Identification de répétitions dans les navigations au sein d'un catalogue de santé

A. Pauchet[1] — M. El Abed[2] — T. Merabti[2] — É. Prieur[2] — T. Lecroq[2] — S.J. Darmoni[2]

[1] *INSA Rouen - LITIS EA 4108, BP08, 76801 Saint-Etienne du Rouvray, France*
contact : alexandre.pauchet@insa-rouen.fr

[2] *Université de Rouen - LITIS EA 4108, 76821 Mont-Saint-Aignan, France*

RÉSUMÉ. *Nous présentons dans cet article un algorithme d'extraction de comportements récurrents durant la consultation de ressources au sein du catalogue de santé CISMeF. Nous proposons pour cela d'utiliser la structure de données appelée arbres des suffixes, appliquée aux fichiers log de CISMeF. Parallèlement à cela, nous nous intéressons à l'identification de ressources pertinentes pour une requête donnée en construisant un ensemble de ressources syntaxiquement et sémantiquement proche des ressources consultées au cours de la navigation. L'idée sous-jacente étant, à partir de la consultation d'une ou plusieurs ressources, de proposer une liste de liens susceptibles de contenir l'information recherchée par l'utilisateur.*

ABSTRACT. *In this article we aim at introducing an algorithm designed to extract reccurrent navigation behaviours within CISMeF, a catalogue of Health Resources. We propose to use the data structure called tree suffixes, applied to the log files collected from CISMeF. At the same time, we are interested in the identification of resources relevant to a query by building a set of resources syntactically and semantically close to the resources visited during the user navigation. The idea is, when consulting one or more resources, to propose a list of ressources that could contain the information sought by the user.*

MOTS-CLÉS : *Fouille de données internet, extraction de motifs de navigation, arbres des suffixes*

KEYWORDS: *Web Log Mining, navigation pattern extraction, suffix trees*

1. Introduction

Internet représente actuellement un énorme volume de données, qu'il est souvent difficile à appréhender pour les utilisateurs. En particulier, la recherche d'une information précise est souvent longue et fastidieuse, malgré les nombreux moteurs de recherche, catalogues, annuaires et index existants. La problématique sous-jacente est de fournir rapidement et efficacement à un utilisateur un accès à l'information qu'il désire, lors d'une recherche d'information sur Internet.

Un moteur de recherche s'appuie sur une indexation efficace des ressources identifiées sur internet de façon manuelle (pour les annuaires) ou automatique. Cette indexation consiste en l'identification de mots considérés comme caractéristiques pour chaque ressource. La recherche classique d'une information consiste alors, à partir d'une requête, à présenter une liste de ressources par ordre de pertinence supposée, selon l'indexation de la ressource.

Mesurer la pertinence d'une ressource, c'est-à-dire son adéquation à une requête, est donc primordiale car cela conditionne l'ordre d'affichage des ressources proposées lors d'une recherche d'information. Trois grands axes de recherche permettent d'améliorer cette adéquation.

1) Améliorer l'indexation des ressources : les informations décrivant une ressource sont souvent incomplètes ou imparfaites, il s'agit donc là d'améliorer de façon quantitative et qualitative ces informations. De plus, Internet étant extrêmement dynamique, il existe un très grand nombre de ressources non encore référencées ou inaccessibles, qu'il conviendrait d'intégrer aux recherches. Google (Brin *et al.*, 1998) est le meilleur exemple de collecte d'informations (*crawling*) et d'indexation du Web. La collecte et l'indexation automatiques des ressources peuvent aussi être améliorées en utilisant les métadonnées formelles contenues dans les ressources (Weibel *et al.*, 2000), en ce qui est appelé le web sémantique (Berners-Lee *et al.*, 2001).

2) Améliorer la précision de la requête des utilisateurs : une recherche est souvent effectuée sous la forme d'un ensemble de mots clefs, qui ne traduisent pas forcément correctement la pensée de l'utilisateur. (Loisel *et al.*, 2008) par exemple propose un système de dialogue afin d'aider l'utilisateur à préciser sa requête pour qu'elle soit optimale et corresponde exactement à ce qu'il souhaite.

3) Améliorer le processus de recherche lui-même ou, en d'autres termes l'évaluation de la pertinence des ressources proposées. La plupart du temps, cela consiste à appliquer à des données issues d'Internet (pages ou ressources, fichiers de navigation, logs de serveur, *etc.*) des techniques de fouille de données, afin d'en extraire de nouvelles connaissances. C'est cette dernière approche que nous utilisons dans cet article.

Dans le cadre particulier du domaine médical, le projet CISMeF (Catalogue et Index des Sites Médicaux Francophones) (Darmoni *et al.*, 2000) s'adresse en particulier aux professionnels de santé afin de favoriser leur accès à l'information disponible sur Internet. Nous proposons dans cet article trois méthodes complémentaires permettant d'identifier les ressources pertinentes pour une requête, à partir du moment

où une première ressource a été consultée. La première méthode consiste à présenter un ensemble de resources fréquemment consultées par les utilisateurs. La seconde méthode permet de construire un ensemble de ressources syntaxiquement et sémantiquement proches de la ressource consultée. Enfin, la dernière méthode, sur laquelle nous nous étendrons davantage, vise à identifier des répétitions au sein des navigations des utilisateurs CISMeF, dans la consultation de ressources. Cette méthode peut être vue comme une simplification de la recherche de séquences (Masseglia *et al.*, 2004) permettant de bénéficier de la puissance des arbres de suffixes (construction en temps et espace linéaires et extraction des motifs répétés en temps et espace linéaires).

Dans la suite de cet article, la section 2 présente un ensemble de travaux relatifs à notre problématique. La section 3 décrit le contexte de cette étude (CISMeF). La section 4 se concentre sur la recherche de ressources pertinentes pour une requête donnée. La section 5 décrit la recherche de comportements récurrents dans les fichiers log de CISMeF. Enfin, la section 6 conclut cet article en présentant quelques perspectives pour ces travaux.

2. État de l'art

Internet et la fouille de données sont deux domaines de recherche importants et très actifs depuis quelques années. Le premier génère un flux volumineux de données que le second est capable de traiter pour en extraire des connaissances. La combinaison de ces domaines a alors donné naissance à ce qui est appelé "Web Mining" par les anglo-saxons (Kosala *et al.*, 2000; Iváncsy *et al.*, 2006). Ces techniques de fouille de données peuvent être appliquées à trois grands types différents de données :

– au contenu des ressources (*Web Content Mining*),

– à la structure d'organisation des ressources (*Web Structure Mining*),

– à l'utilisation faite par les utilisateurs de ces ressources (*Web Usage Mining*).

Parmi les algorithmes visant à modéliser la topologie du Web, on citera en particulier PageRank (Google) (Brin *et al.*, 1998) et The Clever System (HITS) (Chakrabarti *et al.*, 1999), dont la qualité majeure est d'essayer d'évaluer la pertinence de chaque page à une requête.

Nous nous focalisons ici sur les travaux relatifs *Web Usage Mining*, et en l'occurrence sur l'extraction de comportements à partir de l'analyse de fichiers de navigation. Il s'agit donc de l'application de techniques de fouille de données afin de découvrir des comportements récurrents chez les utilisateurs d'Internet, afin d'en améliorer la navigation. Comme dans toute tâche de fouille de données, le processus se décompose en trois étapes principales : 1) un pré-traitement permettant une mise en forme des données, 2) une recherche de motifs et 3) l'analyse des motifs récurrents observés. (Tanasa *et al.*, 2004) décrit très précisément la première étape de mise en forme des données. Les étapes deux et trois sont souvent indissociables et donc traitées ensemble dans la plupart des travaux.

Les applications orientées Web Usage Mining peuvent être classées en deux catégories distinctes, selon qu'elles cherchent à construire un modèle de l'utilisateur (Langley, 1999) ou bien qu'elles visent à découvrir des ensembles de comportements caractéristiques dans la navigation (Spiliopoulou, 1999). Les connaissances acquises sont alors utilisées pour personnaliser, voire individualiser, un site selon le profil utilisateur, pour optimiser l'architecture du site, pour améliorer les performances du serveur ou enfin dans le but d'optimiser les gains d'un site commercial (personnalisation de publicités, promotions, *etc.*). Nous nous intéressons dans cet article plus particulièrement à la recherche de motifs dans des fichiers log. Cette extraction de motifs vise souvent à identifier les utilisateurs, mais le but final et la manière diffèrent souvent.

(Gao *et al.*, 2004) identifie les utilisateurs d'un site Web à visée commerciale à l'aide de réseaux Bayésiens, afin de retenir au maximum les clients éventuels.

(Benabdeslem *et al.*, 2006) et (Charrad *et al.*, 2006), eux, s'appuient sur des cartes topologiques de Kohonen. (Benabdeslem *et al.*, 2006) propose ces cartes de Kohonen afin de construire une cartographie d'un site Web tel qu'il est perçu par les utilisateurs et de projeter leur navigation sous la forme de trajectoires représentant leur comportement. (Charrad *et al.*, 2006) caractérise les utilisateurs d'un site Internet en se basant sur leurs motifs de navigation. La découverte de motifs de navigation est effectuée en combinant une analyse des correspondances multiples et des cartes de Kohonen.

(Büchner *et al.*, 1999) présente l'algorithme M*i*DAS pour la découverte de séquences d'actions dans des fichiers log. Les mécanismes proposés sont appliqués là aussi à un ensemble de sites de commerce en ligne, dans le but de proposer des promotions et des activités personnalisés.

(Iváncsy *et al.*, 2006) utilisent trois algorithmes différents fondés sur des ensembles de ressources fréquemment consultées, des séquences et des motifs sous forme d'arbres. Il préconise l'utilisation de chacun de ces algorithmes selon le but de l'application (publicités personnalisées, création dynamique de profil utilisateur, *etc.*).

(Yang *et al.*, 2001) applique des méthodes de Data Mining à des fichiers log de navigation sur un même serveur afin d'en extraire un modèle de prédiction des requêtes futures d'un utilisateur. Le but est de stocker en mémoire cache les ressources visées pour réduire le temps de latence entre la requête et l'envoi de la ressource. Le modèle s'appuie sur l'identification de séquences de requêtes fréquemment utilisées et l'algorithme de prédiction utilise un ensemble de règles d'association. Ces travaux ont donné lieu à la mise en place d'un système de stockage de données appelé Netshark (Fang *et al.*, 2006).

3. Contexte : CISMeF et sa terminologie

CISMeF (`http://www.chu-rouen.fr/cismef` ou `http://www.cismef.org`) est l'acronyme de Catalogue et Index des Sites Médicaux Francophones sur l'Internet

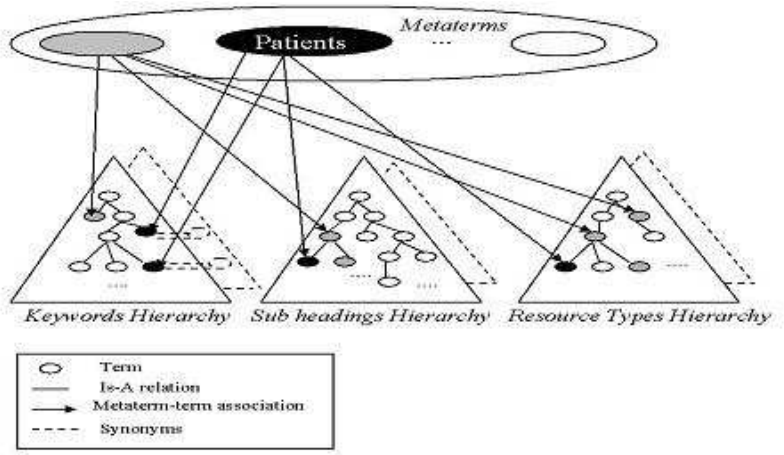

Figure 1. *Liens sémantiques entre les méta-termes et les Types de Ressources, qualificatifs et descripteurs CISMeF*

(Darmoni *et al.*, 2000). Il s'agit d'un portail de santé (Koch, 2000) qui a été conçu pour cataloguer et indexer les sources d'information institutionnelle de santé françaises les plus importantes (environ 50'000 sources actuellement) et ce afin de permettre une recherche plus pertinente pour les professionnels de santé, les étudiants mais aussi les patients, leur familles, et d'une façon encore plus large le cyber-citoyen (Douyère *et al.*, 2004). CISMeF adhère aux principes de qualité de l'information de santé sur l'Internet défini par la Health on the Net (HON) Foundation depuis 10 ans maintenant (Boyer *et al.*, 2007).

CISMeF utilise deux outils standards pour organiser l'information : le thesaurus MeSH (Nelson *et al.*, 2001), utilisé généralement pour indexer les articles scientifiques de la base de données MEDLINE, ainsi qu'un ensemble de méta-données extraites du noyau de Dublin Core[1] (Dekkers *et al.*, 2003). Les méta-données se réfèrent aux informations descriptives des ressources Web et dont les plus importantes sont le titre, l'identifiant, la date, le contenu, le mot clef et le type de ressource. Pour plus de précision, huit méta-données spécifiques à CISMeF ont été ajoutées tels que : pays, institution, ...

La terminologie CISMeF encapsule la version française du thésaurus MeSH dans la mesure où, d'une part, elle représente une extension des concepts déjà existants dans le MeSH et, d'autre part, elle emploie de nouveaux concepts. Dans le but de généraliser le MeSH aux ressources de santé sur Internet, des améliorations ont été réalisées depuis l'année 2000. En plus des descripteurs et des qualificatifs MeSH,

1. `http://www.dublincore.org`

deux nouveaux concepts ont été créés au sein de l'équipe CISMeF : des types de resources qui sont une extension des types de publications de MEDLINE et des méta-termes. La paire (descripteur/qualificatif) décrit le centre d'intérêt de la ressource. Les types de ressources sont utilisés afin de catégoriser la nature de la ressource. Les méta-termes sont, en général, des spécialités médicales ou des sciences biologiques, qui ont un lien sémantique avec un ou plusieurs termes MeSH, qualificatifs et types de ressources (*cf.* figure 1). En juin 2008, nous avions créé 123 méta-termes et 289 types de ressources dans la terminologie de CISMeF. Les listes complètes des méta-termes et des types de resources sont disponibles respectivement aux URL suivantes : `http://doccismef.chu-rouen.fr/liste_des_meta_termes_anglais.html` et `http://www.chu-rouen.fr/documed/typeeng.html`.

4. Ressources pertinentes pour une requête donnée

Lors de l'envoi d'une requête par un utilisateur, la présentation des résultats, lorsqu'elle est correctement réalisée, permet un gain de temps considérable. En effet, il est important de trier les liens vers les ressources satisfaisant la requête selon leur pertinence. Cependant, il est impossible de connaître parmi les ressources que l'utilisateur peut sélectionner, l'importance de chacune d'elle dans son contexte, sans poser la question à l'utilisateur.

Deux approches sont présentées ici. La première approche considère que la pertinence est liée à la fréquence des demandes pour une ressource (*cf.* section 4.1). La seconde approche s'appuie sur le contenu des ressources pour présenter des ressources syntaxiquement et sémantiquement proches (*cf.* section 4.2).

4.1. *Ressources fréquemment demandées*

Cette approche, très classique, est aisée à mettre en œuvre. Les ressources correspondants à une requête sont présentées dans l'ordre de fréquence de leur consultation.

Par exemple, pour la requête « école d'infirmiers.tr », Doc'CISMeF affiche 32 ressources pour cette requête. Le tableau suivant présente les ressources qui ont été demandées pour cette requête, ainsi que leur occurrence, durant la journée du 30/04/2008 (*cf.* tableau 1).

Seules 8 ressources ont été visitées parmi les 32, durant cette journée. L'affichage de ces ressources doit donc se faire suivant l'ordre : 146171, 216501, 101731, 283691, 29711, 155951, 29611, 29661, suivi des 24 ressources restantes.

Cette approche, bien que la plus souvent utilisée dans les moteurs de recherche, présente deux inconvénients majeurs :

1) L'ordre d'apparition des ressources ne tient pas compte du contexte, mais uniquement de l'intérêt de la majorité des utilisateurs pour cette requête. Ainsi, une

Ressource	Occurences
146171	11
216501	10
101731	2
283691	2
29711	2
155951	1
29611	1
29661	1

Tableau 1. *Ressources demandées pour la requête « école d'infirmiers.tr »*

ressource répondant à un grand nombre de requêtes sera considérée comme plus pertinente qu'une ressource plus ciblée mais plus adéquate.

2) Elle a un effet de renforcement : plus une ressource est élevée dans l'ordre de présentation et plus elle sera sélectionnée.

4.2. *Ressources syntaxiquement et sémantiquement proches*

La seconde approche consiste à utiliser les propriétés de l'algorithme "CISMeF related resources" (CISMeF_RRA) (Merabti *et al.*, 2008) développé sur le catalogue CISMeF (Douyère *et al.*, 2004). L'algorithme CISMeF_RRA est inspiré de l'algorithme "Related Citation" de PubMed (Kim *et al.*, 2001), qui consiste à calculer les ressources les plus proches par rapport à une ressource sélectionnée dans le catalogue (*cf.* figure 2).

CISMeF_RRA est défini en utilisant la description de le ressource CISMeF (Titre, Résumé, Mots clés, Types de ressources). L'algorithme combine deux distances pour calculer la similarité entre ressources, une syntaxique (sur l'ensemble des mots du titre et du résumé) et une sémantique (relations sémantiques entre les mots d'indexations de chaque ressource).

L'approche proposée en utilisant CISMeF_RRA est formulée ainsi :

Soit n le nombre de ressources que l'utilisateur a sélectionné à partir d'une requête formulée dans le catalogue (*cf.* figure 3) :

– Si n=1, l'utilisateur se voit proposer les ressources les plus proches en cliquant seulement sur le bouton "Document Proche".

– Si n>1, l'utilisateur se voit proposer l'ensemble constitué de l'union des ressources les plus proches pour toutes les ressources de 2 jusqu'à n, avec un score identique pour chaque ressource.

Figure 2. *Documents proches dans CISMeF*

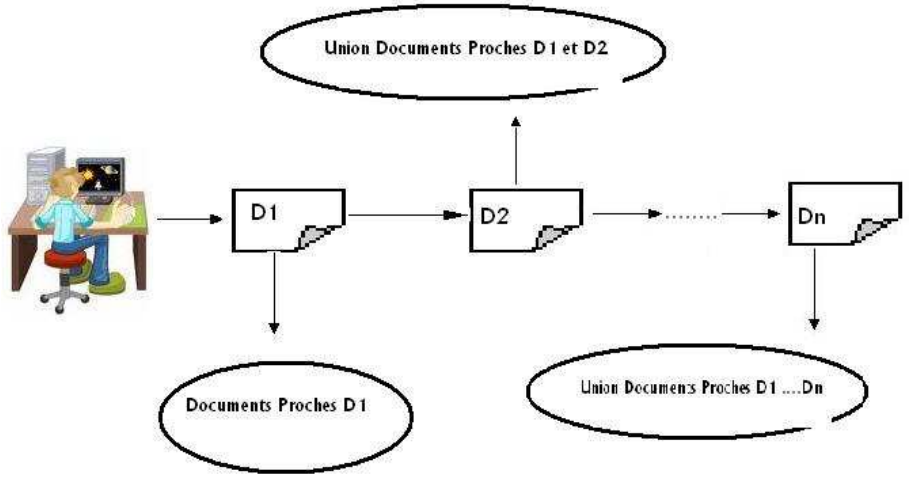

Figure 3. *Schéma représentant l'évolution des ressources proposées à l'utilisateur suivant l'enchaînement de sa recherche*

5. Recherche de comportements récurrents

Dans cette section, nous présentons la recherche de comportements récurrents de consultation de ressources CISMeF. Ces comportements récurrents apparaissent sous la forme de motifs répétés, identifiés dans les fichiers log de CISMeF à l'aide d'arbres des suffixes. Nous avons choisi les arbres des suffixes dans ce contexte car ils peuvent se calculer en temps et espace linéaires et permettent d'effectuer des requêtes en temps linéaire. Nous présentons tout d'abord la structure de donnée utilisée, les arbres des suffixes, puis la préparation des données et enfin les résultats obtenus que nous commenterons.

5.1. *Arbre des suffixes*

5.1.1. *Définitions*

Un *mot* est une suite finie de caractères. La *longueur* d'un mot est son nombre de caractères. On distingue le mot de longueur 0, qu'on appelle le *mot vide* et qui est noté ε. Un mot x de longueur est noté $x[1\,..\,m]$. Un *facteur* de x est une suite de caractères qui apparaissent de manière consécutive dans x. L'unique facteur de x commençant à la position i et se terminant à la position j (avec $1 \leq i \leq j \leq m$) est noté $x[i\,..\,j]$. Un *suffixe* de x est un facteur de la forme $x[i\,..\,m]$ avec $1 \leq i \leq m$

L'arbre des suffixes est une structure qui permet de représenter tous les facteurs d'un mot. Plus formellement, il est défini de la manière suivante :

Définition 1 *L'*arbre non compact des suffixes *d'un mot y est l'automate déterministe ayant un unique état initial appelé la* racine *tel que l'ensemble des états terminaux est l'ensemble des* feuilles *de l'arbre, les autres états sont les* nœuds internes *de l'arbre. Un état terminal correspond à un suffixe du mot. Le langage reconnu par cet automate est l'ensemble des suffixes de y.*

En pratique on ajoute un terminateur (généralement noté \$) à la fin du mot. Ce terminateur ne doit pas posséder d'autre occurrence dans le mot, ce qui fait qu'aucun facteur interne n'est un suffixe. Chaque feuille de l'arbre représente donc un suffixe non vide du mot et chaque suffixe non vide est représenté par une feuille. Les feuilles sont numérotées par la position de début du suffixe qu'elles reconnaissent.

L'arbre de la figure 4 est l'arbre non compact des suffixes pour le mot `tata$`. Les branches d'un tel arbre sont étiquetées par une unique lettre. Le nombre maximal de nœuds d'un arbre non compact des suffixes est quadratique.

Pour compacter l'arbre, on supprime les nœuds internes ne possédant qu'une branche sortante et on concatène les branches.

La figure 5 représente l'arbre obtenu après compaction. Les branches sont donc désormais étiquetées par des facteurs du mot.

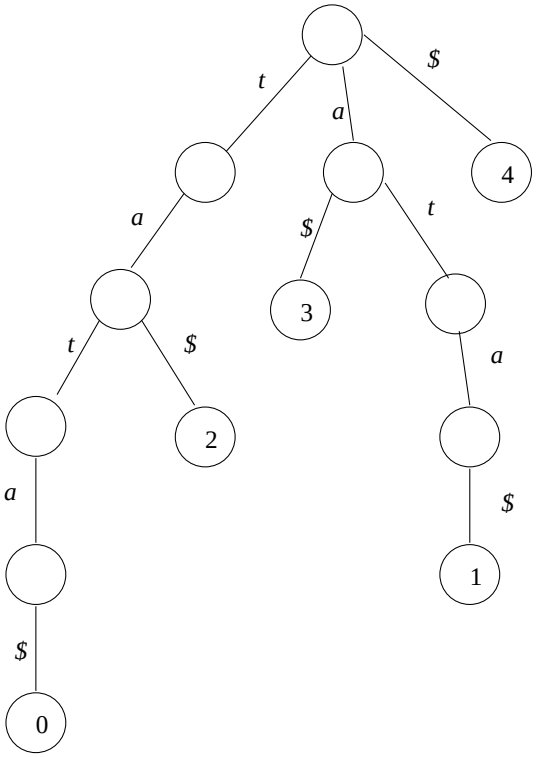

Figure 4. *Arbre non compact des suffixes du mot* `tata$`

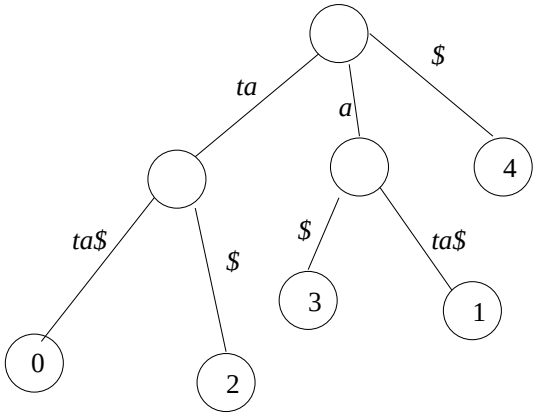

Figure 5. *Arbre compact des suffixes du mot* `tata$`.

Propriété 1 *Chaque nœud p de l'arbre est identifié au facteur qu'il représente, c'est-à-dire à la concaténation des étiquettes des branches du chemin allant de la racine à p. Par conséquent, la racine est identifiée au mot vide ε.*

Pour diminuer l'espace de stockage des étiquettes, on les remplace par des couples (i, ℓ), où i est la position de début du facteur représenté par la branche et ℓ la longueur de ce facteur.

Définition 2 *On appelle* transition *un couple (i, ℓ) représentant une branche de l'arbre des suffixes.*

L'arbre compact des suffixes d'un mot y est noté $\mathcal{A}(y)$. Dans la suite, un arbre des suffixes désigne un arbre compact des suffixes.

Proposition 1 *Le nombre de nœuds internes de l'arbre des suffixes d'un mot de longueur n est au plus $n - 1$.*

Les algorithmes de McCreight (McCreight, 1976), d'Ukkonen (Ukkonen, 1995) et de Farach (Farach, 1997) permettent de construire l'arbre compact des suffixes d'un mot en un temps et un espace linéaires (proportionnelles à la longueur du mot). L'appartenance d'un mot x à l'arbre $\mathcal{A}(y)$ ou de manière similaire savoir si x est un facteur de y peut se faire en temps $O(|x|)$.

5.1.2. *Arbre des suffixes généralisé*

Jusqu'ici nous avons présenté l'arbre des suffixes pour un mot, nous allons voir maintenant l'arbre des suffixes généralisé, que l'on construit pour un ensemble fini de mots.

On ajoute des terminateurs deux à deux différents à chacun des mots de l'ensemble. L'arbre possède donc autant de feuilles qu'il y a de suffixes dans les différents mots. Les feuilles sont alors numérotées par des couples (numéro du mot, position du suffixe dans ce mot).

Exemple

La figure 6 représente l'arbre des suffixes généralisé pour les mots `tata$`, `ttaca#` et `acac§`. La feuille `(2,1)` représente le suffixe débutant à la position 1 dans le mot 2, soit `taca#`. La somme des longueurs des trois mots est 16 (terminateurs compris) et leur arbre des suffixes généralisé comporte 7 nœuds internes correspondant à autant de facteurs répétés. Ainsi `ac` apparaît dans les mots 2 et 3 puisque le sous-arbre enraciné en `ac` possède des feuilles ayant 2 et 3 comme première composante.

L'arbre des suffixes généralisé peut se calculer en temps et espace linéaires (proportionnels à la somme des longueurs des mots). Il possède donc un nombre de nœuds

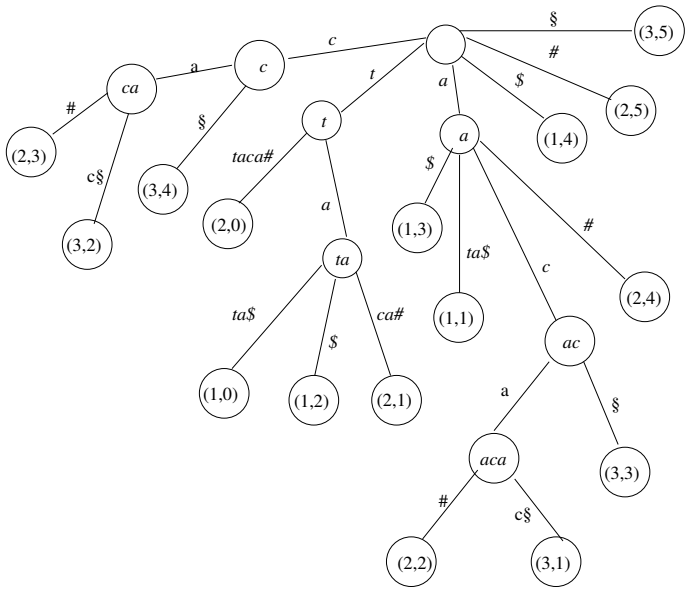

Figure 6. *Arbre des suffixes généralisé pour {*`tata$, ttaca#, acac$`*}.*

interns également proportionnel à cette somme. Ces nœuds internes permettent de détecter les facteurs communs à tous les sous-ensembles de mots appartenant à l'ensemble.

5.2. *Préparation des données*

Préalablement à toute tâche de fouille de données et d'extraction de connaissances, une phase de pré-traitement des données est nécessaire, et souvent prépondérente quant à la qualité des résultats. (Tanasa *et al.*, 2004) identifie précisément 4 phases :

1) fusion des différents fichiers de log,

2) suppression des informations inutiles,

3) structuration des données,

4) stockage dans une base de données.

Ces différentes étapes ont été respectées scrupuleusement. Les étapes 1 et 4 étant triviales, nous ne nous intéressons ici qu'aux étapes 2 et 3. Cela consiste à ne conserver que les informations à traiter, et à extraire les *épisodes* (*i.e.* l'ensemble des requêtes visant à obtenir une information).

À noter que dans la plupart des cas, parmi les informations inutiles à supprimer dans des fichiers log, se trouvent les requêtes issues de *web bots* (ou *spiders*) qui

scannent systématiquement les sites Web afin d'en extraire leur contenu. Les fichiers utilisés ici ne sont pas des log de serveur internet mais des log internes du serveur CISMeF, rendant inutile un tel filtrage.

5.2.1. *Description des données*

Les fichiers log traités résument l'ensemble des événements survenus sur le serveur CISMeF. Ces fichiers contiennent :

– l'adresse IP du client,

– la date et l'heure de la requête,

– le type de recherche : recherche simple ou recherche logique,

– la requête CISMeF correspondante,

– la ressource cible,

– le système d'exploitation et le navigateur utilisé.

Dans un premier temps, nous ne conservons que les requêtes ne portant pas sur une ressource de type image ou multimédia. Puis, afin d'extraire les épisodes, les seules informations analysées sont l'adresse IP du client, la date et l'heure de la requête et la ressource cible.

5.2.2. *Extraction des épisodes*

Une *session* est un ensemble d'actions effectuées par un utilisateur donné au cours de sa visite sur un site web. Cette information est aisément accessible, en particulier dans le cas où la connexion est sécurisée. Dans le cadre de la recherche d'information, il est nécessaire d'identifier non pas les sessions, mais les *épisodes* (Tanasa *et al.*, 2004), c'est-à-dire un ensemble de requêtes effectuées par un utilisateur en vue d'atteindre une ressource ou de récupérer une information.

L'identification d'épisodes dans un fichier log est un problème difficile puisque :

– les requêtes effectuées par différents utilisateurs sont mélangées,

– un même utilisateur peut effectuer des recherches successives, voire simultanées,

– un même utilisateur peut interrompre une recherche et la reprendre plus tard.

La solution la plus souvent utilisée pour identifier des sessions est l'utilisation d'une distance sémantique entre les différences ressources consultées (Tanasa *et al.*, 2004) afin de ne garder au sein d'une même session que la consultation de ressources sémantiquement proches. Cependant, ce type de filtrage est extrêmement dépendant de la distance choisie ; nous avons donc préféré utiliser les deux critères suivants afin d'identifier les différents épisodes :

– Chaque utilisateur a son propre épisode ; les requêtes émises par deux utilisateurs différents (adresses IP différentes) ne sont pas enregistrées dans la même session.

– Pour un utilisateur donné, le temps séparant deux requêtes consécutives ne doit pas dépasser un seuil fixé. En d'autres termes deux requêtes issus d'un même utilisa-

teur mais séparées par un trop grand écart de temps sont considérées comme appartenant à deux épisodes différents.

5.2.3. *Construction du texte*

Chaque ressource atteinte par un utilisateur est identifié de manière unique par un numéro de ressource. Un délimiteur ('/') est utilisé pour séparer les différentes ressources le long d'un épisode.

Par exemple,

```
/59451//303901//170702/
```

est la concaténation d'un épisode durant lequel un utilisateur a successivement appelé les ressources `59451`, `303901` et `170702`.

5.2.4. *Arbre des suffixes, motifs récurrents*

L'arbre des suffixes généralisé des épisodes a été construit en utilisant le module `Tree::Suffix2`, interface Perl de la bibliothèque libre libstree[2], implémentant l'algorithme d'Ukkonen (Ukkonen, 1995), afin d'en extraire les séquences récurrentes. La méthode utilisée est `longest_repeated_substrings`. Un filtrage a alors été appliqué pour que toutes les séquences ne contiennent que des identifiants complets de ressources.

Les séquences récurrentes résultantes correspondent alors à un ensemble de motifs récurrents, constitués de suites de requêtes d'utilisateurs vers des ressources.

5.3. *Résultats*

Les résultats présentés ici proviennent de l'analyse des fichiers log collectés sur le serveur CISMeF durant 22 jours (le 29/04/2008 et du 01/05/2008 jusqu'à 21/05/2008).

5.3.1. *Extraction des épisodes*

En utilisant 10 minutes comme unité de temps pour séparer les différents épisodes, 48168 épisodes ont été mis à jour. Le tableau suivant présente la proportion de chaque épisode en fonction du nombre de ressources consultées (*cf.* tableau 2).

À noter que le nombre total d'épisodes est 48168 et que le nombre maximal de liens visités au cours d'une session est 64.

À partir de ce tableau, nous pouvons conclure qu'un seul lien est visité dans 34005 sessions, deux liens sont visités dans 8254 sessions et ainsi de suite.

Le but principal de la recherche de motifs récurrents est de pouvoir identifier des séquences de requêtes dans le but de proposer un ensemble de resources consultées.

---

2. `http://search.cpan.org/~gray/Tree-Suffix-0.20/`

Nombre de liens visités	Nombre d'épisodes	Proportion
1	34005	70,6%
2	8254	17,1%
3	2940	6,1%
4	1284	2,7%
5	658	1,4%
6	346	0,7%
7	216	0,4%
8	139	0,3%
9	91	0,2%
10	60	0,1%
>10	175	0,4%

Tableau 2. *Tableau de correspondance entre sessions et liens visités*

Or, 70,6% des épisodes ne contiennent qu'une seule requête ; nous utiliserons donc 29,4% des épisodes collectés afin d'en extraire des motifs récurrents.

5.3.2. *Extraction des motifs*

À l'aide de l'algorithme présenté précédemment, un certain nombre de motifs récurrents différents, ont été identifiés (*cf.* figure 3). Le seuil minimum pour considérer qu'un motif est détecté est de 2 occurences.

Longueurs des motifs	2	3	4	5
Nombres de motifs	1557	146	20	4

Tableau 3. *Longueurs et effectifs des différents motifs identifiés*

À partir de ce tableau, il ressort que 1557 motifs différents de longueur 2 ont été extraits, 146 de longueur 3 et ainsi de suite.

Le tableau suivant présente les proportions d'épisodes contenant un motif identifié, et ce en fonction de la longueur du motif (*cf.* tableau 4).

	Episodes contenant un motif			
	longueur 2	**longueur 3**	**longueur 4**	**longueur 5**
Effectifs	4127	326	42	8
% épisodes	8,568	0,677	0,087	0,017
% épisodes (l>1)	29,139	2,302	0,297	0,056

Tableau 4. *Proportions d'épisodes contenant un motif identifié*

Nous présentons en particulier les proportions d'épisodes contenant un motif identifié dans l'ensemble des épisodes constitués d'au moins un accès à une ressource (*cf.*

tableau 4), en l'occurence dans 29,4% de la population totale des épisodes (*cf.* section précédente). En effet, les épisodes ne contenant qu'une unique requête vers une ressource ne doivent pas être comptabilisées puisqu'elle ne peuvent potentiellement contenir aucun motif récurrent. On observe donc que plus de 29% des épisodes de longueur $\geqslant 2$ contiennent au moins un motif de longueur 2.

5.4. *Discussion*

Nous avons pu extraire un nombre total de 48168 épisodes indivisibles de recherche d'information, parmi lesquels seuls 29,4% pouvaient être utilisés pour rechercher des comportements récurrents de navigation car contenant au moins deux accès successifs à des ressources différentes. Il n'est cependant pas possible, sans interroger directement les utilisateurs, de savoir s'ils ont quitté le moteur de recherche de CISMeF parcequ'ils ont obtenu l'information qu'ils recherchaient ou non. Cette remarque, particulièrement importante dans le cas de la visualisation d'une unique ressource s'applique bien évidemment aussi quand un ensemble de ressources est consulté. Quoiqu'il en soit, si l'utilisateur est insatisfait du premier lien qu'il consulte, on peut espérer que lui fournir en sus un ensemble de liens visités par d'autres utilisateurs permettraient de le satisfaire.

Parmi les 14163 épisodes contenant au moins deux consultations de ressource, nous avons extrait un ensemble de motifs récurrents dans le comportement des utilisateurs de CISMeF. Pour cela, nous avons choisi d'utiliser des arbres des suffixes puisqu'ils se calculent en temps et espace linéaires et permettent d'effectuer des requêtes en temps linéaire aussi. Les motifs extraits sont de taille 2 à 5, pour un total de 4503 motifs identifiés. Bien que les proportions de motifs de longueur 3, 4 et 5 restent relativement bas, on observe en revanche que 29,139% des épisodes de longueur $\geqslant 2$ contiennent au moins un motif de longueur deux.

Le principal reproche qui pourrait être fait à cette étude est le volume des données sur lesquelles elle s'appuie : 22 jours de fichiers log sur le serveur CISMeF. Afin d'augmenter le nombre de motifs découverts, il faudrait bien évidemment effectuer le même traitement sur une durée beaucoup plus longue, voir même en continu puisque les arbres des suffixes le permettent. Cependant, les résultats obtenus sont prometteurs et montrent la viabilité de la méthode.

6. Conclusion et perspectives

Dans le contexte de CISMeF, catalogue de sites médicaux, nous avons présenté tout d'abord deux méthodes afin d'identifier les ressources pertinentes pour une requête : un ensemble de resources fréquemment consultées par les utilisateurs et un ensemble de ressources syntaxiquement et sémantiquement proches des ressources consultées. Puis, dans un deuxième temps, nous avons présenté une méthode permettant d'identifier des répétitions dans les navigations des utilisateurs CISMeF, et en

particulier dans les ressources consultées. Ces répétitions, ou motifs, caractérisent des séquences identiques de ressources, consultées par des utilisateurs différents.

En ce qui concerne la recherche de motifs dans les fichiers log de CISMeF, il conviendrait d'appliquer l'algorithme sur un volume de données plus important, c'est-à-dire couvrant une période plus longue que 22 jours, afin d'identifier un plus grand nombre de motifs récurrents. En particulier, l'utilisation de tables des suffixes (Manber *et al.*, 1990; Mäkinen, 2003) à la place des arbres des suffixes permettraient, tout en demandant un temps de traitement plus important, de réduire l'espace de stockage et donc d'augmenter le volume des données traitées. Enfin, il serait intéressant de rechercher des motifs dont les ressources consultées ne le sont pas forcément consécutivement en utilisant des techniques similaires à (Marsan *et al.*, 2001). Par exemple, la figure 7 présente 4 épisodes dont les 3 premiers contiennent le motif `{R5, R7, R8}`.

```
Session 1 : R1 ⟶ R5 ⟶ R4 ⟶ R7 ⟶ R8
Session 2 : R5 ⟶ R7 ⟶ R2 ⟶ R8
Session 3 : R3 ⟶ R1 ⟶ R5 ⟶ R7 ⟶ R8 ⟶ R4
Session 4 : R3 ⟶ R4 ⟶ R2
```

Figure 7. *Sessions contenant un motif (`{R5, R7, R8}`) constitué de consultations de ressources non directement consécutives*

Nous pensons utiliser les travaux présentés ici de deux façons distinctes dans le catalogue de santé CISMeF :

1) Pour un épisode durant lequel une unique ressource x a été consultée, si x faisait partie d'un ou plusieurs motifs de longueur supérieure ou égale à deux, la liste L des autres ressources des motifs débutant par la ressource x doit être affichée. À L sera ajouté l'ensemble des ressources syntaxiquement et sémantiquement proches de x. L ne contiendra évidemment pas de doublons.

2) Pour un épisode durant lequel un ensemble de n ressources $x_1, ..., x_n$ ont été consultées, si $x_1, ..., x_n$ faisaient partie d'un ou plusieurs motifs de longueur supérieure ou égale à n, la liste L des autres ressources des motifs débutant par la série de ressources $x_1, ..., x_n$ doit être affichée. À L sera ajouté l'ensemble des ressources syntaxiquement et sémantiquement proches de $x_1, ..., x_n$, conformément à l'algorithme CISMeF_RRA. Une fois encore, L ne contiendra aucun doublons.

Ansi, ce travail permettra aux nombreux utilisateurs de CISMeF (50.000 par jour ouvré) d'avoir deux solutions pour se voir proposer des ressources corrélées à celles préalablement consultées : a) une méthode fondée sur les répétitions et b) une méthode fondée sur une distance mixte statistique et sémantique.

7. Bibliographie

Benabdeslem K., Bennani Y., « Classification et visualisation des données d'usage d'Internet », *Atelier Fouille du Web - Extraction et Gestion des Connaissances (EGC'06)*, Lille, France, 2006.

Berners-Lee T., Hendler J., Lassila O., « The Semantic Web », *Scientific American*, 2001.

Boyer C., Gaudinat A., Baujard V., Geissbühler A., « Health on the Net Foundation : assessing the quality of health web pages all over the world », *twelveth World Congress on Health (Medical) Informatics (Medinfo'07)*, vol. 12, n° 2, p. 1017-1021, 2007.

Brin S., Page L., « The anatomy of a large-scale hypertextual Web search engine », *Computer Networks and ISDN Systems*, vol. 30, n° 1-7, p. 107-117, 1998.

Büchner A., Baumgarten M., Anand S., Mulvenna M., Hughes J., « User-Driven Navigation Pattern Discovery from Internet Data », *Lecturer Notes in Computer Science*, vol. 1836, p. 74-91, 1999.

Chakrabarti S., Dom B., Gibson D., Kleinberg J., Kumar S., Raghavan P., Rajagopalan S., Tomkins A., « Mining the link structure of the World Wide Web », *IEEE Computer*, vol. 32, n° 8, p. 60-67, 1999.

Charrad M., Ahmed M. B., Lechevallier Y., « Extraction de connaissances à partir de fichiers Logs », *Atelier Fouille du Web - Extraction et Gestion des Connaissances (EGC'06)*, Lille, France, 2006.

Darmoni S., Leroy J., Baudic F., Douyère M., Piot J., Thirion B., « CISMeF : a structured Health resource guide », *Methods of Information in Medicine*, vol. 39, n° 1, p. 30-35, 2000.

Dekkers M., Weibel S., « State of the Dublin Core Metadata Initiative », *D-Lib Magazine*, 2003.

Douyère M., Soualmia L. F., Névéol A., Rogozan A., Dahamna B., Leroy J. P., Thirion B., Darmoni S. J., « Enhancing the MeSH thesaurus to retrieve French online health resources in a quality-controlled gateway », *Health Information and Libraries Journal*, vol. 21, n° 4, p. 253-261, 2004.

Fang X., Sheng O., Gao W., Iyer B., « A Data-Mining-Based Prefetching Approach to Caching For Network Storage Systems », *INFORMS Journal on computing*, vol. 18, n° 2, p. 267-282, 2006.

Farach M., « Optimal Suffix Tree Construction with Large Alphabets », *Proceedings of the 38th IEEE Annual Symposium on Foundations of Computer Science*, Miami Beach, FL, p. 137-143, 1997.

Gao W., Sheng O., « Mining characteristic Patterns to Identify Users », *Proceedings of Workshop on Information Technology and Systems*, vol. 13, 2004.

Iváncsy R., Vajk I., « Frequent Pattern Mining in Web log data », *Journal of Applied Science at Budapest Tech Hungary, Special Issue on Computational Intelligence*, vol. 3, n° 1, p. 77-90, 2006.

Kim W., Aronson A. R., Wilbur J., « Automatic mesh term assignment and quality assessment », *AMIA'01*, p. 319-323, 2001.

Koch T., « Quality-controlled subject gateways : definitions, typologies, empirical overview, Subject gateways », *Online Information Review*, vol. 24, n° 1, p. 24-34, 2000.

Kosala, Blockeel, « Web Mining Research : A Survey », *SIGKDD Explorations : Newsletter of the Special Interest Group (SIG) on Knowledge Discovery & Data Mining, ACM*, 2000.

Langley P., « User modeling in adaptative interfaces », *Proceedings of the 7th International Conference on User Modeling*, 1999.

Loisel A., Chaignaud N., Kotowicz J.-P., « Modeling human interaction to design a human-computer dialog system », *International Conference on Enterprise Information Systems (ICEIS'08)*, Barcelone, Espagne, 2008.

Mäkinen V., « Compact Suffix Array — A Space Efficient Full-text Index », *Fundamenta Informaticae*, vol. 56, n° 1-2, p. 191-210, 2003.

Manber U., Myers G., « Suffix arrays : a new method for on-line string searches », *Proceedings of the 1st ACM-SIAM Annual Symposium on Discrete Algorithms*, San Francisco, California, p. 319-327, 1990.

Marsan L., Sagot M.-F., « Algorithms for extracting structured motifs using a suffix tree with application to promoter and regulatory site consensus identification », *J. of Comput. Biol.*, vol. 7, p. 345-360, 2001.

Masseglia F., Teisseire M., Poncelet P., « Extraction de motifs séquentiels. Problèmes et méthodes », *Ingénierie des Systèmes d'Information*, vol. 9, n° 3-4, p. 183-210, 2004.

McCreight E. M., « A Space-Economical Suffix Tree Construction algorithm », *Journal Algorithms*, vol. 23, n° 2, p. 262-272, 1976.

Merabti T., Pereira S., Letord C., Lecroq T., Dahamna B., Joubert M., Darmoni S. J., « Searching Related Resources in a Quality Controlled Health Gateway : a Feasibility Study », *The XXIst International Congress of the European Federation for Medical Informatics (MIE'08)*, vol. 136, p. 235-240, 2008.

Nelson S. J., Johnson W. D., Humphreys B. L., « Relationships in Medical Subject Headings in Relationships in the organization of knowledge », *Bean and Green*p. 171-184, 2001.

Spiliopoulou M., « Data Mining for the Web », *Proceedings of the 3rd European Conference on Principles of Data Mining and Knowledge Discovery (PKDD '99)*, Springer-Verlag, London, UK, p. 588-589, 1999.

Tanasa D., Trousse B., « Advanced Data Preprocessing for Intersites Web Usage Mining », *IEEE Intelligent Systems*, vol. 19, n° 2, p. 59-65, 2004.

Ukkonen E., « On-line Construction of Suffix Trees », *Algorithmica*, vol. 14, n° 3, p. 249-260, 1995.

Weibel S., Koch T., « The Dublin Core Metadata Initiative », *D-Lib Magazine (Web document)*, 2000. `http://www.dlib.org/dlib/december00/weibel/12weibel.html`.

Yang Q., Zhang H., Li T., « Mining Web Logs for Prediction Models in WWW Caching and Prefetching », *7th International Conference on Knowledge Discovery and Data Mining (KDD'01)*, p. 473-478, 2001.

ANNEXE POUR LE SERVICE FABRICATION
A FOURNIR PAR LES AUTEURS AVEC UN EXEMPLAIRE PAPIER
DE LEUR ARTICLE ET LE COPYRIGHT SIGNE PAR COURRIER
LE FICHIER PDF CORRESPONDANT SERA ENVOYE PAR E-MAIL

1. ARTICLE POUR LA REVUE :
RIA - Numéro Spécial "Intelligence Artificielle et Web Intelligence"

2. AUTEURS :
A. Pauchet[1] — M. El Abed[2] — T. Merabti[2] — É. Prieur[2] — T. Lecroq[2] — S.J. Darmoni[2]

3. TITRE DE L'ARTICLE :
Identification de répétitions dans les navigations au sein d'un catalogue de santé

4. TITRE ABRÉGÉ POUR LE HAUT DE PAGE MOINS DE 40 SIGNES :
Identification de répétitions dans CISMeF

5. DATE DE CETTE VERSION :
21 octobre 2008

6. COORDONNÉES DES AUTEURS :

– adresse postale :

[1] INSA Rouen - LITIS EA 4108, BP08, 76801 Saint-Etienne du Rouvray, France

contact : alexandre.pauchet@insa-rouen.fr

[2] Université de Rouen - LITIS EA 4108, 76821 Mont-Saint-Aignan, France

– téléphone : 00 00 00 00 00

– télécopie : 00 00 00 00 00

– e-mail : Yves Demazeau (CNRS, LIG) et Laurent Vercouter (ENSMSE / G2I)

7. LOGICIEL UTILISÉ POUR LA PRÉPARATION DE CET ARTICLE :
LaTeX, avec le fichier de style `article-hermes.cls`,
version 1.23 du 17/11/2005.

8. FORMULAIRE DE COPYRIGHT :
Retourner le formulaire de copyright signé par les auteurs, téléchargé sur :
`http://www.revuesonline.com`

SERVICE ÉDITORIAL – HERMES-LAVOISIER
14 rue de Provigny, F-94236 Cachan cedex
Tél. : 01-47-40-67-67
E-mail : revues@lavoisier.fr
Serveur web : http://www.revuesonline.com