Construction de Classes de Documents Web

Benjamin Nguyen

INRIA FUTURS
Domaine de Voluceau
78153 Le Chesnay CEDEX
FRANCE
Email : Benjamin.Nguyen@inria.fr
Tél : 33 1 39 63 51 88 Fax :

Iraklis Varlamis, Maria Halkidi, Michalis Vazirgianis

Department of Informatics, Athens University of Economics and Buisiness
76, Patision Street, Athens 10434
GREECE
Email : {varlamis,mhalk,mvazirg}@aueb.gr
Tél : (301) 8203 519 Fax :

Résumé

Dans cet article, nous proposons une nouvelle mesure de similarité entre des documents caractérisés par un ensemble succint de concepts d'une hiérarchie (ontologie), ce qui est le cas de documents web. Cette mesure se marie avec un algorithme de regroupement par densité (DB-SCAN), issu du domaine des bases de données spatiales, que nous avons adapté à nos besoins, afin de permettre la construction de classes de documents. Nous donnons également des résultats expérimentaux qui montrent la pertinence de notre approche.

Abstract

In this paper, we introduce a novel similarity measure between documents, such as web documents, that can be characterized by a small set of concepts, that each belong to a hierarchy (ontology). We use this measure with a slightly modified DB-SCAN clustering algorithm in order to construct clusters of semantically related documents. We also give experimental results that illustrate the quality of our approach.

1 Introduction

En recherche d'information, on caractérise traditionnellement un document par son contenu. [PW00] montrent que dans le cas de pages web il est en fait possible de construire des ensembles succints de l'ordre de 5 à 10 mots, qui caractérisent de manière pertinente chaque page. Nous avons dans nos travaux précédents étendu cette approche pour caractériser chaque document web par un ensemble de concepts d'une ontologie. Pour nous, une ontologie est simplement un arbre (IS-A), mais les résultats de cet article s'étendent tel quels sur un arbre avec d'autres relations entre les termes, et peuvent se généraliser avec des modifications minimes dans le cas où l'ontologie qui serait un treillis ou un graphe acyclique orienté. Nous supposons **donnés** les ensembles de mots-clés et concepts de l'ontologie qui caractérisent chaque document, et ne discuterons pas ici de la manière dont ces ensembles sont générés. Pour plus d'informations sur la manière de construire ces ensembles, le lecteur pourra se reporter à [VNVA02, HNVV02], qui détaillent également l'architecture globale de notre projet, appellé THESUS. L'objectif de cet article est de tenter de trouver une méthode efficace pour résoudre le problème suivant :

> Étant donné un ensemble de documents, dont chacun est caractérisé par un (petit) ensemble de concepts d'une ontologie, trouver une méthode pour regrouper en classes les documents qui ont une sémantique proche.

La réponse classique à un problème de ce type serait d'utiliser des techniques d'IR (Information Retrieval), comme celles décrites dans [SM83]. Toutefois, ces techniques présentent l'inconvénient de se baser sur un appariement **exact** des mot clés. En effet, ces méthodes ne permettent pas de prendre en compte le fait que certains mots peuvent avoir une *proximité sémantique* entre eux. Par exemple, imaginons un document qui serait caractérisé par les mots "serpent" et "désert" et un autre document caractérisé par les mots "aspic" et "Sahara". En utilisant les techniques traditionnelles, ces documents ne seraient en aucun cas considérés comme étant proches. Toutefois, en utilisant une ontologie pour mieux *comprendre* ce que signifie chaque terme, nous pouvons proposer une mesure de similarité bien plus pertinente, et par conséquent l'appliquer à un algorithme de regroupement par densité de manière efficace.

Plan de l'article : Dans la section suivante, nous présentons un court état de l'art, centré sur les mesures de similarité. Nous présentons notre mesure dans la section 3, et l'algorithme de regroupement dans la section 4, puis dans la section 5 nous donnons des résultats expérimentaux, qui témoignent de la pertinence de notre approche.

2 État de l'art

Notre but ici n'est pas de dresser un état de l'art complet sur le domaine de la classification de documents. De nombreux travaux ont déjà été réalisés, nous citerons notamment [Fis87, AGY99]. Le problème que nous considérons est un peu plus spécifique, puisque nous nous intéressons à des documents Web [ZE98], et nous prenons en compte les liens vers ces documents, comme le détaille [Kle99]. En effet, les liens simplifient entre autres la manière de caractériser une page par un nombre restreint de mots-clés, et ce sont notamment sur ces résultats que nous nous basons pour affirmer qu'il est possible de construire un ensemble concis de termes qui vont correctement caractériser un document web. Nous détaillons dans nos travaux précédents [HNVV02] comment en utilisant WordNet [Wor] nous sommes en mesure de construire à partir d'un ensemble de

mots clés qui caractérisent une page, cet ensemble de concepts (5 à 10 environ) d'une ontologie d'un domaine particulier. Les détails complets sur notre système sont donnés dans [HNVV02]. D'autres systèmes de meta-moteurs de recherche s'intéressent aux liens : Kartoo, [kar] représente graphiquement les liens entre sites, Vivissimo, [Viv] construit des classes de documents en temps réel à partir du résultat de requêtes sur des moteurs de recherche classiques.

Les travaux les plus proches des notres sont ceux de [DJ01] qui ont construit une distance sur une ontologie afin de pouvoir indexer des sites web, mais nous n'avons trouvé aucun travail qui porte précisément sur le thème des mesures de similarité entre ensembles d'éléments d'une hiérarchie. Nous détaillons dans les paragraphes suivants les travaux sur les mesures de similarité qui présentent néanmoins une certaine pertinence.

2.1 Notions générales sur les mesures de similarité

Notre but est de construire une mesure de similarité entre documents, ce qui se réduit dans notre cas à trouver une mesure de similarité entre ensembles d'éléments appartenant à une hiérarchie. Nous partageons les mêmes intuitions que [Lin98] en ce qui concerne les propriétés que devrait posséder une telle mesure.

Pour deux ensembles d'éléments A et B :

- La similarité entre A et B est fonction de ce qu'ils ont en commun. Plus ils ont de choses en commun, plus leur similarité sera élevée.
- La similarité entre A et B est fonction de leurs différences. Plus ils ont de différences, plus leur similarité sera faible.
- La valeur de similarité maximale est obtenue lorsque A et B sont identiques, quelque soit le nombre d'éléments qu'ils ont en commun.

2.2 Mesures de similarité entre ensembles

Il existe un certain nombre de mesures de similarité entre ensembles. La plus utilisée, le coefficient de Jaccard est très simple. Soient A et B deux ensembles finis d'éléments. La similarité entre A et B se définit comme : $S_J(A,B) = \frac{|A \cap B|}{|A \cup B|}$. Cette mesure respecte bien les intuitions du paragraphe précédent, mais ne prend en compte que l'appariement exact de termes ; or nous voulons aller plus loin en introduisant une mesure qui prenne en compte la proximité des mots, plutôt qu'une comparaison binaire. [EM97] font une revue de diverses mesures de similarité entre deux ensembles finis de points dans un espace métrique et montrent que toutes ces mesures ont une complexité polynômiale en fonction des éléments de l'ensemble. [GHOS96] s'intéressent également à une dizaine de mesures de similarités entre ensembles, et expliquent leur sémantique, mais toutes sont basées sur le coefficient de Jaccard. Les problèmes d'indexation entre ensembles est également traité dans les travaux de [GGK01], où les auteurs montrent comment grouper des ensembles d'éléments, mais toujours en utilisant le coefficient de Jaccard. Pour finir, notons que la mesure *cosinus* traditionnelle[SM83] de l'IR est aussi une application directe du coefficient de Jaccard. Dans tous ces travaux, l'ensemble de valeurs sur lesquelles la mesure de similarité est calculée est un ensemble **plat**, c'est-à-dire que tous les éléments de l'ensemble sont indépendants les uns des autres, et ceci nous a encouragés à nous efforcer de prendre en compte les relations sémantiques qui pouvaient exister entre les concepts d'une ontologie, tout comme [BFS02] qui s'intéressent dans leur approche médiateur ù la similarité entre concepts appartenant à la même hiérarchie.

2.3 Similarité entre deux éléments d'une ontologie

[RSM, Res95, Res99] proposent diverses méthodes pour calculer la similarité entre deux concepts d'une ontologie, par exemple WordNet, et [Lin98] a effectué une comparaison entre ces méthodes. Il en ressort que la mesure de Wu et Palmer [WP94] est la plus rapide à calculer, tout en restant aussi expressive que les autres, d'où notre choix de cette mesure comme fondement de nos travaux. Sa définition est la suivante :

Étant donné un arbre, et deux noeuds a et b de cet arbre, soit c l'ancêtre commun le plus profond (sachant que la racine est de profondeur 1). La mesure de similarité entre a et b s'exprime alors : $S_{WP}(a,b) = \frac{2 \times Profondeur(c)}{Profondeur(a)+Profondeur(b)}$

Il est immédiat de vérifier que cette mesure respecte les critères de 2.1. Il est possible de définir la distance canonique associée à cette mesure de similarité de la manière suivante :$D_{WP}(a,b) = 1 - S_{WP}(a,b)$. Il est possible de vérifier que D_{WP} est bien une distance, mais nous ne le détaillons pas ici.

3 Une nouvelle mesure de similarité entre ensembles de concepts

Nous avons vu dans la section précédente que les mesures de similarité existantes ne prennent en compte qu'un appariement exact de termes. Dans cette section, nous montrons comment étendre les idées de Wu et Palmer pour proposer une mesure de similarité sur des ensembles de termes d'une ontologie. D'après les études de [EM97, Nii87], aucune mesure n'a été proposée sur le calcul de la similarité entre des ensembles d'éléments d'un espace métrique. Notre cas est même un peu plus général, puisqu'il ne s'agit pas d'un espace métrique, mais simplement d'un espace sur lequel il existe une mesure de similarité.

L'idée fondamentale réside dans l'utilisation d'une mesure de similarité entre les éléments individuels de chaque ensemble, pour estimer de manière adéquate la similarité entre les ensembles eux mêmes. Il est important de souligner que si nous utilisons la mesure de Wu et Palmer comme mesure de base, il est possible d'en utiliser une autre, à condition que la mesure ainsi définie soit une généralisation de la mesure de base.

3.1 Wu et Palmer généralisé aux ensembles

Dans les paragraphes qui suivent, nous allons définir formellement la généralisation de la mesure de Wu et Palmer que nous proposons. Nous commençons par donner les intuitions de cette mesure, qui doit respecter les idées générales sur la similarité présentées en 2.1.

Intuition : Chaque document est représenté par un ensemble de concepts de l'ontologie. Si les ensembles contiennent beaucoup de termes semblables, voir identiques, alors ces ensembles seront très similaires. Au contraire, si ils possèdent beaucoup de termes très différents (très distants dans l'ontologie), leur similarité devra être faible. Dans un premier temps, on cherche pour chaque concept du premier ensemble A, de quel concept du second ensemble B il est le plus proche. Ainsi, même si l'ensemble A est composé de concepts très différents, tant que pour chaque concept de A on trouve un concept de B qui est proche, la similarité va être grande. Dans un deuxième temps, on effectue la même opération pour B, ce qui permet de voir s'il n'y a pas dans B des concepts qui sont très différents de ceux de A, et qui donc feront chuter la similarité, ou bien si tous les concepts de B trouvent un bon appariement avec ceux de A.

Formalisme : Soit Ω une ontologie (dans les exemples qui suivent, nous utilisons WordNet). Ω est un ensemble fini, dont chacun des éléments est un concept. On note en minuscules (a,b) les éléments de Ω (concepts). On note en majuscules (A,B) les sous ensembles de Ω. On note $|A|$ le cardinal de l'ensemble A. Soit $S_{WP}(a,b)$ la mesure de similarité de Wu et Palmer entre deux concepts a et b de Ω. On définit :

$$\zeta(A,B) = \frac{1}{2}(\frac{1}{|A|}\sum_{a\in A}\max_{b\in B}(S_{WP}(a,b)) + \frac{1}{|B|}\sum_{b\in B}\max_{a\in A}(S_{WP}(a,b))) \quad (1)$$

Vérifions les propriétés de $\zeta(A,B)$.

Propriétés : Il est immédiat de vérifier que $\zeta(A,B)$ est une mesure de similarité : i. $\zeta(A,B) = 1$ ssi $A = B$, ii. par construction, $\zeta(A,B) = \zeta(B,A)$. De plus, cette mesure de similarité étend bien la mesure de Wu et Palmer. Si les ensembles sont réduits à un élément par exemple $|A| = \{a\}$ et $|B| = \{b\}$, alors $|A| = |B| = 1$ et $\zeta(A,B) = S_{WP}(a,b)$. Si on considère que la complexité pour calculer $S_{WP}(a,b) = O(1)$, la complexité de cette mesure est égale à $O(|A| \times |B|)$. En réalité, le coût pour calculer S_{WP} dépend de l'ontologie, et dans le pire des cas est ègale à $O(h)$ où h représente la profondeur maximale de l'ontologie. Heureusement, en règle générale, les ontologies sont assez peu profondes, mais très larges. Toutefois, dans tous les cas la complexité S_{WP} est indépendante du nombre d'éléments dans A ou B, et le temps de son calcul peut être sorti comme une constante. Pour des ontologies de taille raisonnable (< 5000 concepts) on peut tout simplement pré-calculer toutes les similarités entre chaque paire de concepts. Notre mesure est donc tout à fait performante, vis-à-vis des mesures proposées dans [EM97] où les meilleures complexités théoriques sont polynômiales dans les nombres d'éléments des deux ensembles.

Exemple : Calculons la similarité entre les ensembles suivants : $A = \{chat,\ CD\}$ et $B = \{felin,\ disque,\ moto\}$. Les similarités de Wu et Palmer sont les suivantes : $S_{WP}(chat,\ felin) = 0.95; S_{WP}(chat, moto) = 0.15; S_{WP}(chat,\ disque) = 0.19; S_{WP}(CD,\ felin) = 0.13; S_{WP}(CD, disque) = 0.83; S_{WP}(CD,\ moto) = 0.28$.Les éléments les plus proches de ceux de A sont $felin$ pour *chat* et *disque* pour CD. Réciproquement, les éléments les plus proches de B sont *chat* pour $felin$, CD pour *disque* et CD pour *moto*. Notons ici qu'un document qui parlerait juste de $felin$ et *disque* serait plus proche du document A, puisqu'ici le fait de parler de *moto* est bien entendu considéré comme moins pertinent.

On a donc :

$$\zeta(A,B) = \frac{1}{2} \times (\frac{1}{|A|} \times (S_{WP}(chat,\ felin) + S_{WP}(CD,\ disque)) +$$
$$\frac{1}{|B|} \times (S_{WP}(felin,\ chat) + S_{WP}(disque,\ CD) + S_{WP}(moto,\ CD)))$$

L'application numérique donne dans ce cas $\zeta(A,B) = 0.78$. Nous donnons une petite table avec quelques exemples. Il est à noter que bien entendu ces valeurs sont modulées selon l'ontologie que l'on utilise. Dans la table 1, nous rappelons que nous avons utilisé WordNet pour calculer les similarités, et non une ontologie d'un domaine bien particulier.

4 Application à un Algorithme de Regroupement par Densité

Dans cette section, nous expliquons comment regrouper dans des classes des documents caractérisés chacun par un ensemble de concepts issus d'une même ontologie, en fonction de

A	B	$\zeta(A,B)$
Chat, CD	Chat, CD	1.0
Chat, Tigre, Lynx	Félin	0.95
Chat, CD	Tigre, Disque, Félin	0.89
Chat, CD	Moto, Disque, Félin	0.78
Microprocesseur	Technologie, Electronique	0.70
Cleopatre	Reine, Egypte	0.60
Espion, Microfilm	Révolution Française, Guillotine	0.11

TAB. 1 – *Similarité entre Ensembles*

leur proximité. Nous montrons que le problème peut se rapprocher, grâce à la définition d'une mesure de similarité, du cas classique de regroupement dans un espace métrique, traité par l'algorithme d'Ester, Kriegel *et al.* DB-SCAN [EKSX96]. Nous donnons une explication intuitive de l'algorithme, mais mais nous n'avons malheureusement pas la place d'expliciter ici l'algorithme complet.

4.1 Explication Intuitive de l'algorithme

Pour faire fonctionner cet algorithme, il est nécessaire de pouvoir évaluer la similarité entre deux documents. Nous avons montré dans la section précédente comment construire une mesure de similarité entre deux documents caractérisés par des ensembles de termes d'une ontologie. On définit deux seuils, $MinSim \in [0;1]$ et $MinDocs > 1$, l'un mesurant la similarité minimale entre documents pour qu'ils soient considérés comme voisins, et l'autre mesurant le nombre minimal de documents par classe. L'algorithme commence par prendre un document au hasard et à regarder combien de documents sont 'proches' de lui, c'est-à-dire combien de documents ont une similarité supérieure à $MinSim$ avec lui. Si plus de $MinDocs$ documents sont proches de lui, alors l'algorithme groupe tous ces documents ensemble, marque le premier document, puis passe à un autre document de ce groupe et effectue la même recherche, en aggrégeant les résultats à la classe qui est en train d'être construite. Lorsqu'il est impossible de progresser (lorsqu'il n'y a plus dans la classe traitée de documents similaires en nombre supérieur à $MinDocs$) la classe est considérée comme achevée, et l'algorithme continue en prenant un autre document (au hasard) qui ne fait pas partie d'une classe et qui n'a pas déjà été traité. L'algorithme se termine lorsque tous les documents ont été traités. Le nom 'regroupement par densité' vient du fait que les classes représentent des zones où la densité de documents est suffisante.

4.2 Modifications de l'algorithme

Un contexte très différent : À l'origine, l'algorithme a été utilisé dans le cadre des bases de données spatiales, où les éléments à regrouper sont des points d'un espace métrique. La mesure utilisée pour évaluer la distance est la mesure Euclidienne. Pour calculer les points qui sont dans le voisinage les uns des autres, les points sont tous stockés dans un R*-Tree [BKSS90]. Il est intéressant de remarquer que le temps de calcul du R*-Tree n'est pas pris en compte dans la complexité de DB-SCAN par ses auteurs. Or, dans notre cas, nous ne sommes pas dans un espace métrique dont on connaîtrait la dimension. Nous ne pouvons donc pas utiliser de R*-Tree. A la place nous pré-calculons la similarité entre les n documents à classer et sauvegardons ces

valeurs dans n listes de longueur n que nous trions en utilisant QuickSort. La complexité pour trier une liste de longueur n étant $O(n \log n)$, la complexité pour trier n listes est $O(n^2 \log n)$. Une fois cette phase de préprocessing achevée, on peut faire tourner l'algorithme en remplaçant le R*-Tree par notre liste triée.

Complexité : La complexité est fonction de la complexité moyenne pour définir des documents voisins d'un autre. Dans notre système, vu que la similarité d'un document avec tous les autres est pré-calculée et présentée dans une liste ordonnée, le temps moyen de parcours pour trouver tous les documents qui ont une similarité $\zeta > MinSim$ est $O(\log n)$ en utilisant une méthode dichotomique, si les similarités entre documents sont stockées en mémoire vive,c'est-à-dire la même complexité que pour un R*-Tree. L'algorithme répète ensuite ceci pour tous les documents, ce qui donne une complexité finale en $O(n \log n)$ pour la seule phase de clustering.

Etiquettage : Bien que grouper des documents entre eux soit une tâche qui contribue à simplifier la recherche en leur sein, il est encore plus intéressant de mettre une étiquette sur chaque classe. Une méthode simple et qui donne néanmoins de bons résultats est la suivante :

- Pour chaque classe, nous construisons U l'union de tous les concepts qui appartiennent au moins à un document de cette classe.
- Pour chaque concept $k_i \in U$ nous calculons la proportion de documents de la classe considérée auxquels il appartient.
- Nous considérons comme pertinent les concepts qui apparaissent dans une proportion supérieure à un certain seuil. Dans nos expériences, nous avons fixé arbitrairement ce seuil à 51%, il conviendrait de faire plus d'expériences pour mesurer l'impact de cette valeur.

5 Expérimentation

Dans cette section, nous présentons les résultats fournis par le système, en utilisant une méthode de test 'aveugle'. Le paradigme ici est que si les résultats de la classification sont bons, alors cela signifie que la mesure de similarité est correcte.

5.1 Similarité

Protocole : Le protocole expérimental est le suivant : parmi tous nos documents (environ 40 000) nous avons sélectionné 20 paires de documents qui pouvaient ou non faire partie de la même classe, et nous avons présenté cette liste à 10 testeurs. Chaque testeur devait dire s'il grouperait ensemble les deux documents ou non, en leur attribuant une valeur allant de 1(pas de rapport) à 5 (quasi identiques). Nous avons calculé pour chaque paire la valeur moyenne de similarité 'humaine' en faisant la moyenne des scores donnés par les différents testeurs. Par exemple, si tous les testeurs donnent un valeur de 5 à la paire, alors la similarité 'humaine' est de 1. Un testeur qui donnerait 4 et tous les autres qui donneraient 5 ferait une similarité 'humaine' de 0.98. Cette valeur est attribuée à chaque document. Nous fixons ensuite le seuil $MinSim$. Si la similarité humaine pour une paire de documents est supérieure à ce seuil, la paire est admise comme 'pertinente'. Si la valeur est inférieure à ce seuil, cela signifie que les êtres humains ne jugent pas ces documents sémantiquement proches. Nous avons ensuite comparé les résultats avec notre système en jouant sur la valeur $MinSim$ de l'algorithme de clustering. Pour notre système, THESUS, les paires de documents pertinentes sont celles où les documents ont été affectés à la même classe, avec le même paramètre $MinSim$. Les résultats montrent le taux de

MinSim = 0.80
Taux de rappel Humain / THESUS : 77%
Taux de précision Humain / THESUS : 80%
Corrélation Humain / Humain : 81%

TAB. 2 – *Résultats pour THESUS (MinSim = 0.80)*

MinSim = 0.90
Taux de rappel Humain / THESUS : 83%
Taux de précision Humain / THESUS : 80%
Corrélation Humain / Humain : 81%

TAB. 3 – *Résultats pour THESUS (MinSim = 0.90)*

rappel et de précision[1] entre les résultats humains, et ceux du système. La corrélation entre les humains eux-mêmes est obtenue en calculant la dispersion des résultats entre testeurs, et peut être comparée aux deux taux précédents.

Résultats : Les résultats dans les tables 2 et 3 montrent que le système est tout aussi fiable qu'un testeur. Ceci montre que la mesure de similarité et l'algorithme de clustering donnent des résultats qui ont du sens. Nous avons effectué la même manipulation en utilisant le système Vivissimo, et donnons les résultats dans la table 4. Pour que la comparaison soit possible, nous avons considéré que Vivissimo donnait comme classés ensemble des documents qu'il plaçait dans un même chemin dans son arbre de classification. Nous voyons que sur ce jeu de documents différents, la corrélation entre testeurs humains est également aux alentours de 80%. Les résultats donnés par Vivissimo sont en revanche bien moins bons que ceux de notre système, il suffit de comparer les taux de rappel et de précision de la Table 4.

Et Jaccard ? Nous ne donnons pas ici de table comparative avec les coefficient de Jaccard car parmi les paires de documents choisies, pratiquement aucune n'avait de terme en commun. Ainsi, la similarité pour chacune des paires de documents en utilisant Jaccard aurait été nulle. C'est en partie cette constatation qui a encouragé nos travaux.

5.2 Etiquetage

Protocole : Nous présentons à chaque testeur les étiquettes (un ensemble de concepts) des classes que notre système a découvertes. Pour un ensemble de documents (environ 50 par testeur) nous lui demandons de mettre le document dans une des catégories générées automatiquement par le système, ou bien de le mettre dans une catégorie *bruit* si il estime qu'aucune étiquette ne correspond. En quelque sorte nous posons au testeur la question, "si vous aviez à mettre une étiquette sur ce document, laquelle choisiriez vous, sachant que vous pouvez choisir de ne pas en mettre du tout si aucune ne vous satisfait". Nous indiquons dans la table 5 le pourcentage de documents qu'un testeur humain 'moyen' (c'est-à-dire en réalité une moyenne sur les testeurs) a mis dans la même catégorie que le système, pour diverses valeurs du paramètre $MinSim$ de l'algorithme. Une corrélation de 80% signifie que sur les 50 documents, 40 auront été classés dans

1. Le *recall* et *precision* de la littérature anglophone sur Information Retrieval. On considère ici que les résultats à ramener sont les n documents au dessus du seuil humain, et que le système retourne m documents, dont p sont corrects. Le *recall* est donné par $\frac{p}{n}$ et la *précision* par $\frac{p}{m}$

MinSim = 0.80
Taux de rappel Humain / Vivissimo : 33%
Taux de précision Humain / Vivissimo : 40%
Corrélation Humain / Humain : 79%

TAB. 4 – *Résultats pour Vivissimo ($MinSim = 0.80$)*

MinSim	0.6	0.7	0.8	0.9
Corrélation humain/THESUS	75%	82%	80%	68%

TAB. 5 – *Etiquetage*

la même classe par l'humain 'moyen' et le système. Ainsi cette méthode nous permet de vérifier la capacité de notre système à générer une étiquette pertinente. En revanche, nous n'avons pas fait d'expérience sur la façon dont les testeurs humains auraient pu regrouper ensemble ces mêmes documents.

Résultats : Les résultats sont donnés dans la Table 5. Soulignons que seuls environ 5% des documents n'ont pas été mis dans une catégorie par notre système. Nous estimons que cette valeur très faible de bruit, et la à une forte corrélation entre les humains et le système concourent à prouver que l'etiquetage est performant. La perte de performance lorsque $MinSim$ est très élevé s'explique par le fait que l'algorithme commence à rater des classes. Pour information, le temps de calcul pour faire tourner l'algorithme sur 39000 document sur un Pentium III, 450MHz, 512 MB RAM est de 45 secondes.

6 Conclusion

Dans cet article, nous avons présenté une nouvelle mesure de similarité entre ensembles de concepts d'une hiérarchie, et avons proposé d'utiliser cette mesure avec l'algorithme DBSCAN. Les résultats obtenus sont probants, et justifient notre approche. Un point important que nous n'avons pas abordé ici est l'hypothèse de base, qui est que les documents web peuvent être représentés par un ensemble concis de concepts. Il est intéressant de noter que dans la construction de cet ensemble, nous nous servons notamment de la mesure définie ici, que nous appliquons sur WordNet, en la considérant comme une *ontologie*.

Références

[AGY99] Charu Aggarwal, Stephen Gates, and Philip Yu. On the merits of building categorization systems by supervised clustering. In *Proceedings of the ACM-SIGKDD*, 1999.

[BFS02] A. Bidault, Ch. Froidevaux, and B. Safar. Proximité entre requêtes dans un contexte médiateur. In *RFIA*, 2002.

[BKSS90] N. Beckmann, H.P. Kriegel, R. Schneider, and B. Seeger. The R*-tree: An efficient and robust access method for points and rectangles. In *Proceedings of the ACM-SIGMOD*, 1990.

[DJ01] E. Desmontils and C. Jacquin. Des ontologies pour indexer un site web. In *Journées Francophones d'Ingénierie des Connaissances*, 2001.

[EKSX96] Martin Ester, Hans-Peter Kriegel, Jord Sander, and Xiaowei Xu. A density based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd ACM-SIGKDD*, 1996.

[EM97] Thomas Eiter and Heikki Mannila. Distance measures for point sets and their computation. *Acta Informatica Journal*, 34:109–133, 1997.

[Fis87] Douglas Fischer. Knowledge acquisition via incremental conceptual clustering. *Machine Learning Journal*, 2:139–172, 1987.

[GGK01] Aristides Gionis, Dimitrios Gunopulos, and Nick Koudras. Efficient and tunable similar set retrieval. In *Proceedings of the ACM-SIGMOD*, 2001.

[GHOS96] J. Green, N. Horne, E. Orlowska, and P. Siemens. A rough set model of information retrieval. *Theoretica Informaticae*, 28:273–298, 1996.

[HNVV02] M. Halkidi, B. Nguyen, I. Varlamis, and M. Vazirgiannis. Thesus: Organising web document collections based on semantics and clustering. Technical report, Verso Technical Report, 2002.

[kar] http://www.kartoo.com/.

[Kle99] J. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46, 1999.

[Lin98] D. Lin. An information theoretic definition of similarity. In *Proceedings of the 15th ICML*, 1998.

[Nii87] I. Niiniluoto. Truthlikeness. 1987.

[PW00] T. Phelps and R. Wilensky. Robust hyperlinks cost just five words each. Technical Report UCB//CSD-00-1091, UC Berkeley Computer Science, 2000.

[Res95] P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of IJCAI*, 1995.

[Res99] P. Resnik. Semantic similarity in a taxonomy: An information based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research*, 11, 1999.

[RSM] R. Richardson, A.F. Smeaton, and J. Murphy. Using wordnet as a knowledge base for measuring semantic similarity between words. Technical report.

[SM83] Gerard Salton and Michael McGill. *An Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.

[Viv] http://www.vivissimo.com/.

[VNVA02] I. Varlamis, B. Nguyen, M. Vazirgiannis, and S. Abiteboul. Effective thematic selection in the www based on link semantics. Technical report, 2002.

[Wor] http://www.cogsci.princeton.edu/ wn.

[WP94] Z. Wu and M. Palmer. Verb semantics and lexical selection. In *Proceedings of the 32nd Annual Meeting of the Associations for Computational Linguistics*, 1994.

[ZE98] O. Zaimir and O. Etzioni. Web document clustering, a feasibility demonstration. In *Proceedings of the ACM-SIGIR*, 1998.