Classification de documents combinant la structure et le contenu.

Samaneh Chagheri*, Catherine Roussey, Sylvie Calabretto*, Cyril Dumoulin*****

**Université de Lyon, CNRS, LIRIS UMR 5205, INSA de Lyon, 7 avenue Jean Capelle, Villeurbanne, France*

Samaneh.chagheri@insa-lyon.fr,sylvie.calabretto@insa-lyon.fr

***Irstea/Cemagref, Campus des Cézeaux, Clermont Ferrand, 24 avenue des Landais, Aubière, France*

Catherine.roussey@irstea.fr

****27, rue Lucien Langénieux, Roanne, France*

Cyril.dumoulin@continew.fr

RÉSUMÉ. *La démocratisation et l'évolution des logiciels de traitements de texte ont révolutionné le monde du document. Les auteurs construisent des documents dits structurés c'est-à-dire dont le contenu textuel s'organise autours de balises. Toutefois, la classification traditionnelle de documents n'utilise que le contenu textuel des documents et ignore les informations de structure. Dans ce papier, nous proposons une nouvelle représentation des documents structurés basée sur un vecteur pondéré associant un mot et une balise. Les poids sont calculés en adaptant les formules TF-IDF et TF-IEF. Cette représentation est construite à partir d'une représentation synthétique du document appelé arbre résumé. Pour évaluer notre approche, nous avons mené plusieurs expérimentations avec un système de classification basé sur le classifieur SVM^{light}. Nous présentons les résultats de nos expérimentations menées sur les corpus REUTERS et INEX.*

Abstract: Developing the text processing applications has revolutionized the world of documents. The author constructs the document as structured document in which the textual content is organized around tags. However, the traditional document classification typically classifies the documents considering the text and ignoring its structural elements. In this paper, we propose a representation method which makes use of structural elements to create the vector of tag and word weighted by an extension of TF-IDF and TF-IEF formula. This representation is constructed from an aggregated tree of XML document. Several experimentations have been made using SVM^{light} as classifier on Reuters and INEX collections.

MOTS-CLÉS : *représentation des documents, classification supervisée, document structuré, Machine à vecteur support, modèle vectoriel, TF-IDF, TF-IEF.*

Keywords: document representation, supervised classification, structured document, Support Vector Machine, Vector Space Model, TF-IDF, TF-IEF.

CORIA 2012, pp. 261–272, Bordeaux, 21-23 mars 2012

1. Introduction

La démocratisation des logiciels de traitement de texte et leur récente évolution ont révolutionné le monde du document. Premièrement, les auteurs construisent des documents dits structurés c'est-à-dire dont les éléments de contenu sont marqués ou décrits par des balises. Deuxièmement, XML est devenu le standard de représentation des documents structurés. Ainsi, la quantité de documents XML échangés atteint un niveau tel que les outils pour rechercher l'information dans ces documents ne suffisent plus. Les outils permettant de classer automatiquement de collections volumineuses de documents XML sont devenus indispensables. Dans nos travaux nous nous intéressons à la classification supervisée de documents XML. Les classes de documents existent et sont apprises par le système de classification à partir d'exemples. Ce type de système est utilisé dans les entreprises où une classification par types des documents rend compte de la diversité des métiers de l'entreprise. Ainsi un système de classification organisera automatiquement la documentation de l'entreprise (manuel utilisateur, rapport de spécification technique).

Les premiers systèmes de classification supervisée de documents utilisaient uniquement le contenu textuel des documents et ignoraient les informations de structure. Nos travaux partent de l'hypothèse que la prise en compte de la structure du document améliore la performance d'un système de classification supervisée.

Dans cet article, nous proposons une nouvelle méthode de représentation des documents XML basée sur le modèle vectoriel de Salton (Salton, 1968) souvent utilisée en classification documentaire. Tout d'abord le document structuré est représenté par un arbre étiqueté dont les étiquettes correspondent aux balises et les feuilles aux textes du document. A partir de cet arbre, un arbre résumé est construit par simplifications successives de la structure et en agrégeant les feuilles contenant le texte. A partir de l'arbre résumé est extrait un vecteur pondéré de couples (<balise> : « mot »). Les poids sont calculés par une nouvelle formule qui est une adaptation de TF-IDF et TF-IEF (Wolff, et al., 2000). SVM est utilisé comme classifieur. SVM est basé sur le classifieur linéaire qui sépare les exemples positifs des exemples négatifs d'un ensemble d'apprentissage. Cette méthode a obtenu de très bons résultats dans plusieurs domaines dont la classification documentaire.

L'article s'organise de la manière suivante : la section 2 propose un état de l'art des différentes représentations documentaires utilisées pour la classification de documents XML. Une description détaillée des différentes étapes nécessaires à la construction du vecteur est expliquée dans la section 3. La section 4 présente les expérimentations que nous avons menées sur deux corpus de test et les résultats obtenus. Enfin, la section 5 conclue en donnant les perspectives de nos travaux.

2. Etat de l'art

Le nombre croissant de documents XML a demandé aux systèmes de classification de s'adapter à un nouveau type de documents. Les systèmes de classification ne travaillent pas directement sur les documents, ils prennent en entrée une représentation des documents plus synthétique. La représentation de document la plus utilisée dans le domaine de la classification est le modèle vectoriel de Salton (Salton, 1968). Dans le contexte des documents structurés il a fallu adapter les modèles de représentation documentaire. Nous pouvons diviser ces modèles en trois groupes :

Le premier groupe de représentations documentaires ne contient que des éléments textuels des documents. Un document est représenté par un vecteur pondéré de mots[1] extraits du texte du document. Les informations de structure peuvent être utilisées pour modifier la pondération des mots : par exemple dans (Despeyroux, et al., 2005) le poids des mots est fonction de l'importance de l'élément structurel dans lequel ils apparaissent. La formule TF-IDF a été adaptée pour les documents structurés sous le nom de TF-IEF (Sauvagnat, 2005) utilisé en RI. Les informations de structure peuvent aussi être utilisées pour filtrer les mots les plus représentatifs du document comme dans (Kim, et al., 2007).

Le second groupe de représentations ne contient que des éléments structurels du document. Les systèmes de classification de ce type ont pour but de classer les documents en fonction de leur DTD ou de leur schéma XML. Ces systèmes sont appelés systèmes de classification structurelle. Par exemple, les travaux de (Wisniewski, et al., 2005) utilisent un réseau bayésien de balises pour générer les DTD associées à une collection de documents XML. Les systèmes de classification structurelle proposés dans les travaux de (Aïtelhadj, et al., 2009) ou de (Dalamagas, et al., 2005) condensent la structure des documents XML dans un arbre, étiqueté par les balises, intitulé arbre résumé.

Enfin, le troisième groupe est composé des représentations documentaires contenant à la fois des éléments structurels et des éléments textuels. Les travaux présentés dans (Doucet, et al., 2002) utilisent trois représentations vectorielles différentes du document : un vecteur pondéré de mots, un vecteur pondéré de balises et un vecteur pondéré de balises et de mots. Les poids des éléments du vecteur sont calculés à chaque fois par la formule TF-IDF.

Les travaux de (Vercoustre, et al., 2006) représentent un document XML par un ensemble de chemins, textuels ou non, extraits de l'arbre XML. Un chemin est la suite des étiquettes des nœuds visités correspondant à un parcours de l'arbre de la racine jusqu'à une feuille. Un chemin textuel est un chemin prolongé par un mot

---

[1] Dans notre état de l'art, un mot représente toutes les sorties possibles d'un traitement linguistique appliquée sur un texte que ce soit la forme graphique (token), le lemme, la racine (stem) d'un mot.

contenu dans la feuille de l'arbre. La représentation documentaire présentée dans (Wu, et al., 2008) utilise aussi les chemins textuels.

Les travaux de (Yi, et al., 2000) représentent un document par un ensemble de chemins associés au vecteur pondéré des mots contenus dans leur feuille. Ceux de (Ghosh, et al., 2008) utilisent deux représentations vectorielles des documents : un vecteur pondéré de mots et un vecteur pondéré de chemins. Les poids sont calculés par la formule TF-IDF.

Les travaux de (Yan, et al., 2008) (Yang, et al., 2010) proposent une extension du modèle vectoriel appelée SLVM (Structured Link Vector Model) où un document est représenté par une matrice pondérée de mots extraits du texte du document. Cette matrice correspond à un ensemble de vecteurs pondérés de mots : un vecteur par balise. Le poids du mot est proportionnel à sa fréquence dans la balise.

Le modèle de représentation des documents que nous proposons appartient au troisième groupe, il s'agit d'un vecteur pondéré de couples (<balise> : « mot ») avec une nouvelle formule de pondération basée sur une adaptation de TF-IDF et TF-IEF. Les couples composant le vecteur sont extraits d'une simplification du document construite en plusieurs étapes.

3. Présentation générale de notre approche

Pour construire un système de classification supervisé de documents structurés, il faut tout d'abord construire une représentation condensée des documents et ensuite entrainer le classifieur sur l'ensemble des représentations issues du corpus d'apprentissage. L'entrainement permet au classifieur d'apprendre le modèle de chaque classe à partir des exemples de cette classe. La qualité des résultats obtenus par le classifieur dépend directement de la pertinence de la représentation des documents : plus la représentation des documents est pertinente, plus le classifieur obtiendra de bons résultats.

Dans cet article, nous proposons une nouvelle représentation vectorielle des documents XML tenant compte à la fois de la structure et du contenu textuel. Rappelons qu'un document XML est représenté par un arbre étiqueté dont les étiquettes des nœuds correspondent aux balises XML et les feuilles de l'arbre contiennent les parties de texte du document. La figure 2 présente un exemple d'arbre correspondant à un document XML de la figure 1.

Un document XML est représenté dans notre approche par un vecteur pondéré de couples (<balise> : « mot »). Une fois les représentations des documents construites un classifieur SVM est appliqué sur l'ensemble des documents des collections d'apprentissage et de test.

Pour construire notre vecteur pondéré trois types d'analyse sont nécessaires :

1. Les balises sont issues d'une analyse de l'arbre du document XML. L'arbre utilisé dans cette analyse est une simplification de l'arbre XML appelé arbre résumé.
2. Les mots sont issus d'une analyse linguistique du contenu des feuilles du document XML.
3. Les poids des couples (<balise> : « mot ») sont calculés à partir d'une adaptation des formules TF-IDF et TF-IEF.

3.1. Analyse structurelle : extraction des balises

Notre approche s'applique sur une collection de documents XML ayant une structure homogène, c'est-à-dire partageant la même DTD ou le même schéma. Notre représentation du document structuré part du principe qu'un auteur va d'abord écrire la structure du document, son plan, puis remplir chaque élément de structure par du texte. Par exemple, pour la rédaction d'un article scientifique, l'auteur choisit le modèle du document dont il a besoin. Il connait déjà l'organisation du document : existence d'un titre, d'un résumé, des noms d'auteurs, des sections, des titres de section etc. Ensuite il écrit le texte associé au titre, au résumé et…

Nous adoptons la représentation traditionnelle des documents XML sous forme d'arbre étiqueté comme le montre la figure 2. A chaque nœud du document correspond un nœud de l'arbre. Les étiquettes des nœuds correspondent aux noms des balises XML (headline, text, p). Un arc entre deux nœuds correspond à une relation d'inclusion. Le contenu textuel des balises est représenté par un nœud feuille de forme rectangulaire, intitulé nœud texte. Les nœuds textes sont toujours des feuilles de l'arbre.

```
<newsitem class="C15" date="1996-08-20" id="root" itemid="2288" xml:lang="en">
   <headline>CompuServe reports loss, cutting work force.</headline>
   <byline>Susan Zimmerman</byline>
   <text>
      <p>CompuServe Corp. Tuesday reported a surprisingly ...</p>
      <p>CompuServe predicted a second-quarter ...</p>
   </text>
   <text>
      <p>CompuServe, founded in...</p>
   </text>
</newsitem>
```

Figure 1. *Exemple d'un document XML du corpus Reuters*

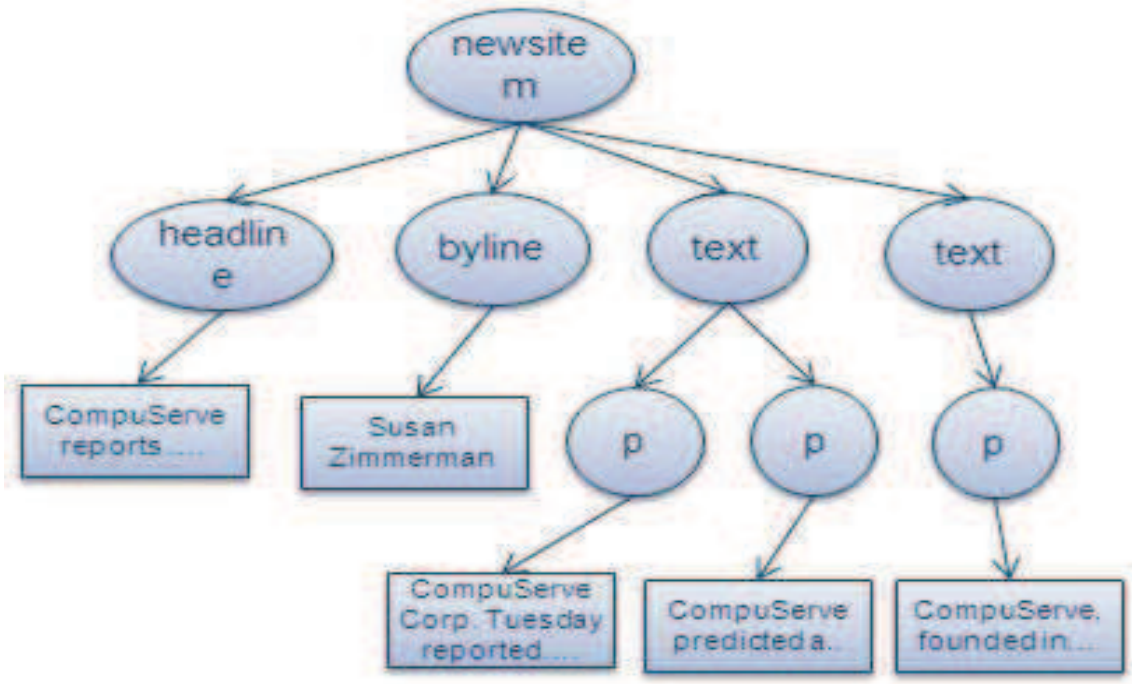

Figure 2. *L'arbre du document XML*

Pour construire l'arbre résumé du document, nous appliquons plusieurs heuristiques pour diminuer le plus possible la taille de l'arbre XML :

1. Nous ne conservons que les branches de l'arbre se terminant par un nœud texte.
2. A partir d'une liste de noms de balises fixée à priori par un expert, nous supprimons les nœuds dont les étiquettes n'appartiennent pas à cette liste ainsi que leurs nœuds enfants.
3. Une fois l'arbre XML nettoyé des balises indésirables, nous supprimons les redondances. Lorsqu'une étiquette se répète plusieurs fois dans une branche de l'arbre, seul le nœud situé le plus proche de la racine est conservé.
4. Ensuite nous agrégeons les chemins identiques. C'est-à-dire que nous fusionnons les nœuds textes qui sont situés sur des branches d'arbres différentes mais dont la séquence d'étiquettes des nœuds est identique.
5. Si deux nœuds portent la même étiquette et sont situés à des profondeurs différentes dans l'arbre, nous les considérons comme différents nœuds avec différents identifications.

Suite à cette analyse structurelle, nous partons de l'hypothèse que dans une collection homogène de documents XML, les nœuds parents des nœuds textes portant le même nom d'étiquette sont toujours situés à la même profondeur. Par exemple dans la collection Reuters les nœuds étiquetés par p sont toujours à une profondeur de 2 dans les arbres résumés.

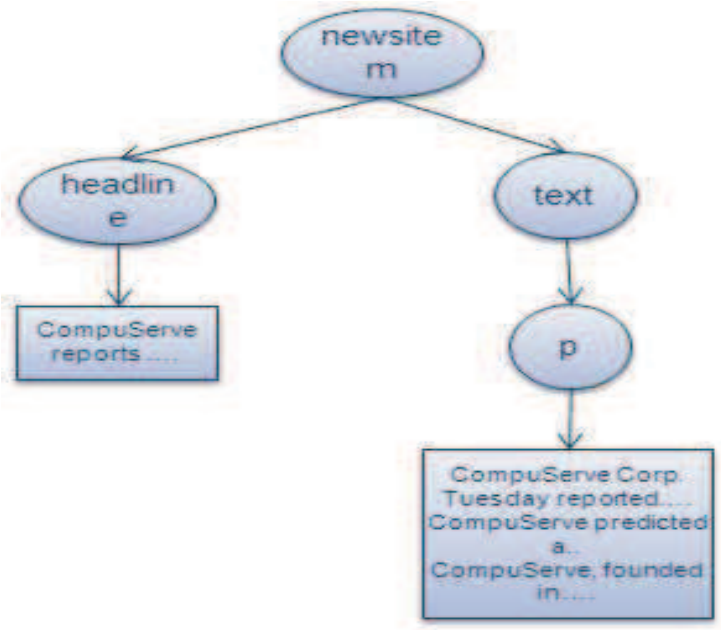

Figure 3. *L'arbre résumé du document XML*

La figure 3 présente l'arbre résumé construit à partir de l'arbre XML de la figure 2. La liste des balises de départ est composée des balises : newsitem, headline, text, p.

3.2. Analyse linguistique : extraction des mots

Sur les nœuds textes de l'arbre résumé, nous appliquons une analyse linguistique pour extraire les lemmes. L'extraction des lemmes est composée des étapes suivantes :

- Extraction des mots composant les nœuds texte.
- Détermination du lemme et de sa catégorie grammaticale pour chacun des mots à l'aide d'un lemmatiseur / analyseur syntaxique.
- Filtrage des lemmes en fonction de leur catégorie grammaticale. Seul les noms et les verbes sont conservés. Les autres lemmes sont supprimés.

Le document est représenté par un ensemble de couples (<balise> : « mot »). <balise> correspond à l'étiquette du nœud parent du nœud texte dans l'arbre

résumé : c'est-à-dire le nom de la balise contenant le texte. Par exemple la représentation du document de la figure 3 contient les couples (<headline> : « report ») et (<p> : « predict »). Maintenant nous allons déterminer le poids de ces couples. La formule de pondération dépend de la profondeur du nœud dans l'arbre résumé.

3.3. Pondération

Nous supposons que l'importance des mots dépend entre autres de la balise dans laquelle ils apparaissent. Pour calculer le poids d'un couple (<balise> : « mot »), nous adaptons la formule TF-IEF de pondération des mots dans les nœuds des documents structurés. Nous supposons que plus un nœud est loin de la racine du document, moins il est représentatif du document. Par conséquent, les poids des couples sont calculés suivant la formule ci dessous:

$$w_{i,e,d} = TF_{i,e,d} * IDEF_{d,e} * IED_{e} \quad [1]$$

$$IDEF_{i,e} = \log \frac{|D_e|}{|D_{i,e}|} \quad [2]$$

$$IED_{e,d} = \log \left(\frac{L_d + 1}{L_{e,d}} \right) \quad [3]$$

Avec :

- $w_{i,e,d}$ représente le poids du mot t_i, apparaissant dans la balise *e*, du document *d*.
- $TF_{i,e,d}$ est le nombre d'occurrences du mot t_i dans la balise *e* du document *d.* Cette valeur représente l'importance locale du mot *i.*
- $IDEF_{i,e}$ représente l'importance globale du mot t_i. Cette valeur est calculée à l'aide des valeurs :
    - $|D_e|$ correspond au nombre de documents dans la collection D ayant une balise *e.*
    - $|D_{i,e}|$ correspond au nombre de documents dans la collection D ayant une balise *e* contenant le mot t_i.
- $IED_{e,d}$ représente l'importance locale de la balise *e*. Cette valeur est calculée à l'aide de :
    - L_d est la profondeur de l'arbre résumé du document *d*
    - $L_{e,d}$ est la profondeur de la balise *e* dans l'arbre résumé du document *d*.

Alors, le vecteur du document XML de la Figure 1 est :

$$d_d = (((< headline >: "compuserve") \mid 1{,}24), ((< p >: "compuserve") \mid 1{,}18), \ldots, ((< p >: "predict") \mid 0{,}83))$$

4. Expérimentations et résultats

Un prototype a été développé en Java pour construire nos vecteurs pondérés à partir de documents XML. Ce prototype utilise la plateforme GATE pour l'analyse linguistique des textes des documents. Le séparateur (tokenizer) proposé dans Gate a été associé à l'analyseur syntaxique (POS tagger) TreeTagger pour extraire les lemmes des nœuds textes. Une fois les vecteurs construits, le classifieur SVM^{light} proposé par (Joachims, 1999) a été utilisé pour l'apprentissage et la classification des documents. Une fois entrainé, SVM ^{light} propose pour chaque nouveau document une des classes préalablement apprises.

4.1. Collections

Les expérimentations ont été menées sur deux collections de documents XML. Une première expérimentation a été menée sur la collection d'articles journalistiques de Reuters intitulés Reuters Corpus 1 (RCV1). Cette collection comprend 800 000 articles courts en anglais. Les articles abordent différents thèmes comme l'information financière, les nouvelles technologies. Chaque article a été annoté manuellement par sujet, par région géographique et par secteur industriel. L'ensemble des documents de la collection suit le même schéma XML.

Pour la collection Reuters, l'expérimentation portait sur les articles catégorisés sous le sujet « GCAT » (Gouvernement/Social). Notre corpus a été construit en sélectionnant 800 articles : 400 articles appartenaient à la classe GCAT et 400 autres articles étaient annotés avec des sujets différents de GCAT. La liste des balises utilisée pour l'analyse structurelle de cette collection est : newsitem, headline, text, p. Cette liste a été construite manuellement après une analyse du contenu de quelques articles.

Une seconde expérimentation a été menée sur une collection INEX 2007 utilisé pour XML Document Mining Track (Denoyer, et al., 2008). Cette collection est composée de 96 000 documents extraits des représentations XML de Wikipédia. Dans cette collection, chaque document appartient exactement à une seule catégorie correspondant à un des portails thématiques de Wikipédia. La longueur et la structuration de ces documents sont beaucoup plus variées que dans le corpus Reuters. Notre corpus est construit en sélectionnant 200 documents dont 100 documents sont associés à la classe Portal: War et 100 autres à la classe Portal : Sexuality. La liste des balises utilisés pour l'analyse structurelle de cette collection est : name, figure, caption, table, section, paragraphe.

4.2. Expérimentations

Sur chacune des collections, deux séries d'expérimentations ont été menées :

- La première intitulée «texte et structure» est une implémentation de notre approche où un document est représenté par un vecteur pondéré de couples (<balise> : « mot »)
- La seconde intitulée «texte » utilise la représentation traditionnelle des documents sous forme de vecteur pondéré de lemmes avec un poids calculé par la formule TF-IDF.

Les performances de notre classifieur ont été évaluées par validation croisée à partir de la méthode intitulée « k-fold cross validation ». La validation croisée est une méthode d'estimation de fiabilité d'un modèle fondé sur une technique d'échantillonnage. Le corpus original est aléatoirement partitionné en k sous-ensembles. Le premier sous ensemble devient le corpus de test et les (k - 1) sous-ensembles constituent le corpus d'entrainement. Le classifieur est évalué sur ce premier échantillon. Ce processus de construction des corpus de test et d'entrainement est répété k fois en choisissant un autre des sous ensembles comme corpus de test. Le corpus d'entrainement est modifié en conséquence. Les résultats obtenus sur les k échantillons sont moyennés pour produire l'estimation finale. Dans nos expérimentations nous avons évalué notre système avec k = 4.

Le tableau 1 présente les résultats de nos expérimentations. Les mesures que nous avons utilisées sont la précision, le rappel et la F-mesure qui combine la précision et le rappel.

Les résultats montrent que la prise en compte de la structure a amélioré la classification uniquement pour la collection Reuters : amélioration de la précision et du rappel. Cette collection contient des documents courts à la structure homogène. Au contraire de la collection INEX, dont les documents ont une structure et une taille beaucoup plus variables, la prise en compte de la structure a détérioré le rappel sans modifier la précision. Ainsi nous devons donc modifier notre approche pour améliorer les résultats sur des collections plus hétérogènes.

Vecteur de caractéristiques	Collection	Précision	Rappel	F-measure
Contenu et structure	Reuters	0.96	0.92	0.94
Contenu seul		0.93	0.85	0.89
Contenu et structure	INEX	0.98	0.58	0.78
Contenu seul		0.98	0.65	0.81

Table 1. *Résultats de la classification*

5. Conclusion et perspectives

Dans cet article nous avons proposé une nouvelle représentation des documents XML tenant compte à la fois de la structure et du contenu textuel pour la classification supervisée. Cette représentation est un vecteur pondéré de couples (<balise> : « mot ») extrait de l'arbre résumé agrégeant le contenu textuel et simplifiant la structuration du document initial. Nous avons aussi proposé une nouvelle formule de pondération de ces couples en adaptant les formules TF-IDF et TF-IEF proposées dans la littérature.

Pour évaluer notre approche, plusieurs expérimentations ont été menées sur les collections Reuters et INEX. Ainsi nous avons pu constater que notre approche améliore les résultats de la classification uniquement sur la collection très homogène de documents.

Comme ces travaux sont menés dans le cadre d'une convention CIFRE avec la société Continew, nos perspectives visent à tester les limites de notre approche sur une collection de documents techniques issus de cette société. Continew assure le stockage et la sécurité de la documentation technique. Ces documents sont écrits par des auteurs différents utilisant des modèles de documents très variés.

6. Bibliographie

Aïtelhadj A., Mezghiche M., & Souam F., « Classification de Structures Arborescentes: Cas de Documents XML », *CORIA 2009*, p. 301-317.

Cortes C., Vapnik V., « Support-vector networks. », Machine learning, 1995, p. 273-297.

Dalamagas T., Cheng T., Winkel K.-J., Sellis T., « Clustering XML Documents Using Structural Summaries », *EDBT*, 2005, p. 547-556.

Despeyroux, T., Lechevallier, Y., Trousse, B., & Vercoustre, A.-M., « Expériences de classification d'une collection de documents XML de structure homogène », *EGC* 2005, p. 183-188.

Doucet A., Ahonen-Myka H., « Naive Clustering of a Large XML Document Collection », *INEX Workshop,* 2002, p. 81-87.

Denoyer L, Gallinari P. «Report on the XML mining track at INEX 2007 categorization and clustering of XML documents» ACM SIGIR Forum, vol. 42, no. 1, Jun. 2008 , p. 79-91.

Ghosh S., Mitra P., « Combining Content and Structure Similarity for XML Document », *ICPR*, 2008, p. 1-4.

Joachims T., « Making large-Scale SVM Learning Practical. Advances in Kernel Methods », *Support Vector Learning, B. Schölkopf and C. Burges and A. Smola (ed.), MIT-Press*, 1999, p. 169-184.

Kim, T.-S., Lee, J.-H., Song, J.-W., & Kim, D.-H., «Similarity Measurement of XML Documents Based on Structure and Contents», ICCS, 2007, p. 902-905.

Salton G., «Search and retrieval experiments in real-time information retrieval». *(C. University, Ed.)*, 1968, p. 1082-1093.

Sauvagnat K. « Modèle flexible pour la recherche d'information dans des corpus de documents semi-structurés ». Thèse de doctorat, Université Paul Sabatier, juin 2005.

Vercoustre A.-M., Fegas M., Lechevallier Y., Despeyroux T., « Classification de documents XML à partir d'une représentation linéaire des arbres de ces documents ». *EGC,* 2006, p. 443-457.

Wisniewski G., Denoyer L., Gallinari P., « Classification automatique de documents structurés. Application au corpus d'arbres étiquetés de type XML ». *CORIA*, *Grenoble*, 2005, p. 52-66.

Wu J., Tang J. (2008). « A bottom-up approach for XML documents classification ». *(ACM, Ed.) ACM International Conference Proceeding Series,* Vol. 299, 2008, p. 131-137.

Yan H., Jin D., Li L., Liu B., Hao Y., « Feature Matrix Extraction and Classification of XML Pages », *APWeb Workshops*, 2008, p. 210-219.

Yang J., Wang S., « Extended VSM for XML Document Classification Using Frequent Subtrees». *INEX*, 2010, p. 441-448.

Yang J., Zhang F., « XML Document Classification Using Extended VSM ». *INEX* 2007, 2008, p. 234–244.

Yi J., Sundaresan N., « A classifier for semi-structured documents ». KDD '00, 2000, p. 340-344.