Titre: Title:	Indexation de documents Web à l'aide d'ontologies
Auteur: Author:	Olivier Gagnon
Date:	2013
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Gagnon, O. (2013). Indexation de documents Web à l'aide d'ontologies [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/1131/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/1131/
Directeurs de recherche: Advisors:	Michel C. Desmarais, & Michel Gagnon
Programme: Program:	Génie informatique

UNIVERSITÉ DE MONTRÉAL

INDEXATION DE DOCUMENTS WEB À L'AIDE D'ONTOLOGIES

OLIVIER GAGNON

DÉPARTEMENT DE GÉNIE INFORMATIQUE ET GÉNIE LOGICIEL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION

DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES

(GÉNIE INFORMATIQUE)

AVRIL 2013

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

INDEXATION DE DOCUMENTS WEB À L'AIDE D'ONTOLOGIES

présenté par : GAGNON Olivier

en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de :

M. ROBILLARD Pierre N., Ph. D., président

M. DESMARAIS Michel C., Ph. D., membre et directeur de recherche

M.GAGNON Michel, Ph. D., membre et codirecteur de recherche

M.ANTONIOL Giuliano, Ph. D., membre

REMERCIEMENTS

- Michel Gagnon et Michel Desmarais, pour leur patience et leur support à la rédaction du présent mémoire.

- Grégoire Lapointe et Julien Gascon-Samson pour leur aide précieuse avec peu de préavis.

RÉSUMÉ

Le Web n'existe que depuis moins de vingt ans. Pourtant, sa portée mondiale lui permet d'être la principale source d'information pour des besoins variés, allant de la recherche scientifique à la recherche d'un bon restaurant vietnamien. Les moteurs de recherche tels Yahoo, Bing et Google raffinent sans cesse leurs algorithmes afin de fournir les meilleurs résultats pour chaque requête des utilisateurs.

La plupart des moteurs de recherche se basent sur des algorithmes d'indexation basés sur la présence ou non de mots-clés correspondant à la requête de l'utilisateur. Le présent mémoire présente différentes expériences pour l'indexation de documents Web à l'aide d'ontologies, des graphes orientés constitués d'entités et de relations propres au domaine du Web Sémantique. Ce mémoire présente ensuite une expérience visant à comparer la qualité des résultats aux requêtes d'information à celle d'une technique d'indexation classique.

Ce mémoire se conclut avec les résultats de l'expérience ainsi que des perspectives d'amélioration pour l'indexation par ontologies.

ABSTRACT

The Web has existed for less than twenty years. Yet, it can be considered as one of the most important sources of information for the most varied of subjects, ranging from scientific research to the location of Vietnamese restaurants. Search engines such as Yahoo, Bing and Google improve regularly their algorithms in order to produce the best results for every possible query made by its users.

Most popular search engines use indexing algorithms based on the presence (or not) of keywords corresponding to the query. This master's thesis presents the experiences that have been made around the world for indexing Web documents with ontologies, which are directed graphs linking entities with relations that are widely used in the domain of Semantic Web. This document then presents an experience that attempts to demonstrate that an ontology-based indexing brings higher quality search results over a classic keyword-based indexing.

This master's thesis concludes with the results of the experience and potential improvements for ontology-based indexing.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES SIGLES ET ABRÉVIATIONS

HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
LSI	Latent Semantic Indexing
OWL	Web Ontology Language
PHP	Hypertext PreProcessor
RDF	Ressource Description Framework
SQL	Standard Query Language
SVD	Singular Value Decomposition
XML	eXtensible Markup Language

INTRODUCTION

Depuis vingt ans, le Web augmente sa taille de façon exponentielle. En 1995, le nombre de pages Web distinctes a brisé le mur du million. En 2010, ce nombre se chiffre maintenant à plus de mille milliards [1]. Trouver de l'information pertinente à nos besoins dans ce nombre sans cesse croissant de documents est une tâche à laquelle les informaticiens se sont attaqués dès le milieu des années 1990, avec les moteurs de recherche bien connus comme *AltaVista*, *Yahoo* et maintenant, *Google*.

Google, avec son algorithme « PageRank », a révolutionné la recherche d'information sur le Web en indexant et en classifiant la plupart des sites Web dans le monde. Lorsqu'un utilisateur soumet une requête aux serveurs de Google, ceux-ci parcourent l'index en un temps record et retournent à l'utilisateur des liens vers des pages Web concernant la requête, ordonnancés selon leur pertinence à la requête. Toutefois, de nombreux sites ont leur propre moteur de recherche afin que les requêtes ne s'appliquent qu'au contenu de leur site et leur performance est souvent de mauvaise qualité.

Par exemple, sur le site de la SAQ (http://www.saq.com), si un utilisateur effectue une recherche avec les mots « vin doux », le moteur de recherche ne retourne aucun produit ayant ces mots dans leur nom. Est-il possible que la SAQ n'ait aucun vin doux à vendre? Cet exemple démontre la limitation des approches classiques d'indexation par mots-clés et le présent document propose une alternative : l'indexation par ontologies.

La problématique de l'indexation des documents sur le Web est divisée en trois différentes étapes qui serviront de cadre au présent document, telle que spécifiée par les auteurs *Hyvõnen, Saarela et Vijaanen* [12], soit l'étape de la formulation du besoin d'information, l'étape de la formulation de l'ensemble de recherche et l'étape finale de la présentation des résultats de la recherche.

1) L’étape de la formulation du besoin d’information

Un problème fondamental de la recherche d’information sur le Web est l’inférence du but de l’utilisateur. En effet, toute requête de l’utilisateur est associée à un but bien précis (par exemple, obtenir des images de loups). Cette recherche d’information doit être interprétée par le moteur de recherche à partir de quelques mots (généralement entre un et trois). L’analyse du but de l’utilisateur est complexe car, pour une même série de mots-clés, un utilisateur pourrait avoir besoin d’images pour une présentation alors qu'un autre utilisateur désire des informations textuelles pour une recherche universitaire. Cette analyse n’est que très peu poussée dans les moteurs de recherche actuels, car elle nécessite une analyse du contexte de la requête ainsi que de l’historique des requêtes. Ce type d’analyse est complexe et repose sur de l’information qui n’est généralement pas disponible avec la requête elle-même.

De plus, pour une même requête, un utilisateur peut avoir besoin de textes, d’images, d’articles de conférences ou de mémoires. Pour répondre à toutes ces requêtes, Google propose des variations de son moteur de recherche afin de donner un contexte aux requêtes. Par exemple, Google Images restreint ses résultats aux images et Google Scholar se limite aux publications scientifiques et universitaires.

2) L’étape de la formulation de l’ensemble de recherche

Un autre aspect important de la recherche sur le Web est l’indexation des documents. L’Internet se compose de milliards de documents et d’images qui ne sont pas définis comme appartenant à un domaine particulier. Comment déterminer le type d’information présente dans un document ou dans une image pour l’associer à un type d’information qu’un utilisateur pourrait tenter de chercher? L’indexation de tous ces documents se base donc sur le point commun qu’ont tous ces documents : les mots. Même les images et les vidéos sont présentement indexés sous forme textuelle. Les créateurs d’images et de vidéos définissent des annotations, ou « tags » qui décrivent le contenu de l’image et ces mots-clés sont ensuite utilisés pour l’indexation de l’image.

Le processus d'annotation demande une intervention humaine et l'indexation d'images basée sur la reconnaissance des formes constitue un important domaine de recherche actuel.

À l'aide des mots présents dans un document, il est possible de déterminer les sujets concernés par un document : si un document contient les mots « loup » ou « loups », il est statistiquement plus probable que ce document soit pertinent pour un utilisateur désirant s'informer sur le sujet des loups qu'un autre document qui ne contient pas du tout ces mots. Il s'agit d'une évidence mais c'est sur ce principe que se base la plupart des moteurs de recherche modernes sur le marché.

Mais que se passe-t-il lorsque un mot a plusieurs sens? Si un utilisateur consulte une page Web sur le site de Cyberpresse (http://www.lapresse.ca) racontant les derniers exploits des Tigers de Détroit (une équipe de baseball professionnel), les articles suggérés par le site comme étant reliés à l'article courant parlent du golfeur Tiger Woods.

Dans ce cas précis, l'indexation par mots-clés considère ces documents comme reliés car ils contiennent tous les deux le mot « Tigers », ce qui est évidemment faux pour un humain mais plus difficile à évaluer pour une machine. L'indexation par ontologies offre une solution à ce problème car cette indexation utilise le contexte du document pour déterminer si le document a un contexte de baseball ou bien de golf. Malgré la ressemblance forte entre ces deux contextes (concepts de joueur, de bâton, de tournoi), les ontologies permettent de déterminer qu'un document qui parle de Tiger Woods a peu de chances d'être relié au baseball.

3) L'étape de la présentation des résultats

Une fois que la requête de l'utilisateur a été analysée et que l'ensemble des documents a été indexé, l'étape finale de la recherche sur le Web est la présentation des résultats. À partir de l'ensemble indexé de documents, lesquels retourner? Dans quel ordre les proposer à l'utilisateur?

Pour une recherche booléenne, où seule la présence ou l'absence d'un mot-clé est utilisée, seule la fréquence permet d'ordonnancer des documents. Ceci s'avère inadéquat car en plus d'avoir plusieurs documents équivalents sur cette base, certains mots n'ont que très peu de poids. Par exemple, une requête comme « recherche ontologie et objets » apportera un très grand nombre de documents sur la base du mot « objets » et un bon nombre aussi celle du mot « recherche ». Au contraire, le mot « ontologie » est ici vraisemblablement celui qui devrait avoir le plus de poids. Une méthode très répandue pour résoudre ce problème consiste à utiliser la méthode TF-IDF, qui tient compte de la rareté d'un mot. Combiné à la fréquence des mots, il est généralement possible d'obtenir un ordre des documents bien déterminé. Cette méthode sera décrite plus en détail dans le chapitre 2 de ce document.

La théorie de l'espace vectoriel est également utilisée pour la recherche d'information. En produisant une matrice avec les lignes représentant les termes et les colonnes représentant les documents, il est ensuite possible de comparer un vecteur contenant les mots de la requête avec les vecteurs représentant les documents. Le cosinus est une mesure courante de cette similarité. Il varie entre 0, qui signifie que le vecteur requête et le vecteur document sont orthogonaux et n'ont aucune similarité et 1, qui implique que le vecteur requête et le vecteur document sont identiques. Ainsi les documents avec la valeur la plus proche de 1 seront les mieux classés lors de la présentation des résultats de la requête à l'utilisateur. Le calcul de la distance cosinus est décrit plus en détail au chapitre 2 du présent mémoire. D'autres techniques d'indexation et de présentation des résultats comme l'indexation sémantique latente (LSI) et l'algorithme PageRank de Google seront également abordés.

La question principale de recherche suivante est posée :

Est-il possible d'améliorer la qualité de la réponse d'une recherche d'information sur le Web avec une méthode d'indexation utilisant des ontologies, comparativement aux méthodes d'indexation plus classiques comme celles basées sur l'espace vectoriel?

Le présent mémoire présente les avancées qui ont été publiées dans le domaine de l'indexation. Tout d'abord les techniques les plus anciennes basées sur le modèle booléen standard sont présentées, suivies par les techniques plus modernes de l'espace vectoriel ainsi que celles utilisant les principes du Web Sémantique ou des ontologies. Ensuite, le cadre expérimental sera défini afin de confirmer ou d'infirmer l'efficacité de la méthode d'indexation par ontologies. Finalement, les résultats de l'expérience seront expliqués et discutés. Des pistes de travaux futurs seront aussi élaborées.

CHAPITRE 1 REVUE DE LITTÉRATURE

Ce chapitre décrit les techniques d'indexation les plus connues dans la littérature. Des expériences menées par d'autres chercheurs qui ressemblent au sujet du présent mémoire ainsi que es notions de calcul de la pertinence d'un document sont également décrites.

Avant de décrire les expériences effectuées par d'autres chercheurs dans le domaine de l'indexation par ontologies, il faut d'abord décrire les concepts importants des deux méthodes d'indexation qui seront comparées dans l'expérience du présent mémoire, soit :

1) Indexation par mots-clés dans le cadre du modèle d'espace vectoriel et en utilisant la mesure TF-IDF et la similarité cosinus.
2) Indexation par mots-clés et expansion de la requête par ontologies.

Les sections suivantes vont décrire la mesure TF-IDF ainsi que les ontologies.

1.1 Mesure TF-IDF

La recherche de documents par mots-clés et booléenne ne permet pas de préciser quels mots-clés sont les plus importants. Prenons par exemple la requête « latex parbox ». Latex est un produit, mais c'est aussi un logiciel de traitement de texte et « parbox » est une commande du logiciel Latex. Cette requête s'applique au cas où un utilisateur se demande comment cette commande fonctionne. Il est possible qu'un document ne contenant pas le terme « latex » soit pertinent s'il contient le terme « parbox ». Par contre, il est fort peu probable que l'inverse soit aussi vrai. Le terme « parbox » devrait donc avoir un poids plus important que « latex ».

Une technique répandue et efficace pour associer un poids à des termes consiste à mesurer leur rareté, telle que définie par la mesure TF-IDF. En effet, le terme « parbox » est très rarement rencontré. Au contraire, le terme « latex » est beaucoup plus fréquent et même son utilisation au sens du système de traitement de texte demeure plus fréquente que celle du terme « parbox ». La mesure de la rareté d'un terme est donnée par le calcul qui suit.

Supposons un ensemble de documents D et la fréquence d'un terme i dans un document j. La mesure de la rareté du terme sera déterminée par sa fréquence inverse dans l'ensemble de documents D (inverse document frequency). L'équation 1 représente ce calcul.

$$\operatorname{idf}(t, D) = \log \frac{|D|}{|\{d \in D : t \in d\}|}$$

Idf(t,D) : Valeur IDF du terme t

D : Nombre de documents dans l'ensemble D

$d \in D : t \in d$: Nombre de documents de l'ensemble D contenant le terme t

Figure 1-1 : Équation 1 - Valeur IDF

Par exemple, si un terme se retrouve dans tous les documents, alors la valeur IDF de ce terme sera égale à log(1) donc 0. À l'inverse, si un terme ne se trouve que dans un seul document, sa valeur IDF sera égale à log(|D|). Ce qui produit l'effet recherché pour atténuer la valeur de mots qui n'ont pas de pouvoir de discernement entre les documents de l'ensemble D.

Pour la valeur TF (term frequency) il suffit de compter la fréquence d'un terme t dans un document particulier d. Une fois ces valeurs calculées, on produit l'équation 2 qui constitue la base de la mesure *TF-IDF* :

$$\operatorname{tfidf}(t, d, D) = \operatorname{tf}(t, d) \times \operatorname{idf}(t, D)$$

Figure 1-2 : Équation 2 - Valeur TF-IDF

Le produit final donne la valeur *TF-IDF* d'un terme t par rapport à un document particulier d dans un ensemble de documents D. Une fois ces valeurs obtenues, il faut déterminer quels documents sont les plus pertinents pour une requête donnée de l'utilisateur. L'utilisation des espaces vectoriels est une façon de calculer la corrélation entre la requête et chacun des documents de l'ensemble D.

1.2 Théorie des espaces vectoriels

La théorie des espaces vectoriels définit la similitude entre une requête de l'utilisateur et chacun des documents d'un corpus par l'angle qu'ils forment dans un espace dont les dimensions sont les termes. Cette théorie considère que chaque document peut être représenté comme un vecteur de termes. Par exemple, le texte « Latex est un logiciel libre » devient un vecteur à cinq dimensions A = ('Latex', 'est', 'un', 'logiciel', 'libre'). À la base, chaque terme a un poids égal (1 si il est présent dans un document, 0 dans le cas contraire). Cependant, ce calcul vectoriel est souvent combiné au calcul TF-IDF afin de donner un poids plus représentatif à chacun des termes selon leur rareté relative dans l'ensemble de documents D.

Avec chaque document converti sous forme vectorielle et pour lequel chaque terme est pondéré selon la transformation TF-IDF, il est possible d'appliquer l'algèbre vectorielle pour déterminer la similitude entre le vecteur représentant le document et le vecteur représentant la requête de l'utilisateur. Dans le cadre du présent mémoire, la mesure utilisée pour déterminer cette similitude est la distance cosinus.

Distance Cosinus

En considérant le vecteur document A et le vecteur requête B, la mesure de la distance cosinus consiste simplement à calculer le cosinus de l'angle entre les deux vecteurs. L'équation 3 montre ce calcul.

$$\cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$

Figure 1-3 : Équation 3 - Cosinus de l'angle entre deux vecteurs

Comme les pondérations TF-IDF pour chaque terme du vecteur sont positives, le cosinus de l'angle entre deux vecteurs peut varier de 0 (les deux vecteurs sont orthogonaux, ils sont totalement différents) jusqu'à 1 (les deux vecteurs sont parallèles). Ainsi les documents dont le cosinus entre leur vecteur et celui de la requête qui se rapprochent le plus de la valeur 1 ont le plus de similitude et seront considérés comme étant les plus pertinents pour la requête. L'exemple de la figure 1-4 montre deux documents représentés par les vecteurs d1 et d2 ainsi que la requête représentée par le vecteur q. Dans cette figure, on voit que l'angle α est plus petit que l'angle θ alors le vecteur document d1 est plus pertinent à la requête représentée par le vecteur q que le vecteur document d2.

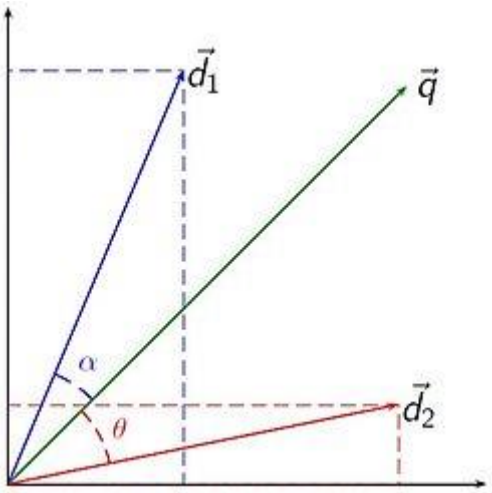

Figure 1-4 : Exemple graphique du calcul de similitude par la distance cosinus

Une faiblesse importante de cette mesure est qu'elle ne tient pas compte de la sémantique des mots. Lors de l'analyse des documents servant à déterminer les valeurs TF et IDF des termes, aucune distinction n'est faite, par exemple, entre l'espèce de serpents des pythons et le langage de programmation Python. Pour tenter de résoudre ce problème de synonymie, l'expérimentation du présent mémoire utilisera une ontologie du domaine des technologies Web. L'utilisation d'une ontologie permet également une expansion sémantique de la requête de l'utilisateur dont les détails sont expliqués dans la méthodologie de l'expérimentation au chapitre deux. L'ontologie est une façon de représenter un domaine de connaissances (biologie, informatique) et d'autres méthodes de représentation des domaines peuvent être utilisées, comme un « thesaurus » contenant des termes et leurs synonymes.

1.3 Qu'est-ce qu'une ontologie?

Depuis plusieurs années, avec la croissance exponentielle de l'Internet, se développent de nouvelles façons de représenter l'information disponible sur le Web. Une de ces façons de la représentation des informations est une pièce centrale de ce document : le *Web sémantique*. Le Web sémantique est un ensemble de méthodes et de technologies permettant aux machines d'analyser l'information disponible des documents sur le Web [13]. Ces technologies sont

multiples et comprennent, entre autres, la technologie RDF et le langage OWL qui permet de définir des ontologies. Le langage OWL est une représentation structurée de données destinées à être lues par des applications automatisées qui est dérivé du langage XML.

Devedzic [3] définit une ontologie comme « une terminologie, les concepts essentiels, leur classification, leur taxonomie, leurs relations et les hiérarchies, ainsi que les axiomes du domaine ». En d'autres termes, en utilisant le langage OWL, il est possible de représenter le sens des termes dans un vocabulaire et les relations entre ces termes. Cette représentation des termes et des relations constitue une ontologie.

```
<Owl:Class about="#script_language">
      <Owl:label>langage de script</Owl:label>
      <Owl:label>script language</Owl:label>
      <Owl:subClassOf>
         <Owl:Restriction>
            <Owl:onProperty resource="#scriptLanguageUsedBy"/>
            <Owl:allValuesFrom resource="#Script"/>
         </Owl:Restriction>
      </Owl:subClassOf>
      <Owl:comment>langages permettant d'effectuer des tâches répétitives (routine) sur des
éléments ayant une caractéristique commune</Owl:comment>
   </Owl:Class>


<Owl:script_language about="#Python">
            <Owl:label>python</Owl:label>
      <Owl:type resource="&owl;Thing"/>
      <Owl:comment>langage de script python</ Owl:comment>
      <Owl:scriptLanguageUsedBy resource="#Fichier_Script_Python"/>
   </Owl:script_language>
```

Figure 1-5 : Exemple de fichier OWL avec un exemple d'ontologie décrivant une classe et une instance de cette classe.

Dans l'exemple de la figure 1-5, on peut voir que la classe « Script Language » a été définie et que « Langage de script » est une étiquette associée à la classe. De plus, il y est aussi

défini que toute instance de la classe devra respecter des règles. Dans l'exemple de la figure 2, la classe « Script Language » est reliée à des instances de la classe « Script » par la relation « Script Language Used By », tel que défini dans la figure 1-5. Également, la ligne « Owl :script_language about="#Python" » définit que le terme « Python » est une instance de la classe « script language ». Il est possible d'ajouter d'autres relations et termes afin d'obtenir une ontologie de plus en plus complète pour représenter le plus fidèlement possible l'univers du domaine.

Avec un moteur d'inférence, il est donc possible pour une machine de comprendre que l'instance « Fichier Script Python » est une instance de la classe « Script » sans que cela soit défini implicitement dans l'ontologie. Avec l'ajout de nouvelles classes et relations, on étend l'univers propre à l'ontologie dans un format qu'une machine peut facilement interpréter.

Comme défini précédemment, une ontologie est un ensemble d'entités et de relations. Ces entités et relations appartiennent à un domaine particulier, comme la biologie, le sport ou l'histoire. Des experts définissent un vocabulaire d'entités et de relations de plus en plus descriptif du domaine générique. Par exemple, une ontologie portant sur le domaine de la biologie pourrait contenir des concepts tels "maladie" et "corps humain" ainsi que des relations entre ces concepts comme "infecte".

Dans l'optique du Web Sémantique, la construction d'ontologies est un processus collaboratif sur lequel se penchent les experts d'un domaine. Ceux-ci sont les meilleures personnes pour concevoir une ontologie complète sur leur domaine de compétence. De plus, si plus d'ontologies sont conçues avec l'optique de l'indexation, la puissance d'un indexeur utilisant ces ontologies augmentera car plus de termes et de relations seront reconnus et indexés. Il est très important qu'une ontologie utilisée avec l'objectif de permettre d'indexer des documents soit la plus complète et la moins ambiguë possible, car ces caractéristiques vont avoir un impact sur la qualité de l'indexation. Si l'ontologie utilisée ne contient pas suffisamment de concepts importants pertinents au domaine ou manque de relations entre les concepts, l'indexation en sera affectée de façon négative.

1.4 Ontologies et recherche d'information

Plusieurs auteurs représentent la connaissance du domaine sous forme d'une ontologie. Par exemple, Paralic et Kostial [15] ont tenté l'expérience de comparer les résultats d'une requête portant sur la fibrose kystique en utilisant trois techniques de mesure : indexation par mots-clés avec la mesure TF-IDF, l'indexation sémantique latente (mieux connue sous l'acronyme LSI) ainsi que l'approche par ontologies.

LSI est une technique d'indexation qui se base sur la théorie des espaces vectoriels (section 1.2). Il s'agit de prendre la matrice termes-documents (un tableau contenant tous les vecteurs de termes) et d'y appliquer une transformation SVD [27] pour réduire la matrice à quelques dimensions. Une fois la matrice réduite, son contenu représente les facteurs latents qui représentent l'appartenance des termes à ces facteurs. Une matrice séparée lie ensuite les documents à ces mêmes facteurs. Finalement, la distance cosinus est utilisée pour comparer le vecteur des termes de la requête avec les vecteurs résultants LSI pour chaque document. Un exemple de cette technique d'indexation se trouve au chapitre trois du tutoriel LSI de Thomo [26].

L'approche ontologique des auteurs se limite à l'indexation manuelle des 1239 documents en leur attribuant les concepts d'une ontologie du domaine de la fibrose kystique que le document contient. Ensuite, l'utilisateur doit obligatoirement utiliser les étiquettes des concepts de l'ontologie comme les mots-clés de sa requête.

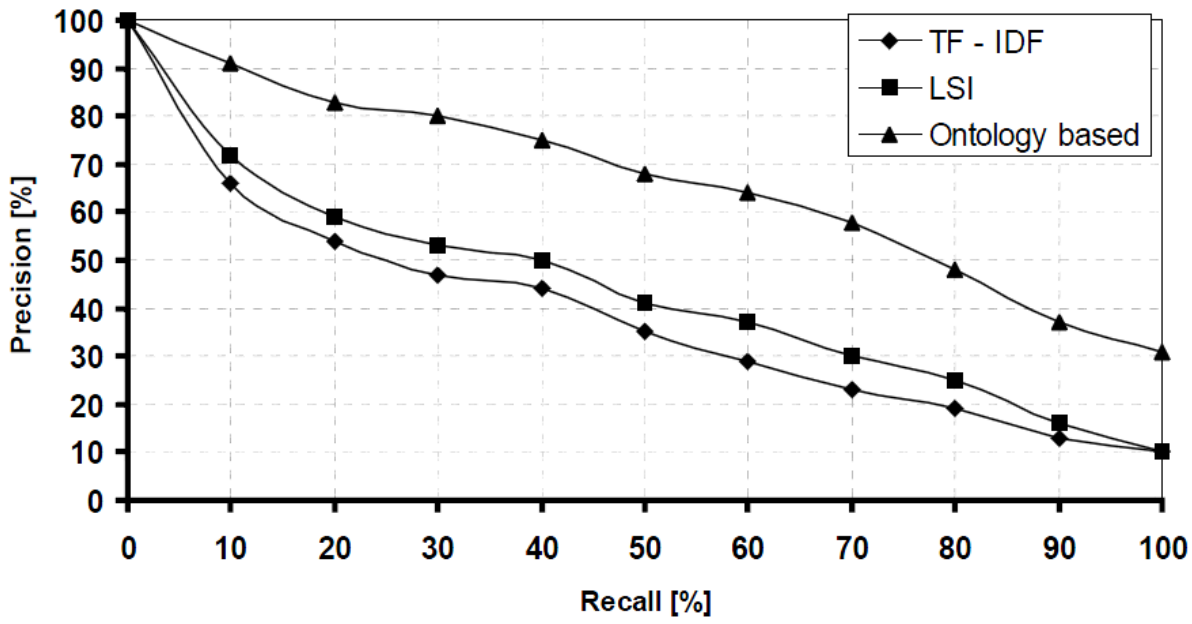

Figure 1-6 : Graphique des résultats de l'expérience de Paralic et Kostial

Les résultats de l'expérimentation de Paralic et Kostial démontrent que l'approche par ontologies produit de meilleurs résultats que des approches classiques comme la mesure TF-IDF. En effet, pour obtenir un rappel de 100%, soit un ensemble de documents contenant tous les documents pertinents à la fibrose kystique dans l'ensemble de 1239 documents, les approches classiques ne parviennent qu'à obtenir une précision de 10% tandis que l'approche par ontologies obtient une précision de 30%. Dans le cas de l'approche LSI, comme illustré à la figure 1-6, les résultats sont une faible amélioration comparativement à la mesure TF-IDF. Cela signifie que l'approche par ontologies, selon leurs résultats, permet de mieux discerner les documents pertinents de ceux qui sont non pertinents au domaine de la fibrose kystique. Une explication des calculs du taux de rappel et de la précision est décrite en détail à la section 1.5 du présent mémoire.

Un problème soulevé par Paralic et Kostial consiste justement à effectuer le rapprochement entre la requête de l'utilisateur et les concepts de l'ontologie. L'ontologie contient un ensemble fini de concepts reliés à un domaine de connaissance du créateur de l'ontologie, tandis qu'une requête de l'utilisateur, même si elle concerne le domaine, ne correspondra que

rarement à un concept présent dans l'ontologie. Par exemple, si l'ontologie contient le concept « animal de compagnie » et que l'utilisateur tente une recherche avec les mots-clés « animaux domestiques », il est important que le moteur de recherche utilisant une ontologie comprenne que l'un est le synonyme de l'autre. La correspondance entre les mots-clés d'une recherche et les concepts d'une ontologie est un domaine de recherche complexe que les auteurs n'ont pas abordé et ils se sont limités à faire la correspondance manuellement.

D'autres auteurs, comme Vallet, Fernandez et Castells [16] ont tenté une expérience qui se rapproche beaucoup de celle du présent mémoire. Ils ont constaté que des algorithmes d'indexation sémantique comme SEAL [21] et TAP [22] ont de la difficulté à classer les documents dans un ordre logique de pertinence. Les algorithmes parviennent à déterminer si un document est pertinent ou non à une requête, mais ils ne parviennent pas à déterminer si un document est plus pertinent qu'un autre. Ainsi, lors d'une recherche d'information, les moteurs de recherche sémantiques actuels trouvent les documents ayant un lien sémantique avec la requête mais ceux-ci ne sont pas ordonnancés en ordre de pertinence, ce qui cause un problème important si on essaie de construire un moteur de recherche sémantique qui va indexer des centaines de milliers de documents. Un ensemble de résultats de milliers de documents pour une requête, sans aucun ordonnancement, est pratiquement inutile pour l'utilisateur. Les moteurs de recherche sémantiques actuels comme SWSE (http://www.swse.org) ou Swoogle (http://swoogle.umbc.edu) ne font que recenser les documents concernant le concept sans les ordonnancer et le présent document propose une technique qui améliore la présentation des résultats à l'utilisateur.

Vallet, Fernandez et Castells (2005) ont construit une ontologie de base avec 3 superclasses :

1) Concepts
2) Catégories
3) Types de documents

La superclasse « Concepts » possède, comme enfants, toutes les classes représentant des concepts pertinents à l'utilisateur, comme « peinture », « sculpture » ou « œuvre d'art » si on veut que l'algorithme d'indexation se concentre sur les documents portant sur le domaine des arts. La superclasse « Catégories » représente des concepts qui ne seront jamais instanciés mais qui seraient utiles pour catégoriser les documents, comme « Culture », « Sports » ou « Politique ».

Il est important de noter ici que Vallet, Fernandez et Castells utilisent cette catégorisation pour résoudre les problèmes de synonymie. Si un texte contient le mot « Madrid », l'algorithme d'indexation va se baser sur la catégorie sous laquelle le document est instancié pour déterminer si il s'agit de la ville, de l'équipe de soccer ou du restaurant québécois. Cette façon de procéder est intéressante mais elle impose une limitation que les auteurs soulignent. En effet, la catégorisation et l'annotation des documents se fait manuellement par un expert du domaine de l'ontologie. Pour un ensemble restreint de documents cela fonctionne, mais pour des centaines de milliers de documents, l'annotation manuelle devient une tâche impensable.

Finalement, la troisième catégorie, « Types de documents », est aussi un outil d'aide pour la catégorisation pour définir le genre de chaque document qui sera indexé. Ainsi les documents seront classés sous des concepts comme « Journal », « Message » ou encore « Rapport ». Lors de la recherche, si l'utilisateur limite sa recherche aux articles de journaux, cette catégorisation des documents permettra de répondre à cette spécification.

Aussi, les documents sont annotés lorsque des concepts de l'ontologie sont présents dans le document. Lorsque plusieurs concepts de l'ontologie correspondent à un mot, comme « les iris » qui peuvent être une peinture de Vincent Van Gogh ou bien les fleurs, la correspondance se fera selon la catégorie qui a été assignée manuellement au document, comme la botanique ou la peinture. Cette approche manuelle pour faire la distinction entre des concepts distincts mais qui s'écrivent de la même façon est mentionnée par les auteurs comme peu pratique sur un grand

ensemble de documents comme un algorithme d'indexation se devrait de faire pour offrir un moteur de recherche sémantique complet sur le Web.

Le présent mémoire s'attaque directement à ce problème. Avec une ontologie d'un domaine particulier, il est possible de déterminer avec une bonne précision quels documents concernent le domaine sans avoir à annoter préalablement le document. Si un algorithme d'indexation n'a besoin que d'une ontologie pour déterminer le sujet d'un document, cet algorithme pourrait indexer des centaines de milliers de documents si des ontologies lui sont fournies. Cela n'élimine pas l'intervention humaine pour l'indexation, mais le succès de l'expérimentation du présent mémoire démontre qu'il est possible de réduire le travail requis pour obtenir un algorithme d'indexation universel performant sur des millions de documents.

Après un essai préliminaire, Vallet, Fernandez et Castells se sont rapidement rendus compte qu'il est difficile d'annoter automatiquement un document à l'aide de concepts d'une ontologie. L'ontologie ne peut contenir tous les synonymes et les déclinaisons d'un concept sans complètement alourdir la structure et le travail du créateur.

Finalement, les auteurs ont testé leur système avec 4 requêtes prédéfinies avec une ontologie de 143 concepts et 1144 instances sur 2039 documents de nouvelles. Ils ont mentionné que les requêtes d'un utilisateur ne peuvent pas toujours correspondre directement à des concepts dans une ontologie mais ils ne proposent aucune solution.

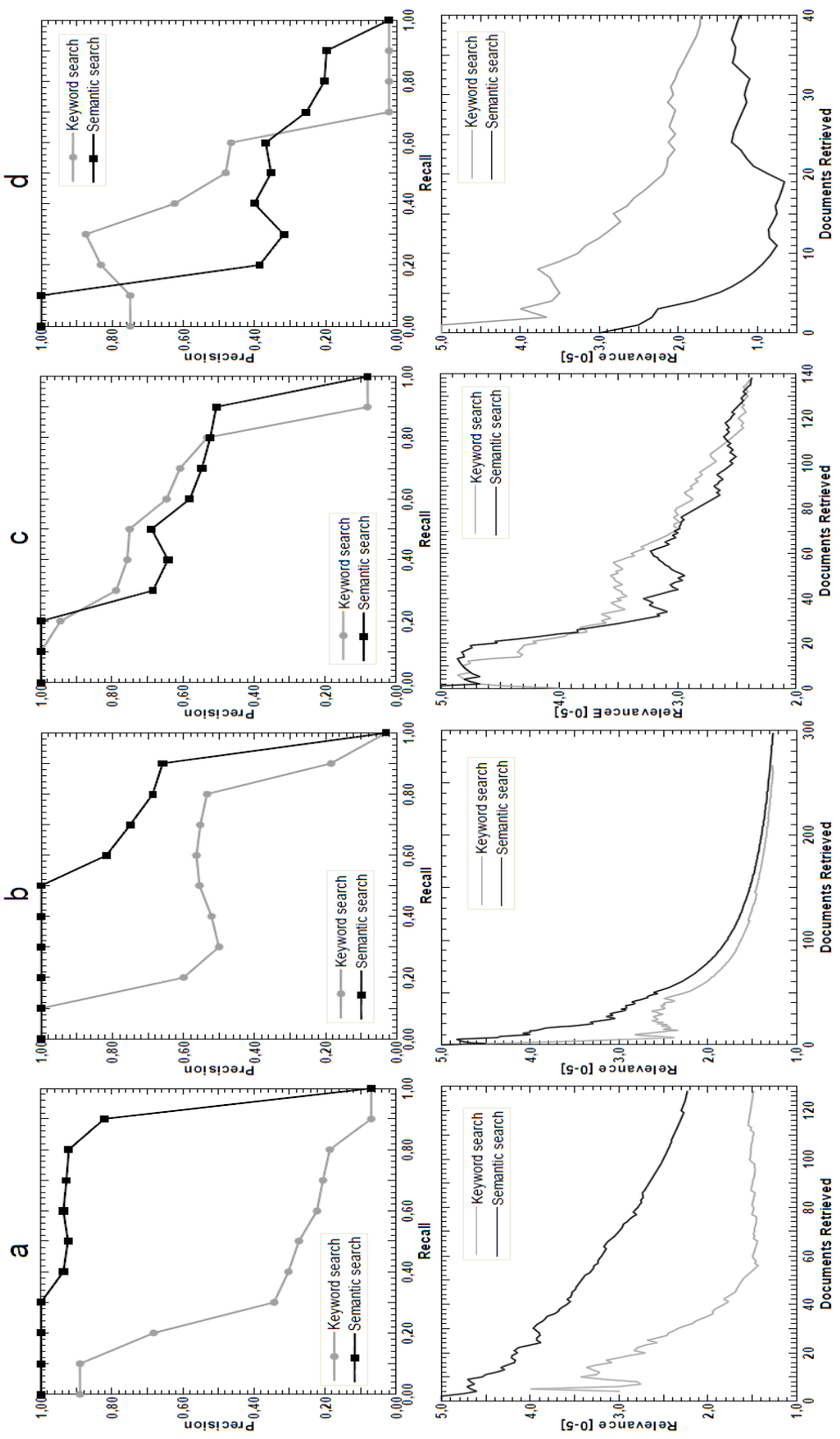

Figure 1-7 : Graphiques des résultats de l'expérience de Vallet et al.

Les graphiques de l'expérience de Vallet et al. (figure 1-7) expriment les résultats des auteurs pour chacune des 4 requêtes de l'expérimentation. Les graphiques du haut expriment le lien entre le rappel (de 0 à 100%) et la précision (de 0 à 100%). Les graphiques du bas expriment le nombre de documents retrouvés par chacune des deux techniques d'indexation testées en fonction de leur pertinence moyenne (de 1 à 5) à la requête jugée par des experts du domaine. Les lignes grasses représentent les résultats de la méthode ontologique et les lignes minces représentent les résultats d'une recherche par mots-clés standard. Les graphiques A et B démontrent que la méthode ontologique est plus précise que la méthode par mots-clés seulement si l'ontologie représente entièrement le domaine. Comme celle-ci est incomplète, on remarque une perte importante de précision lorsque le rappel passe de 90 à 100%. Leurs expériences démontrent clairement que la force de la méthode ontologique est directement dépendante de la qualité et de la complétude des ontologies attachées à l'algorithme d'indexation. Les auteurs ont délibérément testé cet aspect pour la requête C en retirant plusieurs concepts clés de l'ontologie et la performance diminue au point de n'être pas plus efficace qu'une recherche par mots-clés. La requête D porte sur la désambiguïsation des termes et les résultats montrent que une ontologie même bien construite ne peut contenir tous les concepts qui sont présents dans les documents. Cette faiblesse rend la méthode ontologique moins performante qu'une recherche par mots-clés.

D'un point de vue plus général, les grandes difficultés de l'indexation ontologique sont relevées par Gabor Nagypàl [17]. L'auteur analyse également les résultats de la recherche de Vallet, Fernandez et Castells et établit une synthèse des faiblesses de leur approche et des techniques d'indexation actuelles, soit :

1) La variété du langage naturel : les concepts ont plusieurs façons de s'écrire, ont des synonymes qui nuisent à l'analyse textuelle.

Il est vrai qu'il est impossible que l'ontologie contienne chaque façon possible d'écrire un concept. Toutefois, la lemmatisation du contenu des documents permet de regrouper plusieurs

façons d'écrire un même concept, par exemple le pluriel. Les synonymes les plus courants peuvent être ajoutés manuellement sans trop demander une charge de travail significative lorsque la lemmatisation ne permet pas d'obtenir le même lemme de base, par exemple « livre » et « bouquin ».

2) Les concepts abstraits de haut niveau : Plusieurs concepts pertinents pour des recherches, comme par exemple « Guerre d'Irak » ou « Révolution Industrielle » ne sont pas inclus spécifiquement dans des documents, ce qui empêche les approches par mots-clés utilisant des mesures comme TF-IDF, de retrouver ces documents.

Il est évident que les documents concernant la révolution industrielle ne contiendront pas tous les mots du concept « révolution industrielle » ou une variation du concept, mais il serait normal de prioriser les documents qui contiennent les mots du concept. Si aucun document ne contient les mots du concept de haut niveau, il serait donc pertinent de proposer des documents ayant le potentiel de d'être reliés au concept.

Le moteur de recherche sémantique Lexxe (http://www.lexxe.com) utilise les concepts abstraits pour enrichir la recherche : si un utilisateur recherche des informations sur la vitesse d'une bicyclette, il peut utiliser un concept ontologique prédéfini par les programmeurs (environ 500 concepts abstraits sont définis) pour trouver des documents qui ne seraient pas retrouvés avec une méthode de recherche traditionnelle par mots-clés. Par exemple, une recherche avec les mots clés « Speed : Bicycle » va retourner les documents contenant les mots-clés « speed » et « bicycle » mais également les documents qui contiennent des instances du concept ontologique « speed », comme « 18 mph » ou « 40 km/h ».

3) Mauvaise exploitation des relations ontologiques : Dans les approches actuelles, une relation ontologique cruciale et très utile comme « PartOf » n'est pas prise en compte. Un document parlant de l'Allemagne ne sera pas considéré si une requête concerne l'union européenne même si l'Allemagne en fait partie.

La mauvaise utilisation des relations ontologiques, est un point important développé dans le présent mémoire. La présence du concept recherché dans un document est importante, mais la présence d'autres mots de l'ontologie informe l'algorithme d'indexation qu'il prend une bonne décision d'annoter le document à ce concept. Si un document contient le mot « Java », la présence d'autres concepts clés comme « Internet » ou « langage » indiquent à l'algorithme d'indexation que le concept « Java » du document est potentiellement davantage rattaché au domaine de l'informatique que celui des cafés. La présence d'autres mots comme « achat » ou « café » ou « épicerie », ou l'absence complète d'autres mots sur l'informatique informerait l'algorithme d'indexation que son annotation ferait peu de sens.

4) Mauvaise gestion de la temporalité : Si une requête d'un utilisateur demande des informations sur le 19ème siècle, un document parlant des guerres napoléoniennes couvrant la période 1806-1815 est sans doute pertinent, mais ne sera pas retrouvé à moins de contenir le mot ou concept « 19ème siècle ».

La mauvaise gestion de la temporalité amène un aspect important que les ontologies pourraient améliorer comparativement à l'indexation par mots-clés. Avec une ontologie du domaine de l'histoire, il est pertinent qu'un document couvrant la bataille de Waterloo en 1815 soit retourné si un utilisateur demande des informations sur les guerres napoléoniennes. Les ontologies permettent de définir les années qui font partie d'un concept temporel (Guerres napoléoniennes) et ainsi retourner des documents qui concernent des événements précis qui se sont produits dans la période demandée par l'utilisateur (bataille de Waterloo), ce qui se rapproche du second point de Nagypàl.

En conclusion, plusieurs systèmes d'indexation qui utilisent des ontologies ont été testés. Utilisés directement ou en combinaison avec des méthodes d'indexation par mots-clés et des mesures comme TF-IDF, ces systèmes démontrent que l'utilisation d'une ontologie pour indexer un ensemble de documents constitue une amélioration notable comparativement à des approches

qui ne l'utilisent pas. La méthode décrite dans le présent mémoire propose une méthode d'indexation par ontologies basée sur les points suivants :

- Une formule du calcul de la force représentative d'un concept par rapport à un document

La méthode d'indexation par ontologies proposée dans le présent mémoire décrit la formule de calcul qui permet de déterminer la force représentative d'un document par rapport aux concepts présents dans une ontologie. Cette formule produit un score de force représentatif pour chaque terme et chaque document basé sur neuf paramètres qui sont décrits aux chapitres trois et quatre du présent mémoire.

- Exploitation des relations ontologiques pour l'indexation

Des auteurs comme Nagypàl [17] et Vallet et al. [16] proposent une telle utilisation mais l'ampleur de la tâche de l'analyse d'une ontologie même simple les a tous menés a analyser d'autres aspects d'une méthode d'indexation ontologique. Le présent mémoire propose une analyse simple des relations à l'intérieur d'une ontologie afin de déterminer si une telle analyse augmente la précision du moteur de recherche.

1.5 Évaluation de la qualité des résultats d'une recherche sur le Web

Une question importante demeure afin de pouvoir répondre à la question de recherche de l'introduction : comment déterminer la *qualité* des résultats d'une recherche sur le Web ? Un utilisateur moyen jugerait de la qualité d'un moteur de recherche avec les caractéristiques suivantes : la réponse à la requête est pertinente à ce que qu'il a demandé, la réponse contient un ensemble suffisant de documents pour obtenir une vue d'ensemble du sujet au besoin et la réponse a été fournie dans un temps raisonnable.

Nous ne consacrons pas d'efforts à la performance des algorithmes en termes de temps et de temps et de ressources de calcul. Cet aspect ne sera pas évalué tout en assumant qu'il restera dans les limites de l'acceptable pour l'utilisateur moyen. De plus, le temps de réponse est souvent dépendant de la complexité des algorithmes et de la puissance du matériel disponible pour répondre aux requêtes, des éléments qui sont en dehors du cadre du présent travail.

Nos évaluations analyseront la pertinence des réponses à la requête de l'utilisateur ainsi que de s'assurer que le plus grand nombre de documents pertinents soit retournés. Retourner à l'utilisateur le milliard de milliards de documents Web n'aurait aucune utilité et n'en retourner qu'un seul, même si c'est le plus pertinent, ne satisferait probablement pas l'utilisateur complètement.

Chez Google, la mesure de la qualité est un domaine qui n'est pas du tout automatisé [19]. Malgré le potentiel de pouvoir analyser les millions de requêtes faites chaque jour sur le moteur de recherche par un algorithme, les ingénieurs de la compagnie considèrent qu'une analyse manuelle faite par des employés de la compagnie constituent la meilleure mesure de qualité disponible actuellement. Cette approche est adoptée pour l'expérimentation du présent mémoire : le jugement final de la pertinence ou non d'un document à une requête revient à des analystes humains experts du domaine. Ce jugement permettra ensuite de mesurer la qualité d'une recherche à l'aide des calculs du rappel et de la précision [20]. Ces deux calculs sont décrits en détail dans les deux sections suivantes.

Taux de rappel d'une recherche

Une mesure courante de la qualité des résultats retournés par une requête est le rappel. Cette mesure représente la proportion de documents pertinents qui ont été retrouvés dans l'ensemble global de documents, D. Le taux de rappel d'une recherche peut être mesuré par l'équation suivante :

$$\text{Rappel} = \frac{\text{Nb d'éléments pertinents retrouvés}}{\text{Nb d'éléments pertinents dans l'ensemble D}}$$

Figure 1-8 : Équation 4 - Rappel

Par exemple, sur un ensemble de 100 documents, si 50 documents sont pertinents à une recherche et que la méthode en recense 40, le taux de rappel de la méthode se situera à 80%. Il sera ainsi possible de comparer la qualité entre la méthode *TF-IDF* et la méthode par ontologies en comparant leur taux de rappel. Mais cette méthode entraîne une autre question : comment déterminer objectivement la pertinence d'un document par rapport à une recherche? Malheureusement, la pertinence ou non d'un document restera toujours subjective aux yeux de l'utilisateur, mais la méthode pour réduire ce problème est décrite en détail au chapitre trois de ce document.

Précision d'une recherche

Un autre aspect important de la qualité d'une recherche sur le Web est la précision de la réponse pour une requête donnée. La précision est exprimée sous la forme suivante :

$$\text{Précision} = \frac{\text{Nb d'éléments pertinents retrouvés}}{\text{Nb d'éléments retrouvés}}$$

Figure 1-9 : Équation 5 - Précision

Par exemple, pour un ensemble D de 100 documents, si une requête produit un ensemble de documents R, constituant la réponse du moteur de recherche, de 50 documents et que 45 d'entre eux sont jugés pertinents, la précision de la méthode de recherche est donc de 90%. Il sera donc possible de comparer la qualité des méthodes de recherches par rapport à leur précision.

La précision et le rappel sont intimement liés. Un moteur de recherche peut obtenir un rappel de 100% en retournant tous les documents de l'ensemble D pour toutes les requêtes. Il est donc important qu'une haute valeur du taux de rappel (équation 4) soit accompagnée d'une haute valeur du taux de précision (équation 5) pour obtenir un ensemble de résultats R de qualité. Les graphiques du haut de l'expérience de Vallet, Fernandez et Castells montrent ce lien entre la précision et le rappel : plus on s'approche d'un taux de rappel de 100% (tous les documents pertinents à la requête de l'ensemble de documents D sont inclus dans l'ensemble de résultats R), plus le taux de précision chute. De nombreux documents non pertinents sont inclus dans l'ensemble R afin d'obtenir tous les documents pertinents, ce qui diminue la précision.

1.6 État de l'art de l'indexation de documents Web

Présentement, la plupart des indexeurs utilisent une variante de la méthode booléenne afin de déterminer la pertinence d'un document. C'est-à-dire que un document est considéré comme un ensemble de mots et si des mots de la requête sont dans un document, ce document sera considéré pertinent. Cette approche comporte des lacunes et l'indexation par ontologies permet d'en résoudre deux en particulier : (1) Elle n'analyse pas le contexte des documents et (2) il faut que le ou les mots de la requête soient présents dans un document pour qu'il soit considéré pertinent.

1) Les indexeurs actuels ne tiennent pas compte du contexte des documents, uniquement des mots.

Les indexeurs actuels basés sur les mots-clés utilisant la mesure TF-IDF ou des variantes de celle-ci s'appuient sur l'hypothèse que les mots d'un document pris de façon isolée suffisent à déterminer si un document concerne un sujet ou non. Mais cette approche atteint vite ses limites lorsqu'elle tente d'indexer des textes sur la base de quelques mots sans ternir compte de leur contexte. L'ordre des mots n'a pas d'importance dans le calcul de pertinence d'un document.

L'analyse du contexte des documents requiert de ne plus traiter un document comme un ensemble de mots à traiter indépendamment, mais comme plusieurs ensembles séparés par la ponctuation ou des conjonctions. Par exemple, les textes « Le serveur, d'origine amérindienne apache, m'apporte un délicieux café Java » et « Voici la procédure pour faire fonctionner votre code Java sur un serveur Apache » sont difficiles à différencier pour un algorithme d'indexation qui ne tient pas compte de l'ensemble des mots d'un document, c'est-à-dire du contexte qui entoure les mots-clés de la requête. Comme décrit dans l'article de *Hyvõnen, Saarela et Vijaanen* [12], la formulation du besoin d'information est une problématique importante lors de la conception d'un algorithme d'indexation. La mise en contexte du document devient alors cruciale pour différencier les documents et augmenter la précision de l'indexation et en conséquence, la précision de la recherche.

Pour répondre au problème de la mise en contexte, les moteurs de recherche actuels comme *Google* utilisent une source d'information très efficace : les hyperliens crées par des humains. En effet, pour déterminer si un document est pertinent à une requête donnée, si des milliers de pages réfèrent à ce document, le document en question peut ensuite être considéré comme pertinent à la requête de l'utilisateur. Son score de pertinence en sera ainsi augmenté, il aura préséance sur les autres documents lors de recherches subséquentes sur le sujet. Cette approche est très efficace et l'algorithme *PageRank* [14] de *Google* l'automatise avec succès. Serait-il possible d'automatiser complètement le processus de mise en contexte?

Avec les ontologies, il est possible de concevoir un ensemble de termes et de relations qui peuvent servir à déterminer le contexte du document de façon automatique. Par exemple, une ontologie composée de relations et de termes liés à l'informatique et une autre concernant les produits de consommation permettraient de conclure que le premier texte « Le serveur, d'origine marocaine, m'apporte un délicieux café Java » concerne les produits de consommation à cause de la grande proximité du mot « café » présent dans l'ontologie comme un produit de consommation. Les probabilités que le second texte concerne l'informatique sont beaucoup plus grandes à cause de la présence du mot « code », qui est présent dans l'ontologie comme une entité de programmation. Plus des ontologies complètes seront développées, plus il sera possible de bien mettre en contexte un document et de renforcer la précision d'une recherche Web en utilisant des ontologies au lieu d'uniquement se baser sur les mots.

2) La majorité des indexeurs actuels exigent la présence des mots dans un document pour déterminer la pertinence.

Pour indexer un document par rapport à un ou plusieurs mots, les méthodes classiques booléennes ont besoin de la présence de ces mots dans le document pour effectuer le lien avec les recherches des utilisateurs. Si un utilisateur fait une recherche en utilisant les mots « le chien », la méthode booléenne combinée avec le TF-IDF saura donner un poids très faible au déterminant

« le » étant donné sa présence dans une grande proportion de documents. Toutefois, il ne retournera que les documents contenant le mot « chien ». Mais cette façon d'indexer pose problème en ne se basant que sur la présence et la fréquence des mots pour en déterminer l'importance.

En effet, si on compare deux documents avec les contenus : « Le chien est un animal de la famille des canidés » et « Le *Canis Domesticus* est un animal de la famille des canidés », seul le premier sera compris dans le résultat de la requête ayant pour sujet : « chien ». Il est pourtant clair que les deux documents sont pertinents à la recherche, mais le second document, ne contenant pas le mot, sera exclu du résultat de la requête.

Une approche par ontologies reconnaîtrait qu'un texte contenant des termes comme « canidé », « animal » et « canis domesticus » a d'excellentes probabilités d'intéresser un utilisateur voulant s'informer sur le chien. Une ontologie d'animaux bien construite avec le concept de « chien » comme étant une sous-classe de la classe « animal » avec une relation de famille avec « canidé » et de nomenclature latine « canis domesticus » permet immédiatement de cerner le sujet du document sans que le mot recherché soit présent dans le document. Une telle approche aurait l'intérêt d'augmenter le taux de rappel d'une recherche sans en diminuer la précision.

Suite aux développements et concepts présentés dans ce chapitre, la question de recherche exprimée au chapitre un peut être transformée en l'hypothèse de recherche suivante :

La méthode d'indexation par ontologies produit-elle une recherche de meilleure qualité en termes de *précision* et de *rappel* lorsque comparée à une indexation par mots-clés utilisant le calcul TF-IDF pour un même ensemble de documents?

Cette hypothèse est l'objet d'une expérimentation décrite au chapitre deux.

CHAPITRE 2 MÉTHODOLOGIE

Ce chapitre décrit la méthode utilisée pour produire les résultats de l'étude comparative entre une méthode classique des espaces vectoriels (mots-clés avec la mesure TF-IDF) et l'approche avec ontologies.

2.1 Conception de l'approche classique des espaces vectoriels (calcul TF-IDF)

Une fois l'ensemble de documents D déterminé, un logiciel a été conçu pour utiliser la méthode des mots-clés avec la mesure TF-IDF pour indexer les documents de l'ensemble.

Pour utiliser cette méthode, il faut construire un tableau de vecteurs représentant chacun des documents. Il faut donc extraire chacun des mots présents dans tous les documents du corpus d'expérimentation. Il est important de lemmatiser ou segmenter (*stemming* en anglais) chacun des termes des documents pour que les différentes façons d'écrire un seul et même concept soient réduites au même terme. Par exemple, « bonbons » représente le même concept que « bonbon ». Pour cette étude, un lemmatiseur a été utilisé, celui du paquetage « tm » de R [23] qui est basé sur l'algorithme de Porter. Par exemple, si on analyse un ensemble de dix millions de documents et que mille d'entre eux contient le mot « céréales », la valeur IDF du terme sera le logarithme de 10000000/1000, soit 4. Si un document de cent mots contient le terme « céréales » trois fois, la valeur TF-IDF sera donc de 12. Cette mesure sera utilisée pour comparer les résultats à ceux de l'indexation par ontologies au chapitre quatre.

Prenons un exemple pour illustrer le fonctionnement de la méthode avec un ensemble D de quatre documents :

Document 1 : « Un grand réseau est une version plus étendue d'un petit réseau »

Document 2 : « La construction de réseaux informatiques est complexe »

Document 3 : « Joyeux noël et bonne année! »

Document 4 : « J'ai construit un réseau informatique, un réseau téléphonique sans fil et un réseau social. Mais sur un tout autre sujet, mon opinion sur la grève étudiante est la suivante : La hausse des droits de scolarité est nécessaire pour rattraper le temps perdu depuis 30 ans. Toutefois, un gouvernement aussi corrompu doit être chassé du pouvoir dans un avenir immédiat. »

Avec cet ensemble D, l'algorithme calcule la valeur TF-IDF de chaque terme pour chacun des quatre documents de l'ensemble. Par exemple, si on applique directement l'équation 2 à chacun des documents pour le terme « réseau », les résultats sont :

Pour le document 1 : TF = 2, IDF = log(4/3) = 0.125, TF x IDF = 0.25

Pour le document 2 : TF = 1, IDF = log(4/3) = 0.125, TF x IDF = 0.125

Pour le document 3 : TF = 0, IDF = log(4/3) = 0.125, TF x IDF = 0

Pour le document 4 : TF = 3, IDF = log(4/3) = 0.125, TF x IDF = 0.375

Ensuite, la similarité cosinus permet de comparer chacun des termes de la requête de l'utilisateur avec chacun des documents de l'ensemble D pour ensuite produire un score de similarité. La valeur la plus élevée de similarité sera associée au document ayant le plus de termes similaires avec la requête en tenant compte du poids relatif des termes calculé avec TF-IDF. La qualité de ces résultats sera comparée à ceux produits par l'indexation par ontologies.

Voici un exemple très simple de distance cosinus avec un seul document et une requête :

Document : Julie aime Paul plus que Linda aime Paul.

Requête : Mélanie apprécie Paul plus que Julie aime Paul.

Chaque terme a un poids de valeur 1 (cette valeur est généralement modifiée par le calcul TF-IDF) ce qui produit les deux vecteurs suivants :

Document : [Julie(1), aime(2), Paul(2),plus(1),que(1),Linda(1),Mélanie(0),apprécie(0)]

Requête : [Julie(1),aime(1),Paul(2),plus(1),que(1),Linda(0),Mélanie(1),apprécie(1)]

Avec l'équation 3 utilisée dans l'utilitaire de Applied Software Design [24], le cosinus entre le vecteur document et le vecteur requête est d'environ 0.822 ce qui donne un angle d'environ 35 degrés entre les deux vecteurs. Comme l'angle est plus près de 0 degrés (similitude parfaite) que de 90 degrés (documents totalement différents), on peut affirmer que la requête et le document ont une certaine similitude.

2.2 Conception de l'approche par ontologies

Dans le cadre de l'expérimentation de la méthode ontologique, une ontologie portant sur le domaine générique des technologies Web a été construite dans le cadre du présent mémoire avec plus de 50 classes et relations liées au domaine des technologies Web. Avec cette ontologie, il est possible d'expérimenter l'indexation de documents selon leur degré de pertinence relatif au domaine. Une ontologie est construite avec un éditeur tel que Protégé et un script PHP la convertit en une structure de données. Une fois cette structure complétée, l'algorithme d'indexation l'utilise pour l'indexation de documents texte et produire un ordonnancement des documents pour une requête donnée.

Pour la construction de la structure de données à partir du fichier de l'ontologie, il est important de bien analyser toute l'information que l'ontologie procure.

```
<Class about="#Script">
  <label>script</label>
     <subClassOf resource="#items_on_server"/>
     <subClassOf>
       <Restriction>
          <onProperty resource="#appliedOn"/>
          <allValuesFrom  resource="#items_on_server"/>
       </Restriction>
```

```
      </subClassOf>
      <subClassOf>
        <Restriction>
          <onProperty resource="#hostedOn"/>
          <allValuesFrom resource="#web_server"/>
        </Restriction>
      </subClassOf>
      <comment>fichier executable pour l'exécution de routines web</comment>
</Class>
<Class about="#script_language">
   <label>langage de script</label>
   <label>script language</label>
      <subClassOf>
        <Restriction>
          <onProperty resource="#scriptLanguageUsedBy"/>
          <allValuesFrom resource="#Script"/>
        </Restriction>
      </subClassOf>
      <comment>langages permettant d'effectuer des tâches répétitives (routine) sur des documents ayant
une caractéristique commune</comment>
</Class>
<script_language about="#Python">
 <label>python</label>
      <type resource="&owl;Thing"/>
      <comment>langage de script python</comment>
      <scriptLanguageUsedBy resource="#Script_Python"/>
</script_language>
<ObjectProperty about="#scriptLanguageUsedBy">
      <range resource="#Script"/>
      <domain resource="#script_language"/>
</ObjectProperty>
```

Figure 2-1 : Partie de l'ontologie d'expérimentation sur les technologies Web

Avec les informations de la figure 2-1, représentées dans le format OWL, il est possible d'utiliser un parseur XML pour extraire les informations utiles pour l'indexation, comme les étiquettes(« labels ») contenant les mots qui représentent une classe ou une instance, ou bien les relations parent-enfant comme « Python » qui est défini comme une instance de la classe « Script Language » (la balise « Class » est remplacée par une balise de la classe mère). Une relation parent-enfant entre deux classes, comme entre « Script » et « Fichiers sur un serveur » est représentée par la balise « subClassOf » avec l'attribut « resource ». Les autres liens entre les classes sont représentés par le contenu des balises « ObjectProperty ».

Par exemple, un parseur XML adapté aux ontologies pourra obtenir les informations suivantes :

-- Il existe deux classes, l'une nommée "script_language" et l'autre nommée "script"

-- Ces deux classes sont reliées entre elles par une relation nommée "scriptLanguageUsedBy"

-- Pour la classe "script", le mot français représentant ce concept est "script"

-- Pour la classe "script_language", le mot français représentant ce concept est "langage de script"

-- Il existe une instance de la classe "script_language", nommée "Python"

-- Le mot représentant l'instance "Python" est "python"

-- etc…

Toutes ces informations doivent être représentées dans un format qui est utilisable par un algorithme pour l'indexation. Un graphe directionnel est un format idéal pour la représentation, dont un exemple partiel est fourni à la figure 2-2. Le graphe représentant l'ontologie est construit à de 3 différentes composantes :

1) Les classes représentant les concepts et les mots qui représentent la classe.

2) Les instances des classes et les mots qui représentent l'instance.

3) Les relations entre les classes et les instances.

Ensuite, cette représentation est traduite sous forme de triplets (instance/classe, relation, classe). Par exemple, la figure 2-1 produit plusieurs triplets dont (Python, InstanceOf, Script_Language), ce qui représente la relation suivante : « Python est une instance de la classe 'langage de script' ».

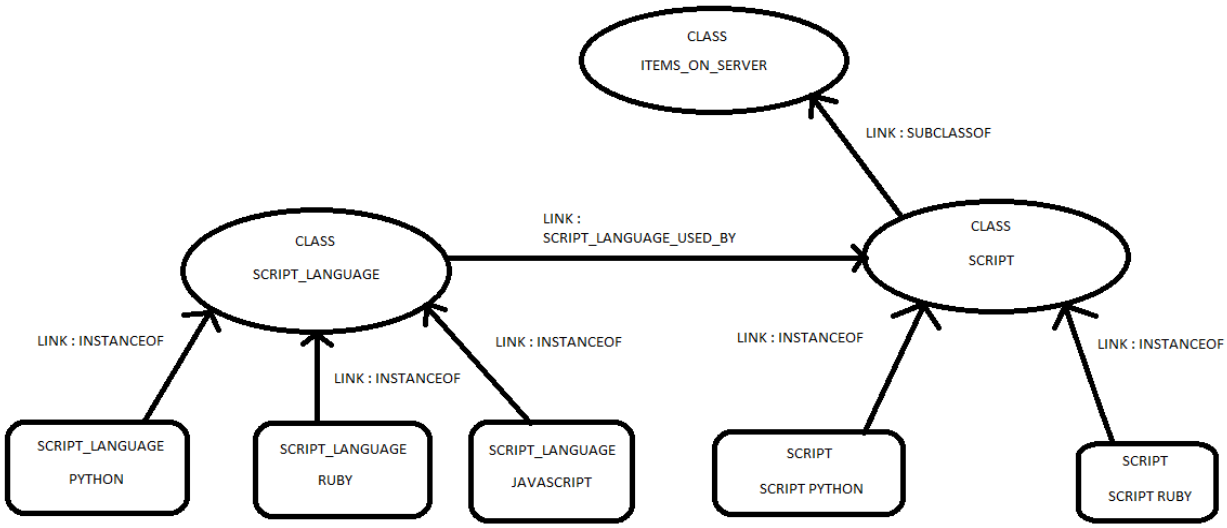

Figure 2-2 : Représentation graphique partielle de l'ontologie sur les technologies Web

Une fois les relations entre classes extraites de l'ontologie, il faut découper chaque document de l'ensemble D en mots comme dans le cas de l'algorithme TF-IDF. Toutefois, le découpage se limite aux mots qui font partie des étiquettes des classes ou des instances des classes de l'ontologie. Tous les autres termes qui ne sont pas des classes ou instances de classes sont ignorés. Par exemple, avec une ontologie des pays du monde ayant chaque nom de pays représenté comme une instance de la classe « pays », le découpage des documents se limite aux termes « Canada », « Brésil », « France », « pays » et ainsi de suite. Si l'ontologie des pays contient une instance nommée « Arabie Saoudite » et qu'un document contient uniquement le mot « Arabie », le mot sera conservé pour l'indexation car il fait partie de l'étiquette d'une instance de l'ontologie.

Ce découpage produit un tableau de termes où chaque élément correspond à un terme et les informations nécessaires au calcul de la pertinence. Un élément contient les informations suivantes :

1) Le mot trouvé.

2) La classe parent ou l'instance de classe auquel il appartient (dans le cas des classes composées de plusieurs mots. Par exemple « Langage de script »).

3) L'identificateur unique du document dans lequel le mot a été trouvé.

4) La fréquence de la classe ou de l'instance reliée au mot.

5) La fréquence du mot.

6) La longueur du document.

Par exemple, le document 1928 analysé avec la partie de l'ontologie de la figure 2-2 contenant le texte « Le langage Python » produira trois entrées dans le tableau :

a) « langage, langage de script, 1928, 0, 1, 3 »

b) « Python, langage de script, 1928, 0, 1, 3 »

c) « Python, Python, 1928, 1, 1, 3 »

Ainsi, une entrée de tableau peut représenter soit la présence d'une partie d'une classe composée de plusieurs mots (a), une inférence sur la classe parent causée par la présence d'une instance de la classe enfant (b), ou la présence entière d'une instance d'une classe (c). Cette simplification est adéquate pour les fins de la recherche.

Avec les relations de l'ontologie et l'ensemble des termes retrouvés dans les documents, il est possible d'effectuer l'indexation des documents et d'ordonnancer la pertinence d'un document par rapport à une requête sur une base sémantique. Il faut définir un ensemble de règles qui vont déterminer si les informations fournies par l'ontologie ont un impact positif ou négatif sur la pertinence d'un document par rapport à une requête d'un utilisateur. L'information fournie par l'ontologie permet une mise en contexte du document (1) et donne aussi un pouvoir d'inférence à

l’algorithme d’indexation à l’aide de la distance ontologique (2). Ces deux caractéristiques sont les principales innovations de la méthode par ontologies et sont décrites en détail aux sections 2.3 et 2.5.

2.3 Mise en contexte

Lors de l'analyse du document, les méthodes classiques d’indexation se basent principalement sur la présence ou non du mot recherché par l'utilisateur pour déterminer la pertinence du document. Une approche d’espace vectoriel se base sur la fréquence du mot, sa fréquence dans l'ensemble des documents classifiés et la transformation TF-IDF, ainsi que sur la taille du document pour déterminer un ordonnancement entre les documents déterminés comme pertinents.

L'approche par ontologies permet d’effectuer une mise en contexte des termes en associant les documents qui les contiennent à des thématiques. En utilisant une ontologie contenant les termes et les relations propres à un domaine, tous les documents pertinents au domaine pourront être ordonnancés selon leur degré de pertinence à une recherche donnée par un utilisateur.

L'approche ontologique se base également sur des techniques connues d’indexation. Si le sujet d'un texte est sans équivoque, l'utilisation de l'ontologie produit des informations supplémentaires redondantes et son utilisation apporte du renfort par rapport aux méthodes d’indexation traditionnelles. En plus de considérer la fréquence des termes comme les méthodes traditionnelles, elle analyse le contexte du document pour confirmer que le terme correspond à un domaine précis.

TitreCours: Introduction aux technologies XML

DescriptionCours: Utilisation des technologies XML pour la gestion, le stockage, la distribution des documents d'affaires sur le Web. Intégration de XML aux bases de données et aux applications existantes.

Figure 2-3 : Description textuelle du cours IFT1152

Le document de la figure 2-3, une description de cours, est un exemple type de document dont le sujet est facilement identifiable à l'aide de n'importe quelle technique d'indexation utilisant la fréquence de mots pour en déterminer le sujet. La présence du mot XML dans la section du titre ainsi que sa présence à trois reprises dans un document de seulement trois phrases, permet de déterminer que le concept clé du document est le XML. Une mesure comme le TF-IDF, appliquée uniquement sur ce document, donne la valeur la plus élevée au terme « XML » si on exclut les déterminants comme « le » ou « la ».

Par contre, les documents ne sont pas tous aussi clairs par rapport aux sujets qu'ils contiennent. Par exemple :

TitreCours: Conception de sites Web dynam. et transact.

DescriptionCours: Conception de sites Web complexes pour la génération dynamique de contenu et la gestion d'interactions avec les utilisateurs. Présentation générale de l'architecture du Web et du protocole HTTP (HyperText Transfer Protocol). Structure d'un document HTML (HyperText Markup Language). Mise en forme d'un document HTML par l'utilisation de CSS (Cascading Style Sheet). Paradigmes de conception propres aux systèmes Web. Programmation du cote serveur. Gestion d'une session sur un site Web. Élements de sécurité pour les sites Web. Présentation du format XML (Extended Markup Language) et du langage de transformation de documents XSL (Extended Stylesheet Language). Programmation du cote client par le biais de scripts exécutes par le navigateur Web. Interface avec une base de données relationnelle. Notions de performance et de sécurité. Notions de validation et de test de sites Web dynamiques et transactionnels.

Figure 2-4 : Description textuelle du cours LOG4420

Le document de la figure 2-4, une autre description de cours, est générique et couvre plusieurs sujets. Une technique classique d'indexation, comme les mots-clés avec la mesure TF-IDF, classifierait le document sous tous les mots importants selon leur fréquence, ce qui donnerait plus d'importance au terme « Web ». Cette façon de faire donne des informations importantes sur le sujet du document, mais l'utilisation d'une ontologie permet d'approfondir les informations que le texte fournit. Dans le cas du document de la figure 2-4, avec une ontologie, l'indexation détermine que le document contient beaucoup de termes qui concernent les technologies Web mais que certains termes n'ont pas de lien direct entre eux(comme « base de données » et « CSS ») alors que d'autres sont reliés par une relation ontologique(« XSL est un langage de transformation de documents XML ». Cette haute fréquence de termes de l'ontologie montre que le contexte du document est fort probablement les technologies Web (http, CSS, XML, XSL, web, base de données).

La section suivante de ce document décrit en détail les relations entre les classes de l'ontologie qui vont avoir un impact sur le calcul de pertinence d'un document.

2.4 Mots-clés représentatifs de l'ontologie

Comment déterminer si un texte est pertinent à une recherche plutôt qu'à une autre en se basant sur une ontologie? Comme l'ontologie est remplie de mots représentant ces classes, la première façon de calculer la pertinence du document est de vérifier la fréquence du mot recherché, mais également de tous les mots de l'ontologie présents dans le document.

Par exemple, dans le cas du document de la figure 2-3, la fréquence du mot "XML" dans le document ainsi que son positionnement dans la section du titre permet de déterminer que le sujet principal du document est le langage XML. Mais se fier uniquement sur la présence ou non du terme recherché peut être très efficace, mais aussi causer des erreurs de classification dans de nombreux cas.

TitreCours: Zoologie des reptiles

DescriptionCours: Évolution des reptiles. Étude des différences et caractéristiques entre différentes espèces de reptiles (serpents, amphibiens, lézards). Étude de l'anatomie des serpents québécois (pythons, couleuvres) et des grands serpents (boas, cobras). Dissection d'un python.

Figure 2-5 : Description textuelle du cours 2-407-300

La description de cours de la figure 2-5 constitue un exemple de la limite de la capacité d'analyse des méthodes classiques. Python est un langage de script utilisé dans plusieurs systèmes. Le terme "python" est toutefois un terme ambigu, il peut représenter le langage de script ou, comme dans le texte de la figure 2-5, une espèce de serpent.

L'utilisation d'une ontologie permet de résoudre cette ambiguïté. Si le texte ne contient qu'un seul mot présent dans l'ontologie, celui recherché par l'utilisateur, et qu'un autre texte contient le mot recherché ainsi que deux ou plusieurs mots qui sont également présents dans l'ontologie, il est beaucoup plus probable que le second texte soit pertinent à la recherche. Par exemple, le texte « Apprentissage de la programmation de scripts avec le langage Python » contient des termes appartenant au domaine des technologies Web tels que « script », « programmation » et « python » alors que le texte de la figure 2-5 ne contient que « python ». Le texte de la figure 2-5 devient moins pertinent car le terme est isolé.

Des textes comme « Le suisse est un rongeur vivant dans les forêts du Canada » ou « La Suisse est un pays d'Europe utilisant le franc suisse comme monnaie » sont faciles à différencier si un contexte ontologique les entoure. Par exemple, des triplets relationnels (classe/instance, relation, classe/instance cible) comme (suisse, PaysDans, Europe) ou (suisse, AnimalEspece, rongeur) permettent de distinguer les polysèmes (mots ayant plusieurs sens pour la même écriture).

La première force de la méthode ontologique est la résolution de tels cas simples d'ambiguïtés pour déterminer le contexte du document. La probabilité qu'un document concerne les technologies Web est statistiquement plus importante si des mots-clés propres aux technologies Web s'y retrouvent.

2.5 Distance ontologique

La présence ou non de mots-clés de l'ontologie dans un document donne de bonnes informations sur le contexte du document, mais cette présence a des effets très variables sur la pertinence du document par rapport à un besoin d'information de l'utilisateur. Une ontologie peut couvrir un domaine très vaste de connaissances mais le besoin de l'utilisateur est souvent très précis, touchant une infime partie des connaissances de l'ontologie. Par exemple, une ontologie contenant des connaissances du domaine de la biologie pourrait être divisée en sections bien distinctes, comme la biologie cellulaire, la biologie végétale ainsi que la biologie animale et humaine. Si un utilisateur est intéressé par le processus de la photosynthèse des végétaux, le moteur de recherche va rechercher les documents ayant le degré de pertinence le plus élevé pour la photosynthèse. Pour calculer cette pertinence avec l'ontologie, il faut vérifier si le document contient d'autres termes de l'ontologie pour en déterminer la pertinence. Si c'est le cas, il faut vérifier quelle sorte de relation existe entre les termes trouvés dans le document.

C'est dans l'objectif de bien cibler les documents pertinents que la notion de distance ontologique est introduite. La distance ontologique consiste en l'ordre des liens entre deux termes de l'ontologie. Une relation entre deux termes (relation parent-enfant, une instance avec sa classe mère ou bien 2 classes reliées par une relation) est d'ordre 1 alors qu'une relation fraternelle est d'ordre 2 entre deux instances avec la même classe mère.

Par exemple, si un document contient les mots "langage de script", "script", "ruby", "HTML" et "python" et que l'ontologie de la figure 2-2 est impliquée, le système détecte que "python" est une instance du concept "langage de script". Cela implique qu'une relation directe

d’ordre 1 concerne ces deux termes. Donc, si un utilisateur fait une recherche sur l'un de ces deux termes, le document sera plus pertinent qu'un autre document n'ayant pas de relations directes entre les termes. De la même façon, le système détecte la relation directe entre les mots "script" et "langage de script" par leur relation nommée "scriptLanguageusedBy" représentée dans la figure 2-2. Il est tout à fait logique qu'un document ayant des termes liés par des relations soient plus pertinents pour les termes concernés. Finalement, comme le terme "ruby" est une instance de la classe "langage de script", le système détecte une relation fraternelle (c’est-à-dire une relation enfant-parent-enfant) avec le terme "python", une relation d’instance de classe avec "langage de script" et une relation d’ordre 2 avec "script".

Le résultat final est une sorte de toile liant les termes présents dans le document.

Pour la classe « Langage de script » : présence de termes représentant la classe. Présence de termes représentant deux instances de la classe, soit « Python » et « Ruby ». Présence de termes représentant une classe reliée d’ordre 1 via la relation « scriptLanguageUsedBy ». Présence du terme « HTML » représentant une instance avec une distance ontologique d’ordre 3 ou plus.

Pour la classe « Script » : présence du terme représentant la classe. Présence de termes représentant des instances avec une distance ontologique d’ordre 2, soit « Python » et « Ruby ». Présence de termes représentant la classe « langage de script » reliée d’ordre 1 via la relation « scriptLanguageusedBy ». Présence du terme « HTML » représentant une instance avec une distance ontologique d’ordre 3 ou plus.

Pour l’instance « Python » : présence du terme représentant la classe. Présence de termes représentant des instances liées par une relation fraternelle « Ruby ». Présence de termes représentant une classe avec une distance ontologique d’ordre 2 « Script ». Présence du terme « HTML » avec une distance ontologique d’ordre 3 ou plus.

Pour l'instance « Ruby » : présence du terme représentant la classe. Présence de termes représentant des instances liées par une relation fraternelle « Python ». Présence de termes représentant une classe avec une distance ontologique d'ordre 2 « Script ». Présence du terme « HTML » représentant une instance avec une distance ontologique d'ordre 3 ou plus.

Pour l'instance « HTML » : présence du terme représentant la classe. Présence des termes « Python », « Ruby », « langage de script », « script » représentant des classes et des instances avec une distance ontologique d'ordre 3 ou plus.

Cela implique que pour chacun des 4 premiers termes, les liens avec trois autres termes augmente le score de pertinence. Toutefois, la présence du terme « html », concept présent dans une ontologie du Web, n'augmentera pas de façon significative le score de pertinence des quatre autres termes car il n'est relié à aucun d'entre eux par une relation d'ordre 1 ou 2. Les liens d'ordre 3 et plus sont jugés faibles dans le cadre du présent mémoire. Sa présence implique que le document a plus de chances de concerner les technologies Web mais pas un domaine plus précis comme les langages de script. En abordant la situation de la façon inverse, un document qui contient le terme « HTML » une seule fois mais des dizaines de fois les termes reliés aux langages de script, le document n'aura probablement pas une forte pertinence pour quelqu'un recherchant des informations sur le langage HTML. L'information est considérée comme étant diluée par rapport aux autres concepts présents dans le document.

CHAPITRE 3 MÉTHODE ONTOLOGIQUE

Ce chapitre décrit en détail la conception de la méthode ontologique. Il aborde la description du corpus de documents pour l'expérimentation ainsi que le processus menant à la formule du calcul de pertinence des documents pour la méthode. Une expérience pour juger la pertinence des documents du corpus y est décrite.

3.1 Définition du corpus d'expérimentation

Nous utiliserons un ensemble de 24694 documents pour tester chacune des deux méthodes d'indexation. 16335 de ces documents contiennent le titre et une description sommaire d'un cours qui est enseigné dans une des quatre plus grandes institutions universitaires francophones de l'île de Montréal, soit l'UQAM (Université du Québec À Montréal), le HEC (Hautes Études Commerciales), l'Université de Montréal et l'École Polytechnique de Montréal. 8359 autres documents contiennent un résumé d'une page Web du site fr.wikiversity.org dans le domaine de l'informatique. Cet ensemble est de taille raisonnable pour utiliser la méthode TF-IDF ainsi que la méthode par ontologies proposée. En effet, comme mentionné au chapitre un, le calcul TF-IDF dépend du nombre de documents pour son calcul IDF. De plus, ces documents de description de cours contiennent une grande concentration de mots qui ont une importance significative pour déterminer le sujet et le contexte du document. Des exemples de documents sont fournis aux figures 2-3, 2-4, 2-5 et 2-6.

```
TitreCours: Reseaux informatiques

DescriptionCours: Classification des reseaux. Techniques de commutation.
Architectures technologiques de transmission. Tramage, detection
d'erreurs, controle du flot et controle d'erreurs par retransmission.
Architecture des reseaux : modele par couches, relations entre les
couches et primitives de controle. Protocoles des reseaux locaux :
Ethernet et reseaux sans fil. Architecture technologique TCP/IP
(Transport Control Protocol/Internet Protocol) : modele, adressage,
protocoles, routage, gestion du trafic, services et qualite des services.
Applications de TCP/IP. Architecture technologique ATM (Asynchronous
Transfer Mode).
```

Figure 3-1 : Exemple de fichier contenant la description d'un cours

Ces documents, tous en français, sont ensuite individuellement analysés par chacune des deux techniques d'indexation (mots-clés avec la mesure TF-IDF et par ontologies) pour être traités afin d'en extraire l'information contenue et ainsi indexer chacun des fichiers.

3.2 Calibration de la pertinence

Les critères de pertinence de la méthode ontologique sont définis comme suit :

1) La fréquence dans un document du terme recherché

2) La taille du document

3) La fréquence dans un document des termes représentant des classes ou instances reliés au terme recherché (liens d'ordre 1 et 2)

4) La fréquence dans un document des termes représentant des classes ou instances éloignées dans l'ontologie (liens d'ordre 3 et plus)

5) Les inférences des classes vers les instances et vice-versa. Si une classe/instance est présente dans un document mais que ses instances/classe mère ne le sont pas, ses instances/classe mère sont considérées comme également présentes.

Il faut maintenant déterminer le processus de calibration, soit le calcul du score de pertinence basé sur l'ensemble des 5 éléments. Quel poids donner à chacun des 5 critères?

Quelques principes nous guident. Un document qui ne contient qu'un seul terme de l'ontologie est le cas le plus simple à analyser pour l'algorithme d'indexation par ontologies. Si le terme est peu fréquent dans un document, il est probable que le document est peu ou pas pertinent à une recherche d'information concernant le terme. Plus le terme se répète dans un document, plus il est probable qu'il soit le sujet central du document et devient pertinent à la recherche d'information. Si le terme est le seul concept de l'ontologie présent dans le document, sa

pertinence doit donc être très basse initialement et augmenter proportionnellement selon sa fréquence dans le document, ainsi la méthode ontologique ressemble aux méthodes classiques dans ces circonstances. Un document contenant le terme « XML » 3 fois sera plus pertinent qu'un autre document qui ne le contient qu'une seule fois.

Ensuite, si le terme recherché est présent et que d'autres termes de l'ontologie sont également présents, la pertinence pour chaque terme doit diminuer. Il est logique que, si deux termes de l'ontologie sont présents dans un document à fréquence égale, la probabilité que le document soit pertinent pour chacun des deux termes est plus grande que si un document comprend dix termes de l'ontologie. Par exemple, si l'utilisateur fait une recherche sur le concept « XML », un document contenant « Apprentissage des technologies XML et le langage XSL » sera considéré comme plus pertinent qu'un document contenant « Apprentissage des technologies XML, PHP, JavaScript, le protocole HTTP et le langage XSL » car le concept recherché est dilué dans un ensemble plus grand de termes de l'ontologie dans le second document.

Toutefois cette diminution de la pertinence doit être compensée si les autres termes du document sont reliés aux termes de la requête. Par exemple, le document contenant « Apprentissage des technologies XML et le langage XSL » est très pertinent à la fois pour des requêtes portant sur « XML » ou « XSL ». Leur présence dans le même texte et leur relation ontologique renforcent la pertinence du document comparativement à un document comme « XML et bases de données » qui est également pertinent pour une requête portant sur le « XML » mais moins que le premier document étant donné l'éloignement entre les classes dans l'ontologie.

Le score de pertinence sera donc dilué par le nombre de termes de l'ontologie éloignés des termes de la requête présents dans un document. Un document comprenant beaucoup de termes n'est pas pertinent pour chacun des termes, il est probablement un document touchant à beaucoup de sujets sans les approfondir. Par contre, si les termes sont directement reliés au terme recherché avec des relations de l'ontologie comme dans l'exemple du paragraphe précédent, l'effet sera inversé et la présence des autres termes augmentera le score de pertinence.

La méthode ontologique produit donc un score de pertinence qui est calculé selon les critères suivants, en ordre de priorité :

1) La fréquence du(des) terme(s) de la requête dans un document.

2) La fréquence des termes dans un document reliés par une relation parent-enfant ou fraternelle aux termes de la requête (ordre 1 ou 2).

3) La fréquence des termes de l'ontologie dans un document.

Et sera diminué par les critères suivants, en ordre de priorité :

1) La fréquence de termes de l'ontologie présents dans le document sans relation avec les termes de la requête (ordre 3 ou plus).

2) La taille du document qui contient le terme analysé

Le poids relatif de chaque critère est une estimation objective basée sur la méthode, alors l'expérimentation comprendra une approche expérimentale d'optimisation des valeurs pour produire l'ensemble de résultats pertinents. Une formalisation des critères est proposée à la section 3.3.

3.3 Expérimentation

Pour l'expérimentation, il est important de définir clairement les objectifs de la méthode de recherche pour ensuite déterminer une procédure pour mesurer l'efficacité de la méthode. Comme défini dans le chapitre précédent, la méthode ontologique attribue une valeur pondérée à chaque caractéristique du document qui se rapproche ou s'éloigne de la requête de l'utilisateur. Par exemple, si l'utilisateur désire trouver tous les documents qui concernent la technologie XML

en utilisant le mot-clé « XML » pour sa requête, la méthode ontologique va traiter chacun des documents et faire un calcul de pertinence selon des critères précis comme la fréquence du mot recherché, la présence de concepts reliés ou non reliés à la requête, la taille du document, etc…

L'objectif final de l'indexation par ontologies est de fournir un ensemble D de documents ordonnancés selon leur degré de pertinence. L'expérimentation est considérée comme un succès si la méthode d'indexation ontologique obtient une meilleure qualité de recherche, sous forme de taux de rappel et taux de précision, comparativement à la méthode par mots-clés utilisant le calcul TF-IDF et la distance cosinus.

La formule (1) ci-dessous définit le calcul de pondération de différents facteurs qui entrent en jeu pour déterminer la pertinence tels que décrits dans la section précédente (3.2). Ces cinq critères sont représentés par neuf paramètres. Cette différence entre le nombre de critères et le nombre de paramètres s'explique par le fait que le critère de la fréquence des termes peut s'exprimer de plusieurs façons. Tous les termes représentant une classe peuvent être présents ou seulement une partie. Une pondération distincte doit être attribuée pour chacun des cas spécifiques à chacun des cinq critères de la section précédente.

$$P_{i,j} = (w_1 F_{m,j} R_i + w_2 F_{i,j} + w_3 F_{Hi,j}) (1-C) + w_4 F_{Pk,j} + (w_5 F_{i,j} R_i) (C) + w_6 F_{o,j} + w_7 F_{c,j} - w_8 F_{n,j} - w_9 T_j$$

$$C = 1 \text{ si } R_i = 1, \text{ sinon } C = 0$$

$w_{1\ldots9}$: Poids de chacun des 9 paramètres déterminés expérimentalement

$P_{i,j}$: Score de pertinence d'un concept (i) dans un document (j).

$F_{m,j}$: Fréquence d'un terme (m) qui n'est qu'une partie du concept (i) (dans le cas où le

concept (i) est constitué de plusieurs mots, ex : Langage de script) du concept (i) dans un document (j).

R_i : Ratio du concept (i) (1 / Nombre de mots de plus de 2 lettres composant le concept (i)). Dès qu'un concept (i) est constitué de plus qu'un mot, R_i sera inférieur à 1.

$F_{i,j}$: Fréquence du concept (i) dans un document (j).

$F_{Hi,j}$: Fréquence (H) où au moins 2 mots d'un concept (i) composé de plusieurs mots sont retrouvés dans une même phrase dans un document (j).

$F_{Pk,j}$: Fréquence (P) d'un concept (k) de l'ontologie qui est relié par une relation parent-enfant au concept (i) dans un document (j).

$F_{o,j}$: Fréquence d'un concept de l'ontologie (o) dans un document (j).

$F_{c,j}$: Fréquence d'un concept de l'ontologie (c) relié par une relation d'ordre 1 ou 2 au concept (i) dans un document (j).

$F_{n,j}$: Fréquence d'un concept de l'ontologie (n) relié par une relation d'ordre 3 ou plus au concept (i) dans un document (j).

T_j : Taille du document (j) en nombre de caractères.

Figure 3-2 : Formule 1 - Formule initiale du calcul de pertinence de la méthode ontologique

Les valeurs des différents paramètres seront optimisées pendant l'expérimentation pour trouver les valeurs qui donnent la meilleure qualité de recherche. La qualité de la recherche sera mesurée par le degré d'accord de la pertinence telle que déterminée par trois experts du domaine. Pour chaque requête, ces experts ont fourni les documents les plus pertinents.

Toutefois, la méthode ontologique propose des défis intéressants en ce qui concerne les concepts constitués de plus d'un mot. En effet, des concepts comme « langage de script » ou « standard ISO de programmation » ont un plus grand potentiel de liens ontologiques car chacun des mots a potentiellement ses propres relations dans l'ontologie. Par exemple, pour le concept « langage de script », les mots clés sont « langage » et « script ». Comment l'analyseur doit réagir si un document ne contient que le mot « langage » ? Uniquement le mot « script », mais trois fois ? Les deux mots du concept, mais séparés dans le document ? Toutes ces questions ne se posent pas dans le cas des concepts constitués d'un seul mot, mais complexifient beaucoup la tâche d'analyse de la pertinence. La formule (1) s'applique à la fois aux concepts composés d'un seul mot et à ceux constitués de plusieurs mots et une analyse sera faite sur les résultats entre les concepts d'un seul mot et ceux de plusieurs mots. Dans le cas d'un concept composé de plusieurs mots, dès qu'un mot de plus de 2 lettres du concept est présent dans le document, le concept est considéré comme étant présent dans le document.

Un échantillon représentatif de l'ontologie doit être défini pour déterminer la meilleure version de la formule (1) du calcul de pertinence. Un ensemble G de seize concepts sera utilisé pour optimiser chacun des neuf paramètres (Wi) de la formule (1). Pour déterminer la valeur optimale pour chacun des paramètres, il suffit de déterminer lesquels, parmi les dix premiers résultats de l'algorithme pour une requête portant sur l'un des seize concepts présélectionnés, sont jugés pertinents à la requête. Des graphiques seront ensuite produits afin de déterminer quelle valeur du paramètre produit les résultats les plus précis pour les seize concepts.

Pour chaque paramètre (Wi), la technique classique du « Hill-Climbing » est utilisée. Un paramètre à la fois sera modifié, pour trouver la valeur optimale de celui-ci qui procure les meilleurs résultats, avant de faire varier un autre paramètre. Une fois que tous les paramètres auront été optimisés, le résultat final sera considéré comme un optimum local. Répéter cette technique à plusieurs reprises, en modifiant l'ordre dans lequel les paramètres seront modifiés, produira une liste d'optimums locaux dans l'espace multidimensionnel. Le meilleur résultat parmi tous ces optimums sera considéré comme la forme finale et optimisée de la formule du

calcul de la pertinence et sera utilisée pour l'expérimentation finale sur l'ensemble D de concepts tel que défini au chapitre précédent.

La liste des neuf paramètres qui seront utilisés pour l'optimisation est telle que définie dans la formule (1) avec les variables (Wi), soit :

- (1) Fréquence d'un terme dans le document qui n'est qu'une partie d'un concept composé de plusieurs mots (Valeur initale : 150 * (Nombre de mots du concept trouvés / Nombre de mots composant le concept))
- (2) Fréquence du concept composé de plusieurs mots dans le document. (Valeur initiale : 150)
- (3) Fréquence où au moins 2 mots d'un concept composé de plusieurs mots sont retrouvés dans une même phrase d'un document. (Valeur initiale : 25 * (Nombre de mots du concept retrouvés dans une même phrase / Nombre de mots composant le concept)
- (4) Fréquence d'un concept de l'ontologie relié par une relation parent-enfant au concept. (Valeur Initiale : 50)
- (5) Fréquence du concept composé d'un seul mot dans le document. (Valeur Initiale : 40)
- (6) Fréquence d'un autre concept l'ontologie dans le document. (Valeur Initiale: 30)
- (7) Fréquence d'un autre concept de l'ontologie présent dans le document et relié par une relation d'ordre 1 ou 2. (Valeur Initiale: 7.5)
- (8) Fréquence d'un autre concept de l'ontologie présent dans le document et relié par une relation d'ordre 3 ou plus. (Valeur Initiale: -2)
- (9) Taille du document en nombre de caractères. (Valeur Initiale : -0.01/caractère)

La variation portera sur chacun des neuf paramètres et la valeur optimale du premier paramètre (W1) sera utilisée pour calculer la valeur optimale du second (W2) et ainsi de suite. Pour s'assurer que l'ordre dans la variation des paramètres ne fausse les données en produisant un optimum local, cette technique sera répétée avec un ordre de variation des paramètres différent

pour augmenter la probabilité de tomber sur l'optimum global que produirait la méthode du gradient.

Quels mots-clés de recherche seraient idéaux pour optimiser la formule? Les concepts jugés pertinents pour l'optimisation doivent couvrir à la fois le plus de mots et avoir certains mots en commun afin de vérifier si la méthode ontologique est capable de faire la distinction entre les concepts même si les mots sont similaires. Les concepts utilisés seront donc :

- Script
- XML
- Langage de script
- Langage de programmation Web
- Service Web
- Système de base de données
- Outil de gestion de configuration
- Langage de base de données
- Protocole de communication Web
- Serveur Web
- Java
- Javascript
- HTTP
- applet
- SQL
- HTML

3.4 Optimisation de la formule du calcul de la pertinence

Les valeurs initiales, telles que spécifiées dans la liste des neuf paramètres qui seront optimisés, produisent un ensemble de résultats qui servira de base pour le processus

d’optimisation. L’objectif est de varier chacun des neuf paramètres et d’augmenter la qualité des ensembles de résultats pour les seize concepts. L’analyse de la qualité d’un ensemble de résultats d’une recherche est effectuée par des experts du domaine des technologies Web. Prenons un exemple détaillé pour illustrer ce calcul. Les experts effectuent des recherches avec le moteur de recherche ontologique et obtiennent des résultats comme l’exemple suivant :

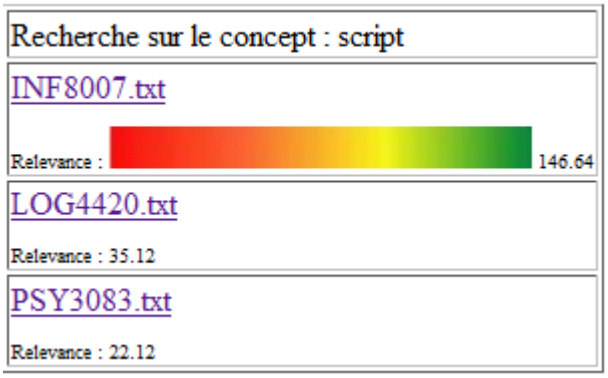

Figure 3-3 : Présentation graphique des résultats pour une requête

Dans le cas d’une requête portant sur le concept « Script », seul 3 documents contiennent le concept mais ces trois documents comportent des degrés de pertinence très variables par rapport à la requête. La barre qui se termine dans le ton de vert indique que le premier document est un document très pertinent à la requête alors que l’absence de barre pour les deuxième et troisième documents signifie qu’ils sont peu pertinents.

TitreCours: Langages de script

DescriptionCours: Caractéristiques des langages de script. Principaux langages et domaines d'application. Programmation avec un langage de script : syntaxe, structures de contrôle, structures de données, communication interprocessus et communication avec une base de données, modules clients et serveurs.

Figure 3-4 : Document 1 - INF8007.txt

TitreCours: Conception de sites web dynam. et transact.

DescriptionCours: Conception de sites web complexes pour la génération dynamique de contenu et la gestion d'interactions avec les utilisateurs. Présentation générale de l'architecture du web et du protocole HTTP (HyperText Transfer Protocol). Structure d'un document HTML (HyperText Markup Language). Mise en forme d'un document HTML par l'utilisation de CSS (Cascading Style Sheet). Paradigmes de conception propres aux systèmes web. Programmation du cote serveur. Gestion d'une session sur un site web. Éléments de sécurité pour les sites web. Présentation du format XML (Extended Markup Language) et du langage de transformation de documents XSL (Extended Stylesheet Language). Programmation du cote client par le biais de scripts exécutés par le navigateur web. Interface avec une base de données relationnelle. Notions de performance et de sécurité. Notions de validation et de test de sites web dynamiques et transactionnels.

Figure 3-5 : Document 2 - LOG4420.txt

TitreCours: Mémoire humaine

DescriptionCours: Principales approches à l'étude de la mémoire. Système de traitement, d'entreposage, de recouvrement d'information. Mémoire a court terme, a long terme, épisodique, sémantique. Mnemotechniques, plans, scripts et schémas. Théories de l'oubli.

Figure 3-6 : Document 3 - PSY3083.txt

Ces trois documents représentent bien les différents niveaux de pertinence qu’un analyseur humain pourrait attribuer à différents documents. Dans le cas du document 1, il est évident que le cours s’adresse a des étudiants voulant apprendre les langages de scripts et, par incidence, les scripts. Le document 1 est donc très pertinent pour une requête portant sur les scripts.

Le document 2 est un exemple de document ou la pertinence à la requête est plus nuancée. Il contient effectivement le mot « script » mais il est un concept parmi tant d'autres. En conséquence, le cours concerné par le document 2 semble contenir des informations sur les scripts, mais il couvre tant d'autres sujets que la proportion d'information concernant le sujet de la requête de l'utilisateur est minimisée. Ce document se révèle donc comme étant moyennement pertinent à une requête portant sur le concept « Script ». Il est potentiellement utile mais d'autres options plus focalisées sur les scripts seraient sans doute plus utiles pour l'utilisateur.

Finalement, le document 3 constitue une sorte de piège pour la méthode ontologique ou n'importe quel algorithme de classification de documents Web. Il contient le concept « Script » mais il est fort improbable qu'il s'agisse du concept relié à l'informatique. Pour la méthode ontologique, le concept est solitaire dans un ensemble de mots qui n'ont pas de lien avec les domaines des technologies Web et du génie logiciel. Ainsi, la probabilité que ce document puisse satisfaire les besoins d'un utilisateur intéressé par les scripts est minimale.

La version optimisée de la formule du calcul de la pertinence (1) devra conserver la capacité de bien différencier les documents pour les concepts à un seul mot et les concepts composés de plusieurs mots pour s'assurer que la méthode ontologique s'applique sur toutes les formes possibles de concepts. Le tableau des résultats sera donc séparé en deux sections : l'une pour les concepts composés d'un seul mot et l'autre pour les concepts composés de plusieurs mots.

Comme point de départ pour l'optimisation de la formule, les valeurs initiales ont été estimées sur la base d'une pré-expérimentation et d'un jugement subjectif sur leur importance. Les paramètres sont classés en trois catégories :

1) Paramètres critiques

La présence du concept recherché (paramètres W1 W2 et W4) dans un document procure beaucoup d'information sur la pertinence du document. Ces paramètres doivent donc avoir un poids élevé dans la formule. Le poids est réduit dans le cas des concepts à un seul mot car les probabilités de synonymie et d'utilisation hors contexte sont plus élevées que les concepts à mots multiples.

2) Paramètres secondaires

Les paramètres secondaires (paramètres W3, W5, W6, W7 et W8) permettent de déterminer si un document est pertinent ou non à un concept si les paramètres critiques ne parviennent pas à déterminer la pertinence simplement par la présence des mots-clés dans le document.

3) Paramètre de bris d'égalité

La taille du document (paramètre W9) est un paramètre qui sert simplement à réduire le score de pertinence des documents longs. Un document long a plus de chances de couvrir beaucoup de sujets non reliés à la recherche de l'utilisateur. La présence unique du mot-clé dans un document de 300 ou 30000 mots a un impact sur la pertinence du document reflétée avec ce paramètre. La valeur initiale est très basse et n'aura d'impact que sur les très gros documents.

Avec cette logique de division des paramètres en 3 groupes d'importance, des valeurs initiales ont été déterminées pour les neuf paramètres (voir formule 1). En conséquence, la méthode pour déterminer la valeur optimale consiste à faire varier le paramètre autour de la valeur par défaut et de déterminer combien de documents dans la liste des résultats sont pertinents au concept recherché.

Des points sont perdus lorsque des documents pertinents sont omis de l'ensemble des résultats ou si un document pertinent n'a pas été classé plus haut qu'un document jugé non pertinent. Par exemple, si l'ensemble des 10 résultats de l'algorithme concernant le concept « Langage de programmation Web » produit les résultats suivants :

Figure 3-7 : Ensemble de dix résultats pour une requête

Si un document jugé comme potentiellement pertinent en est exclu (Ex : LOG4420) et qu'un document de l'ensemble de résultats comme IFT3065 est jugé non pertinent, le score de

classification pour cette requête sera de 9 car 9 documents retournés sur 10 sont pertinents. La valeur optimale d'un paramètre sera établie en le faisant varier jusqu'à obtenir la valeur maximale du score de classification. En répétant ce processus pour chacun des neuf paramètres (Wi) et sur tout l'ensemble G, il sera possible de déterminer la meilleure valeur pour chacun des neuf paramètres.

3.5 Évaluation de la pertinence par des experts du domaine

Avant l'expérience d'optimisation, il faut déterminer quels documents sont pertinents pour chacun des 16 concepts. L'évaluation d'un seul expert, soit l'auteur du mémoire, est trop subjective. Ainsi l'évaluation de la pertinence ou non des documents du corpus a été effectuée par trois experts du domaine soit :

1) Expert 1, bachelier en ingénierie logicielle. Homme de 29 ans.
2) Expert 2, étudiant au doctorat en ingéniérie logicielle à l'université McGill et chargé de cours à l'École Polytechnique. Homme de 26 ans.
3) Expert 3, directeur des technologies de l'information chez Amigo-Express, homme de 35 ans.

Chaque expert a évalué tous les documents qui contenaient au moins un terme qui constitue un des 16 concepts de l'expérimentation soit environ 200 documents. Ils avaient ensuite la tâche de déterminer si chaque document répondait à la question suivante par oui ou par non :

Ce document décrit-il un cours qui a, au moins, un minimum de potentiel de m'en apprendre davantage sur les termes de ma requête?

Si la majorité (au moins 2 experts sur 3) répondent à la question par l'affirmative pour un document et un concept donné, ce document est évalué comme pertinent au concept.

L’expérience d’optimisation devra ensuite s’appliquer à tenter de retrouver le plus grand nombre de ces documents.

Tableau 3.1 : Évaluations de pertinence des documents pour chaque concept par trois experts du domaine

Concept	Documents jugés pertinents par l’expert 1	Documents jugés pertinents par l’expert 2	Documents jugés pertinents par l’expert 3	Documents jugés pertinents par la majorité des experts
Script	12	11	3	11
XML	14	14	11	14
HTML	10	22	10	10
Java	10	6	10	10
Javascript	10	7	10	10
HTTP	10	4	10	10
SQL	11	23	10	11
Applet	3	3	3	3
Langage de script	15	18	11	15
Langage de programmation Web	14	14	15	14
Service Web	2	2	5	2

Système de base de données	20	21	13	20
Outil de gestion de configuration	8	8	7	8
Langage de base de données	17	28	Non-évalué	17
Protocole de communication Web	10	16	6	10
Serveur Web	7	17	2	7
Total de documents pertinents aux 16 concepts				172

On remarque que, même avec une question de départ identique et des connaissances du domaine similaires, les experts obtiennent des résultats de pertinence très constants pour plusieurs concepts. Les experts ont un recoupement presque parfait et les documents pertinents des experts les plus sélectifs se retrouvent tous dans l'ensemble des documents pertinents de l'expert ayant la notion la plus élargie de pertinence. Seul le seuil de pertinence propre à chaque expert varie. Ces résultats démontrent que, pour une même requête, les utilisateurs ont des attentes différentes qui complexifie la tâche d'indexation afin qu'elle puisse satisfaire le plus d'utilisateurs possible.

Il est important de préciser que, pour la suite de l'expérimentation, que seuls les dix premiers résultats retournés par chacune des deux méthodes d'indexation (espace vectoriel et ontologies) seront vérifiés pour valider leur appartenance au groupe des 172 documents jugés pertinents par la majorité des experts. Ce choix se justifie par une recherche de l'entreprise Google sur ses propres utilisateurs [25] qui conclut que moins de 3% des liens retournés au-delà des 10 premiers sont cliqués par les utilisateurs.

Ce choix implique également que le calcul de la précision est simplifié. La méthode des espaces vectoriels retourne plus de 400 documents pour une seule requête et le calcul de la précision tel que défini dans l'équation 5 nécessiterait l'analyse de chacun de ces documents par un expert afin de déterminer si ils sont pertinents ou non à la requête, ce qui est une tâche trop importante pour cette expérimentation. En limitant le nombre de documents retournés pour une recherche à 10, le nombre maximal possible de documents retournés est donc de 160 (16 requêtes, 10 résultats par requête). Le nombre maximal de documents pertinents pouvant être retournés au total sur les 16 requêtes est de 140.

Le maximum possible de documents pertinents retournés est inférieur à 160 parce que plusieurs concepts, comme « Service Web » et « Applet » n'ont que quelques documents pertinents. À l'opposé de ces deux concepts, le concept « Langage de base de données » a 17 documents pertinents mais seulement 10 documents peuvent être retournés alors l'efficacité maximale est déjà atteinte si les 10 documents retournés sont pertinents. Ce raisonnement appliqué aux 16 concepts établit la valeur 140.

CHAPITRE 4 EXPÉRIENCE D'OPTIMISATION

Ce chapitre décrit l'expérience menant à l'optimisation de la formule du calcul de la pertinence (formule 1).

4.1 Expérience d'optimisation

4.1.1 Paramètre W1 : Présence des parties d'un concept à termes multiples dans un document.

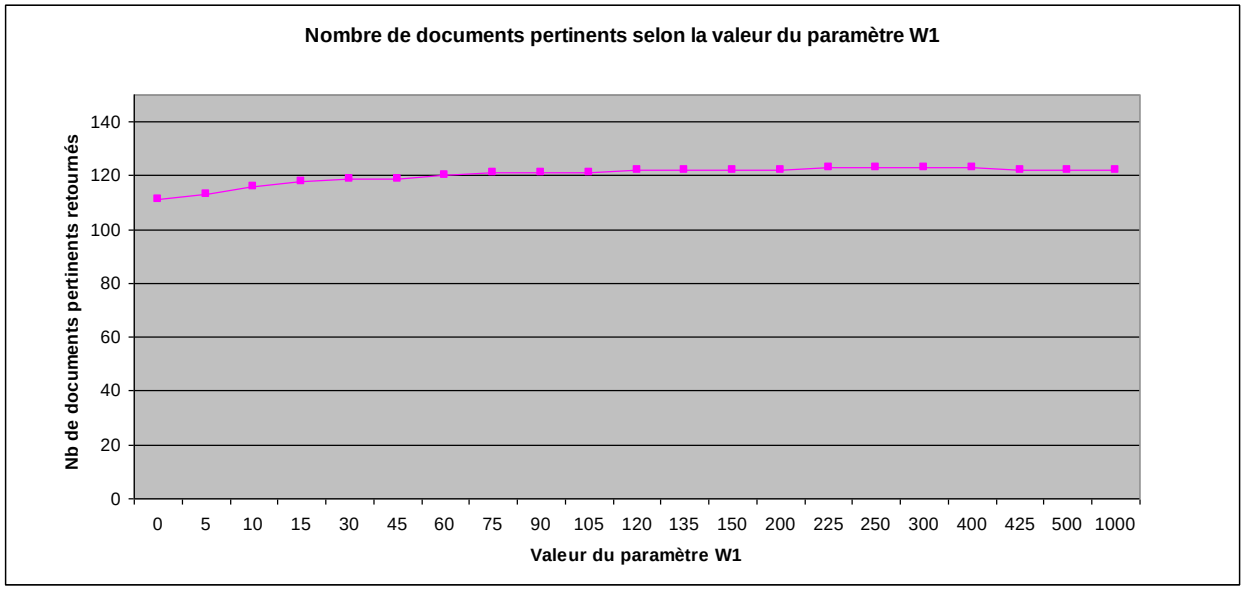

Figure 4-1 : Graphique des résultats sur la variation du paramètre W1

Il est important de noter que la précision (nombre de documents retournés) reste constante à 147 pendant toute l'expérience d'optimisation alors celle-ci n'est pas incluse dans les graphiques.

Les résultats de l'expérimentation du paramètre W1 indiquent que le paramètre ne semble pas affecter la performance au-delà d'une certaine valeur (environ 150). Ce comportement est en partie expliqué par le corpus de documents. Il y a très peu de documents contenant des termes de l'ontologie qui ne sont pas pertinents au domaine des technologies de l'Internet. Dès qu'un

document contient des termes qui constituent une partie d'un concept composé de plusieurs mots, il est considéré comme pertinent pour de grandes valeurs de W1. Ces documents ressortent pour des grandes valeurs de W1 mais sont trop peu nombreux pour affecter la performance du paramètre.

Un plus grand nombre de documents portant sur des thèmes variés ferait sans chuter la performance du paramètre pour des valeurs élevées au lieu de simplement plafonner.

Si la valeur du paramètre est à 0, 111 documents pertinents sont retournés pour les 16 requêtes de l'expérimentation. La performance du paramètre W1 est maximale pour une valeur entre 225 et 425 qui retourne 123 documents pertinents. La valeur 325 est donc retenue pour la suite de l'expérimentation.

4.1.2 Paramètre W2 : Présence intégrale d'un concept composé de plusieurs termes dans un document.

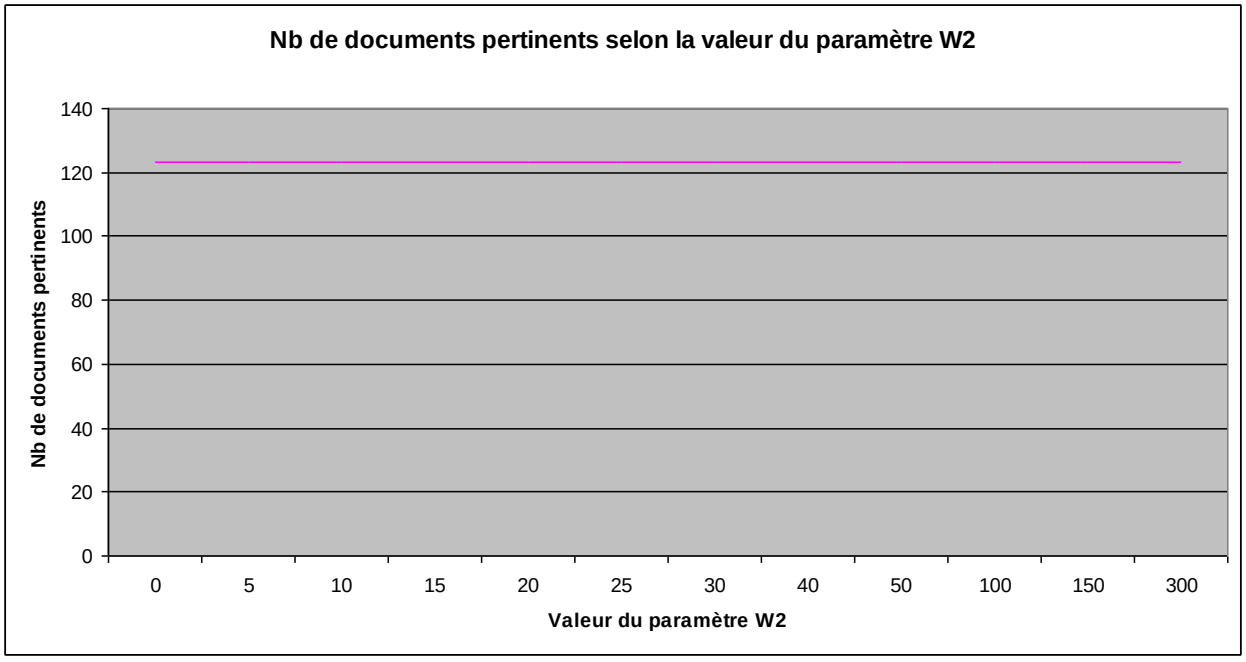

Figure 4-2 : Graphique des résultats sur la variation du paramètre W2

Les résultats de l'expérimentation du paramètre W2 indiquent que la performance du paramètre est nulle. Ce comportement est causé par le fait que très peu de documents contiennent exactement un concept de l'ontologie. Ils les contiennent souvent soit en partie ou répartis en plusieurs endroits dans le document. Ces cas plus fréquents sont affectés par les paramètres W1 et W3. Le nombre de documents pertinents retournés par la formule demeure à 123. L'expérimentation suggère que le paramètre est superflu et peut être retiré de la formule de calcul de pertinence (1).

4.1.3 Paramètre W3 : Présence de deux ou plus parties d'un concept constitué de plusieurs termes dans une même phrase dans un document.

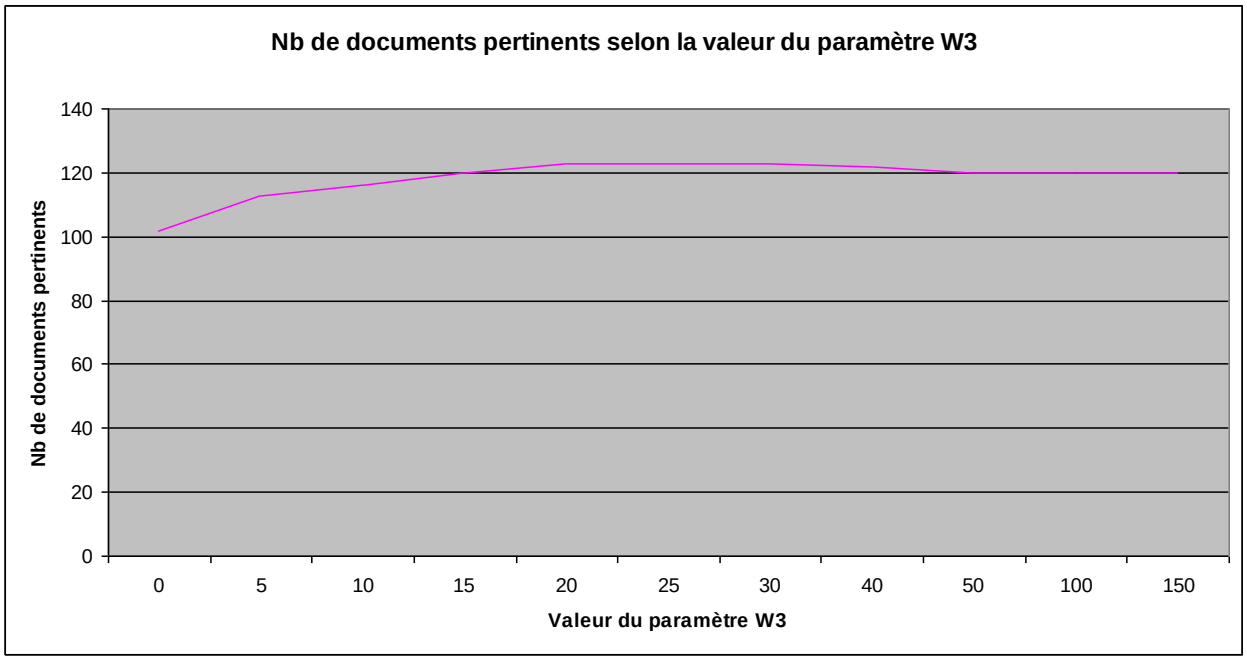

Figure 4-3 : Graphique des résultats sur la variation du paramètre W3

Si le paramètre W3 a une valeur nulle, seulement 102 documents pertinents sont retournés par la formule de calcul de pertinence (1). Toutefois, dès qu'on attribue une valeur au paramètre, le nombre de documents pertinents retournés augmente et atteint une valeur maximale lorsque le

paramètre prend comme valeur entre 20 et 30. Pour la valeur 25, 123 documents pertinents sont retournés. La valeur 25 est donc retenue pour la suite de l'expérimentation.

Il est intéressant de noter que d'attribuer une valeur élevée pour le paramètre W3 ne change que très peu les résultats car très peu de documents contiennent des parties de concepts de l'ontologie dans une même phrase qui ne sont pas reliés au domaine des technologies Web et du génie logiciel. Un ensemble de documents plus varié contenant des termes pouvant mener a plus d'ambiguïté produirait une courbe plus prononcée pour des valeurs extrêmes du paramètre W3.

4.1.4 Paramètre W4 : Inférence sur les concepts reliés (parent/enfant) présents dans un document.

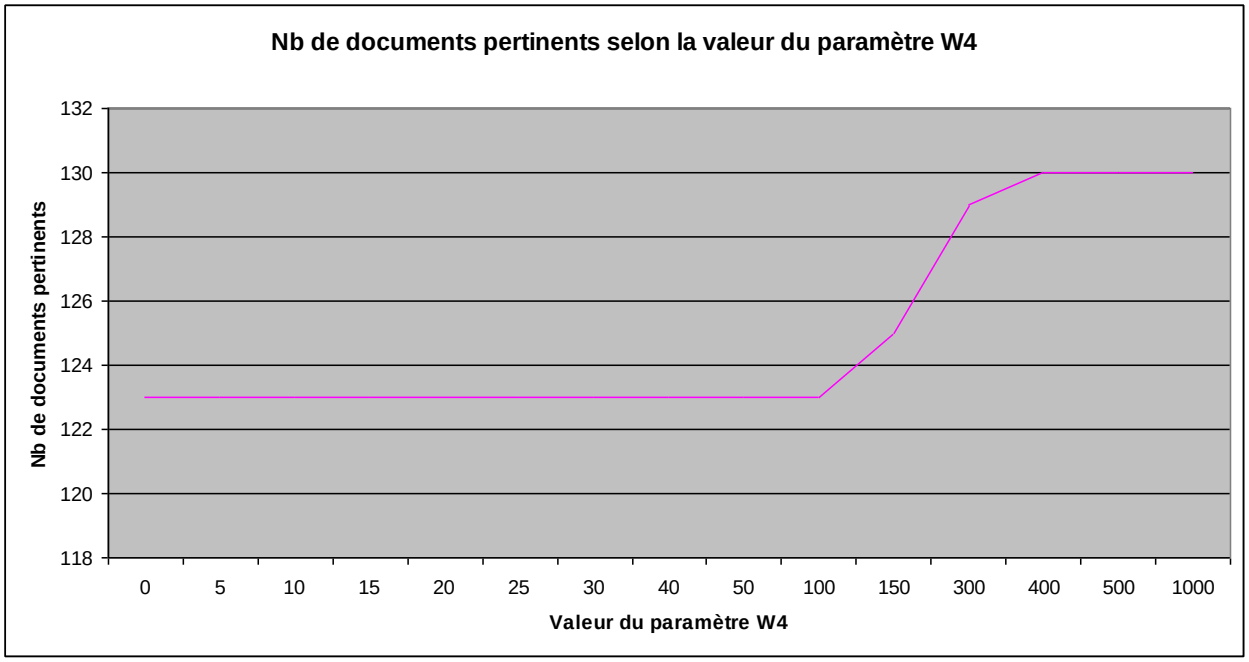

Figure 4-4 : Graphique des résultats sur la variation du paramètre W4

Les résultats de l'expérience sur le paramètre W4 indiquent que l'utilisation d'ontologies améliore la performance de la formule de calcul de pertinence (1). Sans l'utilisation d'inférences,

soit une valeur du paramètre de 0, le nombre de documents pertinents demeure à 123 alors qu'avec une valeur du paramètre élevée le nombre de documents pertinents retournés augmente rapidement avant de plafonner à 130 documents pertinents retournés.

Les résultats indiquent que l'inférence est infaillible et qu'elle ne fait qu'améliorer les requêtes mais il faut préciser que le corpus d'expérimentation ne contient pas de documents qui peuvent produire des ambiguïtés. Par exemple, aucun document du corpus ne parle des pythons ce qui crée une ambiguïté avec le langage de script du même nom. Ainsi l'inférence de l'ontologie « Python est un langage de script » ne sera jamais fausse et améliore la performance du paramètre.

Une approche demandant à l'utilisateur de préciser quel concept il recherche est à considérer afin que les inférences soient les plus efficaces que possible. Sinon le paramètre doit être ajusté pour limiter le nombre d'inférences incorrectes. La valeur 400 pour le paramètre W4 est conservée pour la suite de l'expérimentation.

4.1.5 Paramètre W5 : Fréquence d'un concept constitué d'un seul terme dans un document.

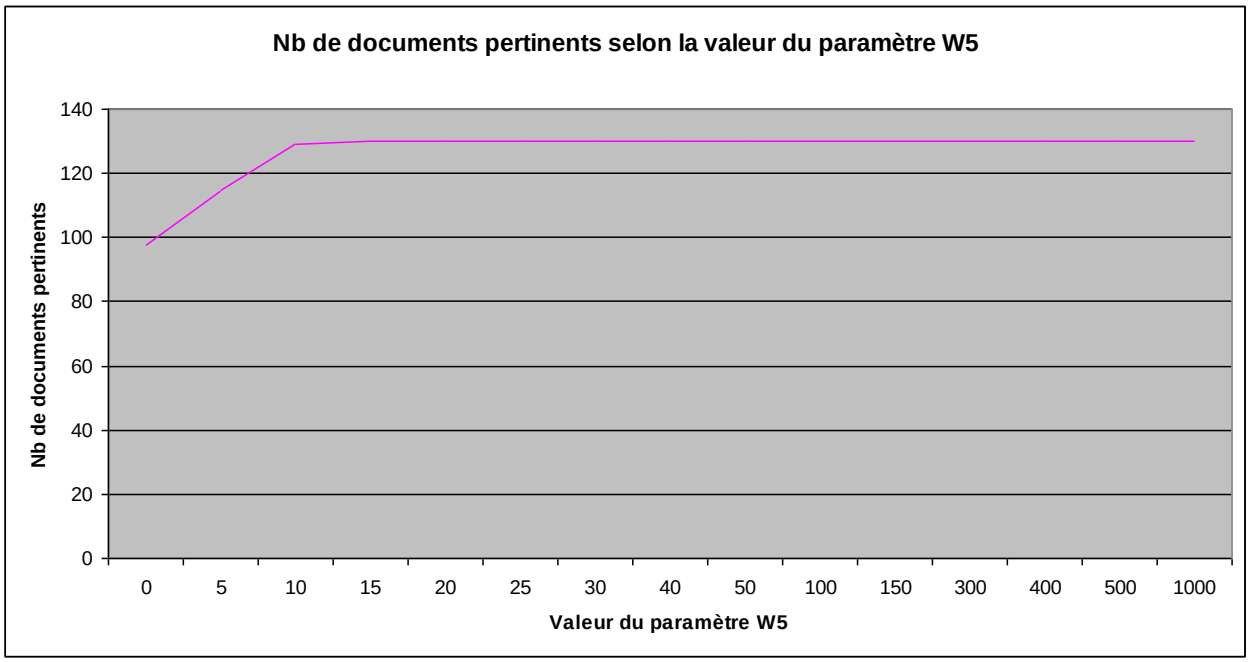

Figure 4-5 : Graphique des résultats sur la variation du paramètre W5

Les résultats de l'expérience sur le paramètre W5 indiquent que la fréquence du concept démontre clairement la pertinence du document à la requête. Même des valeurs très élevées du paramètre n'affectent pas la performance, ce qui ressemble aux résultats du paramètre W5.

Dans ce cas également, le fait que il n'y ait pas de documents avec des termes pouvant causer des ambiguïtés comme « script de théâtre » ou « le café Java » améliore la performance du paramètre. Avec un ensemble de documents variés, la performance du paramètre W5 serait affectée négativement pour de grandes valeurs et l'importance des paramètres W6 et W7 pour résoudre ces ambiguïtés sera plus importante que dans l'expérimentation du présent mémoire. Avec cette perspective, la valeur du paramètre qui sera conservée pour la suite de l'expérimentation sera de 15. Il s'agit de la valeur minimale où 130 documents pertinents sont retournés par la formule (1).

4.1.6 Paramètre W6 : Présence d'un autre concept de l'ontologie dans le document.

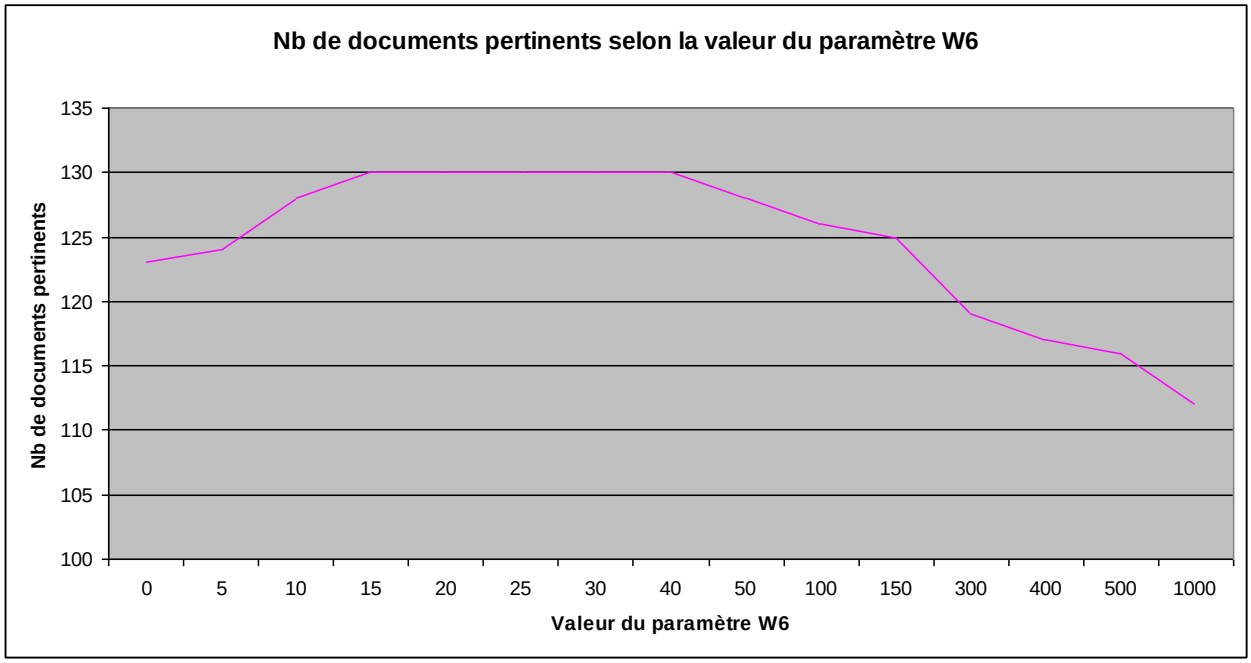

Figure 4-6 : Graphique des résultats sur la variation du paramètre W6

L'expérimentation sur le paramètre W6 montre que la présence d'un autre concept de l'ontologie confirme la pertinence du document pour les requêtes.

Le paramètre perd son efficacité dès que sa valeur excède 40. Comme le nombre de documents pertinents retournés est de 130 pour des valeurs entre 25 et 40, la valeur 32.5 est conservée pour le paramètre pour la suite de l'expérimentation.

4.1.7 Paramètre W7 : Fréquence d'un autre concept de l'ontologie dans le document et relié par une relation d'ordre 1 ou 2.

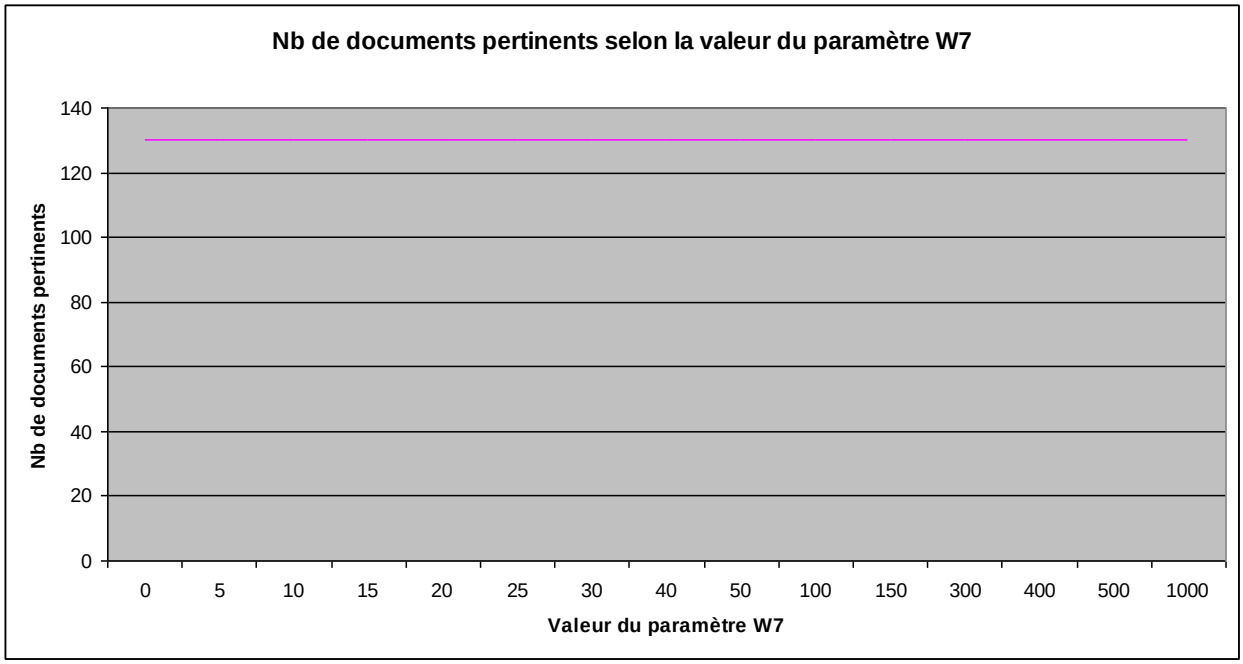

Figure 4-7 : Graphique des résultats sur la variation du paramètre W7

Les résultats de l'expérience sur le paramètre W7 indiquent que d'attribuer des points de pertinence aux documents qui contiennent des concepts reliés à celui de la requête est inutile.

Il faut noter que ces résultats sont influencés par le faible nombre de documents qui contiennent deux concepts ou plus reliés entre eux par une relation ontologique. De plus, le paramètre W4 attribue déjà des points de pertinence à la plupart des documents susceptibles d'être influencés par le paramètre W7.

Ce paramètre est donc retiré de la formule de calcul de la pertinence (1) mais une autre expérience avec plus de documents sera nécessaire pour confirmer ou infirmer son utilité dans le calcul de pertinence d'un document.

4.1.8 Paramètre W8 : Pénalité pour la fréquence d'un autre concept de l'ontologie dans le document et relié par une relation d'ordre 3

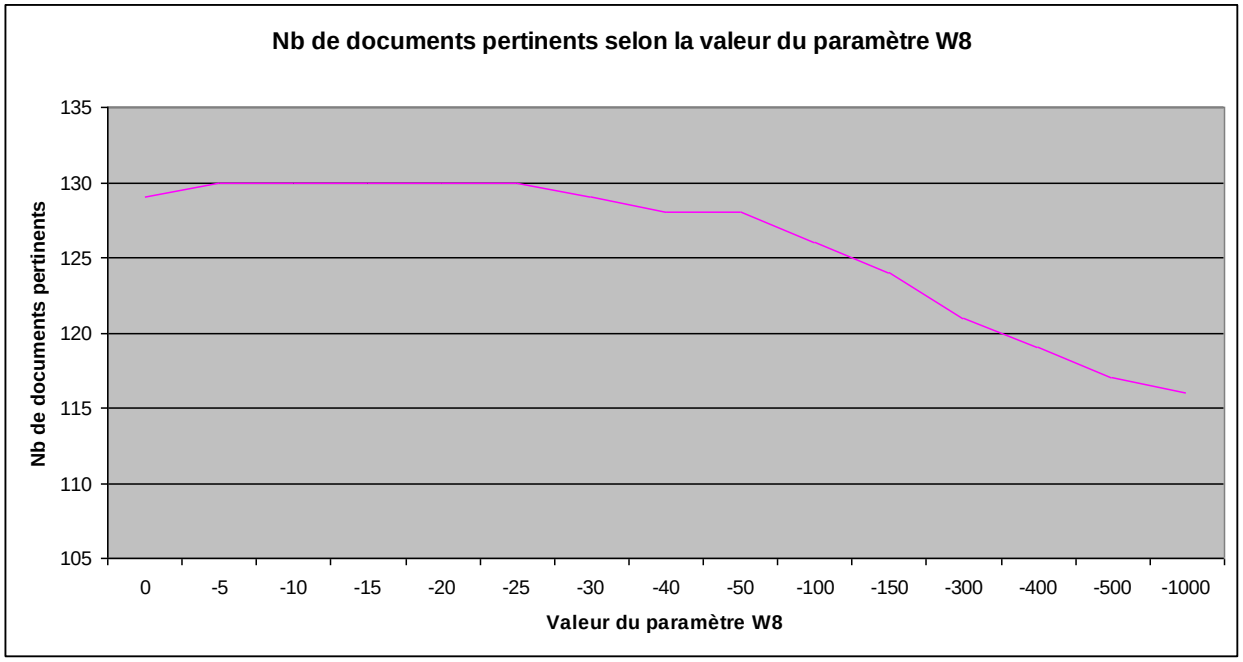

Figure 4-8 : Graphique des résultats sur la variation du paramètre W8

Les résultats de l'expérience sur le paramètre W8 indiquent que la pénalité de pertinence pour un document contenant un autre concept de l'ontologie mais non relié à la requête a un impact minimal sur la performance globale de la formule.

Le paramètre conserve sa performance maximale pour des valeurs entre -5 et -25, la valeur -15 est donc utilisée pour la suite de l'expérimentation.

Le peu d'impact du paramètre s'explique en partie à cause du corpus de documents utilisé pour l'expérimentation. Le premier concept trouvé dans un document augmente la pertinence (paramètre W6) et peu de documents contiennent beaucoup de concepts de l'ontologie ce qui explique les résultats sur le paramètre W8.

4.1.9 Paramètre W9 : Taille du document en nombre de caractères

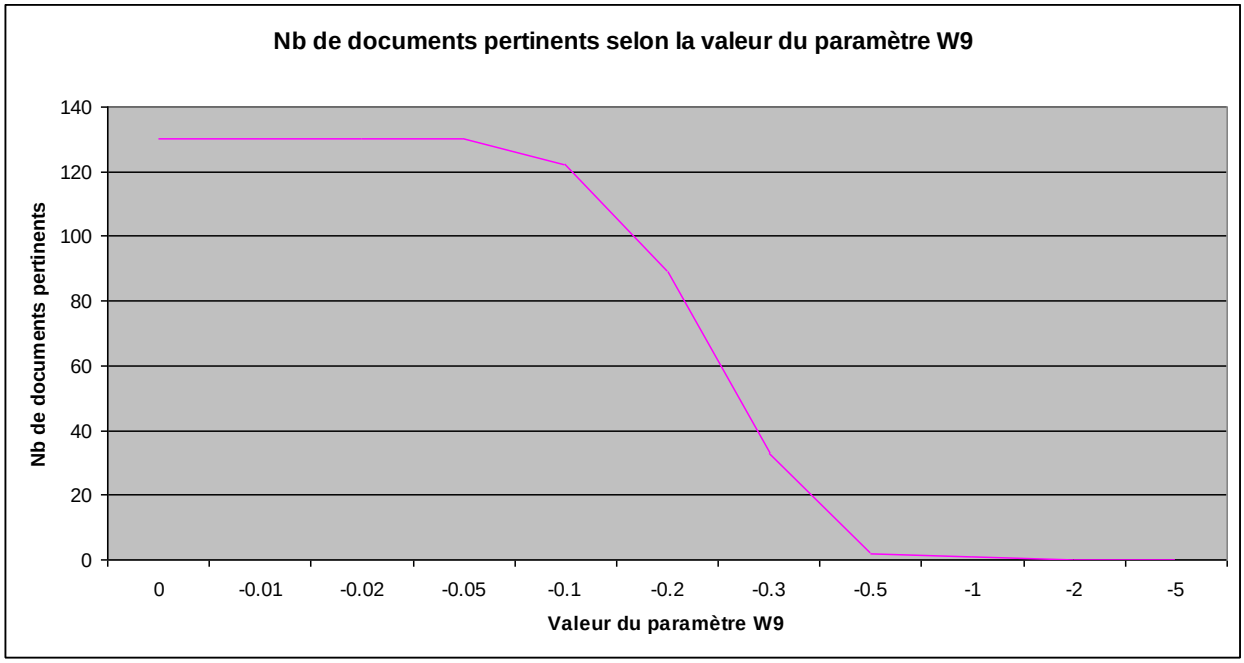

Figure 4-9 : Graphique des résultats sur la variation du paramètre W9

Finalement, les résultats de l'expérimentation sur le paramètre W9 montrent que la taille du document influence sa pertinence mais uniquement de façon négative. Il peut servir pour le bris d'égalité entre deux documents pertinents mais ce paramètre ne peut déterminer si un document est pertinent ou non à une requête.

Ce paramètre est donc retiré de la formule de calcul de la pertinence (1). Ce paramètre est plus utile dans une optique de classification (ordonnancer les documents pertinents en ordre de pertinence) que l'optique d'indexation du présent mémoire (déterminer si un document est pertinent ou non).

La formule finale du calcul de pertinence est donc :

$$P_{i,j} = (150\ F_{m,j}\ R_i + 25\ F_{Hi,j})\ (1\text{-}C) + 400\ F_{p,j} + (15\ F_{i,j}\ R_i)\ (C) + 32.5\ F_{o,j} - 15\ F_{c,j}$$

$$C = 1 \text{ si } R_i = 1, \text{ sinon } C = 0$$

Formule 2 : Formule finale du calcul de pertinence de la méthode ontologique

La formule optimisée (2) retourne 130 documents pertinents sur un total de 137 alors que la formule initiale (1) n'en retourne que 122.

Il est important de noter que l'exercice d'optimisation a été répété en optimisant les paramètres dans un ordre différent, soit en débutant par le paramètre W9 jusqu'au paramètre W1 et également dans un ordre aléatoire (qui fut W6, W1, W4, W8, W9, W5, W3, W7, W2) et ces deux exercices supplémentaires n'ont pas donné une formule finale aussi performante en termes du nombre de documents pertinents retournés.

CHAPITRE 5 RÉSULTATS ET DISCUSSION DE L'EXPÉRIMENTATION

Ce chapitre discute des résultats de l'expérience du précédent chapitre. Des tableaux expliquent en détail la performance comparative entre la méthode des espaces vectoriels et la méthode ontologique. Des explications sur ces résultats sont également décrites.

5.1 Résultats de l'expérimentation

Suite à l'expérience d'optimisation, les résultats de la recherche pour chacun des 16 concepts de la formule optimisée du calcul de la pertinence (2) sont comparés aux résultats de la recherche utilisant le modèle d'espace vectoriel avec la similarité cosinus combiné à la transformation TF-IDF. Le nombre de documents pertinents provient des évaluations des experts de la section 2.10.

Tableau 5.1 : Comparaison des résultats pour les concepts composés d'un seul terme

Concept	Documents pertinents dans l'ensemble D pour le concept	Recherche par ontologies	Recherche par modèle vectoriel avec cosinus et transformation TF-IDF
Script	10	10	9
XML	10	10	10
HTML	10	10	10
Java	10	10	10
Javascript	10	10	10
HTTP	10	4	4
SQL	10	10	10
Applet	3	3	3
Total de documents pertinents retournés	---	67	66
Nombre de documents pertinents dans l'ensemble D	73		
Total de documents retournés		67	67
Rappel		91.78%	90.41%
Précision		100%	98.5%

Tableau 5.2 : Comparaison des résultats pour les concepts composés de plusieurs termes

Concept	Documents pertinents dans l'ensemble D pour le concept	Recherche par ontologies	Recherche par modèle vectoriel avec cosinus et transformation TF-IDF
Langage de script	10	10	5
Langage de programmation Web	10	10	3
Service Web	2	2	0
Système de base de données	10	9	0
Outil de gestion de configuration	8	8	0
Langage de base de données	10	10	0
Protocole de communication Web	10	7	7
Serveur Web	7	7	4
Total de documents pertinents retournés	---	63	19
Nombre de	67		

documents pertinents dans l'ensemble D		
Total de documents retournés	80	80
Rappel	94.02%	27.94%
Précision	78.75%	23.75%

Tableau 5.3 : Comparaison des résultats globaux entre la méthode ontologique et le modèle d'espace vectoriel combiné à la transformation TF-IDF

	Recherche par ontologies	Recherche par modèle vectoriel avec cosinus et transformation TF-IDF
Total de documents pertinents retournés	130	85
Total de documents retournés	147	147
Total des documents pertinents dans l'ensemble D	140	
Rappel	92.85%	60.71%
Précision	88.43%	57.82%

5.2 Discussion des résultats

Suite à l'expérimentation, le constat est que la variation de chacun des paramètres menant à l'optimisation n'a pas le même impact dans le cas où l'utilisateur recherche des informations sur un concept à un seul mot, tel que « XML » ou « Javascript », que dans le cas des concepts composés de plusieurs mots comme « langage de script » ou « système de base de données ».

En effet, l'expérimentation révèle que la méthode ontologique produit le même ensemble de documents pertinents aux recherches qu'une méthode classique par mots-clés utilisant le calcul TF-IDF lorsque appliquée aux concepts à un seul mot. Une hypothèse est que, pour les concepts à un seul mot, leur présence dans un texte garantit pratiquement la pertinence du document dans le domaine des technologies Web.

Toutefois, pour les concepts composés de plusieurs mots, la méthode ontologique produit des résultats supérieurs au calcul TF-IDF. La méthode ontologique a retrouvé 63 des 67 documents pertinents à un concept de plusieurs mots alors que le modèle d'espace vectoriel combiné à la transformation TF-IDF s'est limité à correctement retourner 19 des 67 documents concernés. De plus, la méthode ontologique aurait obtenu un taux de rappel de près de 100% si l'ontologie avait défini « ARP » comme une instance du concept « protocole de communication web » car les documents pertinents manquants dans les résultats de la méthode ontologique contenaient le terme « ARP » . Cette seule instance manquante de l'ontologie a affecté le taux de rappel d'environ 5% ce qui indique que la puissance de la méthode ontologique est fortement influencée par la complétude de l'ontologie.

Le seul document pertinent, portant sur un concept composé de plusieurs mots, qui n'a pas été retourné par la méthode ontologique révèle une lacune de la méthode. Pour la requête « système de base de données », un document contenant six fois les termes « système de base de connaissances » obtient un score de pertinence très élevé. Ce score est causé par les paramètres W1 et W3.

La méthode ontologique, telle que définie dans le présent mémoire, n'attribue pas des poids différents aux différents mots qui constituent un concept. Alors, une concentration importante de mots « systèmes de base » a exactement le même impact qu'une concentration importante des mots « base de données ». Il est évident, pour un analyste humain, que la seconde concentration est beaucoup plus pertinente au concept initial que la première. Pour un ordinateur, cette distinction est beaucoup plus difficile à faire.

Il faut également noter la faible performance des deux algorithmes pour les documents pertinents au concept « http ». Ce cas précis est causé par le fait que les évaluateurs ont repéré plusieurs documents qui concernent les protocoles de communication mais qui ne contenaient pas le terme « http ». Pour la méthode d'espace vectoriel ce résultat est normal car le terme recherché n'est pas présent. Dans le cas de la méthode par ontologies, cela est causé parce que l'algorithme ne va impliquer l'inférence « http est un protocole de communication web » seulement si les termes représentant la classe mère sont tous présents dans le document. Comme ces documents ne contenaient pas le terme « Web », l'inférence n'a pas été faite alors qu'elle aurait permis de retrouver plusieurs documents pertinents supplémentaires. Faire une telle inférence est facile pour un évaluateur humain mais un ordinateur peut facilement faire de fausses inférences. Par exemple, pour le concept « langage de base de données » qui a une instance nommée « SQL », si le document contient les termes « langages de base », faire l'inférence que « SQL » est potentiellement un sujet du document est dangereux et mènera probablement à des faux positifs. Dans le cadre du présent mémoire, les inférences sur les instances ne sont faites que si tous les termes de la classe mère sont présents et cela a causé une perte de performance pour les résultats du terme « http ». Une expérience supplémentaire sur ce problème précis est nécessaire.

La méthode ontologique demande un travail important précédant l'indexation du corpus des documents, mais son pouvoir d'inférence ainsi que son analyse sémantique du texte donnent des résultats d'égale ou de meilleure qualité qu'une méthode d'indexation classique utilisant le calcul TF-IDF autant sur le taux de rappel que sur la précision sur un même corpus de documents.

CONCLUSION

Suite à l'expérimentation et aux résultats, l'approche par ontologies apporte un gain substantiel sur la précision des requêtes comparativement à une méthode classique par mots-clés. Les résultats sont prometteurs pour une application commerciale de la méthode et les travaux suivants seront nécessaires afin de raffiner la formule d'indexation :

1) Pour une application générale à grande échelle, demander à l'utilisateur de désambiguïser sa demande le cas échéant. Implantation d'un mécanisme de guidage.

Dans l'éventualité de la construction d'un moteur de recherche pour concurrencer la domination de Google, les lacunes de la classification par ontologies relevées par les auteurs Nagypal [17] ou Vallet, Fernandez et Castells [16] décrites au chapitre un du présent mémoire devront être résolues.

Les moteurs de recherche utilisant les ontologies ont besoin d'un contexte afin de bien comprendre la demande de l'utilisateur et celle-ci est difficile à comprendre avec quelques mots. Si un utilisateur effectue une recherche d'information avec uniquement le mot « chat » sur Google, l'algorithme PageRank va retourner les pages les plus populaires du moment et espérer que cela satisfait l'utilisateur. Si celui-ci désirait des informations sur les chats domestiques, il trouvera ces informations en troisième page. Il pourrait reformuler sa requête et rechercher avec les termes « chat domestique » ou « chat animal », ce qui produira les résultats espérés. Il faut noter que l'utilisation de la version française de Google (google.fr) procure un contexte linguistique à la requête et donne priorité aux documents francophones, ce qui donne un score de pertinence plus élevé aux documents de chats domestiques. Ces techniques de désambiguïsation ont toutes un point commun :

La désambiguïsation d'une requête est une tâche laissée à 100% à l'utilisateur avec les moteurs de recherches actuels.

Il serait fort intéressant qu'un moteur de recherche guide l'utilisateur lorsque sa demande est ambiguë. Ainsi, lorsque l'utilisateur entre le terme « chat » sur un moteur de recherche ontologique, ce terme peut représenter l'instance « chat » dans une ontologie d'animaux ou bien l'instance « chat » dans une ontologie de logiciels. Si le moteur de recherche ontologique est composé de ces deux ontologies, le moteur doit guider l'utilisateur à faire un choix parmi les ambiguïtés présentes dans les ontologies afin de préciser sa demande d'information. Dans le cas de « chat », le guidage proposera « chat, animal domestique » ou « chat, logiciel de communication ». Le choix de l'utilisateur permettra ensuite d'appliquer un contexte à la recherche et d'exploiter les forces de la classification ontologique pour une recherche classique avec peu de termes.

Une application courante du guidage lors de la recherche est implantée sur le site Web de Corbeil Électroménagers (http://www.corbeilelectro.com) et le résultat donne des arguments en faveur du guidage : 121% d'augmentation des ventes [18] en ligne lorsque le moteur guide les utilisateurs selon leur termes de recherche.

2) Application commerciale de la méthode à un site de recherche de spéciaux dans les supermarchés

Au cours de la rédaction du présent mémoire et à la lumière des résultats encourageants, l'application commerciale de la classification ontologique s'est révélée un projet envisageable à court terme.

Un projet de site Web qui regroupe les promotions des circulaires de douze chaînes de supermarché et de dépanneurs est en cours et la recherche des promotions se basera sur la

méthode ontologique. Une ontologie de produits alimentaires et ménagers est en construction et va tenter de couvrir la plus grande partie des produits offerts dans un supermarché ou un dépanneur. Si un utilisateur fait une recherche avec le terme « Fruits », le moteur de recherche pourra répondre à la demande et fournir la liste de tous les fruits en promotion même si le mot « fruit » n'est pas présent dans la description de la promotion de la circulaire, comme c'est souvent le cas. Par exemple, « Pommes Macintosh 99 cents /lb » sera retourné comme un document correspondant à une recherche de « fruits » si « pomme » est défini comme une sous-classe de « fruit » dans l'ontologie.

3) Construire une version de la méthode ontologique appliquée à la suggestion de documents pertinents à la page Web courante

Lorsque un utilisateur consulte une page Web, il existe une multitude d'algorithmes conçus pour attirer son attention pour qu'il poursuive sa session sur le site Web ou achète un produit. Par exemple, pratiquement tous les sites de nouvelles ont une section « à lire aussi… » qui regroupe des liens vers des articles reliés à la page courante. Pour l'achat de produits, l'application « AdWords » de Google est omniprésente sur le Web et offre des liens vers des produits reliés au contenu de la page courante. Comme démontré dans l'exemple des Tigers de Détroit et de Tiger Woods dans l'introduction, ces méthodes de suggestion ne sont pas infaillibles et l'utilisation de la classification par ontologie permettrait de mieux comprendre le contexte des documents et suggérer des documents pertinents plus efficacement qu'avec les méthodes classiques par mots-clés. Dans le cas du site de spéciaux de supermarchés proposé au point 2 du présent chapitre, une recherche portant sur les pommes Lobo devrait proposer également des spéciaux sur les pommes Macintosh ou Granny Smith.

4) Amélioration de l'algorithme de rapprochement entre les concepts de l'ontologie et les requêtes des utilisateurs.

Comme démontré par l'expérience de Vallet, Fernandez et Castells, dès que une requête de l'utilisateur ne correspond pas à des concepts de l'ontologie, le moteur de recherche perd son efficacité.

C'est pourquoi une version améliorée de l'algorithme de recherche utilisera des méthodes de rapprochement afin de tenter de convertir la requête afin de l'apparenter à un ou des concepts présents dans l'ontologie du domaine. Par exemple, si l'ontologie contient le concept « cycle de vie du produit logiciel » et que l'utilisateur effectue une recherche avec les termes « cycles de vie logiciel », il faut que l'algorithme détermine quel(s) concept(s) de l'ontologie se rapprochent le plus de la requête afin de produire un ensemble de résultats ayant le meilleur potentiel de correspondre à la recherche d'information. Cela aura pour effet que l'utilisateur n'aura pas à connaître les termes exacts de l'ontologie pour faire sa recherche d'information comme pour la plupart des moteurs de recherche actuels.

5) Simplification de la construction des ontologies : apprentissage automatisé et interface de construction des ontologies.

Lors de l'expérimentation, les 16 expressions utilisées pour comparer la méthode ontologique avec la méthode classique par mots-clés étaient des concepts présents dans l'ontologie. Cet environnement est contrôlé et ne correspond pas entièrement à un environnement de recherche normal ou les utilisateurs ne connaissent pas comment l'ontologie est construite ou comment elle fonctionne.

Comme démontré par l'expérience de Vallet, Fernandez et Castells [16], dès que la requête de l'utilisateur contient des termes qui ne sont pas présents dans l'ontologie, la précision de la méthode ontologique diminue au point d'être moins précise qu'une recherche classique par mots-clés. Vallet et Al. ont tenté d'améliorer leur algorithme en compensant les faiblesses de l'ontologie par une recherche par mots-clés lorsque la recherche ontologique tombait sous un seuil critique de précision (dans les cas où aucun concept de l'ontologie ne correspondait à la

requête). Cette approche est intéressante mais Vallet et Al. précisent que la définition du seuil qui détermine quelle méthode sera invoquée par la requête est un exercice périlleux qui peut affecter la performance de l'une ou de l'autre méthode.

L'utilisation de méthodes complémentaires pour compenser les faiblesses de la méthode ontologique est une piste de solution. Mais il est important, pour assurer l'avenir de la classification par ontologies, de raffiner les algorithmes afin que l'ontologie puisse se construire sans l'intervention directe d'un expert en construction d'ontologies. Une analyse sémantique automatisée des documents pourrait mettre à jour l'ontologie. Par exemple, un document qui contient « Le Canada est un pays d'Amérique du Nord » peut enrichir automatiquement l'ontologie d'une instance « Canada » de la classe « Pays » et d'une relation « EstDansLeContinent » entre l'instance « Canada » et l'instance de la classe « Continent » nommée « Amérique du Nord ». Une validation humaine serait probablement nécessaire afin de réduire le nombre de faux positifs d'un tel algorithme mais cette validation serait moins longue que l'ajout direct des instances et des relations via une interface de construction d'ontologies comme Protégé.

La participation d'experts à l'amélioration des ontologies, via une interface Web et supervisée dans le style de Wikipedia, permet à un plus grand nombre d'utilisateurs de contribuer à l'enrichissement des ontologies et ainsi compenser un peu l'une des plus grandes faiblesses relevées de la méthode de classification ontologique : les concepts manquants lors de la création de l'ontologie.

Également, une version plus élaborée de l'algorithme de recherche permettra d'enrichir l'ontologie automatiquement sans aucune intervention humaine. Par exemple, un site commercial vendant des appareils électroménagers avec une ontologie de base qui contient le concept « Réfrigérateur ».

On peut s'attendre à ce que plusieurs utilisateurs désirant acheter un réfrigérateur n'écrivent pas leurs termes de recherche exactement comme le concept est défini dans l'ontologie. Une façon courante des gens d'écrire ce concept est « frigo » ou « frigidaire ». Au départ, le moteur de recherche ontologique n'a aucune idée de ce que l'utilisateur recherche car, à moins d'avoir été spécifiquement déterminé par l'expert qui a crée l'ontologie, le concept n'est pas présent dans l'ontologie. Toutefois, en analysant le parcours de l'utilisateur sur le site, on peut tenter de comprendre ce qu'il recherche et ensuite enrichir l'ontologie avec ces nouvelles données lorsque le scénario se répète. Si l'utilisateur recherche avec le terme « frigo » et consulte les pages Web des réfrigérateurs du magasin, il est fort probable que le sujet de sa recherche concernait les réfrigérateurs. Si plusieurs personnes font la même expérience, l'ontologie sera enrichie afin de définir « frigo » comme étant un synonyme de « réfrigérateur » et ainsi mieux orienter la recherche des utilisateurs subséquents. L'achat d'un produit aura plus de poids que la simple consultation lorsque il s'agira de déterminer le seuil qui déterminera lorsque l'ontologie sera enrichie avec des nouveaux termes non prévus par le créateur de l'ontologie.

BIBLIOGRAPHIE

[1] Alpert J., Hajaj N. (28 juillet 2008), “We knew the Web was big “ [Billet de blogue], tiré de http://googleblog.blogspot.com/2008/07/we-knew-the-web-was-big.html.

[2] Aroyo. L. et al. (2010), 7th extended semantic web conference, Proceedings part 1 (p. 31-45)

[3] Devedzic. V. (2006), chapitre 2, “Semantic Web and education” (p. 27-69),Belgrade, Serbie et Monténégro, Springer.

[4] Heflin J.”Simple HTML Ontology Extensions”, tiré de la page Web http://www.cs.umd.edu/projects/plus/shoe/search, consulté le 5 décembre 2010.

[5] Auteur inconnu (nom d’utilisateur shuwu83), « IESTwitter Project », basé à l’Université du Washington, tiré de la page Web http://code.google.com/p/iestwitter/ consulté le 5 décembre 2010.

[6] Russell I., Markov Z., Neller T. (2005), « Web Document Classification ».

[7] Kohler J., Phillippi S., Specht M, Ruegg A. (2006), “Ontology based text indexing and querying for the semantic web”, Université de Koblenz, Allemagne.

[8] Van de Maele F. (2006), « Ontology based crawler for the semantic web », mémoire de maîtrise, Université de Bruxelles, Belgique.

[9] Burden J.P.H., Jackson M.S. (2006), « New direction in search engine technology », Université de Wolferhampton.

[10] Hernandez N., Hubert G., Mothe J., Ralalason B. (2008), « RI et ontologies, état de l’art 2008 », Université Paul Sabathier, France.

[11] TF-IDF. (s.d.). Dans Wikipedia, consulté le 5 décembre 2010, tiré de http://en.wikipedia.org/wiki/Tf%E2%80%93idf.

[12] Hyvönen E., Saarela S., Viljanen K. (2004), « Application of ontology techniques to view-based semantic search and browsing », Computer Science Volume 3053 (p. 92-106).

[13] Semantic Web. (s.d.). Dans Wikipedia, consulté le 20 décembre 2010, tiré de http://en.wikipedia.org/wiki/Semantic_web.

[14] PageRank. (s.d.). Dans Wikipedia, consulté le 20 décembre 2010, tiré de http://en.wikipedia.org/wiki/PageRank.

[15] Paralic J. Kostial I. (2003), « Ontology-based Information Retrieval », Information and Intelligent Systems, (p. 22-28).

[16] Vallet D., Fernández M, Castells P. (2005), « An Ontology-based Information Retrieval Model », ESWC (p. 455-470).

[17] Nagypàl G. (2005), "Improving information retrieval effectiveness by using domain knowledge stored in ontologies", OTM workshops, LNCS 3762 (p.780-789).

[18] Guidyu, March 8th 2011 news, tiré de http://www.guidyu.com/en/main-nav/media/march-8th-2011/, consulté le 20 mars 2011.

[19] Rajaraman A. (11 juin 2008), « How Google measures search quality » [Billet de blogue]. Tiré de http://anand.typepad.com/datawocky/2008/06/how-google-measures-search-quality.html, consulté le 18 avril 2011.

[20] Precision and recall. (s.d.). Dans Wikipédia, consulté le 18 avril 2011, tiré de http://en.wikipedia.org/wiki/Precision_and_recall.

[21] Maedche A., Staab S., Stojanovic N., Studer R., Sure Y. (2001), "SEmantic PortAL – The SEAL approach", Spinning the semantic Web (p. 317-359).

[22] Guha R., McCool R. (2003), Miller E., "Semantic Search", International World Wide Web Conference, (p. 700-709).

[23] Feinerer I., (28 janvier 2013), « Introduction to the TM package, text mining in R », tiré de http://cran.r-project.org/web/packages/tm/vignettes/tm.pdf, consulté le 18 avril 2013.

[24] Applied Software Design, Cosine Similarity Calculator, tiré de http://www.appliedsoftwaredesign.com/archives/cosine-similarity-calculator, consulté le 18 avril 2013.

[25] Dejarnette,R., (24 janvier 2012) « Click-through rate of top 10 search results » [Billet de blogue],tiré de https://www.internetmarketingninjas.com/blog/search-engine-optimization/click-through-rate/, consulté le 24 mars 2013.

[26] Thomo A. (2009), Latent Semantic Analysis (Tutorial), tiré de http://www.engr.uvic.ca/~seng474/svd.pdf, (p.4-7).

[27] Singular Value Decomposition. (s.d.). Dans Wikipédia, consulté le 24 mars 2013, tiré de http://en.wikipedia.org/wiki/Singular_value_decomposition.