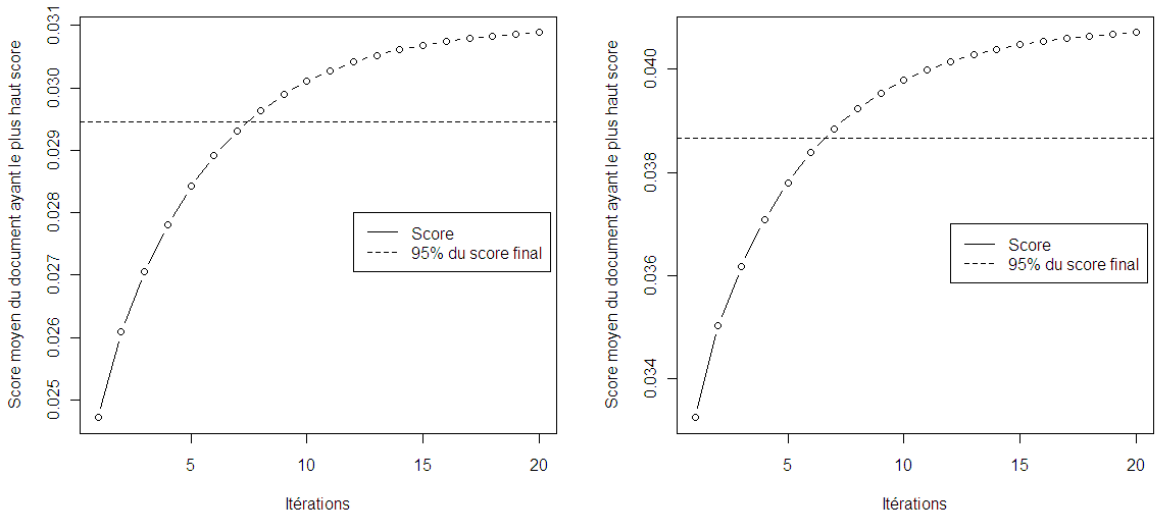


Figure 50 : évolution moyenne du plus haut score document/requête en fonction des itérations pour 76 requêtes de CISI (à gauche) et pour les 225 requêtes de Cranfield (à droite)

Ce qui nous intéresse est de déterminer une condition d'arrêt en limitant les risques de tomber sur des minima/maxima locaux. Dans cette optique, nous pouvons nous baser sur l'évolution des scores pour stopper les itérations. La figure 44 du rapport Delta max / Score max indique que l'écart entre un score et lui-même à l'itération suivante est inférieur à 1 % de ce score à partir de la 10ème itération. La figure 50 montre l'évolution du score du document ayant eu le meilleur score pour une

requête donnée. Ce score dépasse 95% de sa valeur finale dès la 8ème itération pour CISI et dès la 7ème pour Cranfield.

Ces deux conditions éventuelles d'arrêt concernent la convergence numérique des scores SimRank et peuvent être envisagées comme condition d'arrêt sur critère mathématique. Pour la RI, les mesures choisies pour évaluer les systèmes de recherche sont basées sur le l'ordre des documents. Il est donc préférable d'étudier l'évolution de l'ordonnancement au fil des itérations pour déterminer une condition d'arrêt expérimentale car ce qui nous intéresse le plus n'est pas réellement la convergence des scores mais plutôt l'impact des scores sur l'ordre des documents. Il est possible que, dans certaines conditions, certains calculs de score de similarité document / requête nécessitent plus d'itérations que le nombre que nous allons choisir. Cela est moins grave qu'il n'y paraît, en effet si la convergence n'est pas réellement atteinte, alors deux documents continuent d'alterner leurs positions. S'ils alternent leurs positions alors leurs scores SimRank respectifs sont très proches, ce qui a un impact minime sur le résultat.

1.1.3 Etude de l'évolution des différences dans l'ordonnancement de documents en fonction des itérations

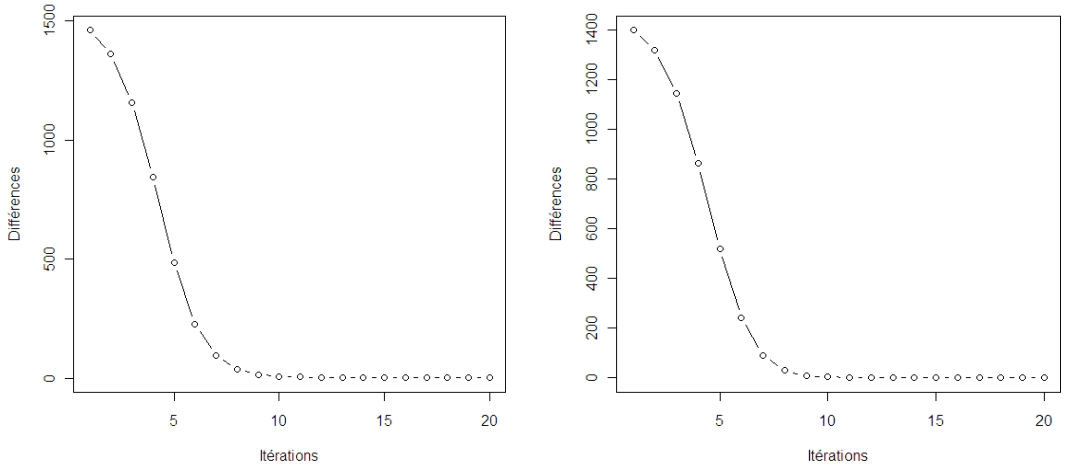


Figure 51 : moyenne des différences entre ordonnancements en fonction des itérations pour 76 requêtes de CISI (à gauche) et pour les 225 requêtes de Cranfield (à droite)

Les graphiques de la figure 51 représentent l'évolution du nombre de différences (de documents classés différemment) dans l'ordonnancement au fil des itérations. Ce nombre de différences est calculé en incrémentant un compteur à chaque fois que le document retourné a changé de rang dans l'ordonnancement précédent. Ainsi, deux ordonnancements peuvent être complètement différents tout en étant très proches. Par exemple l'ordonnancement 1,2,3,4,5 a cinq différences avec 5,1,2,3,4. Nous aurions pu étudier l'évolution des différences de rang, c'est-à-dire examiner la distance en termes de rangs qui séparent un document à une itération et ce même document à l'itération suivante. De la même manière que le nombre de différences dans l'ordonnancement, la différence de rang doit décroître au fur et à mesure que l'on se rapproche de la convergence. Quand la convergence est sur le point d'être atteinte, les documents ne changent pas ou changent peu de rang, les variations de rang sont alors de simples permutations de positions entre un rang donné et ce rang à plus ou moins une position.
Le nombre de différences dans l'ordonnancement évolue du nombre total de documents, quand l'ordonnancement est complètement différent du précédent, à zéro quand les documents ont le même rang dans les deux ordonnancements.

Les différentes figures précédentes indiquent que notre algorithme converge expérimentalement. Il converge rapidement (7-8 itérations), cela montre une possible applicabilité à la RI.

1.2 Etude des constantes de propagation

La constante C1 est un coefficient utilisé à chaque fois que la similarité entre deux documents est calculée. Il est un facteur dont le but est de modérer la moyenne des similarités des termes que contiennent les documents comparés. Réciproquement, la constante C2 est un coefficient utilisé à chaque fois que la similarité entre deux termes est calculée. Il est un facteur dont le but est de modérer la moyenne des similarités des documents dans lesquels les termes apparaissent. Nous avons choisi d'examiner l'influence des constantes de propagation C1 et C2, sur la moyenne des précisions quand 10 documents sont restitués, puis sur la moyenne des MAP et enfin sur la moyenne des meilleures mesure F pour les deux corpus étudiés.

1.2.1 Influence des constantes de propagation sur la précision à 10 documents restitués

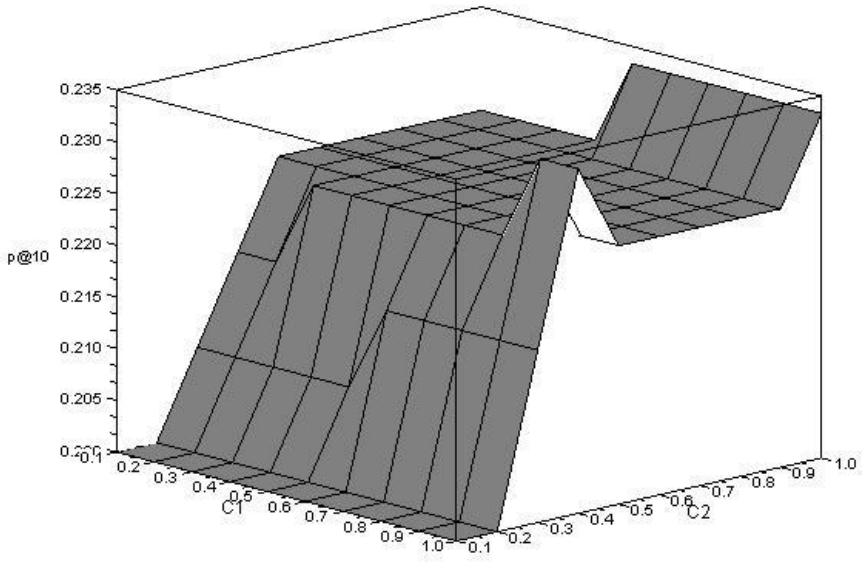


Figure 52 : moyenne des précisions à 10 documents restitués sur Cisi en fonction de C1 et C2

La figure 52 indique que pour maximiser la précision à 10 documents retrouvés sur Cisi, deux plages sont maximales : la première pour C2=0,4 et C1=0,9 ; la seconde pour C2=0,9 et C1 variant de 0,4 à 0,9.

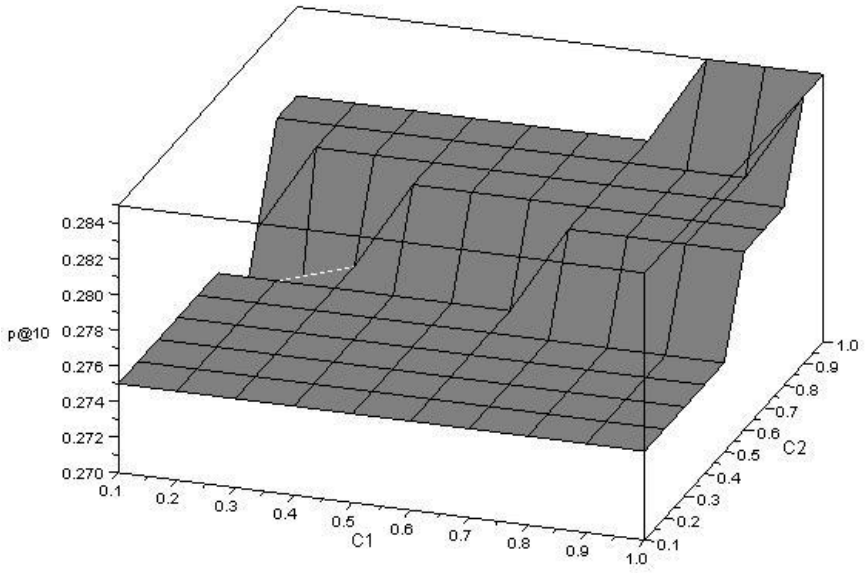


Figure 53 : moyenne des précisions quand 10 documents sont restitués sur Cranfield en fonction de C1 et C2

La figure 53 montre que la précision à 10 documents restitués sur Cranfield est maximale quand C1 et C2 sont élevés (tous deux compris entre 0,9 et 1). Cela est contradictoire avec les résultats sur Cisi. Néanmoins p@10 est sujette à variation : un seul document de plus restitué dans les 10 premiers modifie grandement ce graphique. C'est pourquoi, nous allons regarder la moyenne des MAP qui est une mesure basée sur l'ensemble des documents restitués par une méthode donnée – cf. figure 49 et figure 50.

1.2.2 Influence des constantes de propagation sur la MAP

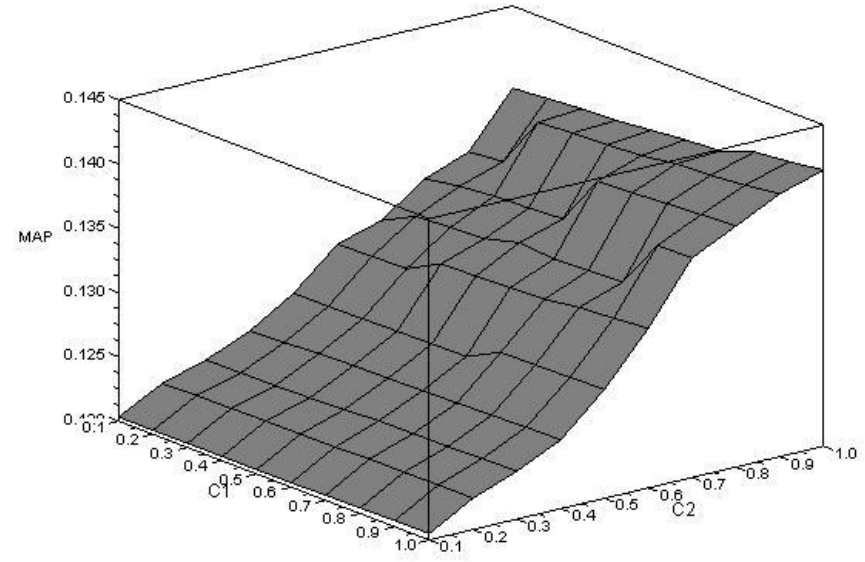


Figure 54 : moyenne des MAP sur Cisi en fonction de C1 et C2

La figure 54 indique clairement que la moyenne des MAP est maximale pour Cisi avec C1 et C2 compris entre 0,9 et 1.

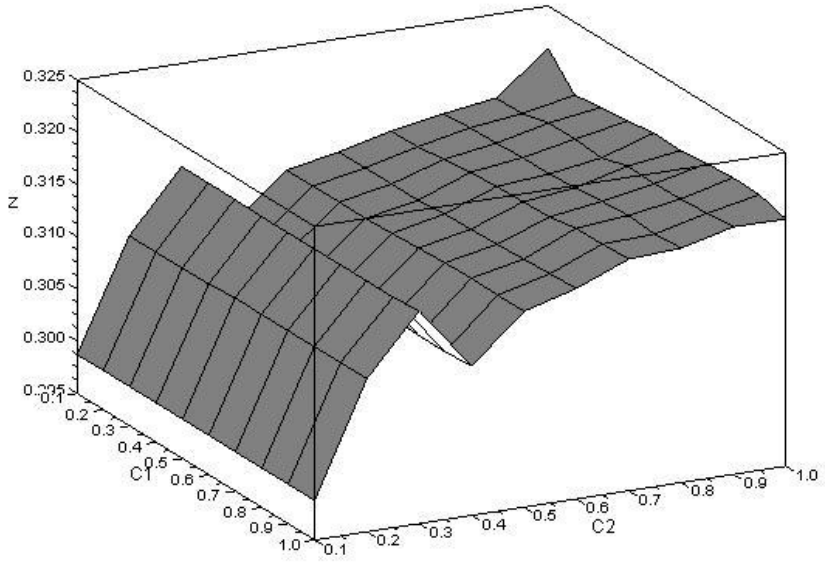


Figure 55 : moyenne des MAP sur Cranfield en fonction de C1 et C2

Comme pour le corpus Cisi, C2 semble avoir plus d'influence sur la MAP moyenne que C1. Ainsi la moyenne des MAP est maximale quand C2 vaut 0,9.

1.2.3 Influence des constantes de propagation sur la mesure F

Nous avons calculé la moyenne des meilleures mesures F en fonction de C1 et C2 pour avoir une idée sur la combinaison qui maximise la mesure F.

Pour obtenir la moyenne des meilleures mesures F, on calcule toutes les mesures F obtenues pour une requête donnée en faisant varier le nombre de documents restitués de 1 au nombre total de documents. On examine alors le nombre de documents restitués maximise la mesure F pour chaque requête. Ce nombre indique la meilleure coupe à effectuer sur l'ordre des documents pour retourner la mesure F la meilleure possible. L'opération est réitérée pour chacune des requêtes. On obtient alors pour chaque requête le nombre de documents à retrouver, puis on calcule la moyenne.

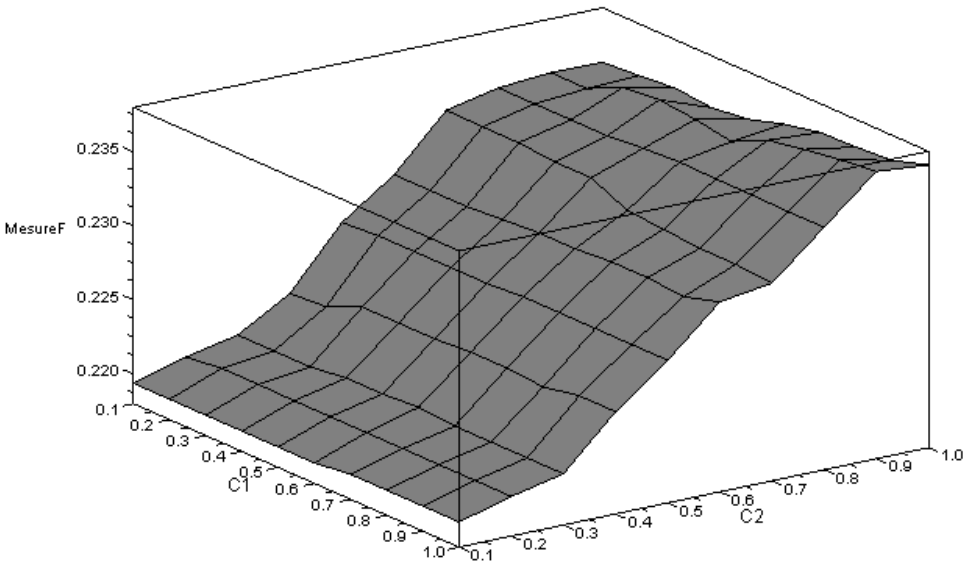


Figure 56 : moyenne des meilleures mesures F sur Cisi en fonction de C1 et C2

La figure 56 montre que pour la mesure F moyenne, C2 influe de façon significative, C1 influe de façon moindre. Comme pour la moyenne des MAP, la meilleure plage est obtenue avec C2=0,9 et C1 compris entre 0,4 et 1.

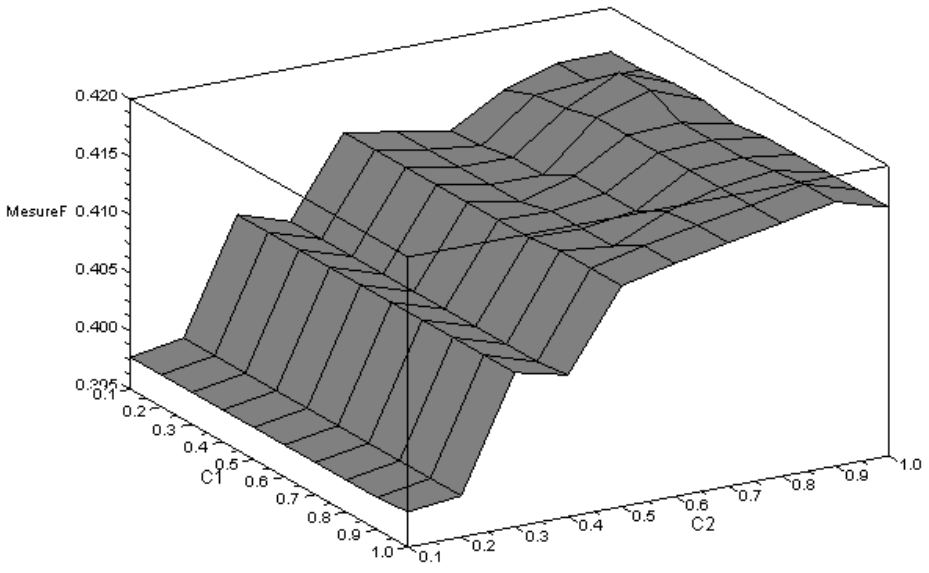


Figure 57 : moyenne des meilleures mesures F sur Cranfield en fonction de C1 et C2

La figure 57 indique, comme la figure 55, que la courbe est maximale pour un C2 élevé.

Les tests des paramètres C1 et C2 réalisés sur les collections Cisi et Cranfield nous laissent penser que l'on peut donner à C2 une valeur comprise entre 0,9 et 1 ; la valeur attribuée à C1 a moins d'importance et peut être comprise entre 0,4 et 1. Une des raisons possible de cela est que les termes sont plus nombreux que les documents et donc que les deux coefficients de propagation ne jouent pas un rôle tout à fait symétrique. Au regard de ces résultats, nos choix concernant les paramétrages de C1 et C2 semblent valables car dans la plage de valeur qui permet d'obtenir les meilleurs résultats.

1.3 Comparaisons avec les mesures Cosinus et Okapi

Nous avons choisi de comparer notre mesure de similarité structurelle à d'autres mesures : la mesure Cosinus, la mesure Okapi BM25. Pour ces évaluations, C1 et C2 sont fixés à 0,95. Le nombre d'itérations effectuées est de 10. Pour réaliser les calculs du Cosinus et de Okapi BM25 nous utilisons la librairie Lemur. Les documents et les requêtes sont radicalisés selon l'algorithme Porter.

Les paramètres utilisés pour Okapi sont : k1=1,2 ; k3= 7 ; b=0,75. – cf. chapitre 4 le modèle probabiliste.

Nous comparons les trois mesures à l'aide de la MAP, des moyennes des précisions à 5, 10, 30, 100 documents restitués, des moyennes des R-précisions et des mesures F à 5, 10, 30, 100 documents restitués.

1.3.1 Résultats obtenus à la *Mean Average Precision*

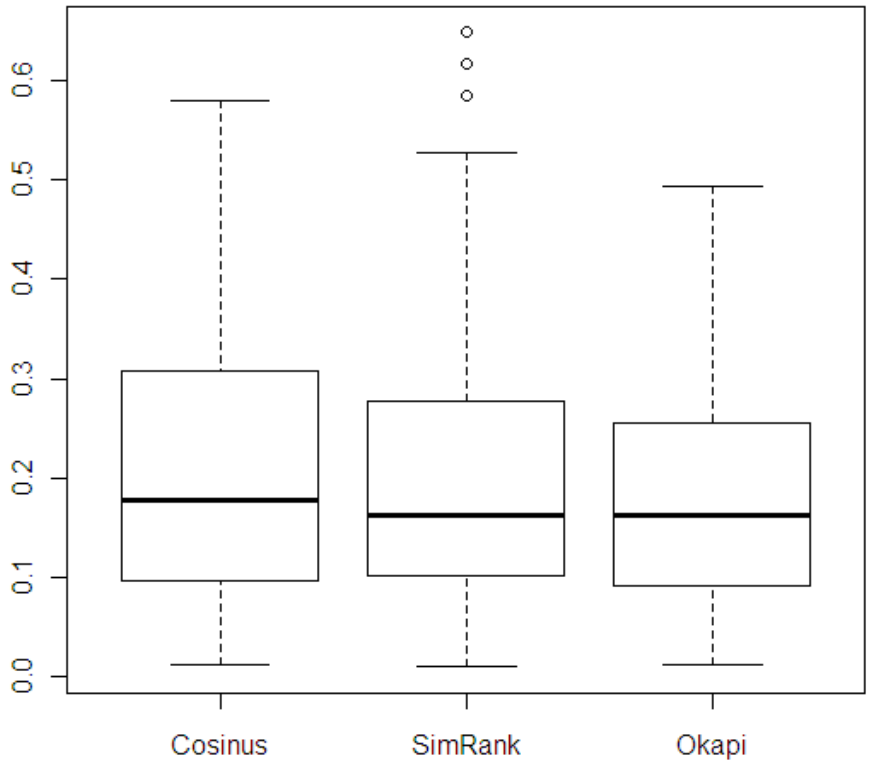


Figure 58 : représentation des MAP pour Cosinus, SimRank et Okapi sur Cisi

La figure 58 permet de mesurer la distribution des MAP pour chacune des trois mesures.

La représentation en boîte à moustaches consiste à représenter une série de données par une ligne verticale en pointillés, sur laquelle se positionne un rectangle séparé en deux par une ligne noire horizontale. La hauteur de la ligne verticale en pointillés informe par sa taille de la distance qui sépare la valeur la plus basse de la série et la valeur la plus haute de la série. Le rectangle indique la répartition de 75 % des valeurs de la série étudiée autour de la MAP moyenne représentée par la ligne noire horizontale. Les ronds éventuels au-dessus des boîtes à moustache représentent des valeurs extrêmes de la série.

La première constatation est que Cosinus obtient la meilleure moyenne, devant SimRank, devant Okapi. Globalement les répartitions sont comparables – les médianes sont très proches (à noter que SimRank obtient trois excellentes MAP) représentées par les trois ronds au-dessus de la boîte à moustaches.

Nous allons maintenant regarder les résultats obtenus sur Cranfied – cf. figure 54.

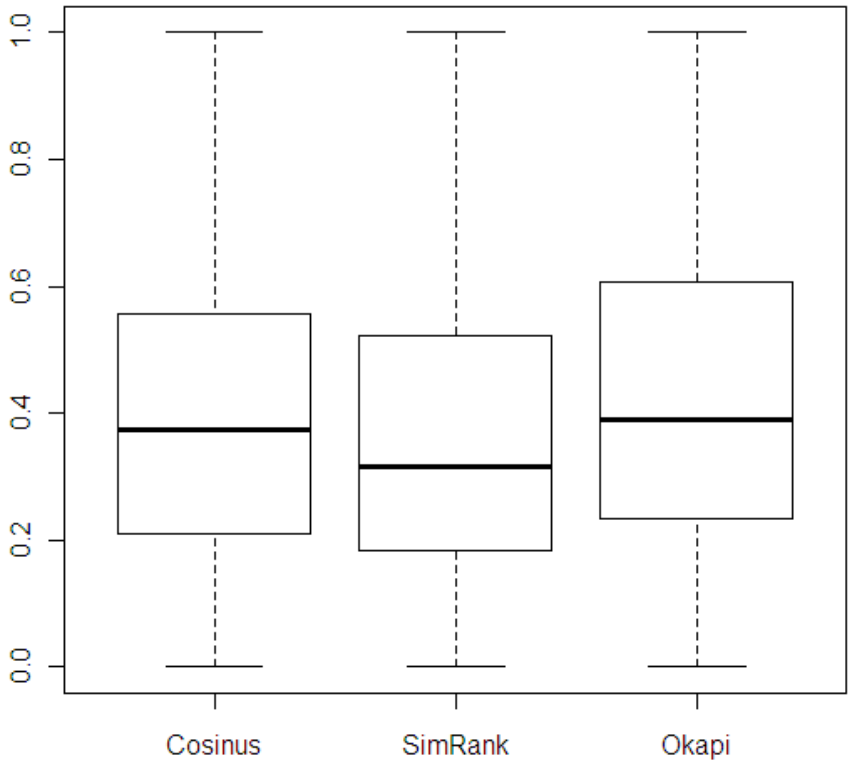


Figure 59 : représentation des MAP pour Cosinus, SimRank et Okapi sur Cranfield

La figure 59 montre que la MAP obtenue par Okapi est en moyenne meilleure que celle obtenue par Cosinus qui elle-même est meilleure que celle obtenue par SimRank. Bien que notre mesure soit inférieure aux mesures témoin, elle obtient des résultats ainsi que les distributions comparables. Une partie de ces résultats a donné lieu à publication [Champclaux et al., 2007].

A titre de comparaison, nous avons reporté des résultats obtenus sur ces mêmes collections par d'autres méthodes : une méthode d'expansion de requête utilisant les algorithmes génétiques (présentée dans [Cummins et al., 2005]) obtient une MAP moyenne de 0,41 sur Cranfield et de 0,23 sur Cisi. Le modèle de [Truong et al., 2008] basé sur les graphes bipartites obtient une MAP moyenne

de 0,31 sur Cranfield et de 0,21 sur Cisi. Les résultats présentés indiquent une MAP moyenne de 0,37 sur Cranfield et de 0,20 sur Cisi pour SimRank.

1.3.2 Résultats obtenus à la R-précision et à la précision à 5, 10, 30, 100 documents restitués

Le tableau suivant présente les moyennes des p@5, p@10, p@30, p@100 ainsi que la moyenne des R-précisions sur Cisi.

	Cosinus	SimRank	Okapi
p@5	0,347	0,344 (-0,8%)	0,313 (-10%)
p@10	0,313	0,3 (-4%)	0,292 (-7%)
p@30	0,231	0,217 (-6%)	0,217 (-6%)
p@100	0,148	0,142 (-4%)	0,143 (-3%)
R-Prec	0,132	0,226 (+40%)	0,214 (38%)

Tableau 3 : précisions à 5, 10, 30, 100 documents restitués et R-Prec moyennes pour SimRank, Cosinus et Okapi sur Cisi

Les moyennes des p@5, p@10, p@30, p@100 placent le Cosinus devant le SimRank, lui-même devant Okapi. Le SimRank obtient néanmoins la meilleure moyenne des R-précisions devant les deux mesures témoin.

Examinons les résultats obtenus avec Cranfield :

	Cosinus	SimRank	Okapi
p@5	0,426	0,384 (-10%)	0,441 (+3%)
p@10	0,296	0,276 (-7%)	0,301 (+1%)
p@30	0,149	0,143 (-4%)	0,147 (-1%)
p@100	0,059	0,0597 (+1%)	0,058 (-1%)
R-Prec	0,367	0,345 (-6%)	0,407 (+9%)

Tableau 4 : précisions quand 5, 10, 30, 100 documents sont restitués et R-Prec moyennes pour SimRank, Cosinus et Okapi sur Cranfield

Le tableau 4 positionne notre méthode en 3ème position derrière Cosinus et Okapi pour les précisions à *n* et la R-précision.

1.3.3 Résultats à la mesure F à 5, 10, 30, 100 documents restitués

Le tableau suivant présente les moyennes des f@5, f@10, f@30 et f@100 sur Cisi :

	Cosinus	SimRank	Okapi
f@5	0,116	0,109 (-6%)	0,090 (-28%)
f@10	0,149	0,145 (-2%)	0,132 (-12%)
f@30	0,195	0,182 (-7%)	0,179 (-8%)
f@100	0,189	0,184 (-2%)	0,187 (-1%)

Tableau 5 : mesures F moyennes quand 5, 10,30, 100 documents sont restitués pour Cosinus, SimRank et Okapi sur Cisi

Le tableau 5 nous montre que le Cosinus obtient les meilleures mesures F, devant le SimRank et Okapi. SimRank obtient une meilleure mesure F quand 5, 10 et 30 documents sont restitués que Okapi. Les écarts entre les différentes méthodes sont faibles. Si l'on regarde plus en détail, SimRank améliore notablement 10 des 76 requêtes par rapport à Cosinus, ces 10 requêtes ont la particularité d'avoir un nombre de termes supérieur à la moyenne.

Examinons les résultats sur Cranfield :

	Cosinus	SimRank	Okapi
f@5	0,337	0,307 (-9%)	0,362 (6%)
f@10	0,323	0,304 (-6%)	0,335 (3%)
f@30	0,227	0,219 (-3%)	0,226 (0%)
f@100	0,108	0,108 (0%)	0,106 (-1%)

Tableau 6 : mesures F moyennes quand 5, 10, 30, 100 documents sont restitués pour Cosinus, SimRank et Okapi sur Cranfield

Le tableau 6 montre qu'en termes de mesure F en fonction du nombre de documents restitués, Okapi obtient de meilleurs résultats à la mesure F, devant Cosinus et devant SimRank pour 5, 10 et 30 documents restitués. A 100 documents restitués, la mesure F de SimRank et de Cosinus dépassent de 2% celle de Okapi.

Pour avoir une idée générale en termes de précision et de rappel des résultats obtenus par les trois mesures que nous évaluons, nous allons regarder la courbe précision-rappel en 11 points .

1.3.4 Courbes précision-rappel

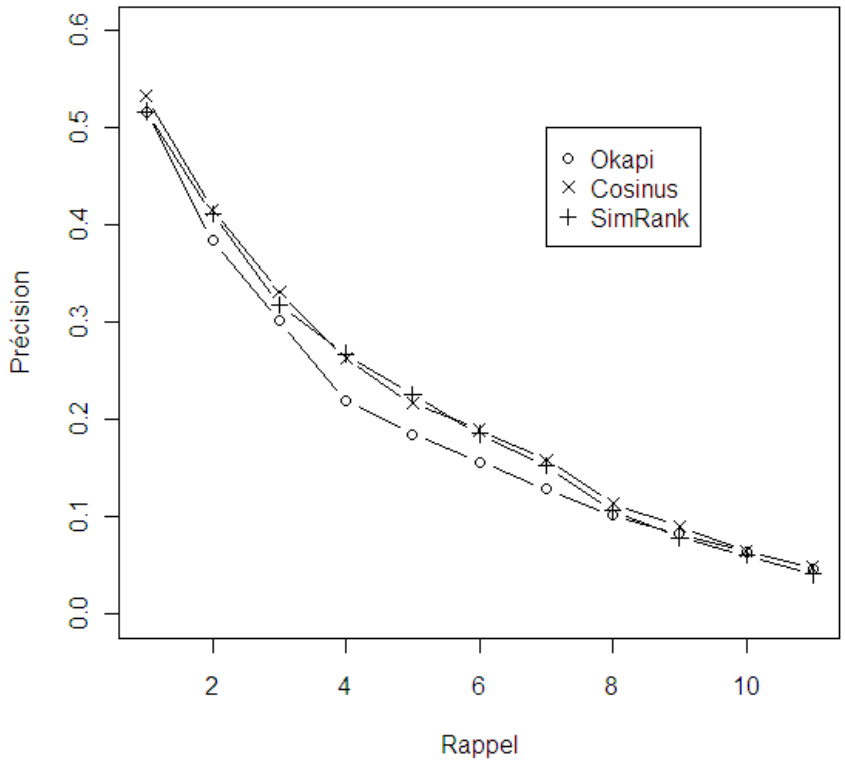


Figure 60 : courbes précision-rappel en 11 points pour Cosinus, SimRank, Okapi sur Cisi

La figure 60 montre que les trois mesures obtiennent des résultats similaires. Le Cosinus obtient les meilleurs scores quand le rappel est inférieur à 10 %. Le SimRank obtient les meilleurs résultats quand le taux rappel est entre 10 % et 30 %.

Les résultats obtenus par les trois mesures deviennent quasi identiques lorsque le nombre des documents restitués augmente.

Examinons maintenant la courbe précision-rappel obtenue sur Cranfield :

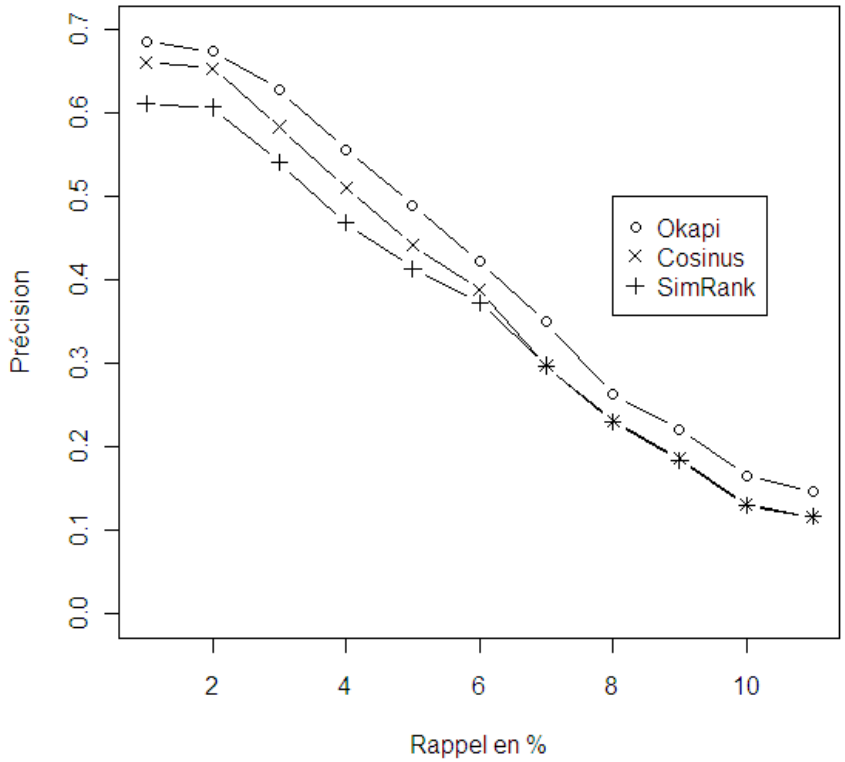


Figure 61 : courbes précision-rappel en 11 points pour Cosinus, SimRank, Okapi sur Cranfield

Les courbes précision-rappel précédentes indiquent une supériorité de Okapi par rapport aux deux autres méthodes. La méthode Cosinus classe mieux que la méthode SimRank en tête de liste, cette différence s'amenuise au fur et à mesure que des documents sont restitués.

La conclusion générale que l'on peut tirer de ces résultats est que notre méthode obtient des résultats parfois inférieurs aux méthodes témoins (MAP), mais parfois montre un avantage (la R-précision sur Cranfield, la mesure F à 100 documents restitués sur Cisi) et se comporte comme les méthodes existantes et peut donc être envisagée pour classer des documents. Les résultats obtenus sont comparables à ceux obtenus par des méthodes traditionnelles, bien que parfois légèrement inférieurs.

A titre de comparaison, nous reportons les résultats présentés dans [Kumar et al., 2009], où est évalué le modèle LSI (*Latent Semantic Indexing)* en comparaison au modèle vectoriel et au modèle vectoriel généralisé. Le modèle LSI est un modèle vectoriel qui par réduction de l'espace document-terme de départ produit un espace dans lequel les documents et les termes similaires sont plus proches. Ce modèle exploite la structure sémantique Latente [Furnas et al., 1988].

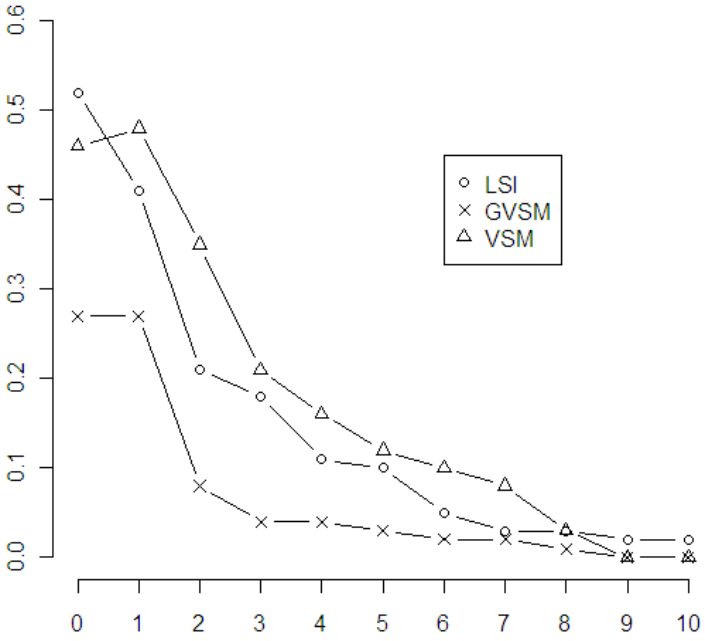


Figure 62 : courbes précision-rappel en 11 points pour trois méthodes (LSI, VSM, GVSM) sur Cranfield [Kumar et al., 2009]

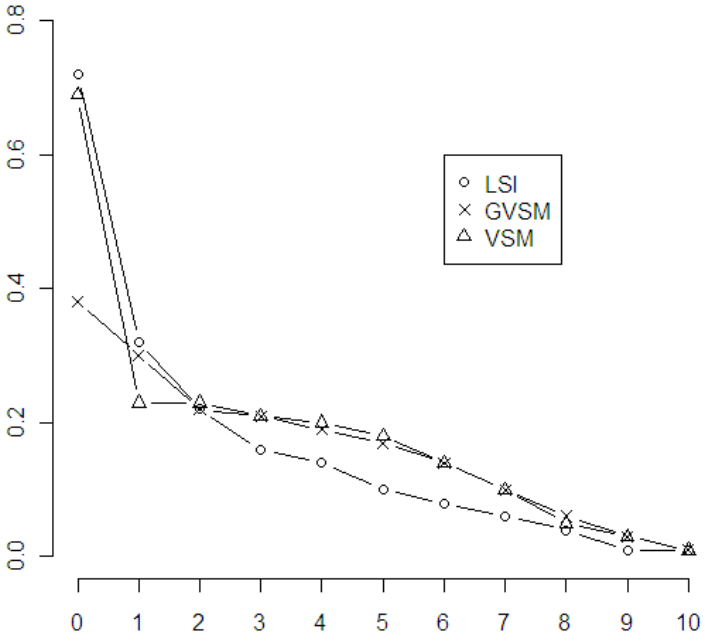


Figure 63 : courbes précision-rappel en 11 points pour trois méthodes (LSI, VSM, GVSM) sur Cisi [Kumar et al., 2009]

La figure 62 montre que les modèles étudiés (LSI, VSM, GVSM) obtiennent des résultats comparables et légèrement inférieurs à ceux que nous avons présenté sur Cranfield.
La figure 63 montre que la méthode LSI permet d'obtenir une précision à rappel faible (inférieur à 0,3) supérieure à la précision obtenue par les méthodes que nous avons présentées, la précision quand le rappel est élevé est quant à elle plus faible.

1.3.5 Significativité des résultats obtenus lors de la comparaison des méthodes

Afin de vérifier la significativité des résultats obtenus lors de la comparaison des différentes méthodes, nous devons utiliser un test statistique. Le choix du test dépend de la distribution des données. Les tests paramétriques s'appliquent sur des données suivant une distribution normale alors que les tests non paramétriques sont eux adaptés à des données qui suivent une distribution non normale. Nous avons donc analysé la distribution des résultats obtenus.

Sur le corpus Cisi, ni les précisions à 5, 10, 30 et 100 documents restitués ni les précisions f@5, f@10, f@30, f@100 ne suivent une distribution normale pour les méthodes SimRank, Cosinus ou Okapi ($p < 0.05$; test de Shapiro-Wilk). En revanche, la précision R-prec pour la méthode Okapi ($p = 0.09$; test de Shapiro-Wilk) et la mesure F pour les méthodes SimRank et Cosinus ($p=0.08$, $p=0.06$; test de Shapiro-Wilk) suivent une loi normale.

Sur le corpus Cranfield, la précision à 5 documents restitués ne suit pas une distribution normale pour la méthode SimRank, contrairement aux méthodes Cosinus et Okapi ($p=0.10$, $p=0.33$; test de Shapiro-Wilk). Pour les autres précisions, aucune n'a de distribution normale ($p < 0.01$; test de Shapiro-Wilk).

Par conséquent, la plupart des distributions étant non normales, un test non paramétrique a été utilisé pour comparer les précisions obtenues selon les méthodes SimRank, Cosinus et Okapi.

Les différents critères retenus prennent une valeur réelle qui varie entre 0 et 1. En termes statistiques, la valeur varie sur une échelle dite ordinale. Un échantillon pour une mesure donnée est l'ensemble des valeurs obtenues sur ce critère pour l'ensemble des requêtes d'une collection. D'une méthode à l'autre, les échantillons sont indépendants. Pour ces raisons, nous avons utilisé un test de Mann-Whitney. Ce test permet de classifier les valeurs puis de comparer leurs rangs pour deux échantillons indépendants.

	SimRank		Cosinus		Okapi		p SimRank vs Cosinus	p SimRank vs Okapi
	Moyenne [ET]		Moyenne [ET]		Moyenne [ET]			
p@5	0.34	[0.30]	0.34	[0.27]	0.31	[0.25]	0.84	0.68
p@10	0.30	[0.24]	0.31	[0.23]	0.29	[0.22]	0.71	0.91
p@30	0.21	[0.16]	0.23	[0.16]	0.21	[0.15]	0.53	0.89
p@100	0.14	[0.11]	0.14	[0.11]	0.14	[0.11]	0.85	0.94
Map	0.20	[0.15]	0.21	[0.15]	0.18	[0.11]	0.73	0.91
R_prec	**0.22**	**[0.15]**	**0.13**	**[0.09]**	0.21	[0.13]	**0.01**	0.92
f@5	0.10	[0.14]	0.11	[0.13]	0.09	[0.08]	0.75	0.86
f@10	0.14	[0.14]	0.14	[0.13]	0.13	[0.10]	0.83	0.80
f@30	0.18	[0.12]	0.19	[0.12]	0.17	[0.10]	0.47	0.87
f@100	0.18	[0.11]	0.18	[0.11]	0.18	[0.11]	0.80	0.90

Tableau 7 : test de Mann-Whitney sur les mesures utilisées pour comparer SimRank avec Cosinus et, SimRank avec Okapi sur Cisi

Les résultats obtenus pour la p@5 ne montrent pas de différence significative entre le SimRank et le Cosinus sur le corpus Cisi. Il en est de même pour la p@10, la p@30, la p@100, la MAP, la f@5, la f@10, la f@30 et la f@100 ($p > 0.05$; test de Mann-Whitney). Concernant la R-précision, celle-ci est significativement supérieure pour le SimRank par rapport au Cosinus ($p = 0.01$; test de Mann-Whitney) sur le corpus Cisi.

	SimRank		Cosinus		Okapi		p SimRank vs Cosinus	p SimRank vs Okapi
	Moyenne [ET]		Moyenne [ET]		Moyenne [ET]			
p@5	**0.38**	**[0.26]**	0.42	[0.27]	**0.44**	**[0.26]**	0.09	**0.01**
p@10	0.27	[0.18]	0.29	[0.18]	0.30	[0.18]	0.20	0.16
p@30	0.14	[0.09]	0.14	[0.09]	0.14	[0.09]	0.48	0.79
p@100	0.05	[0.03]	0.05	[0.03]	0.05	[0.03]	0.82	0.56
Map	**0.36**	**[0.25]**	0.40	[0.25]	**0.43**	**[0.26]**	0.14	**0.01**
R_prec	**0.34**	**[0.23]**	0.36	[0.22]	**0.40**	**[0.24]**	0.20	**0.01**
f@5	**0.30**	**[0.21]**	0.33	[0.21]	**0.36**	**[0.21]**	0.16	**0.01**
f@10	0.30	[0.18]	0.32	[0.19]	0.33	[0.18]	0.25	0.07
f@30	0.21	[0.11]	0.22	[0.12]	0.22	[0.12]	0.51	0.61
f@100	0.10	[0.05]	0.10	[0.05]	0.10	[0.06]	0.67	0.60

Tableau 8 : test de Mann-Whitney sur les mesures utilisées pour comparer SimRank et Cosinus et, SimRank et Okapi sur Cranfield

Concernant la comparaison de SimRank et Okapi sur Cranfield, la p@5, la MAP, R-précision et la f@5 sont significativement meilleures pour Okapi ($p < 0.05$; test de Mann-Whitney). Pour les autres mesures, il n'y a pas de différence significative entre les méthodes.

Les mesures utilisées dans la section précédente pour l'évaluation de notre méthode concernent la qualité de ses réponses. Or, il est intéressant en RI d'être en mesure de faire du filtrage, c'est-à-dire être capable d'associer à un document un score de ressemblance à une requête et de dire si ce score (souvent un nombre réel) fait que le document est pertinent ou non. Le filtrage nécessite de convertir les scores des documents en valeurs binaires ; le document est pertinentou ne l'est pas.

1.4 Adaptation de la méthode au filtrage d'information

Comme nous l'avons suggéré au chapitre 4 notre méthode de tri retrouve tous les documents liés directement ou indirectement à la requête dans la collection. Cela a pour effet de retourner un grand nombre de documents. On peut s'interroger dans le cas d'un très grand nombre de documents restitués si la valeur de similarité attribuée au couple formé par le $n^{ième}$ document et la requête a réellement un sens.

Afin d'être en mesure de dire si un score SimRank obtenu par un document avec la requête fait de lui un document pertinent ou non, nous avons souhaité munir notre algorithme d'un seuil. Les outils d'analyse nous permettent de déterminer expérimentalement quel est le meilleur seuil en se basant sur une mesure donnée. Ainsi, nous avons choisi la mesure F qui combine précision et rappel pour évaluer une liste de documents retournés. Après avoir noté et classé les documents à l'aide de notre méthode, le nombre de documents à retourner pour obtenir la meilleure mesure F possible pour chaque requête est calculé. De cette manière on va déterminer un seuil expérimentalement :

celui qui maximise la mesure F moyenne pour la collection. Ensuite par l'étude des scores obtenus par les documents et la requête, on va rechercher comment on aurait pu trouver le seuil expérimental.

1.4.1 Recherche d'un seuil sur la collection Cisi

Les mesures F obtenues pour les requêtes de Csi sont calculées. Nous observons le nombre de documents à retourner pour avoir la meilleure mesure F pour une requête donnée. Ce nombre est nommé rang de la meilleure mesure F et le score obtenu par le document à ce rang est le seuil absolu. Le rang de la meilleure mesure F est en moyenne égal à 78 avec une variance de 87, ce qui indique une grande variabilité. Le choix d'une coupe basée sur ce rang paraît trop variable d'une requête à l'autre pour être envisagé. On peut alors regarder le score obtenu par le document à ce rang. Le tableau suivant présente le score moyen obtenu au rang de la meilleure mesure F, ainsi que l'écart-type, le score le plus petit et le score le plus grand obtenu sur Cisi.

SimRank moyen à la meilleure mesure F	0,01458
Ecart-Type	0,00331
Minimum	0,00979
Maximum	0,02428

Tableau 9 : score moyen, écart-type, scores minimum et maximum obtenus à la meilleure mesure F

Le tableau 9 indique que le score moyen obtenu est variable : il vaut en moyenne 0,0145 avec une variance de 0,0033. De plus, comme nous l'avions constaté la valeur de ce score dépend du nombre de documents du corpus et de leur taille (leur nombre de termes).

Les scores SimRank varient beaucoup d'une requête à l'autre. De même le score SimRank entre la requête et elle-même varie beaucoup d'une requête à l'autre – voir tableau 10.

SimRank de la requête avec elle-même	0,1078
Ecart-Type	0,0733
Minimum	0,0291
Maximum	0,0242

Tableau 10 : score moyen, écart-type, scores minimum et maximum obtenus par la requête avec elle-même

Les scores SimRank absolus ne nous permettent donc pas de déterminer un seuil, nous nous intéressons donc aux seuils relatifs.

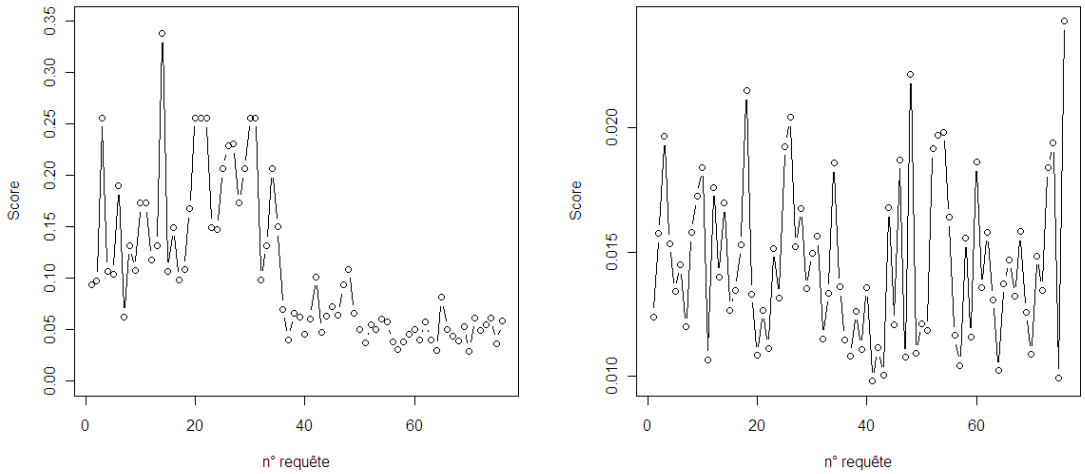


Figure 64 : variation des scores SimRank de la requête avec elle-même (à gauche) et variation des scores SimRank des documents aux rangs desquels la meilleure mesure F est obtenue en fonction des requêtes de Cisi.

Il semble que les variations suivies par les scores des documents et par le score de la requête ne soient pas corrélées. Après ces constatations qui étaient prévisibles, nous avons décidé d'observer le rapport entre le score obtenu par le document au rang de la meilleure mesure F et le score obtenu par la requête avec un clone d'elle-même afin de ramener le score SimRank du document à celui de la requête servant de borne supérieure. L'idée est de ramener l'intervalle des scores compris entre 0 et le score de la requête à un intervalle entre 0 et 1.

Pour observer un éventuel lien entre le nombre de termes de la requête et la hauteur du score du document au rang duquel la meilleure mesure F est obtenue, nous traçons le nuage de points du seuil relatif en fonction du nombre de termes de la requête :

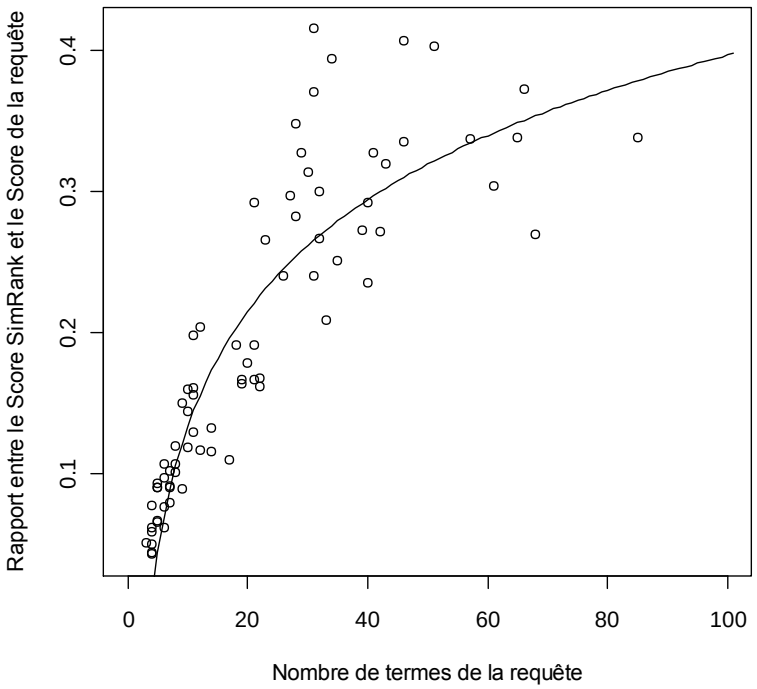


Figure 65 : seuil relatif en fonction du nombre de termes de la requête

Le nuage de points suit une courbe de tendance de type logarithmique.
Son équation, calculée grâce à la courbe de tendance est :

y =0,11 ln(nombre de termes de la requête)-0,11

Il est nécessaire de déterminer si la relation entre le seuil relatif et le nombre de termes est propre à SimRank ou au corpus étudié. Pour cela nous allons reproduire ce raisonnement avec la collection Cranfield. D'autres tests sur d'autres collections sont certainement nécessaires avant de trouver une formule de seuil qui soit performante. Cette expérience constitue un premier pas.

1.4.2 Recherche d'un seuil sur la collection Cranfield

Le rang moyen auquel on obtient la meilleure mesure F est de 21 avec une variance de 77, le rang le plus petit est 1, le rang maximum est 860. Les scores correspondants valent en moyenne 0,025 avec une variance de 0,006. On trace le nuage de points du seuil relatif en fonction du nombre de termes -cf. figure 66.

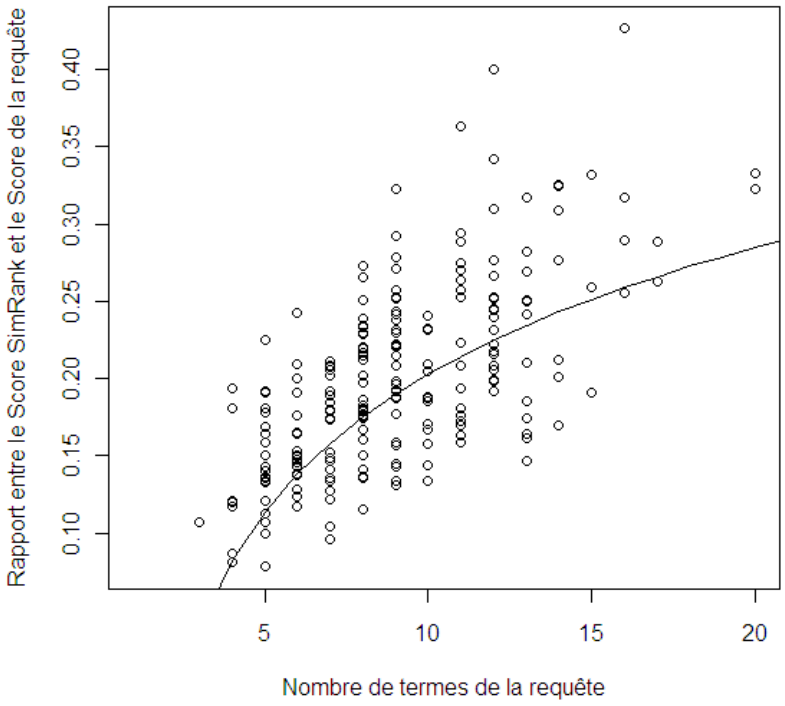


Figure 66 : seuil relatif en fonction du nombre de termes de la requête

Il est intéressant de constater que la courbe de tendance est également de la forme « a ln(x) + b » avec a =0,111.

Nous sommes alors en mesure de proposer une première méthode de filtrage qui consiste à sélectionner les documents dont le SimRank vérifie :

$SimRank(d,q) > (0{,}111 \times ln\,(nbt(q)) - b) \times SimRank(q,q)$

Avec *nbt(q)* le nombre de termes de la requête.

Dans l'attente d'une analyse plus approfondie, il est possible pour choisir la valeur de *b* de prendre par exemple une valeur intermédiaire entre celle qui convient pour Cisi et celle qui convient pour Cranfield. D'autres choix sont évidemment possibles.

1.4.3 Comparaison de SimRank avec seuil à Cosinus avec Seuil

Nous avons souhaité construire un dispositif expérimental permettant de comparer le SimRank muni d'un seuil au Cosinus muni d'un seuil. Cette fois, seuls les documents ayant un score supérieur au seuil défini pour la mesure utilisée sont restitués.

Pour SimRank, le seuil défini précédemment est choisi avec le paramètre *b* égal à 0,08, ce qui correspond à la moyenne du *b* optimal calculé pour le corpus Cisi et de celui calculé pour le corpus Cranfield.

Pour le Cosinus, les scores absolus par un document ont un sens géométrique ; il n'est pas nécessaire de les relativiser. Pour choisir le seuil utilisé par le Cosinus, le rang permettant d'obtenir la meilleure mesure F possible pour chaque requête est mémorisé, pour chacun des deux corpus étudiés Cisi et Cranfield. La coupe qui permet d'obtenir la meilleure mesure F moyenne sur l'ensemble des requêtes de Cisi est obtenue quand le cosinus des documents et de la requête est supérieur à 0,11 ; pour Cranfield, la meilleure coupe est obtenue quand on retourne les documents ayant un cosinus avec la requête supérieur à 0,26. Nous retiendrons ces deux seuils.

	SimRank+seuil	Cosinus+seuil	Ecart %
Cranfield	0,249	0,32	-22
Cisi	0,150	0,2	-25

Tableau 11 : mesures F moyennes pour SimRank muni d'un seuil et Cosinus muni d'un seuil optimal sur Cisi et Cranfield

Le tableau 11 indique que notre méthode muni d'un seuil obtient des résultats clairement inférieurs à ceux obtenus par la méthode Cosinus munie d'un seuil. Notre seuil utilise un paramètre dont la valeur à été fixée arbitrairement.

Nous avons également comparé notre méthode munie d'un seuil à la méthode Cosinus munie d'un seuil intermédiaire, correspondant à la moyenne des deux seuils utilisés dans l'expérimentation précédente : 0,18. Nous choisissons un tel seuil non pas parce que nous pensons que la moyenne entre deux seuils utilisées sur deux collections différentes a un sens mais plutôt pour ne favoriser ni l'une ni l'autre des collections.

	SimRank+seuil	Cosinus+seuil intermédiaire	Ecart %
Cranfield	0,249	0,245	1,6
Cisi	0,150	0,125	20

Tableau 12 : mesures F moyennes pour SimRank muni d'un seuil et Cosinus muni d'un seuil intermédiare Cisi et Cranfield

Le SimRank avec seuil obtient une mesure F moyenne légèrement supérieure à celle obtenue par le Cosinus avec seuil intermédiaire. Ces résultats montrent que notre méthode permet de couper finement un retour de SRI. Le SimRank est apparu légèrement inférieur au Cosinus en termes de MAP, ce qui nous permet de dire qu'il classe légèrement moins bien, mais semble fournir des scores plus utilisables au sens de la coupe.

L'analyse statistique des distributions montre que les mesure F calculées ne suivent pas une distribution normale ($p < 0.05$; test de Shapiro-Wilk). Pour étudier la significativité de la comparaison de SimRank+seuil à Cosinus+seuil, nous utilisons le test de Mann-Whitney :

	SimRank+seuil	Cosinus+seuil	Cosinus+seuil intermédiaire	SimRank Vs Cosinus+seuil intermédiaire P	SimRank Vs Cosinus+seuil intermédiaire P
	Moyenne [ET]	Moyenne [ET]	Moyenne [ET]		
Cisi	0,150 [0.14]	0,2 [0.12]	0,125 [0.13]	0.23	0.51
Cranfield	0,249 [0.15]	0,32 [0.2]	0,245 [0.15]	0.79	0.84

Tableau 13 : test de Mann-Whitney sur les mesures utilisées pour comparer SimRank+seuil et Cosinus+seuil intermédiaire

Le tableau 13 indique que les deux méthodes sont comparables car il n'y a pas de différence significative dans les résultats.

1.5 SimRank utilisé à la suite d'un tri : Cosinus puis SimRank

Jusqu'à présent, nous avons utilisé notre méthode comme un trieur de documents en fonction de requêtes. Dans un deuxième temps, nous avons muni notre méthode d'un seuil permettant de définir la pertinence et respectivement la non pertinence d'un document par rapport à une requête donnée. Ainsi, notre méthode de tri est devenue une méthode de filtrage.

Nous avons souhaité explorer une autre piste en prévision de l'utilisation de notre algorithme sur de grandes collections à savoir utiliser notre méthode consécutivement à une première phase de tri réalisé par une méthode usuelle. Nous proposons une méthode qui consiste pour une requête donnée à effectuer un filtrage des documents du corpus par rapport à cette requête à l'aide de la mesure Cosinus, puis de sélectionner les documents restitués et de les trier à l'aide de notre méthode. Notre but est de constater de quelle manière notre algorithme reclasse les documents reconnus comme pertinents par la méthode de la première phase. Ces évaluations ont donné lieu à publication [Champclaux et al., 2008][Champclaux et al., 2009].

1.5.1 Résultats obtenus à la Mean Average Précision

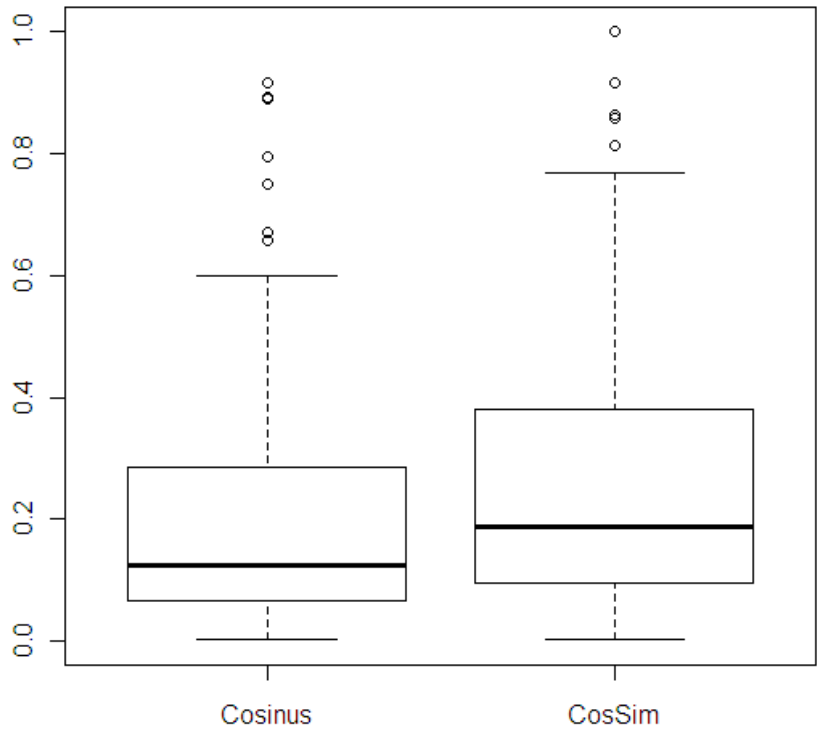


Figure 67 : représentation des MAP pour Okapi et OkaSim sur Cranfield

La figure 67 indique une amélioration de la combinaison des méthodes par rapport à la méthode Cosinus. Notre méthode améliore la MAP pour la majorité des requêtes du corpus (33 requêtes ne sont pas améliorées). Les scores sont plus répartis par rapport à la moyenne avec notre méthode qu'avec la méthode Cosinus.

1.5.2 Résultats obtenus à la R-précision et à la précision à 5, 10, 30, 100 documents restitué

	Cosinus	CosSim	Ecart %
p@5	0,2053	0,2568	25,10
p@10	0,1542	0,2053	33,14
p@30	0,0927	0,1182	27,47
p@100	0,0469	0,0531	13,16
R-Prec	0,1880	0,2419	28,68

Tableau 14 : précisions quand 5, 10, 30, 100 documents sont restitués et R-Prec moyennes pour Cosinus et CosSim sur le corpus Cranfield

Le tableau 14 indique de bons résultats selon tous les critères étudiés, notre méthode améliore la R-Prec d'environ 30 % par rapport au Cosinus. L'amélioration de la précision est supérieure en tête de liste (p@10) et s'approche des résultats de Cosinus au fur et à mesure que les documents sont restitués ce qui est normal car on restitue exactement les documents trouvés par le Cosinus.

L'analyse statistique des distributions montre que les mesure F calculées ne suivent pas une distribution normale ($p < 0.05$; test de Shapiro-Wilk). Comme précédemment, nous utilisons le test de Mann-Whitney :

	Cosinus	CosSim	p
	Moyenne [ET]	Moyenne [ET]	
p@5	**0,2053 [0.21]**	**0,2568 [0.22]**	**<0.01**
p@10	**0,1542 [0.13]**	**0,2053 [0.15]**	**<0.01**
p@30	**0,0927 [0.06]**	**0,1182 [0.07]**	**<0.01**
p@100	**0,0469 [0.03]**	**0,0531 [0.03]**	**0.02**
R-Prec	**0,1880 [0.18]**	**0,2419 [0.21]**	**<0.01**
MAP	**0,1929 [0.18]**	**0,2534 [0.21]**	**<0.01**

Tableau 15 : test de Mann-Whitney sur les mesures utilisées pour comparer CosSim et Cosinus

Le tableau 15 indique que la méthode CosSim est significativement meilleure que Cosinus.

1.5.3 Courbes précision-rappel

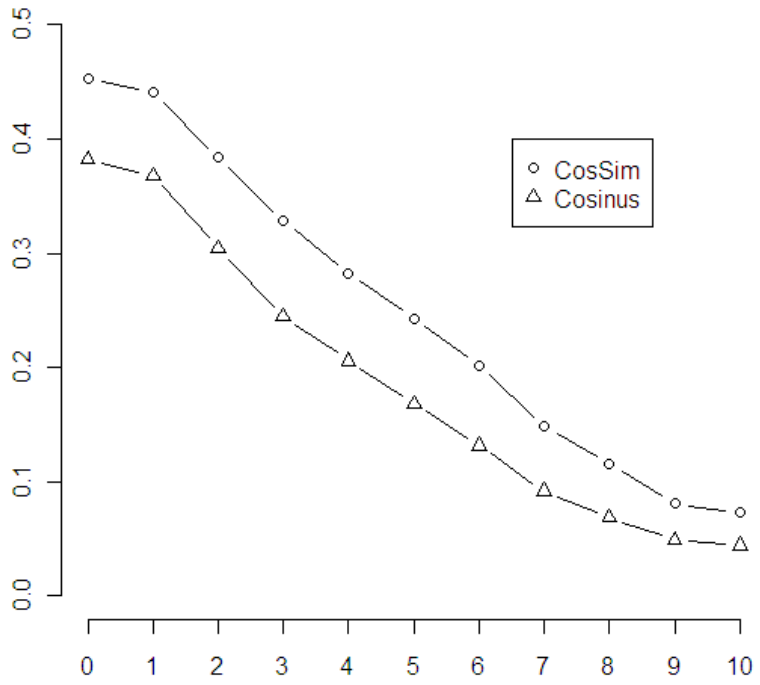


Figure 68 : courbe précision-rappel pour Cosinus et CosSim sur le corpus Cranfield

La figure 68 confirme les résultats, la méthode CosSim améliore la méthode Cosinus seule. Ces résultats nous confortent dans l'usage de notre méthode en deux phases.

1.5.4 Influence de la pondération des termes sur SimRank en deux phases

Les précédents tests montrent que l'usage des similarités structurelles peut améliorer un ordonnancement Cosinus. Nous nous sommes intéressés à l'influence de différentes pondérations définies dans [Salton, 1988]. Notre méthode se base sur les relations entretenues par les documents, ces relations sont conceptuellement représentées par les arcs du graphe représentant l'ensemble corpus et requête. Le poids d'un terme dans un document est représenté par une valeur sur l'arc qui lie le document au mot dans le graphe. Il nous semble évident que notre méthode est sensible à la pondération, reste à le vérifier expérimentalement. Nous allons examiner quatre pondérations différentes :

- *CosSim 0ou1 (CS_0ou1)*: les termes du document sont pondérés à 1 s'ils apparaissent dans le document et 0 sinon.
- *CosSim Tf* (CS_*Tf*): les termes du document sont pondérés par leur fréquence d'apparition dans le document.
- *CosSim TfIdf (CS_TfIdf)*: les termes des documents sont pondérés en utilisant *tf.idf*.
- *CosSim TfcNfx (CS_TfcNfx)*: les termes du document sont pondérés en utilisant *tf.idf* et une normalisation 2, les termes de la requête sont pondérés en utilisant la fréquence des termes normalisée (le facteur *tf* est normalisé par rapport au *tf* maximum, puis est à nouveau normalisé de manière à prendre une valeur entre 0,5 et 1).

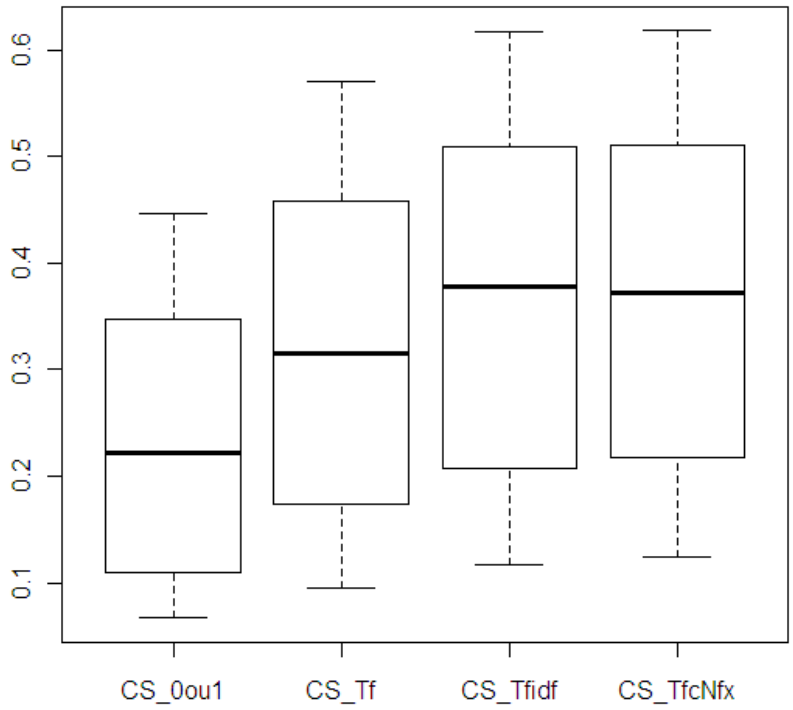


Figure 69 : représentation des MAP pour Cosinus pondéré par tf et CosSim avec différentes pondérations sur Cranfield

Les boites à moustaches de la figure 69 montrent que notre méthode est sensible à la pondération. Les meilleurs résultats en termes de MAP sont obtenus avec une pondération *Tfc-Nfx*. Les résultats obtenus par une pondération *Tf.Idf* sont très proches de la meilleure pondération.

	R-Prec	Ecart par rapport à CS_Tf.idf en %
CS_0ou1	0,232	-33,1
CS_*Tf*	0,308	-11,2
CS_Tfldf	0,347	0
CS_TfcNfx	0,351	+1,2

Tableau 16 : R-précisions pour CosSim avec quatre pondérations différentes

Le tableau 16 permet de constater la même chose : notre méthode est sensible à la pondération, la pondération *TfcNfx* obtient les meilleurs résultats, suivie de près par la pondération *Tf.idf*.

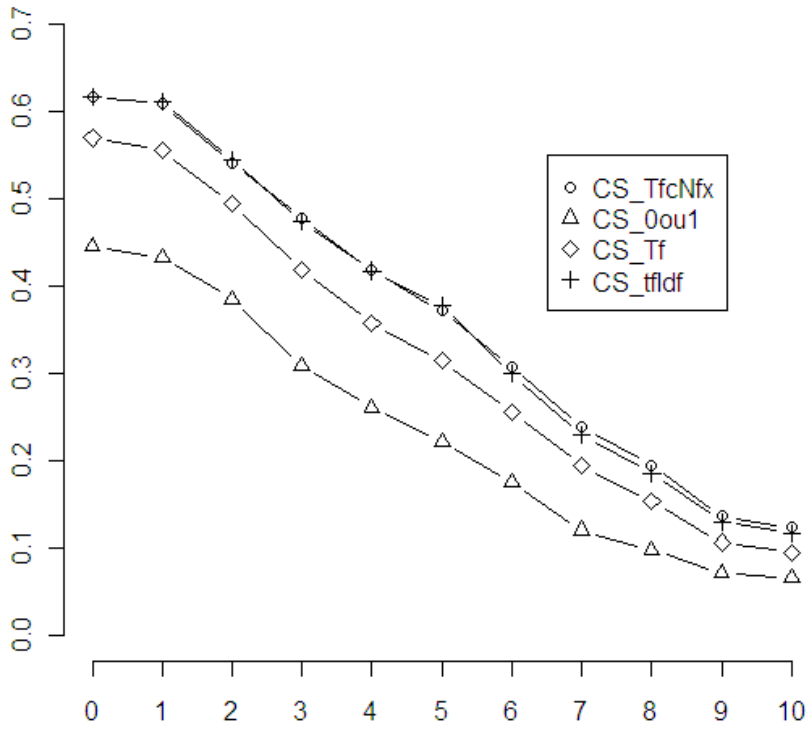


Figure 70 : courbe précision-rappel pour CosSim avec 4 pondérations différentes.

Les courbes précision-rappel de la figure 70 indiquent encore l'intérêt de pondérer les termes : les meilleurs résultats sont obtenus avec Tf.idf et Tfc-Nfx. Cela indique que le choix d'utiliser *tf.idf* pour pondérer les termes est un bon choix.

L'analyse statistique des distributions montre que touts les mesures utilisées ne suivent pas une distribution normale ($p < 0.05$; test de Shapiro-Wilk). Comme précédemment, pour étudier la significativité de la comparaison de CosSim à Cosinus, nous utilisons le test de Mann-Whitney :

		Moyenne [ET]		Gain (%) [IC]		P
R-Prec	*CS_0ou1*	**0,232**	**[0.20]**	**-11.5**	**[-15.6 ; -7.5]**	**<0.01**
	CS_Tf	0,308	[0.22]	-1.9	[-4.1 ; 0.1]	0.08
	CS_Tfidf	0,347	[0.23]	–	–	–
	Tfcnfx	0,351	[0.24]	0.08	[-1.0 ; 1.1]	0.92
MAP	*CS_0ou1*	0,246	[0.20]	**-12.6**	**[-16.9 ; -8.3]**	**<0.01**
	CS_Tf	0,331	[0.24]	-2.0	[-4.3 ; 0.2]	0.06
	CS_Tfidf	0,372	[0.25]	–	–	–
	Tfcnfx	0,377	[0.25]	0.1	[-1.0 ; 1.2]	0.84

Tableau 17 : test de Mann-Whitney sur les mesures utilisées pour comparer CS_Tfidf à CS_0ou1, à CS_Tf, à CS_Tfcnfx

Le tableau 17 indique que *CS_0ou1* est significativement inf§rieur à *CS_Tfidf*. A noter que le *p* est très proche de 0,05 pour *Tf*, ce qui signifie que *CS_TfIdf* est presque significativement meilleur que *CS_Tf*. La différence entre *CS_Tfidf* et *CS_TfcNfx* n'est pas significative.

2. SimRank sur une grosse collection

2.1 Okapi puis Cosinus et Okapi puis SimRank

La complexité de l'algorithme que nous proposons ne permet pas son application à de grandes collections telle que la collection TREC ad hoc 1998 qui compte 317 997 documents totalisant 520 452 termes uniques et l'algorithme tel qu'il est actuellement programmé peut prendre en entrée une matrice de l'ordre de 5 000 × 10 000.

La collection TREC est une référence dans le domaine de la RI, nous avons souhaité l'exploiter afin de poursuivre notre étude de l'algorithme SimRank.

Pour s'adapter aux contraintes de l'algorithme, nous nous sommes inspirés de la méthode MAC/FAC (*Many are called few are choosen*) en réalisant une expérimentation en deux phases. La méthode HITS, qui utilise le résultat d'un moteur de recherche pour constituer un graphe de départ avant d'associer à ses nœuds un score hub et un score autorité, nous a également inspiré. Nous retenons, pour chacune des 50 requêtes TREC, les 1 500 premiers documents restitués par la méthode Okapi qui est une méthode efficace de RI. Ainsi, nous constituons un nouveau corpus de tests d'une taille comparable aux deux corpus étudiés précédemment Cranfield et Cisi. Cependant, ce corpus se distingue de ces derniers puisqu'en retenant les 1 500 documents les mieux classés d'un corpus de 370 000 documents, les 1 500 documents ont tous une similarité minimale par rapport à la requête, ce qui n'est pas le cas des deux autres corpus. Après constitution d'un corpus pour une requête donnée, nous associons à chaque nœud du graphe représentant le corpus un seul score : le score SimRank.

Pour chacune des requêtes nous disposons donc d'un corpus composé des 1 500 premiers documents retournés par Okapi. Ce corpus compte en moyenne 20 000 termes uniques. Notre but étant d'étudier les propriétés de tri de notre algorithme en se basant sur les relations entretenues par les documents, nous supprimons les termes apparaissant dans un seul document. Cette

suppression des termes ayant une fréquence par document égale à leur fréquence dans la collection (sur l'ensemble des documents) nous permet de ramener les corpus d'étude à une taille convenable pour notre algorithme (1 500 × 10 000).

Il est nécessaire de pouvoir comparer l'ordonnancement basé sur les relations entre documents et termes à une autre méthode. Pour cela, il nous a semblé logique de comparer la similarité structurelle à la similarité directe représentée par le Cosinus des documents avec la requête. La similarité définie par le Cosinus est la similarité directe dans un espace vectoriel. Ainsi nous comparons deux méthodes d'ordonnancement, la méthode Cosinus et la méthode SimRank.

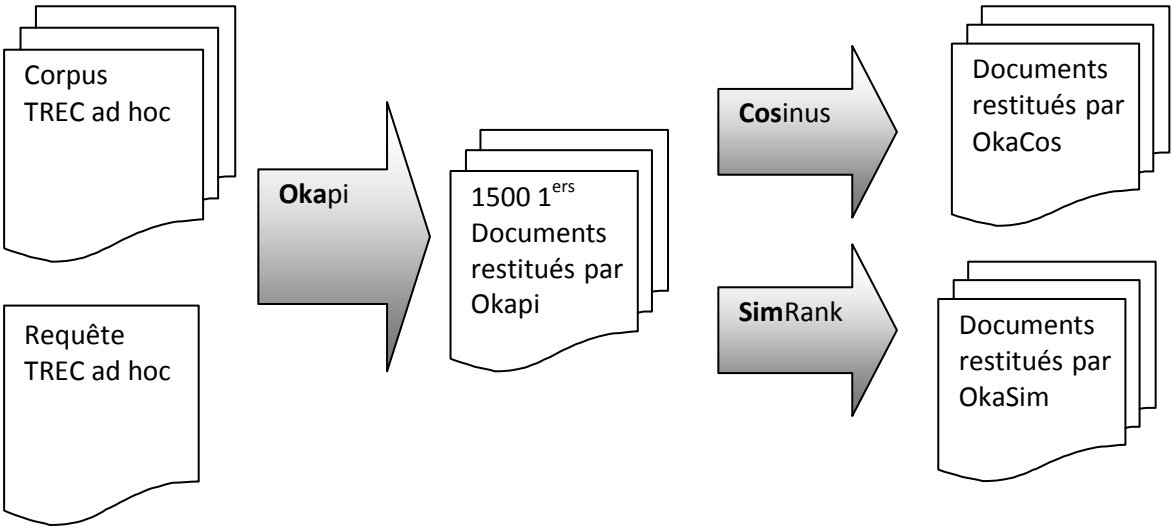


Figure 71 : schéma des expérimentations sur TREC : OkaCos et OkaSim

2.1.1 Résultats obtenus à la Mean Average Precision

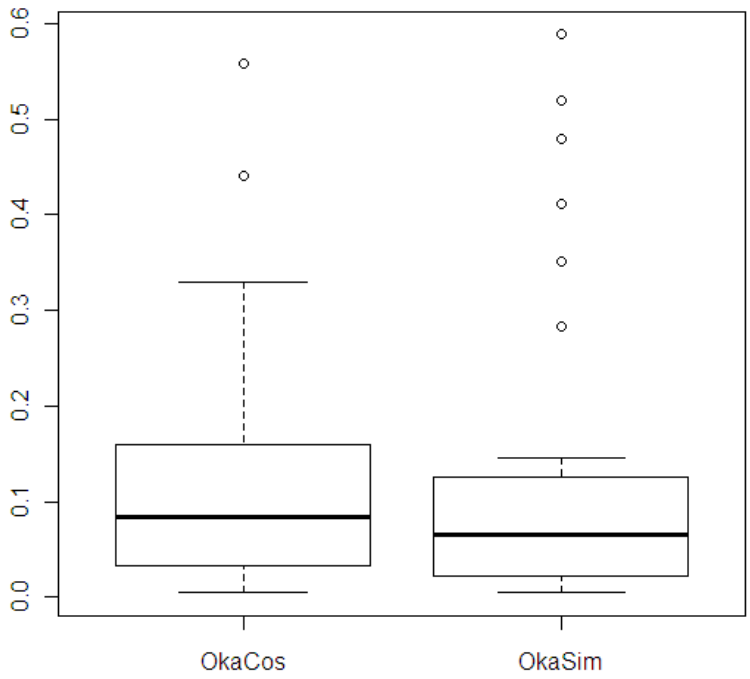


Figure 72 : représentation en boîtes à moustaches des MAP pour OkaCos et OkaSim sur TREC

La figure 72 indique de meilleurs résultats pour le Cosinus, au détriment de notre méthode. Néanmoins SimRank obtient de meilleurs résultats pour 10 requêtes et obtient des résultats identiques pour 17 requêtes sur 50.

2.1.2 Résultats obtenus à la précision à 5, 10, 30, 100 documents restitués

	OkaCos	OkaSim	Ecart %
p@5	0,2	0,132	-34
p@10	0,176	0,130	-26
p@30	0,126	0,090	-28
p@100	0,085	0,068	-19
R-Prec	0,125	0,097	-22

Tableau 18 : précisions quand 5, 10, 30, 100 documents sont restitués et R-Prec moyennes pour OkaCos et OkaSim sur TREC

Le tableau 18 montre une très nette supériorité du Cosinus par rapport à notre méthode. Le Cosinus majore de 25 % environ chacune des mesures retenues. Néanmoins ces mesures ne reflètent que le comportement en tête de liste.

Pour avoir une idée du comportement général de la précision en fonction du rappel, nous allons regarder la courbe précision-rappel.

2.1.3 Courbe précision-rappel

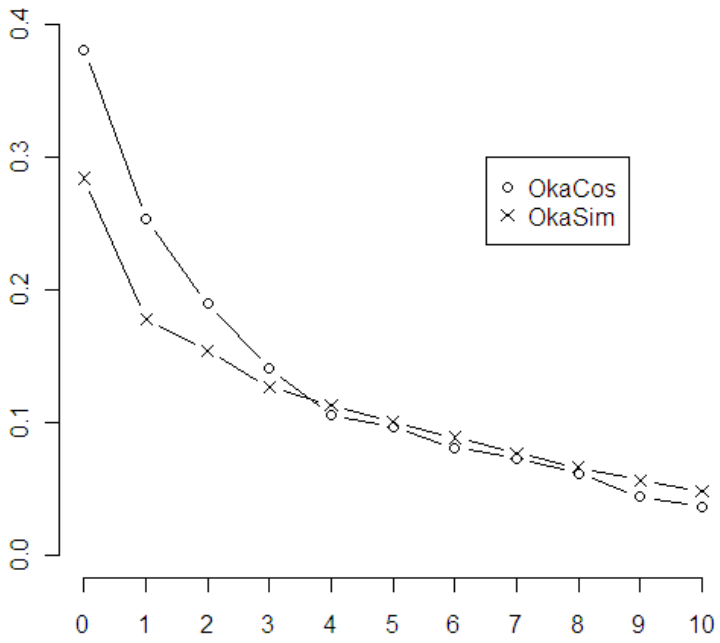


Figure 73 : courbe précision-rappel pour OkaCos et OkaSim sur TREC

La courbe précision-rappel de la figure 73 donne une vue du comportement global de notre algorithme par rapport au Cosinus. Ainsi, nous pouvons remarquer que notre méthode obtient des résultats plus faibles à petit rappel (quand peu de documents pertinents sont restitués) et pour 40 % de rappel (40 % des documents pertinents restitués), notre méthode dépasse le Cosinus. Nous interprétons cela de la manière suivante : le Cosinus semble plus performant pour trouver les documents du corpus ayant une forte ressemblance directe à la requête. Il semble que le SimRank permette quant à lui, de faire remonter (en termes de rang dans la liste des documents restitués) les documents pertinents ayant un faible Cosinus. Comme nous l'avons constaté au fil de notre étude, le SimRank identifie également les documents ayant une similarité directe avec la requête, mais utilise ces documents pour distribuer des points de score aux documents auxquels ils sont reliés. Cela fait monter le score de documents reliés structurellement et réciproquement, cela fait également diminuer le score (et par conséquent le rang) des documents reliés directement, ainsi les documents ayant une forte ressemblance directe à la requête sont moins bien classés par SimRank que par Cosinus.

2.1.4 Significativité des mesures utilisées pour comparer les deux méthodes

L'analyse statistique des distributions montre que toutes les mesures utilisées ne suivent pas une distribution normale ($p < 0.05$; test de Shapiro-Wilk). Comme précédemment, pour étudier la significativité de la comparaison de OkaSim à OkaCos, nous utilisons le test de Mann-Whitney.

	OkaCos		OkaSim		p
	Moyenne [ET]		Moyenne [ET]		
p@5	0,2	[0.24]	0,132	[0.18]	0.14
p@10	0,176	[0.18]	0,130	[0.18]	0.10
p@30	0,126	[0.13]	0,090	[0.11]	0.10
p@100	0,085	[0.08]	0,068	[0.08]	0.11
R-Prec	0,125	[0.11]	0,097	[0.13]	0.09
MAP	0,122	[0.11]	0,106	[0.13]	0.13

Tableau 19 : test de Mann-Whitney sur les mesures utilisées pour comparer OkaSim et OkaCos

Le tableau 19 indique que les mesures utilisées ne permettent pas de différencier significativement les résultats des deux méthodes. Ce résultat était prévisible au regard des courbes précision-rappel qui se croisent : OkaCos est meilleur à taux de rappel inférieur à 40 %, OkaSim est meilleur à taux de rappel supérieur à 40 %. Une méthode n'est donc pas significativement meilleure que l'autre.

2.2 Cosinus puis Cosinus et Cosinus puis SimRank

Nous avons souhaité utiliser Okapi comme 1ère phase de notre méthode. Cette méthode est reconnue et présente l'avantage d'être différente de la mesure Cosinus et de la mesure SimRank. Ainsi nous proposons la méthode suivante :

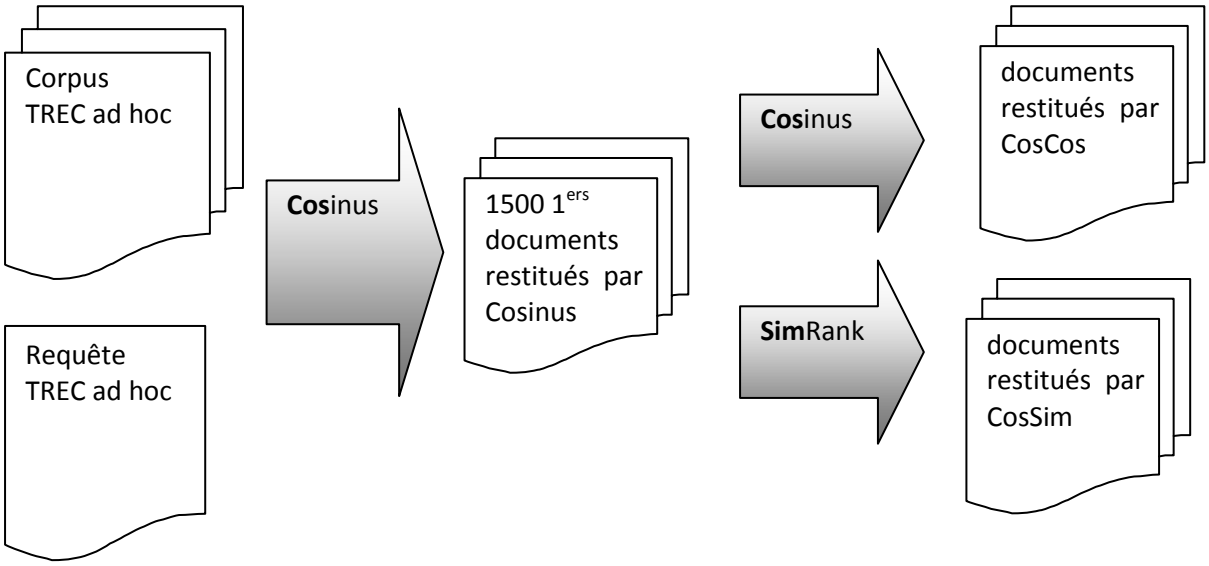


Figure 74: schéma des expérimentations sur TREC : CosCos et CosSim

2.2.1 Résultats obtenus à la Mean Average Precision

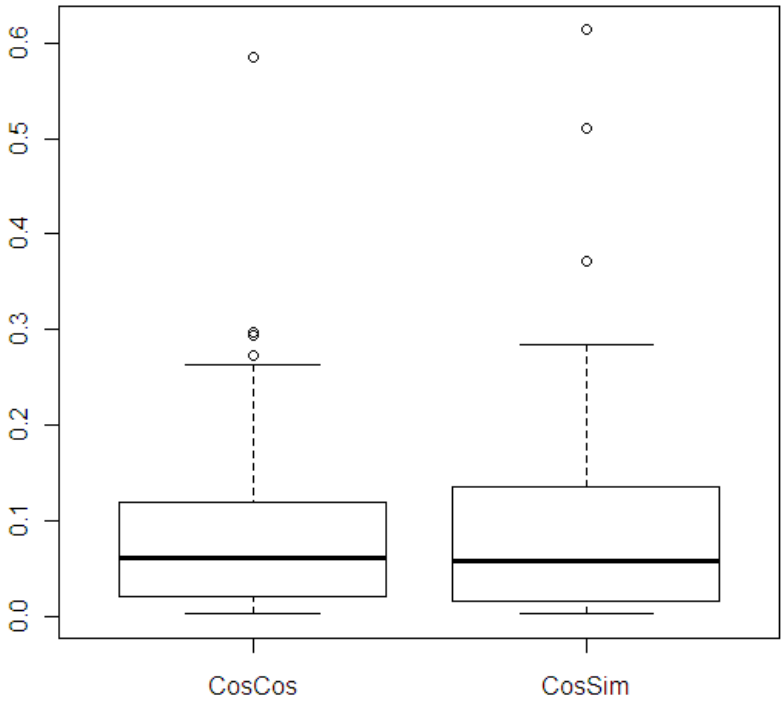


Figure 75 : représentation des MAP pour CosCos et CosSim sur TREC

Les MAP des deux méthodes CosCos et CosSim sont inférieures à celle de OkaCos et OkaSim de l'expérimentation 1 ; cela s'explique par le fait que Okapi classe mieux les documents que la méthode Cosinus pour ce corpus. Le but recherché ici n'est pas d'évaluer les performances, mais de comparer le comportement de notre méthode à celle de la méthode Cosinus, de manière à mettre en rapport notre similarité structurelle avec la similarité vectorielle. En termes de MAP, les deux mesures sont proches, avec un avantage de 7 % en moyenne pour notre méthode.

2.2.2 Résultats obtenus à la précision à 5, 10, 30, 100 documents restitués

	CosCos	CosSim	Gain
p@5	0,132	0,108	-18,1
p@10	0,124	0,112	-9,6
p@30	0,0786	0,084	6,7
p@100	0,0588	0,065	10,5
R-Prec	0,1029	0,109	6,4

Tableau 20 : précisions quand 5, 10, 30, 100 documents sont restitués et R-Prec moyennes pour CosCos et CosSim sur TREC

Le tableau 20 montre une supériorité de la mesure Cosinus en tête de liste (p@5, p@10) par rapport à notre méthode, puis à partir de 30 documents restitués, notre méthode prend l'avantage. La R-précision moyenne obtenue avec notre méthode est également supérieure à celle obtenue avec Cosinus.

2.2.3 Courbe précision-rappel

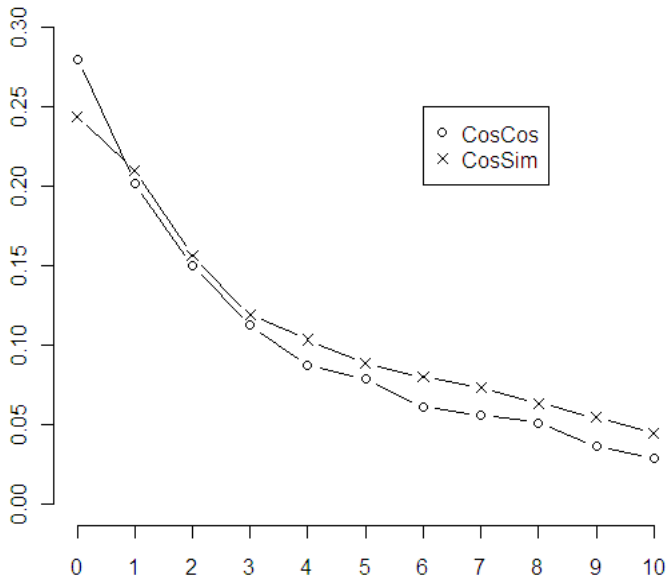


Figure 76 : courbe précision-rappel pour CosCos et CosSim sur TREC

Les courbes précision-rappel en 11 points de la figure 76 montrent une supériorité de la méthode CosCos par rapport à la méthode CosSim quand le rappel est compris entre 1 et 15 %. Au-delà de 15 %, la méthode CosSim prend l'avantage et creuse cet avantage au fur et à mesure que le rappel augmente. Cela nous permet de constater un effet du SimRank qui est de faire monter le rang de documents pertinents ayant une faible ressemblance directe.

2.2.4 Significativité des mesures utilisées pour comparer les deux méthodes

L'analyse statistique des distributions montre que touts les mesures utilisées ne suivent pas une distribution normale ($p < 0.05$; test de Shapiro-Wilk). Nous utilisons le test de Mann-Whitney :

	CosCos	CosSim	p
	Moyenne [ET]	Moyenne [ET]	
p@5	0,132 [0.21]	0,108 [0.16]	0.47
p@10	0,124 [0.17]	0,112 [0.17]	0.55
p@30	0,0786 [0.09]	0,084 [0.10]	0.88
p@100	0,0588 [0.06]	0,065 [0.07]	0.82
R-Prec	0,1029 [0.11]	0,109 [0.13]	0.94
MAP	0,0954 [0.11]	0,1022 [0.12]	0.98

Tableau 21 : test de Mann-Whitney sur les mesures utilisées pour comparer CosSim et CosCos

Le tableau 21 indique que les mesures utilisées ne permettent pas de différencier significativement les résultats des deux méthodes. CosCos est meilleur à taux de rappel inférieur à 15 %, CosSim est meilleur à taux de rappel supérieur à 15 %. Une méthode n'est donc pas significativement meilleure que l'autre.

Conclusion du chapitre Expérimentations

Nous avons expérimenté la méthode sur différentes collections. Notre étude s'est orientée vers les collections Cranfield et CISI, qui présentent l'avantage d'être de taille réduite et s'adaptent aux contraintes de notre algorithme. Nous avons étudié dans un premier temps la convergence de notre algorithme en situation réelle, nous avons déterminé expérimentalement que 7 à 8 itérations permettent d'atteindre la convergence et nous avons proposé différents réglages des paramètres de notre algorithme. Dans un second temps nous avons comparé la méthode SimRank aux méthodes Cosinus et Okapi. Sur le corpus Cisi, notre méthode obtient des résultats comparables à ceux des deux autres méthodes et dans certains cas les dépasse. Sur le corpus Cranfield, c'est Okapi qui obtient les meilleurs résultats, ceux obtenus par notre méthode sont légèrement inférieurs. Les analyses statistiques ont montré que les résultats entre les méthodes ne sont pas significatifs, à part pour la R-précision qui est à l'avantage du SimRank. Ces expérimentations nous ont permis de conclure que notre algorithme est utilisable pour ordonner des documents en fonction de requêtes.

Ensuite, nous avons muni notre algorithme de tri d'un algorithme de sélection permettant d'obtenir un paquet de documents pertinents plutôt que de simplement ordonner et retourner tous les documents en fonction d'une requête donnée. L'étude des scores nous a permis de mettre en évidence – pour les deux collections étudiées uniquement – un lien entre la taille de la requête et le score obtenu par le document au rang duquel la mesure d'évaluation retenue est maximale. Ce lien nous a permis de déterminer un seuil calculé. Ce seuil est du type $ax + b$. Les tests effectués nous ont permis de déterminer a, x et b pour chacune des deux collections. Il s'est avéré que les paramètres a et x du seuil était identiques pour les deux collections étudiées : a vaut 0,11 et x est le logarithme népérien du nombre de termes de la requête. Le paramètre b est différent d'une collection à l'autre : -0,11 pour Cisi, -0,04 pour Cranfield. Afin de proposer un seuil générique, nous avons choisi de garder a et x et de fixer b à une valeur entre la valeur de b calculée sur Cisi et celle calculée sur Cranfield. Nous avons évalué cette méthode de filtrage en la comparant au Cosinus muni d'un seuil calculé expérimentalement de manière à optimiser la mesure F utilisée pour évaluer les réponses. Notre méthode munie d'un seuil obtient de moins bons résultats que le Cosinus muni d'un seuil calculé sur les deux collections Cisi et Cranfield. Néanmoins il serait intéressant et nécessaire de tester ce seuil sur d'autres collections.

Après avoir étudié les propriétés de tri et de filtrage de notre algorithme, nous avons expérimenté une utilisation différente de notre algorithme, en l'utilisant en tant que « ré-ordonnanceur. L'idée est de trier les documents par rapport aux requêtes à l'aide d'une méthode de tri dite légère, de retenir les documents reconnus pertinents par cette méthode et de reclasser ces documents avec le SimRank. Les résultats obtenus avec Cosinus puis SimRank sur Cranfield montrent une amélioration de 20 % par rapport à l'usage du Cosinus seul.

Enfin, nous avons utilisé la collection TREC ad hoc 1998 pour compléter l'étude de notre algorithme. Notre étude confirme l'idée intuitive de notre algorithme qui est de restituer des documents pertinents ayant une ressemblance directe à la requête faible.

CONCLUSION GENERALE

Inspiré par des travaux de psychologie cognitive, nous avons souhaité introduire en RI le concept de similarité structurelle. La notion de similarité est centrale en RI. En effet, le but d'un SRI est de trouver les documents similaires à une requête formulée par un utilisateur. Le SRI pour répondre aux utilisateurs doit être en mesure de comparer les documents disponibles et la requête. Cette comparaison se fait le plus souvent sur la base des attributs communs aux documents et à la requête : les termes. L'hypothèse faite dans cette thèse est que l'utilisation seule des attributs communs pour la mise en correspondance des documents et de la requête est insuffisante, c'est-à-dire qu'elle ne permet pas forcément de trouver tous les documents pertinents par rapport à une requête. Notre point de vue est qu'il est possible qu'un document soit pertinent pour une requête donnée sans pour autant contenir les termes ou une partie des termes de la requête. Nous suggérons l'idée que des documents similaires à des documents reconnus similaires à la requête peuvent également être pertinents. Ainsi nous proposons d'utiliser les similarités structurelles pour traduire cette idée. Conceptuellement, nous utilisons les graphes : un document donné d'un corpus est considéré comme un nœud auquel sont connectés les nœuds associés à ses termes. La requête est un nœud particulier du graphe représentant le corpus. Il s'agit alors d'étudier les connexions indirectes du réseau de nœuds documents connectés au nœud représentant la requête.

Etudier le réseau de nœuds documents connectés à la requête correspond à donner à ces nœuds directement connectés une valeur de similarité de départ (la similarité directe) proportionnelle à la quantité et au poids des termes communs puis à propager les valeurs représentant la ressemblance au fil des liens document-terme afin d'attribuer à des documents indirectement connectés à la requête une certaine valeur de ressemblance à cette même requête. Nous avons défini un algorithme itératif qui traduit cette idée. Nous avons défini la similarité structurelle entre un document et une requête comme la moyenne des similarités des termes qui les composent. Réciproquement, la similarité structurelle entre termes est définie comme la moyenne des similarités entre les documents qui contiennent ces termes. Ainsi, nous avons créé une mesure de similarité structurelle entre documents (et entre termes), appelée SimRank et définie comme la limite atteinte par les valeurs propagées dans le graphe quand elles cessent d'évoluer. Après avoir défini cette mesure, nous avons testé ses propriétés de convergence d'une part et d'autre part ses propriétés de tri de documents par rapport à des requêtes. L'étude de notre algorithme sur différentes collections nous a permis de déterminer une partie de ses avantages ainsi qu'une partie de ses limites. Les résultats montrent que la mesure est comparable à des mesures existantes, ce qui est plutôt encourageant. Les principales limites de notre méthode sont liées à sa complexité. En effet, sa grande complexité ne permet pas de la mettre en œuvre pour de grandes collections documentaires, néanmoins nous avons proposé une façon de l'utiliser de manière à exploiter les similarités structurelles sur de grandes collections. Cette méthode consiste à effectuer une première sélection de documents pertinents à l'aide d'une méthode légère, puis de reclasser ces documents avec notre algorithme. Les résultats sont encourageants et nous permettent d'être confiants quant à l'intérêt d'une telle méthode. Les similarités structurelles ne s'opposent pas aux similarités directes comme on pourrait le penser au premier abord. En fait, les similarités structurelles découlent des similarités directes et par propagation enrichissent celles-ci d'informations structurelles non disponibles au départ. Le SimRank fonctionne pour trouver des documents pertinents ayant une ressemblance à la requête faible voire nulle. Ceci nous permet d'envisager différents usages des similarités structurelles : soit directement pour classer des documents, soit pour « repêcher » des documents ayant une faible ressemblance directe. De plus si la similarité structurelle n'est pas possible (car trop de biais), il est possible que son usage en deux temps le soit.

PERSPECTIVES

Nous envisageons plusieurs suites à nos travaux :

Perspectives à court terme :

- Notre algorithme semble avoir des propriétés qui le rendent performant au filtrage. Nous devons toutefois approfondir l'étude du seuil. Il faudrait déterminer les combinaisons de facteurs intervenant sur les scores calculés et si possible déterminer le paramètre *b* de la méthode de seuil par rapport à ces facteurs. Cela nécessite une analyse statistique. Pour être valable une étude statistique doit être effectuée sur un grand échantillon de données. Il faudrait donc réitérer les analyses effectuées sur le seuil avec d'autres collections adaptées aux contraintes, CACM par exemple.
- Une autre poursuite de type statistique qui pourrait être intéressante serait de déterminer par analyse de graphe si notre algorithme mérite d'être utilisé pour calculer les similarités entre documents pour une requête donnée. Nous avons vu sur les collections étudiées que parfois le Cosinus obtient de meilleurs résultats, mais pas toujours. Une analyse du graphe constitué par les documents pertinents par rapport à la requête pourrait peut-être permettre de déterminer des critères permettant de prédire l'efficacité du SimRank sur tel ou tel type de graphe.
- Toujours dans le but de caractériser les données de départ pour déterminer au préalable l'éventuelle efficacité de notre algorithme sur ces données, il serait intéressant de se pencher sur la typologie des requêtes étudiées [Mothe, 1995][Englmeier et al., 2006]. L'analyse des caractéristiques des requêtes pour lesquels notre algorithme est performant peut nous amener à découvrir des propriétés particulières de notre algorithme comme son efficacité sur certain type de requêtes.
- Nous souhaitons nous intéresser aux différences entre les tris produits par la mesure Cosinus et les tris produits par la mesure SimRank. En effet, le SimRank propage la similarité directe pour trouver les documents similaires indirectement. Cette similarité indirecte repose donc sur la similarité directe. Dans l'objectif de caractériser l'information purement structurelle, nous pouvons imaginer une formule du type « Cosinus – αSimRank » dont le résultat serait l'information liée aux relations indirectes. Avant d'arriver à une telle formule, il faut rendre les valeurs Cosinus et les valeurs SimRank comparables, ce qui n'est pas le cas actuellement. Si la mise à échelle n'est pas possible avec les formules (9) et (10) du chapitre 4, il faudra alors envisager des modifications, comme des ajouts de facteurs, ou des changements plus profonds comme la remise en question de la normalisation. Les formules actuelles utilisent la norme 1, si elles sont modifiées alors la convergence de l'algorithme est à prouver à nouveau et les analyses de vitesse de convergence sont à étudier à nouveau. Si nous résolvons les problèmes rencontrés, nous pouvons imaginer un SimRank adaptatif qui prendrait le meilleur du Cosinus ou le meilleur du SimRank pour attribuer un score de similarité à un document par rapport à une requête. Cette idée poursuit le même but que l'analyse statistique des caractéristiques du graphe représentant les documents et la requête.

Perspectives à long terme :

- Dans l'optique de l'utilisation du SimRank pour trier des documents par rapport à des requêtes, il faudrait faire évoluer nos formules de manière à prendre en compte le concept de réinjection de pertinence. La réinjection de pertinence consiste dans le modèle vectoriel à modifier le poids des termes des documents pertinents pour que les vecteurs représentant

les documents soient plus proches du vecteur requête. Par analogie, en portant cette méthode aux graphes il serait possible d'injecter de la similarité soit aux nœuds termes des documents reconnus comme pertinents soit aux nœuds documents directement. Réciproquement il serait possible d'injecter de la différence soit aux nœuds termes des documents reconnus non pertinents, soit aux nœuds documents directement. L'analyse l'impact de ces injections sur les scores de similarité document-requête pourrait alors être utile.

— Nous souhaitons également utiliser notre algorithme dans d'autres domaines représentables par des graphes. Un avantage de notre algorithme est qu'il retrouve beaucoup de documents liés à la requête même très indirectement. Cela semble être un désavantage en RI, mais pourrait bien être un avantage dans d'autres domaines. Considérons par exemple un graphe dont les deux types de nœuds sont les nœuds représentant des entreprises et les nœuds représentant la signature d'un brevet, ou d'un contrat industriel, il existe un lien entre un brevet et une entreprise si l'entreprise a déposé un brevet – ou signé un contrat industriel. Notre algorithme trouvera les entreprises directement liées à cette signature, mais surtout il découvrira les sociétés jouant un rôle indirect en tant que sous-traitant par exemple et il est possible que dans ces sociétés nous trouvons l'information nécessaire pour anticiper l'apparition d'une signature, ou du dépôt d'un brevet. Ce genre d'étude s'apparente à de la recherche de signaux faibles en veille stratégique [Dkaki et al., 1997].
— Une autre propriété de notre algorithme est sous-utilisée en RI : nous avons jusqu'à présent utilisé le SimRank d'une manière orientée vers les documents, le but de la RI étant de retrouver des documents pertinents par rapport à la requête, ce point de vue était le bienvenu. Pour déterminer la ressemblance entre un document et une requête, le SimRank procède au calcul de l'inter-ressemblance entre chaque document de la collection. Pour calculer la ressemblance entre deux documents, le SimRank se base sur la similarité entre les termes contenus dans ces documents. Deux informations sont disponibles à la fin du calcul des similarités : l'inter-similarité entre chaque terme de la collection et l'inter-similarité entre chaque document, requête comprise. L'inter-similarité entre chaque terme intervient dans le calcul des similarités inter-documents mais une fois le calcul effectué cette information est perdue. Or il se peut que des domaines utilisent la ressemblance structurelle entre termes. Notre algorithme pourrait alors être utilisé comme une mesure de corrélation de termes. Cette mesure refléterait la « corrélation structurelle ». Imaginons que nous disposions d'une base documentaire, comme celle d'un organe de presse, qui stocke des documents sur une longue période. Notre mesure appliquée aux termes de la collection peut déterminer ceux qui sont structurellement liés. Si la base compte suffisamment de documents sur une période suffisamment grande on peut analyser le lien structurel qui unit deux termes au fil du temps. Par exemple, le mot *Cyclone* est associé au mot *Katrina*, comme a pu l'être le mot *Hugo* quelques années auparavant. Ce lien aura certainement tendance à diminuer avec le temps et la mesure SimRank peut servir pour illustrer cela.

Bibliographie

[Adamson, 1974] Adamson, G., Boreham J. *The use of an association measure based on character structure to identify semantically related pairs of words and document titles*. In Information Storage and Retrieval, 10, p. 253–60, 1974.

[Ahuja et al., 1993] Ahuja, R.K., Magnanti, T.L., Orlin, J.B. *Network Flows : Theory, Algorithms, and Applications.* Prentice Hall, Englewood Cliffs, New Jersey, 1993.

[Aiello et al., 2000] Aiello, W., Chung, F., and Lu, L.. *A random graph model for massive graphs*. In Proceedings of the thirty-second annual ACM symposium on Theory of computing, p. 171-180.ACM Press, 2000.

[Alon et al., 1999] Alon, U., Surette, M.G., Barkai, N., Leibler, S. Nature (London) 397, p. 168-171, 1999.

[Andrews, 1971] Andrews, K. *The Development of a Fast Conflation Algorithm for English.* Dissertation for the Diploma in Computer Science, Computer Laboratory, University of Cambridge, 1971.

[Attneave, 1950] Attneave, F. *Dimension of similarity.* Journal of psychology. 63, p. 566-556. 1950.

[Baccini et al. 2010] Alain Baccini, Sébastien Déjean, Nongdo Désiré Kompaore, Josiane Mothe. *Analyse des critères d'évaluation des systèmes de recherche d'information.* Dans : *Technique et Science Informatiques*, Hermès Science Publications, 2010 (à paraître).

[Barnden, 1994] Barnden, J. A. *On the connectionist implementation of analogy and working memory matching.* In J. A. Barnden, & K. J. Holyoak (Eds.). Advances in Connectionist and Neural Computation Theory Volume 3: Analogy, metaphor, and reminding. Norwood, NJ: Ablex Publishing Corporation, 1994.

[Bassok et al., 1997] Bassok, M. and Medin, D.L. *Birds of a feather flock together: Similarity judgments with semantically rich stimuli*. In journal of Memory & Language, vol. 36, p. 311-336, 1997.

[Bearman ey al., 2004] Bearman, P., Moody, J., Stovel, K. Chains of Affection : The Structure of Adolescent Romantic and Sexual Networks. American Journal of Sociology, vol. 110, n°1, p. 44-91, 2004.

[Becker, 1969] Becker, J. D. (1969). *The modeling of simple analogic and inductive processes in a semantic memory system*. In Proceedings of IJCAI-69, Washington, DC, p. 655-668, 1969.

[Belew, 1989] Belew, R. K. *Adaptive information retrieval: Using a connectionist representation to retrieve and learn about documents*. In *ACM SIGIR Proceedings*. p. 11-20, 1989.

[Belkin et al., 1992] Belkin, N. & Croft, B. *Information Filtering and Information Retrieval: Two Sides of the Same Coin*. In *Communication of the ACM*, 1992.

[Bennet et al., 1998] Bennett, C. H., Gacs, P., Li, M., Vitanyi, P., & Zurek, W. *Information distance*. In IEEE Transactions on Information Theory, IT-44, p. 1407–1423, 1998.

[Berge, 1958] Claude Berge, Théorie des Graphes et ses Applications, Dunod, Paris 1958.

[Berger et al., 1996] Berger, A. L., Della Pietra, D., Stephen A. and Della Pietra, V. J., A Maximum *Entropy Approach to Natural Language Processing*, Computational Linguistics, vol. 22(1), p. 39-71, 1996.

[Bharat et al., 1998] Bharat K. and Herzinger M., *Inproved algorithms for topic dsitillation in hyperlinked environnements*. In Proceedings of the 21st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'98), p. 11-104, 1998.

[Bisson, 2000] Bisson G., *La similarité: une notion symbolique/numérique*. Apprentissage symbolique-numérique (tome 2). Eds Moulet, Brito. Editions CEPADUES, 2000.

[Blondel et al., 2004] Blondel V.D., Gajardo A., Heymans, M. *A Measure of Similarity between graphe vertice : Applications to Synonym extraction and web searching*. SIAM rev.46'4, p.647-666, 2004.

[Bookstein, 1983] Bookstein A.. Outline of a general probabilistic retrieval model. *Journal of Documentation*, vol. 39(2): p. 63-72, 1983.

[Borg et al. 2005] Modern Multidimensional Scaling: Theory and Applications (Second Edition). Ingwer Borg and Patrick jf Groenen. Springer, New York, 2005.

[Bornholdt et al., 2003] Bornholdt S., Schuster H.G., editors. Handbook of graphs and networks – from the genome to the internet. Wiley-VCH, 2003.

[Bornholdt et al., 2004] Stefan Bornholdt, Konstantin Klemm *Topology of biological networks and reliability of information processing* CoRR q-bio.MN/0409022, 2004.

[Boulet, 2009] Boulet Romain, Comparaison de graphes : applications à l'étude d'un réseau de sociabilité paysan au moyen âge. Thèse de doctorat de l'université de Toulouse, (France), 2009.

[Bracke, 1998] Bracke Danièle, Vers un modèle théorique du transfert: les contraintes à respecter. Revue des sciences de l'éducation, Vol. XXIV, n° 2, p. 235 à 266, 1998.

[Brin et al., 1998] Sergey Brin and Lawrence Page, *The anatomy of a large-scale hypertextual Web search engine*. Source, Computer Networks and ISDN Systems archive, vol. 30 , Issue 1-7, April 1998.

[Burge, 1998] Burges, C., *A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery,* vol. 2(2), p. 121-167, 1998.

[Callan et al., 1995] Callan, J.P., Croft, W.B., Broglio, J. *TREC and TIPSTER experiments with INQUERY*, Information Processing & Management, vol. 31(3), p. 327-343, 1995.

[Carbonell, 1983] Carbonell J.G. *Learning by Analogy: Formulating and Generalizing Plans from Past Experience, Machine Learning: An Artificial Intelligence Approach*, Volume I, Morgan Kaufmann Publishers Inc., Los Altos, California, 1983.

[Carriere et al., 1997] Carriere J. and Kazman R., *Webquery : Searching and visualizing the web through connectivity*, In Proceedings of the six International World Wide Web, p 701-711, 1997.

[Caroll et al., 1974] Carroll, J. D., & Wish, M. *Models and methods for three-way multidimensional scaling* In D. H. Krantz, R. C. Atkinson, R. D. Luce, & P. Suppes (Eds.) Contemporary developments in mathematical psychology vol. 2, p. 57-105, 1974.

[Champclaux et al., 2007] Yael Champclaux, Taoufiq Dkaki, Josiane Mothe. *Utilisation des similarités structurelles pour l'évaluation de la pertinence en Recherche d'information.* Dans : *Colloque Veille Stratégique Scientifique et Technologique (VSST 2007)*, *Marrakech (Maroc)*, *21/10/07-25/10/07*, VITA, (support électronique), 2007.

[Champclaux et al., 2008] Yael Champclaux, Taoufiq Dkaki, Josiane Mothe. *Enhancing high precision using structural similarities.* Dans : *IADIS International Conference WWW/Internet (ICWI 2008)*, *Freiburg, Germany*, *13/10/08-15/10/08*, p. 494-498, 2008.

[Champclaux et al., 2009] Yael Champclaux, Taoufiq Dkaki, Josiane Mothe. *Enhancing high precision by combining okapi BM25 with structural similarity in an information retrieval al system.* Dans : *International Conference on Enterprise Information Systems (ICEIS 2009)*, *Milan*, *05/05/09-10/05/09*, INSTICC Press, p. 279-285, 2009.

[Chemlal et al., 2006] Chemlal Soulaiman and Cordier Françoise, *Structures conceptuelles, représentation des objets et des relations entre les objets*. Canadian journal of experimental psychology, vol. 60, n°1, p. 7-23, 2006.

[Chen et al., 1990] H. Chen, V. Dhar, *Online query refinement on Information Retrieval System: a process of searcher/system interactions*, Conference on Research and Development in Information Retrieval (SIGIR), p. 115-133, 1990.

[Chevallet et al., 1998] Jean Pierre Chevallet and Yves Chiaramella, *Experiences in IR Modeling using Structured Formalisms and Modal Logic*, in Information Retrieval, Uncertainty and Logics, Fabio Crestani, Mounia Lalmas, Cornelis Jost "Keith" van Rijbergen, University of Glasgow Scotland, Kluwer Academic Publisher, chapter 3, p39-72, 1998.

[Chiaramella et al.1990] Y. Chiaramella, J.Y. Nie *A retrieval model based on an extended modal logic and its application to the RIME experimental approach*. 13th ACM- SIGIRconference, ed. J.-L. Vidick, Brussels, p. 25-43, 1990.

[Chrisment et al., 2006] Chrisment C., Dkaki T., Dousset B., Karouach S., Mothe J., *Combining Mining and Visualization Tools to Discover the Geographic Structure of a Domain*. Computers, Environment and Urban Systems Journal Elsevier, vol. 30(4): p. 460-484. 2006.

[Cleverdon, 1962] Cleverdon, C. W. *Report on the testing and analysis of an investigation into the comparative efficiency of indexing systems.* Cranfield, U.K.: College of Aeronautics. Aslib Cranfield Research Project, 1962.

[Cooper, 1971] Cooper, W. *A definition of relevance for information retrieval*. Dans *Information Storage and Retrieval*, 1971.

[Cooper, 1988] Cooper, W. Getting Beyond Boole, *Information Processing & Management,* p. 24:3, 1988.

[Cornuejols, 2002] Cornuejols Martine, 2002. *L'organisation des réseaux sémantiques verbaux et imagés : De l'analyse psycholinguistique au traitement automatique du langage dans Bulletin de linguistique appliquée et générale*. n° 27, p. 39-56, 2002.

[Crestani et al., 1998] Crestani, F., Lalmas, M., van Rijsbergen, C.J., Campbell L.: ``*Is This Document Relevant? ... Probably'': A Survey of Probabilistic Models in Information Retrieval*. ACM Computing. Surveys. vol. 30(4), p. 528-552, 1998.

[Crestani, 1995] Crestani F. *Implementation and evaluation of a relevance feedback device based on neural networks*. In J. Mira and J. Cabestany, editors, From Natural to Artificial neural Computation: International Workshop on Artificial Neural Networks, vol. 930 of Lecture Notes in Computer Science, p. 597–604. Springer-Verlag, Malaga, Spain, June 1995.

[Crestani et al., 2001] Fabio Crestani et Mounia Lalmas *Logic and uncertainty in information retrieval*. Dans *Lecture Notes in Computer Science*, 2001.

[Croft, 1995] Croft, W.B. *What Do People Want from Information Retrieval?* D-Lib Magazine, http://www.dlib.org/dlib/november95//11croft.html, November 1995.

[Croft et al., 2003] Croft, W. B., & Lafferty, J. (Eds.). Language modeling for information retrieval. In No. 13 in Information Retrieval Book Series. Kluwer, 2003.

[Croft, 2007] Croft, W.B. *Learning about Ranking and Retrieval Models. Keynote Speech. In :SIGIR 2007 workshop on* learning to rank for Information Retrieval, 2007.

[D'Amato et al., 1988] D'Amato, M.R. & Van Sant, P. *The person concept in monkeys (Cebus Apella)*. Journal of Experimental Psychology: Animal Behavior Processes, 14, p. 43-55, 2007.

[Dawson, 1974] Dawson, J.L. *Suffix Removal and Word Conflation*. ALLC Bulletin, Michaelmas p. 33-46, 1974.

[Decker et al., 1999] Decker, S., Erdmann, M., Fensel, D., Studer, R. *Ontobroker : ontology based access to distributed and semi structured information* in R. Meersman et al., editor, DS-8: Semantic Issue in multimedia Systems. Kluwer academic publisher, 1999.

[Dempster et al., 1977] Dempster, A. P., Laird, N. M. et Rubin, D. B. *Maximum likelihood from incomplete data via the em algorithm (with discussion)*. Journal of the Royal Statistical Society, B 39, p. 1–38, 1977.

[Dervin et al., 1983] Dervin, B. and Nilan M. *Information needs and uses* in M. Williams(Ed.), Annual Review of Information Science & Technology, vol. 21, p. 1-25. White Plains, NY: Knowledge Industry, 1986.

[Diday, 1991] Diday, E. Des objets de l'analyse des données à ceux de l'analyse des connaissances. Dans *Induction Symbolique et Numérique à partir de données*. CEPADUES, 1991.

[Dkaki, 1997] T.Dkaki, B. Dousset, J. Mothe *Mining information in order to extract hidden and strategical information*, 5th International Conference on Computer Assisted Information Retrieval, RIAO 97, p. 32-51, 25-27, Montreal, 1997.

[Dousset, 2009] Dousset B., Extraction de l'information implicite par analyse textuelle de sites internet en UNICODE. Dans : Colloque Veille Stratégique Scientifique et Technologique (VSST 2009), (support électronique), 2009.

[Englmeier et al., 2006] Kurt Englmeier, Gilles Hubert, Josiane Mothe, *Distinguer les requêtes pour améliorer la recherche d'information XML.* Conférence en Recherche d'Information et applications (CORIA 2006), p. 41-52, 2006.

[Erdös et al., 1959] Erdös P. and A. Rényi A., *On random graphs*. Publicationes Mathématicae, vol. 6, p. 290-297, 1959.

[Eskeridge, 1994] Eskeridge, T. *A hybrid model of continuous analogical reasoning*. In Holyoak, K. and Barnden. J. (Eds.) Advances in Connectionist and Neural Computation Theory. Volume 2: Analogical Connections, NJ:Ablex, p. 207-246, 1994.

[Evans, 1968] Evans, T. *A program for the solution of a class of geometric analogy intelligence test questions*. In Semantic Information Processing, p. 271-353. Cambridge, MA: MIT Press, 1968.

[Fagan, 1987] Fagan, Joel .L. *Experiments in automatic phrase indexing for document retrieval: A comparison of syntactic and non-syntactic methods*. Doctoral thesis, Report p. 87-868, Department of Computer Science, Cornell University, Ithaca, NY; September 1987.

[Fagan, 1988] Fagan, Joel L. *Experiments in Automatic Phrase Indexing for Document Retrieval: A Comparison of Syntactic and Nonsyntactic Methods*. Unpublished Ph.D. dissertation, Cornell University. 1998.

[Ferguson, 1994] Ferguson, R. W. 1994. *MAGI: Anology-based encoding using symmetry and regularity*. In Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society, NJ:LEA, p. 283-288, 1994.

[Fleischner, 1990] H. Fleischner, *Eulerian Graphs and Related Topics*. In Annals of Discrete Mathematics 45, North-Holland, Amsterdam (ISBN 0-444-88395-9), 1990.

[Forbus et al., 1994] Forbus, K. & Ferguson, R. & Gentner, D. *Incremental Structure-Mapping*. Proceedings of the Sixteenth Annual Meeting of the Cognitive Science Society. NJ: LEA, p. 313-318, 1994.

[Forbus et al., 1995] Forbus, K. D., Gentner, D., & Law, K. *MAC/FAC: A model of similarity-based retrieval*. Cognitive Science, 19, p. 141-205. (Abridged version to be reprinted in Cognitive Modeling, by T. Polk & C. M. Seifert, Eds., in press, Boston: MIT Press), 1995.

[Fox et al., 1988] Fox, E. & Koll, M. *Partial Enhanced Boolean Retrieval: Experiments with the SMART and SIRE Systems*, Information Processing & Management, p. 24:3, 1988.

[Frakes et al., 1992] Frakes, W.B. *Stemming Algorithms*. In: Frakes, W.B., Baeza-Yates, R. (eds.): InformationRetrieval Data Structures and Algorithms. Prentice Hall, New Jersey p. 131-160, 1992.

[French, 2002] French, R. M. The Computational Modeling of Analogy -Making. *Trends in Cognitive Sciences, 6*(5), p. 200-205, 2002.

[Fuhr, 1989] Fuhr, N. *Models for retrieval with probabilistic indexing*. Information processing and management, vol. 25(1), p. 55–72, 1989.

[Furnas et al., 1988] G. Furnas, S. Deerwester, S. Dumais, T. Landauer, R. Harshman, L. Streeter, and K. Lochbaum. *Information retrieval using a singular value decomposition model of latent semantic structure*. In Proceedings of ACM SIGIR 88, p. 465–480, 1988.

[Gaines et al., 1994] Gaines, B.R., Shaw, M.L.G. *Concept maps indexing multimedia knowledge bases*. In AAAI94 Workshop : Indexing and reuse in multimedia systems, Menlo Park, CA, 1994.

[Gaines et al., 1995], Gaines, B.R., Shaw, M.L.G. *Collaboration through concept maps.* In Proceedings of CSCL95, Computer Supported Cooperative Learning, Bloomington, 1995.

[Garner, 1974] Garner W.R. *The processing of information and structure*. New York Wiley, 1974.

[Gardenfors, 2000] Gardenfors P. *Conceptual spaces : the geometry of though*. Cambridge, MIT Press.

[Gati et al., 1984] Gati, I., & Tversky, A. *Weighting common and distinctive features in perceptual and conceptual judgments*. Cognitive Psychology, vol. 16, 341-370, 1984.

[Gaume et al., 2006] Gaume B., Duvignau K. et Mas J-M. *Ballades Aléatoires dans les Petits mondes Lexicaux*. Technologies langagières et apprentissage des langues. ACFAS, Montréal, 2006.

[Gaussier et al., 2002] Gaussier, E., Goutte, C., Popat, K., Chen, F. *A hierarchical model for clustering and categorising documents. Advances in Information Retrieval*. Proceedings of the 24th BCS-IRSG European Colloquium on IR Research ECIR-02, Glasgow. Lecture Notes in Computer Science 2291, p. 229-247, Springer, 2002.

[Gentner 1983] Gentner, D. Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, 7, 155-170. (Reprinted in A. Collins & E. E. Smith (Eds.), *Readings in cognitive science: A perspective from psychology and artificial intelligence*. Palo Alto, CA: Kaufmann), 1983.

[Gentner et al. ,1989] Gentner, D. *The mechanisms of analogical learning*. *In* S. Vosniadou et A. Ortony (dir.), *Similarity and analogical reasoning,* p. 199-241. Cambridge: Cambridge University Press. 1989.

[Gentner et al., 1997] Gentner, D., & Markman, A. B. *American Psychologist*, vol. 52, p. 45-56, 1997.

[Gibson et al., 1998] Gibson, David and Kleinberg, Jon and Raghavan, Prabhakar *Inferring Web communities from link topology.* In Proc. 9th ACM Conference on Hypertext and Hypermedia, 1998.

[Goldsmith et al., 1999] Goldsmith, J., Reutter, T. *Automatic Collection and Analysis of German Compounds*. In: Busa, F., Mani, I., Saint-Dizier, P. (eds.): The Computational Treatment of Nominals: Proceedings of the Workshop COLING-ACL '98. COLING-ACL, Montreal p. 61-69, 1999.

[Goldstone, 1991] Goldstone, R. L. Similarity. in R.A. Wilson & F. C. Keil (eds.) MIT encyclopedia of the cognitive sciences. MIT Press: Cambridge, MA. 1991.

[Goldstone, 1994] Goldstone R.L. Similarity, interactive interaction, and mapping. Journal of ExperimentalPsychology : Learning, Memory and cognition, vol. 20, p. 3-28, 1994.

[Goldstone, 1995] Goldstone, R.L. *Mainstream and avant-garde similarity*. Psychologica Belgica, vol. 35, p. 145-165, 1995.

[Goldstone et al., 2005] Goldstone, R. L, & Son, J. *Similarity*. In K. Holyoak & R. Morrison (Eds.). *Cambridge Handbook of Thinking and Reasoning*. Cambridge: Cambridge University Press. p. 13-36, 2005.

[Goldstone et al., 2009] Goldstone R.L., Day S., Son J.Y. (in press). Comparison. In B. Glatzeder, V. Goel, & A. von Müller (Eds.) On thinking: Volume II, towards a theory of thinking. The Parmenides Foundation. 2009.

[Green, 1990] Green, A. What do we mean by users need? British Journal Academic Librarianship 5, p. 65-78, 1990.

[Grossman et al., 98] David Grossman and Ophir Frieder *Ad Hoc Information Retrieval: Algorithms and Heuristics*, Kluwer Academic Publishers, 1998.

[Guillaume et al., 2004] Jean-Loup Guillaume and Matthieu Latapy. *Bipartite structure of all complex networks*. Information Processing Letters (IPL), vol. 90 p. 215-221, 2004.

[Guiraud, 1967] les Structures étymologiques du lexique français, Larousse, Paris, 1967.

[Halford, 1992] Halford, G. S *Analogical reasoning and conceptual complexity in cognitive development*. *Human Development,* vol. 35, p. 193-217, 1992.

[Hampton, 1995] Hampton J.A. *Testing the prototype theory of concepts*. J. Mem. Lang. vol. 34: p. 686–708, 1995.

[Harter, 1975] Harter, S. An algorithm for probabilistic indexing. Journal of the American Society for Information Science vol. 26(4), p. 280–289., 1975.

[He, et al., 2008] Chuan He; Cong Wang; Yi-Xin Zhong; Rui-Fan Li. *A survey on learning to rank*. In Proceedings of the 2008 International Conference on Machine Learning and Cybernetics, vol. 3, p.1734-1739, 2008.

[Herrnstein et al., 1964] Herrnstein, R. J. & Loveland, D. H. *Complex visual concepts in the pigeon*. Science, 146, p. 149-151, 1964.

[Hiemstra, 2002] Djoerd Hiemstra *Term-specific smoothing for the language modeling approach to information retrieval: the importance of a query term*. SIGIR 2002: p. 35-41, 2002.

[Hintzmann, 1986] Hintzmann, D.L. *Schema abstraction in a multiple trace memory model*. PsychologicalReview, vol. 93, p. 411–428, 1986.

[Hofstadter, 1984] Hofstadter, D. R. *The Copycat project: An experiment in non determinism and creative analogies*. MIT AI Memo No. 755, Cambridge, MA, 1984.

[Holyoak et al., 1987]Holyoak, K. J ., & Koh, K. *Surface and structural similarity in analogical transfer. Memory and Cognition*, vol. 15(4), 332-340, 1987.

[Holyoak et al., 1989] Holyoak, K. J. et Thagard, P. *Analogical mapping by constraint satisfaction*. Cognitive Science*, 13*, p. 295-355, 1989.

[Holyoak et al., 1995] Holyoak, K. J. et Thagard, P. *Mental leaps: Analogy in creative thought.* Cambridge, MA: MIT Press, 1995.

[Horgan et al., 1989] Horgan D. D., Millis K., Neimeyer R. A. *Cognitive reorganization and the development of chess expertise*. International Journal of Personal Construct Psychology, 2, p. 15-36, 1989.

[Hubel et al., 1968] Hubel, D. H., & Wiesel *Receptive fields and functional architecture of monkey striate cortex*. Journal of Physiology, vol. 195, p. 215-243, 1968.

[Hull, 1996] Hull, D. *Stemming algorithms - A case study for detailed evaluation*. Journal of the American Society for Information Science 47 p. 70-84, 1996.

[Hummel, 1997] Hummel, J. E. & Holyoak, K.J. *Distributed representations of structure: A theory of analogical access and mapping*. Psychological Review, vol. 104, p. 427-466. 1997.

[Hummel, 2000] Hummel, J.E. *Where view-based theories break down: The role of structure in shape perception and object recognition*. In E. Dietrich and A. Markman (Eds.). Cognitive Dynamics: Conceptual Change in Humans and Machines. Hillsdale, NJ: Erlbaum, 2000.

[Hummel, 2001] Hummel, J.E. *Complementary solutions to the binding problem in vision: Implications for shape perception and object recognition*. Visual Cognition, 8, p. 489-517, 2001.

[Iamnitchi et al., 2004] Iamnitchi, A., Ripeanu, M., et Foster, I. *Small-wold file-sharing communities*. INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies, vol.2, p. 952-963, 2004.

[Ingwersen, 1991] Ingwersen P., Intermediary function in Information retrieval interaction. Copenhagen Buisness School. Faculty of Economics, Compenhagen : Samfundsliterrature. Doctoral dissertation, 1991.

[Jacquemin, 1999] Jacquemin, C., Tsoukermann, E. *NLP for term variant extraction: synergy between morphology, lexicon, and syntax*. In: Strzalkowski, T. (ed.) Natural Language Information Retrieval. Kluwer, p. 25-74, Dordrecht 1999.

[Jaccard, 1901] Jaccard, P. *Distribution de la flore alpine dans la Bassin de Dranses et dans quelques regions voisines.* Bulletin de la Société Vaudoise des Sciences Naturelles 37: 241–272, 1901.

[Jakobson, 1963] Jakobson, R. Fant, G., & Halle, M. *Preliminaries to speech analysis: the distinctive features and their correlates*. Cambridge, MA: MIT Press, 1963.

[James, 1890] James. The principle of psychology. Dover, New York, Original work published 1890.

[James, 1985] James, W. Psychology : The briefer course. Notre Dame. University of Notre-Dame Press, 1985.

[Jani, 2000] Jani, N. & Levine, D. *A Neural Network Theory of Proportional Analogy-Making.* Neural Networks, 13, p. 149-183, 2000.

[Jeh, 2002] Glen Jeh, Jennifer Widom: SimRank: a measure of structural-context similarity. KDD. p. 538-543, 2002.

[Jensen et al., 1995] T.R. Jensen, B. Toft, *Graph Coloring Problems.* Wiley, ISBN 0-471-02865-7, New York, 1995.

[Jelinek, 1997] Jelinek,F. *Statistical methods for speech recognition*. MITPress, Cambridge, 1997.

[Joachims et al., 2005] Joachims, T., L. Granka, B. Pan, H. Hembrooke, and G. Gay *Accurately interpreting clickthrough data as implicit feedback*. In Proceedings of the 28th ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'05), p. 154–161, 1997.

[Kaski et al., 1998] Samuel Kaski, Jari Kangas, and Teuvo Kohonen *Bibliography of Self-Organizing Map (SOM) Papers: 1981-1997*, Neural Computing Surveys, 1: p. 102-350, 1998.

[Katz et al., 1963] Katz, J. J., & Fodor, J. *The structure of semantic theory*. Language, vol. 39, p. 170-210, 1963.

[Kheirbek, 1995] Amar Kheirbek, Yves Chiaramella *Integrating hypermedia and information retrieval with conceptual graphs*. In HIM95, Konstanz, Germany, Avril 95.

[Kleinberg, 1998] J. Kleinberg *Authorative sources in a hyperlinked environnement*, Proc. ACM-SIAM Symposium on Discrete Algorithms, 1998. Also appears as IBM Research Report RJ 10076(91892) May 1997.

[Kling, 1971] Kling, R. *A paradigm for reasoning by analogy*. Artificial Intelligence, 2, p. 147-178, 1971.

[Klingbiel, 1973] Klingbiel, P.H. Machine aided indexing of technical literature. Information Storage and Retrieval 9:2; p. 79-84; February 1973.

[Knap et al., 1984] Knapp, A.G., & Anderson, J.A. *Theory of categorization based on distributed memory storage*. Journal of Experimental Psychology: Learning, Memory and Cognition, vol. 10(4), 616-637, 1984.

[Kokinov, 1994] Kokinov, B. 1994. *The context-sensitive cognitive architecture DUAL*. In Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society. NJ:LEA, 502-507, 1994.

[Komatsu, 1992] Komatsu, L. K. 1992. *Recent views of conceptual structure*. Psychological Bulletin, 112, p. 500-526, 1992.

[Koskenniemi et al., 1996] Koskenniemi, K. *Finite-state morphology and information retrieval*. In: Proceedings of the ECAI-96 Workshop on Extended Finite State Models of Language ECAI, Budapest, Hungary 42-5, 1996.

[Kraaij et al., 1996] Kraaij, W., Pohlmann, R. *Viewing stemming as recall enhancement*. In: Proceedings, 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '96) Zurich p. 40-48, 1996.

[Krovetz, 1993] Krovetz, R. *Viewing morphology as an inference process*. In: Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval p. 191-202, 1993.

[Krumhansl, 1978] Krumhansl, C. L. *Concerning the applicability of geometric models to similarity data: The interrelationship between similarity and spatial density*. Psychological Review, vol. 85, p. 450-463, 1978.

[Kruschke , 1992] Kruschke JK. *ALCOVE: an exemplar-based connectionist model of category learning*. Psychol. Rev.99: p. 22–44, 1992.

[Kuehne et al., 2000] Kuehne, S. E., Forbus, K. D., Gentner, D., & Quinn, B. *SEQL: Category learning as progressive abstraction using structure mapping*. Poster session presented at the Twenty-Second Annual Conference of the Cognitive Science Society, Philadelphia, PA, 1984.

[Kuhlthau, 1990] Kuhlthau, C.; Turock, B.; George, M. & Belvin, R. *Validating a model of the search process: a comparison of academic, public and school library users*, Library & Information Science Research, p. 12:1, 1990.

[Kuhlthau, 1993] Kuhlthau, Carol. *Seeking Meaning: A process approach to Library and information services*. Norwood, NJ: Ablex. 1993.

[Kumar et al., 2000] R. Kumar, P. Raghavan, S. Rajagopalan, D.Sivakumar, A. Tomkins, and E. Upfal. *Stochastic models for the web graph*. In Proceedings of the 42st Annual Symposium on Foundations of Computer Science, p. 57. IEE Computer Society, 2000.

[Kumar et al., 2009] Kumar, Ch., Aswani, S., Srinivas. *On the Performance of Latent Semantic Indexing Based Information Retrieval*. In Journal of Computing and Information Technology (accepted), 2009.

[Kwok, 1989] Kwok, K. L. *A neural network for probabilistic information retrieval*, In *ACM SIGIR Proceedings,* p. 21-30, 1989.

[Lancaster, 1968] Lancaster, F. W. *MEDLARS: Report on the evaluation of its operating efficiency*. American Documentation, vol. 20, p. 119–148, 1968.

[Lance et al., 1967] Lance G.N., Williams W.T. *A General Theory of Classification Sorting Strategies: 1=Hierarchical Systems, 2=Clustering systems*. Computer Journal vol. 9-10, p. 373-380, 1967.

[Lange et al., 1993] Lange, T. E., & Wharton, C. M. *Dynamic memories: Analysis of an integrated comprehension and episodic memory retrieval model*. In Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society. Hillsdale, NJ: Erlbaum, 1993.

[Lassaline, 1996] Lassaline, M. E. *Structural alignment in induction and similarity*. Journal of Experimental Psychology: Learning, Memory, and Cognition, vol. 22, p. 754–770, 1996.

[Latapy, 2007] Matthieu Latapy, Mémoire d'habilitation à diriger des recherches, Université Pierre et Marie Curie (UMPC – Paris6), 2007.

[Latapy et al., 2008] Matthieu latapy, Clémence Magnien, Nathalie Del Vecchio *Basic Notions for the Analysis of Large Two-mode Networks*. Social Networks 30 (1), p. 31-48, 2008.

[Lawler et al., 1987] Lawler, E.L, Lenstra, J.K., Rinnooy Kan, A.H.G, Shmoys D.B. *The Traveling Salesman Problem: a Guided Tour of Combinatorial Optimization*. J. Wiley and sons, ISBN 0-471-90413-9, New York, 1987.

[Lenon, 1981] Lennon, M., Pierce, D.C., Willett, P. *An evaluation of some conflation algorithms*. Journal of Information Science 3 177-183, 1981.

[Lesk, 1969] Lesk, M.E. *Word-word associations in document retrieval systems*. American Documentation 20:1; 27-38; p. 11-16, January 1969.

[Lewis et al., 1989] Lewis, D., Croft, W.B. and Bhandaru, N. *Language-oriented information retrieval*. Int. J. Intell. Syst. 4, 285-318.

[Lewis, 1990] Lewis D. D. and Croft.W. B. *Term clustering of syntactic phrases*. University of Massachusetts, Colins Technique Report p. 90-71, 1990.

[Lhun, 1955] Luhn, H.P. *A new method of recording and searching information*. American Documentation vol. 4:1; p. 14-16; 1955.

[Li et al., 1997] Li, M., & Vitanyi, P. An introduction to Kolmogorov complexity and its applications, (2nd ed.). NewYork: Springer-Verlag, 1997.

[Li et al., 2001] Li, M., Li, X., Ma, B., & Vitanyi, P. 2001. *Normalized information distance and whole mitochondrial genome phylogeny analysis*, submitted for publication.

[Lin, 1989] Lin, X. *A fuzzy model of document representation based on neural nets*. In ASIS 1989 Doctoral Research Forum. Available from the author, College of Library and Information Services, University of Maryland, College Park, MD 20742, 1989.

[Littré] Dictionnaire de la langue française de Emile Littré 2eme édition (1872-1877).

[Liu et al., 2002] Liu, X., Gong, Y., Xu, W. *Document Clustering with Cluster Refinement and Model Selection Capabilities*. ACM/SIGIR Conference on Research and Development in Information Retrieval, Tampere, Finlande, p. 191-198, 2002.

[Loiseau, 2004] Recherche flexible d'information par filtrage flou qualitatif. Thèse de doctorat, Université Paul Sabatier, Toulouse, décembre 2004.

[Lovász, 1986] L. Lovász, M.D. Plummer, *Matching Theory*, Annals of Discrete Mathematics 29, North-Holland, ISBN 0-444-87916-1 - et aussi Akadémia Kiadó, Budapest, 1986.

[Mandelbrot, 1965] Mandelbrot, Benoît. Information Theory and Psycholinguistics. in B.B. Wolman and E. Nagel. *Scientific psychology*. Basic Books. 1965.

[Malt, 1989] Malt, B.C. *An one-line investigation of prototype and exemplar strategies in classification*. Journal of Experimental Psychology: Learning, Memory and Cognition, vol. 15(4), p. 539-555. 1989.

[Marcionini, 1992] Marchionini, G. Interfaces of End-User Information Seeking, *Journal of American Society for Information Science*, vol. 43 (2), p. 156-163, 1992.

[Marcotorchino, 1991] Marcotorchino F. *La classification automatique aujourd'hui : bref aperçu historique applicatif et calculatoire*. Publications Scientifiques et Techniques d'IBM. Numéro 2. p. 35-94, 1991.

[Marcus 1991] Marcus, R. *Computer and Human Understanding in Intelligent Retrieval Assistance*, American Society for Information Science, 28, 1991.

[Marcus, 1994] Marcus, R. *Intelligent Assistance for Document Retrieval Based on Contextual, Structural, Interactive Boolean Models*. In Proceedings RIAO'94: Intelligent Multimedia Information Retrieval Systems and Management, Oct., 1994.

[Margulis, 1993] Margulis,E. *Modelling documents with multiple poisson distributions*. Information Processing and Management, vol. 29, p. 215–227, 1993.

[Markman et al., 1993] Markman A.B., Gentner D. *Structural alignement during similarity comparison*. Cognitive Psychology,vol. 25, p. 431-467, 1993.

[Maron et al., 1960] Maron, M. and Kuhns J. *On relevance, probabilistic indexing and information retrieval*. *Journal of the ACM*, vol. 7: p. 216-244, 1960.

[Marr et al., 1978] Marr, D., & Nishihara, H.K. *Representation and recognition of three dimensional shapes*. Proceedings of the Royal Society of London, Series B, 200, p. 269-294, 1978.

[Marshall, 1997] Marshall, J. and Hofstadter, D. *The Metacat Project: A Self-Watching Model of Analogy Making*. Cognitive Studies: Bulletin of the Japanese Cognitive Science Society, special issue on similarity and analogical reasoning, vol. 4(4), p. 57-71, 1997.

[Martin, 1996]Martin, Ph. *Exploitation de graphes conceptuels et de documents structurés et hypertextes pour l'acquisition de connaissances et la recherche d'informations.* Ph.D thesis, University of Nice - Sophia Antipolis, France, October 14, 1996.

[Martin et al., 1999] Martin Ph., Eklund P. *Embedding Knowledge in web documents : CGs versus XML-based metada languages*. In ICCS 1999 7th Conference on Conceptual Structures, Springer Verlag. LNAI 1640 p. 230-246, 1999.

[McGraw, 1995] McGraw, G. *Letter Spirit (part one): Emergent High-Level Perception of Letters Using Fluid Concepts.* Ph.D. dissertation, Indiana University, Bloomington, http://www.cogsci.indiana.edu/farg/mcgrawg/thesis.html, 1995.

[Medin et al., 1978] Medin, D. L ., & Schaffer, M. M. *Context theory of classification*. Psychological Review, vol. 85(3), p. 207-238, 1978.

[Medin et al., 1989] Medin, D. et Ortony, A. *Psychological essentialism*. *In* S. Vosniadou et A. Ortony (dir.), *Similarity and analogical reasoning* p. 179-195. Cambridge: Cambridge University Press, 1989.

[Medin et al., 1993] Medin, D., Goldstone, R. L. et Gentner, D. Respect for similarity. *Psychological Review, vol. 2*, p. 254-278, 1993.

[Melançon, 2006] Melançon, G. *Just how dense are dense graphs in the real world ? A methodological note*. In E. Bertini, C. Plaisant, et G. Santucci (Eds.), BELIV Workshop (AVI Conference), Venice, Italy, p. 75–81. ACM Press, 2006.

[Metzler, 2004] Metzler, D. and W. Croft. *Combining the language model and inference network approaches to retrieval*. Information Processing and Management, vol. 40(5), p. 735–750, 2004.

[Milgram et al., 1969] Milgram S., Travers J. *An experimental study of the Small World Problem*, Sociometry, vol 32, no.4, p. 425-443, 1969.

[Miller et al., 1990] Miller G.A., Beckwith R., Fellbaum C., Gross D. & Miller K. (1990). *Five Papers on WordNet*. CSL Report 43, Cognitive Science Laboratory, Princetown University, July 1990.

[Miner, 1987] Miner E., 1987. *Some Theoretical and Methodological Topics for Comparative Literature*, Poetics Today 8, p.137, 1987.

[Mitchell, 1993] Mitchell, M. *Analogy-making as Perception: A computer model*. Cambridge, MA: MIT Press, 1993.

[Miyamoto, 1990] Miyamoto Sadaaki *Fuzzy sets in information retrieval and cluster analysis*. Theory and decision library, Kluwer Academic Publishers, Dordrecht, Boston, London, 1990

[Moers, 1961] Mooers, Calvin. 1961. *From a point of view of mathematical etc. techniques*. In R. A. Fairthorneed. *Towards information retrieval*, p. xvii-xxiii. Butterworths, 1961.

[Mothe, 1994] Mothe, J. *Search mechanisms using a neural network-Comparison with the vector space model.* 4th RIAO Intelligent Multimedia Information Retrieval Systems and Management, Vol.1, p. 275-294, New York, 1994.

.

[Mothe,1995] Mothe, J. Tanguy, L. *Linguistic features to predict query difficulty - a case study on previous TREC campaigns* SIGIR, Predicting query difficulty - methods and applications workshop, p 7-10, 2005.

[Mozer, 1984] Mozer, M. C. *Inductive Information Retrieval Using Parallel Distributed Computation.* Research Report. San Diego, CA: University of California at San Diego, June 1984.

[Navarro et al., 2004] Navarro, D.J., & Lee, M.D. *Common and distinctive features in stimulus representation: A modified version of the contrast model*. Psychonomic Bulletin & Review, 11(6), p. 961–974, 2004.

[Nogier, 1991] Nogier J.F. Génération automatique de langage et graphes conceptuels, Hermès, Paris, 1991.

[Nosofsky, 1984] Nosofsky J. *Choice, Similarity, and the context theory of classification*. Journal of Experimental Psychology: Learning, Memory, and Cognition. vol. 10, p. 104-114, 1984.

[Nosofsky, 1991] Nosofsky R.M. *Stimuli bias, asymetric similarity, and classification*. Cognitive psychologie, 23, p. 94-140, 1991.

[Nosofsky, 1992] Nosofsky R.M. *Similarity scaling and cognitive process models*. Annual review of psychologie, 43, p. 25-53, 1992.

[Novick, 1988] Novick, L. R. *Analogical transfer, problem similarity and expertise*. Journal of Experimental Psychology: Learning, Memory and Cognition,14(3), p. 510-520, 1988.

[Ogilvie et al., 2003] Ogilvie, P., and Callan J. *Combining Document Representations for Known Item Search*, *SIGIR*, 2003.

[Ortony, 1979] Ortony, A. *Beyond literal similarity*. Psychological Review, 86, p. 161–180, 1979.

[Osherson, 1990] Osherson, D. N. *Categorization*. In D. N. Osherson & E. E. Smith (Eds.), Thinking: an invitation to cognitive science. Cambridge, MA: MIT Press, 1990.

[Paice, 1996] Paice, C.D. *Method for evaluation of stemming algorithms based on error counting*. Journal of the American Society for Information Science 47 (8) p. 632-49, 1996.

[Pearl,1988] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann: San Mateo CA, 1998.

[Premack, 1983] Premack, D. *The code of man and beasts*. Behavioral and Brain Sciences*, 6*, p. 125-167, 1983.

[Quillian, 1968] Quillian, R. *Semantic memory.* Semantic information processing, p. 227-270, 1968.

[Quine, 1969] Quine W. V. Ontological Relativity and Other Essays. New York, Columbia University Press. 165 p., 1969.

[Reed, 1972] Reed S.K. *Pattern recognition and categorization*. Cogn. Psychol. 3: p. 382–407, 1972.

[Reitman, 1965] Reitman, W. R. *Cognition and thought: an information processing approach*. New York, NY: John Wiley and Sons, 1965.

[Richard, 1995] Richard, J-F Les activités mentales. Comprendre, raisonner, trouver des solutions. Paris, A.Colin, 1995.

[Rijsbergen, 1977] C. J. van Rijsbergen. *A theoretical basis for the use of co-occurrence data in information retrieval*. Journal of Documentation, 33: p. 106-119, 1977

[Rijsbergen, 1979] C. J. van Rijsbergen *Information Retrieval*, 2nd ed. Butterworths: London, 1979.

[Rijsbergen, 1981] C. J. van Rijsbergen, D. Harper and M. Porter. *The selection of good search terms*. Information Processing and Management. 17. p. 77-91, 1981.

[Rips, 1989] Rips, L. J. *Similarity, typicality, and categorization*. In S. Vosniadou et A. Ortony (dir.), Similarity and analogical reasoning, p. 21-59. Cambridge: Cambridge University Press, 1989.

[Rips et al., 1993] Rips L.J., Collins A. *Categories and ressemblance*. Journal of Experimental Psychology : General, 122, p. 468-486, 1993.

[Robertson, 1981] Robertson, S.E. *The methodology of information retrieval experiment. Information Retrieval Experiment*. In K. Sparck Jones, Ed. Chapt. 1, p. 9-31. Butterworths, 1981.

[Robertson et al., 1982] S. Robertson, M. Maron, and W. Cooper *Probability of relevance: a unification of two competing models for document retrieval*. Information Technology: Research and Development, 1: p. 1-21, 1982.

[Robertson et al., 1994] Robertson, S. E. and S. Walker *Some simple effectiveapproximations to the 2-Poisson model for probabilistic weighted retrieval*. In Proceedings of ACM SIGIR 1994. p. 232-241, 1994.

[Robertson et al., 2000] Robertson, S.E., Walker, S. *Microsoft Cambridge at TREC-9: Filtering track*, TheNinth Text REtrieval Conference (TREC-9), National Institute of Standards and Technology, Gaithersburg,MD, p. 73-86, November 13-16, 2000.

[Rocchio, 1971] Rocchio, J., *Relevance feedback in information retrieval*. In Salton, G., editor, The SMART Retrieval System: Experiments in Automatic Document Processing, p. 313-323. Prentice-Hall, Englewood Cli s, NJ, USA. 1971.

[Rosch, 1975] Rosch, E. *Cognitive reference points*. Cognitive Psychology, 7, p. 532-547. 1975.

[Ross , 1987] Ross, B. H. 1987. *This is like that: The use of earlier problems and the separation of similarity effects*. Journal of Experimental Psychology : Learning, Memory and Cognition, 13(4), p. 629-639, 1987.

[Salton et al., 1983] Salton, G. and McGill, M.J. Introduction to Modern Information Retrieval. McGraw-Hill, 1983.

[Salton et al., 1983 b] Salton, G.; Buckley, C.; Yu, C.T. An evaluation of term dependence models in information retrieval. Lecture Notes in Computer Science, in: Salton, G.; Schneider, H.J., eds. 146, Berlin: Springer-Verlag, p. 151-173, 1983.

[Salton et al., 1983 c] Salton, G., Fox E., and Wu, H. *Extended boolean information retrieval*. Communications of the ACM, 31(2), p. 1002–1036, November 1983.

[Salton, 1986] Salton, G. *Another look at automatic text-retrieval systems*. Commun. ACM 29, 7, p. 648-656, July 1986.

[Salton, 1987] Gerard Salton, Chris Buckley, *Term Weighting Approaches in Automatic Text Retrieval*, Cornell University, Ithaca, NY, 1987.

[Salton et al., 1988] G. Salton and C. Buckley On the use of spreading activation methods in automatic information retrieval. *11th ACM-SIGIR Conference*. p. 147-160, 1988.

[Salton et al. 1990] Gerard Salton, Chris Buckley: *Improving retrieval performance by relevance feedback*. JASIS 41(4): p. 288-297, 1990.

[Savoy, 1993] Savoy, J. *Stemming of French words based on grammatical categories* Journal of the American Society for Information Science, 44(1), p. 1-9, 1993.

[Schamber, 1996] Schamber, L. *What is a Document ? Rethinking the Concept in Uneasy Times*. In : *J.* American Society for Information Science, vol. 47, n°9, 1996, p. 669-671, 1996.

[Scholtes, 1994] Scholtes J.C. *Neural networks in information retrieval in a libraries context.* EC/PROLIB/ANN Contract, M.S.C. Information Retrieval Technologies B.V., The Netherlands, 1994.

[Schutz et al., 1973] Schutz, A. and Luckmann, T. *Structures of the Life World*. Northwestern University Press, Evanston, Ill., Ed. ACM, New York, Sept. 1990, p. 45-61, 1973.

[Schvaneveldt, 1985] Schvaneveldt, R. *Measuring the structure of expertise*. International Journal of Man-Machine Studies, 23, p. 699-728, 1985.

[Seglen, 1992] Seglen P. O. *The skewness of science*. Journal of the American Society for Information Science, 43, p. 628-638, 1992.

[Selberg, 1997] Selberg E. *Information Retrieval Advances using Relevance Feedback. UW Dept. of CSE General Exam. 1997.*

[Shastri, 1993] Shastri, L. & Ajjanagadde, V. *From Simple Associations to Systematic Reasoning: A connectionist representation of rules, variables and dynamic bindings using temporal synchrony*. Behavioral and Brain Sciences, 16, p. 417-494, 1993.

[Shepard, 1962] Roger Shepard *The analysis of proximities: Multidimensional scaling with an unknown distance function*. Psychometrika, Springer, vol. 27(2), p. 125-140, June 1962.

[Simon et al., 2002] Simon, D., & Holyoak, K. J. *Structural dynamics of cognition: From consistency theories to constraint satisfaction*. Personality & Social Psychology Review, 6, p. 283-294, 2002.

[Singley et al., 1989] Singley, M . K . R., Anderson, J. *The Transfer of Cognitive Skill*. Cambridge, MA: Harvard Univ. Press, 1989.

[Sjoberg, 1972] Sjoberg L. A *Cognitive theory of similarity*. Goteborg Psychological reports. 2(10), 1972.

[Sloutsky et al., 2001] Sloutsky, V.M., Lo, Y.-F., & Fisher, A.V. *How much does a shared name make things similar? Linguistic labels, similarity, and the development of inductive inference*. Child Development, 72(6), p. 1695-1709, 2001.

[Smail, 1994] Smaïl M. *Raisonnement à base de cas pour une recherche évolutive d'information : Prototype Cabri-n. Vers la définition d'un cadre d'acquisition de connaissances*. Thèse de doctorat, Université Henri Poincaré - Nancy I, Octobre 1994.

[Smeaton, 1986] Smeaton, A.F. *Incorporating syntactic information into a document retrieval strategy: An investigation*. Proc. 1986 ACM-SIGIR Conference on Research and Development in Information Retrieval, Pisa, Italy, Association for Computing Machinery, New York; p. 103-113, 1986.

[Smeaton, 1992] Smeaton A. *Progress in the application of natural language processing to information retrieval tasks*. The Computer Journal, vol. 35, p. 268-278, 1992.

[Smith et al., 1998] Smith, E.E., Patanalo, A.L., & Jonides, J. *Alternative strategies of categorization. Cognition*, 65, p. 167-196, 1998.

[Smolensky, 1990] Smolensky, P. *Tensor product variable binding and the representation of symbolic structures in connectionist systems*. Artificial Intelligence, 46 (1-2), p. 159-216, 1990.

[Sowa, 1984] Sowa, J. *Conceptual Structures : Information Processing in Mind and Machine*. Addisson-Wesley Publishing Company, Reading MA, 1984.

[Sowa, 2000] Sowa, J. Knowledge Representation: Logical, Philosophical, and Computational Foundations. Brooks Cole Publishing Co., Pacific Grove, CA, 2000.

[Sparck Jones, 1971] Sparck Jones, K. *Automatic Keyword Classillcation for Information Retrieval*. London: Butterworths, 1971.

[Sparck Jones, 1972] Sparck Jones, K. *A statistical interpretation of term specificity and its application.* In Retrieval Journal of Documentation 28:1; p. 11-21, march 1972.

[Sparck Jones, 1974] Sparck Jones, K. *Automatic indexing*. J. Doc. 30, 4, p. 393-432, 1974.

[Sparck Jones, 1991] K. Sparck-Jones. *Notes and references on early automatic classification work*. *SIGIR Forum*, 25(1): p. 10-17, 1991.

[Sparck Jones et al., 2000] SpärckJones, Karen, Walker S.,and Stephen E.Robertson *A probabilistic model of information retrieval: Development and comparative experiments*. Information Processing and Management p.779–808, p. 809–840, 2000.

[Song et al. 1999] Fei Song and W. Bruce Croft *A General Language Model for Information Retrieval*. In Proceedings of the eighth international conference on Information and knowledge management (CIKM), p. 316-321, 1999.

[Steyvers et al., 2005] Steyvers M., Tenenbaum J.B. *The large-scale structure of semantic networks : statistical analysis and a model for semantic growth*. Cognitive science : a multidisciplinary Journal, vol. 29, p. 41-78, 2005.

[Thagard et al., 1990] Thagard, P., Holyoak, K., Nelson, G., & Gochfeld, D. *Analog retrieval by constraint satisfaction*. Artificial Intelligence, 46, p. 259-310, 1990.

[Thibaut, 1997] Thibaut, J.-P. *Similarité et catégorisation.* L'année psychologique, 97, p. 701-736, 1997.

[Thorndike, 1913] Thorndike, E. L. *Educational psychology.* New York: Lemcke and Buechner, 1913.

[Titterington, 1985] Titterington, D. M., U. E. Makov and A. F. M Smith. *Statistical* Analysis of *Finite Mixture Distributions* John Wiley and Sons, 1985.

[Togerson, 1965] Multidimensional scaling of similarity. Psychometrica. vol. 30, p. 379-393, 1965.

[Truong et al., 2008] Dinh Truong, Taoufiq Dkaki, Josiane Mothe, Pierre-Jean Charrel. *GVC: a graph-based Information Retrieval Model.* Dans : *Conférence francophone en Recherche d'Information et Applications (CORIA 2008)*, *Trégastel (France)*, *12-MAR-08-14-MAR-08*, CNRS, p. 337-351, mars 2008.

[Truong et al., 2008] Dinh Truong, Taoufiq Dkaki, Josiane Mothe, Pierre-Jean Charrel. *Information Retrieval Model based on Graph Comparison.* Dans : *Journées internationales d'Analyse statistique des Données Textuelles (JADT 2008)*, *Lyon, France*, *12-MAR-08-14-MAR-08*, vol. 2, Hermès, p. 1115-1126, 2008.

[Turtle et al., 1990] Turtle, H., and Croft, W. B. *Inference network for document retrieval*. Research and Development on Information Retrieval - ACM-SIGIR, Brussels, p. 1-24, 1990.

[Turtle et al., 1991] Turtle, H. and W. Croft. *Evaluation of an inference network-based retrieval model*. ACM Transactions on Information Systems 9(3), p. 187–222, 1991.

[Tversky, 1977] Tversky A. *Features of similarity*. In Readings in Cognitive Science from Psychological Review 84, p. 327-352, 1977.

[Tversky et al., 1982] Tversky, A., & Gati, I. *Similarity, separability, and the triangle inequality*. Psychological Review, vol. 89, p. 123-154, 1982.

[Tversky et al., 1986] Tversky, A., & Hutchinson, J.W. *Nearest neighbor analysis of psychological spaces*. Psychological Review, vol. 93, p.3-22, 1986.

[Vallet, 2005] David Vallet, Miriam Fernandez, and Pablo Castells *An ontology-based information retrieval model*. In Asuncion Gomez-Pérez and Jérôme Euzenat, editors, ESWC, vol. 3532 of Lecture Notes in Computer Science, p. 455–470. Springer, 2005.

[Veloso, 1993] Veloso, M.M. & Carbonell, J.G. *Derivational Analogy in PRODIGY: Automating Case Acquisition, Storage, and Utilization*. Machine Learning, 10(3): p. 249-278, 1993.

[Vignaux, 2003] Georges Vignaux, Essai de définition d'un document, document de travail du RTP-DOC février 2003.

[Vosniadou et al., 1989] Vosniadou, S. et Ortony, A. *Similarity and analogical reasonning: A synthesis*. In S. Vosniadou et A. Ortony (dir.), Similarity and analogical reasoning, p. 1-17. Cambridge: Cambridge University Press, 1989.

[Waller et al., 1979] Waller, W.G.. and Kraft, D.H. *A mathematical model for a weighted Information Processing and Management, Boolean retrieval system*. vol. 15, 5, 235-245, 1979.

[Wasserman et al., 1994] Wasserman, S., and Faust, K. *Social Network Analysis*, Cambridge U.K., Cambridge University Press, 1994.

[Watts et al., 1998] Watts D.J and Strogatz D. H. *Collective dynamics of small-world network*. Nature(London) vol. 393, p. 440-442, 1998.

[Watts et al., 1999] Watts D.J *Small Worlds: The dynamics of networks between Order and randomness*. Princeton University press, 1999.

[Wechsler et al., 1997] Wechsler, M., Sheridan, P., Schäuble, P. *Multi-language text indexing for Internet Retrieval.* In Proceedings of the RIAO Conference on Computer-Assisted Information Retrieval, 1997.

[Willet, 1988] P. Willett. *Recent trends in hierarchic document clustering: A critical review*. Inf. Process. Manage., 24(5): p. 577–597, 1988.

[Xu, 1990] X. Lu, *Document retrieval: A structure approach*. Information Processing & Management 26(2): p. 209-218, 1990.

[Yager, 1988] Yager R. R. *On ordered weighted averaging aggregation operators in multicriteria decisionmaking*, *IEEE Trans. Syst. Man Cybern.*, vol. 18, n° 1, p. 183-190, 1988.

[Yang, 1999] Yiming Yang *An evaluation of statistical approaches to text categorization*. Information Retrieval, vol. 1(1-2): p. 69-90, 1999.

[Zadeh, 1965] L. A. Zadeh *Fuzzy Sets*. Information and control, p. 338-353, 1965.

[Zipf, 1949] G. Zipf. Human Behaviour and the Principle of Least Effort. Addison-Wesley, 1949.

[Zang et al., 1998] Zhang, S, & Markman, A.B. *Overcoming the early entrant advantage: The role of alignable and nonalignable differences*. Journal of Marketing Research, vol. 35, p. 413-426, 1998.

TITLE: An information retrieval model based on graphs and structural similarities for improving information retrieval process

ABSTRACT:

The main objective of IR systems is to select relevant documents, related to a user's information need, from a collection of documents. Traditional approaches for document/query comparison use surface similarity, i.e. the comparison engine uses surface attributes (indexing terms). We propose a new method which uses a special kind of similarity, namely structural similarities (similarities that use both surface attributes and relation between attributes). These similarities were inspired from cognitive studies and a general similarity measure based on node comparison in a bipartite graph. We propose an adaptation of this general method to the special context of information retrieval. Adaptation consists in taking into account the domain specificities: data type, weighted edges, normalization choice. The core problem is how documents are compared against queries. The idea we develop is that similar documents will share similar terms and similar terms will appear in similar documents. We have developed an algorithm which traduces this idea. Then we have study problem related to convergence and complexity, then we have produce some test on classical collection and compare our measure with two others that are references in our domain.

The Report is structured in five chapters: First chapter deals with comparison problem, and related concept like similarities, we explain different point of view and propose an analogy between cognitive similarity model and IR model. In the second chapter we present the IR task, test collection and measures used to evaluate a relevant document list. The third chapter introduces graph definition: our model is based on graph bipartite representation, so we define graphs and criterions used to evaluate them. The fourth chapter describe how we have adopted, and adapted the general comparison method. The Fifth chapter describes how we evaluate the ordering performance of our method, and also how we have compared our method with two others.

KEYWORDS:

Information retrieval, graphs, comparison, structural similarities, Information retrieval model, bipartite graphs.

RESEARCH LABORATORY:

Institut de Recherche en Informatique de Toulouse, UMR 5505 – CNRS ; Université Paul Sabatier, 118 route de Narbonne 31062 Toulouse cedex 9.