Classification automatique de documents structurés. Application au corpus d'arbres étiquetés de type XML

Guillaume Wisniewski — Ludovic Denoyer — Patrick Gallinari

Laboratoire d'Informatique de Paris 6
8 rue du capitaine Scott
75015 Paris
{wisniewski, denoyer, gallinari}@poleia.lip6.fr

RÉSUMÉ. *Le domaine de la Recherche d'Information Structurée (RIS) est un domaine qui émerge avec l'arrivée de données semi structurées comme les documents XML. Ce domaine, à travers l'initiative INEX, concerne principalement le développement de moteurs de recherche documentaire. Aujourd'hui, il est nécessaire de développer des modèles pour le traitement de différentes problématiques dans les documents structurés comme la discrimination ou la restructuration. Dans cet article, nous nous intéressons à la classification automatique de documents XML en fonction de leur régularités structurelles. Nous proposons de modéliser la structure des documents XML par un réseau bayésien qui permet de prendre en compte différentes dépendances entres les unités structurelles du document. Nous présentons les résultats de nos différents modèles sur le corpus INEX et voyons ensuite comment un de nos modèles permet de déterminer un représentant de chacune des classes obtenues sous forme d'une DTD probabiliste.*

ABSTRACT. *The widespread use of XML has urged the need to develop tools to efficiently store, access and organize XML corpus. The INEX initiative has resulted in major improvements in XML retrieval systems, but today, related tasks, like categorization or structure matching, should be investigated. We consider here the problem of clustering XML documents using their structure. In this paper, we propose a Belief networks-based stochastic model which is able to describe different kind of relation between structural elements. We show how these models can be used for the clustering task. We test them both using the INEX corpus and an artificial corpus of XML documents.*

MOTS-CLÉS : *XML, Recherche d'Information Structurée, Classification automatique, Modèle statistique, apprentissage*

KEYWORDS: *XML, Structured Information Retrieval, Clustering, Stochastic Model, Machine Learning*

Introduction

Le développement du document électronique et du Web a vu émerger puis s'imposer des formats de données semi structurées, tels le XML et le XHTML. Ces nouveaux formats décrivent simultanément la structure logique des documents et le contenu de ceux-ci et permettent ainsi de représenter l'information sous une forme plus riche que le simple contenu. Celle-ci est adaptée à des besoins spécifiques qui permettent, par exemple, de faciliter l'accès à l'information ou d'optimiser le stockage et l'interrogation des documents. Avec l'augmentation rapide du nombre de documents semi structurés, il est nécessaire de concevoir de nouveaux modèles de Recherche d'Information (RI) capables de prendre en compte ce nouveau type de données, d'adapter les problématiques de la RI aux documents semi structurés et d'étudier les nouvelles problématiques que ces documents font émerger.

L'initiative INEX ([FUH 02]) étudie la problématique particulière de la recherche documentaire dans des grands corpus de documents XML. Les différents travaux menés dans le cadre cette initiative ont mis en évidence l'importance des problématiques connexes telles le traitement de données structurées hétérogènes. Une autre piste étudiée concerne la problématique de classification automatique qui, dans la RI traditionnelle, permet d'augmenter de manière significative la précision des moteurs de recherche. L'adaptation de cette problématique au traitement des données semi structurées, n'est pas triviale : doit-on considérer que deux documents sont proches lorsqu'ils possèdent des structures similaires, lorsque leur contenu est proche, ou seulement si leur structure et leur contenu sont proches ? La réponse à cette question dépend très fortement des applications considérées.

Les moteurs de recherche développés pour les documents XML sont dédiés au traitement de documents structurellement homogènes et font l'hypothèse que la structure des données est connue a priori par l'utilisateur. En particulier, les modèles développés dans le cadre d'INEX ne savent traiter que les documents dont la structure est régulière (étiquettes des nœuds identiques, même type d'informations...). Ainsi, il est important d'avoir à disposition des outils permettant de classer automatiquement de grands corpus afin de regrouper des documents de structure proche. C'est la problématique à laquelle nous nous intéressons dans cet article.

Étant donné la taille des corpus de documents semi structurés, il est important que les modèles de classification automatique soient capables de traiter de grandes masses de données. De plus, il serait particulièrement utile qu'un modèle de classification automatique puisse extraire une représentation de la structure d'un groupe de documents : celle-ci pourra, par exemple, être utilisée pour qu'un utilisateur puisse interagir avec chacun de ces groupes ou pour permettre le stockage de ce dernier dans une base de données « traditionnelle ».

Dans cet article, nous présentons un modèle génératif de documents semi structurés, décrivant simultanément la structure et le contenu de ces documents. Ce modèle général peut être adapté à différentes tâches. Il a notamment été utilisé avec succès pour la discrimination ([DEN 04a]) et pour la restructuration automatique de docu-

ments XML ([DEN 04b]). Nous nous intéressons ici à la classification automatique de documents structurés et, plus particulièrement, au regroupement des documents ayant une structure proche. Le formalisme des réseaux bayésiens, sur lequel repose notre modèle, permet la prise en compte de différentes relations entre les éléments structurels d'un document. Nous proposons de comparer différentes versions du modèle afin de mieux comprendre les relations pertinentes. Ces modèles sont testés sur la base de documents INEX. Nous voyons ensuite, de manière plus prospective, comment le modèle *grammaire* permet de détecter différentes sources d'information et de déterminer un représentant de chacune des classes. Ces expériences sont réalisées sur un corpus simulés de documents XML.

1. État de l'art

La classification de documents XML est une problématique relativement nouvelle qui n'a reçu que peu d'attention, notamment parce qu'il n'existe pas, à ce jour, de corpus pour cette tâche.

[TER 02] propose d'utiliser la notion d'arbres fréquents pour réaliser un clustering de structures. L'algorithme *TreeFinder* permet de trouver, dans une collection d'arbres étiquetés non ordonnés, les arbres qui sont inclus dans au moins ϵ arbres de la collection. En utilisant différentes définitions de l'inclusion (inclusion stricte, inclusion ne conservant pas l'ordre des nœuds...), les auteurs arrivent à utiliser plusieurs modélisations de la structure d'un document. L'algorithme proposé réalise la classification du corpus selon la structure des documents, et associe à chacune des classes une structure représentative. [DOU 02] propose une méthode de classification utilisant à la fois le contenu et la structure des documents. C'est, à notre connaissance, la seule approche existante utilisant les deux types d'informations. Cette méthode propose de représenter les documents dans un espace vectoriel puis d'utiliser une méthode classique de classification vectorielle (les *k-means*). Les vecteurs représentants les documents sont composés de deux parties décrivant, respectivement le contenu et la structure à l'aide, respectivement, d'un codage tf/idf des mots et des étiquettes. Mais les résultats sont loin d'être concluants et les auteurs pensent qu'ils n'arrivent pas à tirer pleinement parti des informations disponibles. [LEE 02] s'intéresse à une problématique proche de la notre : la classification, non pas de documents, mais de schémas et ne considère donc pas le même type de données que nous. D'autres travaux ont permis de développer des distances d'édition pour les arbres ([NIE 02]) qui pourraient être utilisées à l'aide d'algorithme classique pour la classification de structures. Toutefois, à cause de leur complexité, de telles méthodes ne sont pas adaptées au traitement de grande masse de données.

L'inférence de DTD à partir d'un ensemble de documents XML a généralement été abordée comme un problème d'inférence de structure d'arbre. Les algorithmes développés s'inspirent des techniques d'inférence grammaticale : leur objectif est de synthétiser dans un automate d'arbre l'ensemble des structures des documents du corpus. Une approche récente dans la communauté base de données ([PAP 00]) montre

les limites des DTD pour cette tâche et propose un nouveau langage de description des schémas plus adapté à l'extraction de DTD.

2. Modélisation de la structure d'un document

Nous avons développé un modèle génératif stochastique de documents semi structurés. Ce modèle repose sur le principe suivant : l'auteur va tout d'abord décrire *a priori* la structure (le plan) de son document puis « remplir » chacune de ces entités structurelles. Par exemple, pour la rédaction d'un article scientifique, l'auteur va décider qu'il doit y avoir un titre, un résumé, un certain nombre de sections composées de paragraphes, puis va rédiger le contenu de chacun des ces éléments. Selon la partie du document rédigée il ne va pas utiliser le même vocabulaire : la distribution du vocabulaire dépendra donc de l'entité structurelle.

Nous adoptons la représentation traditionnelle des documents semi structurés sous forme d'arbre ordonné. À chaque nœud du document correspond un nœuds de l'arbre. La figure 1 donne un exemple de représentation d'un document arborescent.

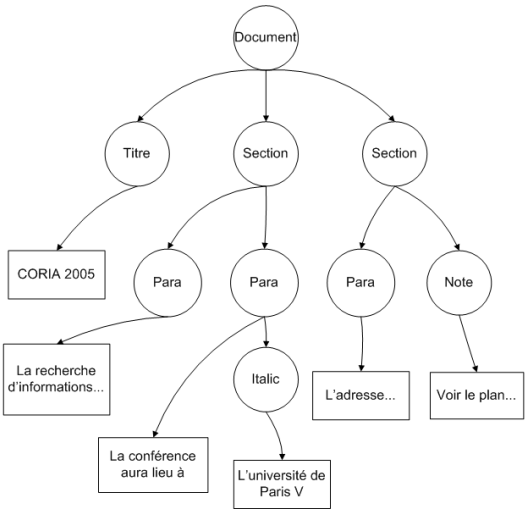


Figure 1. *Exemple de document semi structuré. Les informations de contenu apparaissent dans les rectangles ; les nœuds structurels sont représentés par des cercles dans lesquels apparaissent les étiquettes.*

Étant donné un document d, nous noterons $|d|$ le nombre de ses nœuds. Chaque nœud n_i est composé d'une étiquette s_i et d'un contenu t_i et correspond à une entité structurelle du document (un paragraphe, un titre...). Soit Λ l'ensemble des étiquettes possibles (i.e. : $s_i \in \Lambda$). Le processus génératif décrit précédemment correspond à la modélisation probabiliste suivante (en utilisant un modèle de paramètres θ) : la probabilité de générer un document structuré est le produit de la probabilité a priori de

la structure du document $P(s_1, ..., s_{|d|}|\theta)$ et de la probabilité du contenu du document connaissant sa structure $P(t_1, ..., t_{|d|}|s_1, ..., s_{|d|}, \theta)$. On a donc :

$$\begin{aligned} P(d|\theta) &= P(n_1, ..., n_{|d|}|\theta) \\ &= P\left((s_1, t_1), ..., (s_{|d|}, t_{|d|})|\theta\right) \\ &= P(s_1, ..., s_{|d|}|\theta) \cdot P(t_1, ..., t_{|d|}|s_1, ..., s_{|d|}, \theta) \end{aligned} \quad (1)$$

Ce modèle a été dérivé pour les tâches spécifiques de discrimination de documents structurés ([DEN 04a]) et de restructuration de documents XML ([DEN 04b]). Il a montré sa capacité, d'une part, à traiter ces problématiques avec succès et, d'autre part, à traiter, aussi bien en apprentissage qu'en inférence, une quantité de données importante. Dans cet article, nous nous intéressons à sa capacité à classer automatiquement des structures sans considérer le contenu des documents. La probabilité d'un document ne dépend alors que de sa structure. On a donc :

$$P(d|\theta) = P(s_1, ..., s_{|d|}|\theta) \quad (2)$$

Nous allons proposer différentes modélisations possibles des relations de dépendances entre les unités structurelles d'un document afin de déterminer les dépendances les plus intéressantes pour la classification automatique. Nous avons choisi de modéliser la structure par un réseau bayésien car ce formalisme permet de caractériser les dépendances conditionnelles entre variables aléatoires de manière flexible : à travers plusieurs topologies du réseau, nous allons pouvoir prendre en compte différents types d'informations structurelles. Cependant, il est nécessaire de faire un compromis entre l'expressivité du modèle et sa complexité, pour pouvoir traiter une grande quantité de données.

3. Différents modèles de structure

3.1. *Modèle général*

Soit $(s_1, ..., s_{|d|})$ l'ensemble des nœuds de structure d'un document d. On va considérer que la structure du document est modélisée par un réseau bayésien de N variables aléatoires $X_1, ..., X_N$. Les arcs du réseau seront modélisés par la fonction $pa(X_i)$ qui renvoie l'ensemble des parents de la variables X_i dans le réseau. Nous allons distinguer deux types de variables :

– les variables $S_1, ..., S_{|d|}$ qui correspondent à des nœuds du document modélisé. Ces variables seront à valeur dans Λ.

– les variables $Y_1, ..., Y_{N-|d|}$ permettant de modéliser des dépendances supplémentaires entre les nœuds du documents. Ces variables ont pour but de modéliser des relations plus fines entres les élements de structure du document.

Ainsi, l'ensemble des variables s'écrit $(X_1, ..., X_N) = (S_1, ..., S_{|d|}, Y_1, ..., Y_{N-|d|})$. Nous allons proposer deux familles de modèles :

– la première famille (paragraphe 3.2) correspond à des réseaux « simples » pour lesquels toutes les variables aléatoires correspondent à des entités structurelles du document (i.e. : $N = |d|$). Cette famille permet la modélisation de dépendances directes entre les éléments d'un document.

– la seconde famille (paragraphe 3.3) permet de décrire, à l'aide des variables $Y_1, ..., Y_{N-|d|}$ des dépendances supplémentaires.

3.2. *Modèles de structure de type 1*

On considère un ensemble de variables aléatoires $(S_1, ..., S_{|d|})$ associées à chacune des parties d'un document structuré. La probabilité structurelle du document modélisé par un réseau bayésien d'arcs $pa(S_i)$ est obtenue par le calcul de la probabilité jointe du réseau :

$$\begin{aligned} P(d|\theta) &= P(s_1, ..., s_{|d|}|\theta) = P\left(S_1 = s_1, ..., S_{|d|} = s_{|d|}|\theta\right) \\ &= \prod_{i=1}^{|d|} P(S_i = s_i|pa(S_i), \theta) \end{aligned} \quad (3)$$

La définition de la fonction $pa(S_i)$ permet de prendre en compte certaines relations structurelles.

3.2.1. *Modèle de type « Naive Bayes »*

Le modèle de type *Naïve Bayes* considère l'indépendance des unités structurelles d'un document. Il correspond à un modèle de réseau où la fonction $pa(S_i)$ est la fonction vide. La figure 2 donne le réseau construit pour un tel type de modèle. L'équation 3 se réécrit alors : $P(d|\theta) = \prod_{i=1}^{|d|} P(S_i = s_i|\theta)$. La probabilité $P(S_i = s_i|\theta)$ correspond alors à la probabilité qu'une partie s_i aparraisse dans le document — par exemple qu'il y ait un paragraphe dans un document. C'est un modèle simple de complexité faible, linéaire en fonction du nombre de nœuds du document.

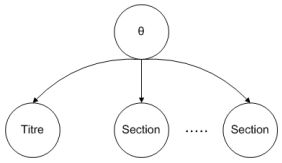


Figure 2. *Modélisation du document par un réseau bayésien de type Naïve Bayes*

3.2.2. *Modèle parent*

Le modèle *parent* vise à modéliser l'information d'inclusion entre les différentes entités structurelles d'un document. Il correspond à un réseau bayésien dans lequel

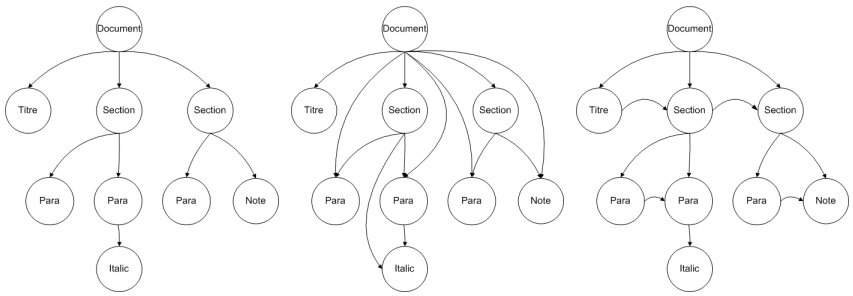
$pa(S_i) = S_j$ si et seulement si le i-ème nœud du document d est le fils de son j-ème nœud. La figure 3(a) donne le réseau correspondant à un tel type de modèle. La probabilité d'un document est alors :

$$P(d|\theta) = \prod_{i=1}^{|d|} P\left(S_i = s_i|pa(S_i)|\theta\right) = \prod_{i=1}^{|d|} P\left(s_i|pere(s_i), \theta\right) \quad (4)$$

où $pere(s_i)$ est la fonction qui revoie l'étiquette du père du nœud i dans le document. La probabilité $P\left(s_i|pere(s_i), \theta\right)$ correspond à la probabilité qu'un nœud d'étiquette s_i soit le fils d'un nœud possédant une étiquette $pere(s_i)$ — par exemple, la probabilité d'avoir un paragraphe dans une section.

3.2.3. *Autres modèles*

Dans la même famille de modèle, nous proposons un modèle *grand-père* qui correspond à la modélisation de descendance d'ordre 2 (figure 3(b)) et le modèle *père-frère* (figure 3(c)) qui correspond à la modélisation de la relation d'inclusion et de la relation de séquentialité : un nœud $paragraphe$ est dans une $section$ et apparaît après une $introduction$. Nous ne détaillons pas ces modèles qui ressemblent beaucoup au modèle de type parent.



(a) Modèle père (b) Modèle père-père (c) Modèle père-frère

Figure 3. *Les réseaux bayésiens décrivant les dépendances conditionnelles entre les différents nœuds du document (la dépendance avec les paramètres θ n'est pas représentée).*

3.3. *Modèle grammaire*

Nous proposons ici un modèle de structure plus complexe qui vise à modéliser la dépendance entre un nœud et l'ensemble de ses fils, contrairement aux modèles précédents qui établissent une dépendance entre chacun des nœuds et leur parent ou

voisin. Nous verrons plus loin que ce modèle permet d'extraire, sous forme d'une DTD probabiliste, un représentant structurel d'un corpus.

Cette modélisation de la structure est inspirée des grammaires probabilistes d'arbre ([CAR 02]). La structure de l'arbre est décrite par une grammaire de type CFG, associant à chaque nœud la liste ordonnée de ses fils. Ainsi, la règle de dérivation $document \rightarrow titre\ section\ section$ indique qu'un nœud d'étiquette $document$ aura trois enfants d'étiquette respective : $titre$, $section$ et $section$. Nous considérons ici le processus génératif dans lequel l'auteur d'un document structuré découpe un document en plusieurs parties étiquetées puis, récursivement, redécoupe chacune de ces parties en sous-parties. Le réseau bayésien correspondant à l'équation précédente est représenté à la figure 4 Soit $\mathrm{enfants}(n_i)$ l'ensemble ordonné des étiquettes des enfants d'un nœud du document. Nous associerons à chaque règle un nœud $Y_i = \mathrm{enfants}(n_i)$ dans le réseau bayésien décrivant le document. Deux types de probabilités seront alors à considérer :

– celles du type $P(Y_i|X_i, \theta)$ qui décrivent la probabilité que l'auteur utilise la règle $X_i \rightarrow Y_i$ pour découper le nœud X_i

– celles du type $P(X_i|Y_i, \theta)$ qui décrivent la probabilité que l'on trouve un nœud X_i sous un nœud Y_i

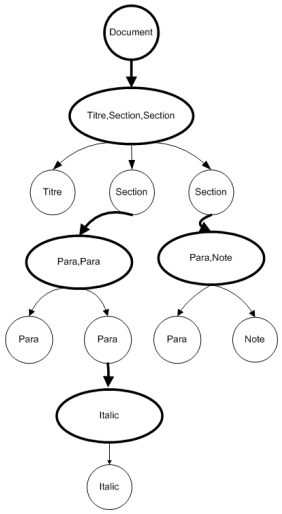


Figure 4. *Le réseau bayésien décrit le modèle grammaire. Les flèches épaisses correspondent aux probabilités $P\,(enfants(n_i)|s_i, \theta)$. Les flèches normales correspondent aux probabilités $P(Y_i|X_j, \theta)$ et ne sont qu'une réécriture permettant de faire le lien avec le document XML.*

Nous supposerons que pour, tout i, $P(X_i|Y_i, \theta) = 1$: le choix de la règle de réécriture détermine de manière certaine les nœuds se trouvant sous un nœud Y. La

probabilité structurelle d'un document calculée par un tel modèle stochastique s'exprime alors par : $P(s_1, ..., s_{|d|}|\theta) = \prod_{i=1}^{|d|} P\left(\text{enfants}(n_i)|s_i, \theta\right)$

4. Classification automatique

Nous avons présenté différents modèles génératifs de structure. Afin d'effectuer la classification automatique d'un corpus, nous avons choisi de modéliser un document structuré par un modèle de mélange.

4.1. *Modèle de mélange*

Soit $C = \{c_1, ..., c_k\}$ l'ensemble des classes dont le nombre k est supposé fixé et connu. Chaque classe est décrite par un ensemble de paramètres θ_{c_i} et correspond à une composante du modèle de mélange. On suppose que les documents sont générés par un mélange de densités à k composantes :

$$P(d|\theta) = \sum_{i=1}^{k} \alpha_{c_i} \cdot P(d|\theta_{c_i}) \tag{5}$$

où les α_{c_i} sont les proportions du mélange ($\sum_{i=1}^{k} \alpha_{c_i} = 1$) et $\theta = \bigcup_{i=1}^{k} \theta_{c_i}$. Les probabilités $P(d|\theta_{c_i})$ correspondent à la probabilité que le document ait été généré par le modèle de la composante c_i. Les paramètres de ce modèle de mélange vont être appris à l'aide de l'algorithme CEM qui est une variante « dure » de l'algorithme EM.

4.2. *L'algorithme CEM*

L'algorithme CEM correspond à un algorithme EM dans lequel une étape supplémentaire (étape C) a été introduite afin d'associer chaque document à une unique classe. Il sera utilisé à l'aide d'une initialisation aléatoire.

Étape E : Dans cette étape, on calcule, pour chaque document$d$ et pour chaque classe (i décrit $[1, k]$), la probabilité $P(d|\theta_{c_i})$ que le document d ait été généré par les paramètres de la classe c_i. Ces probabilités sont déterminées selon les modèles présentés au paragraphe 3.

Étape C : L'étape C permet d'attribuer chaque document du corpus à la classe qui maximise la probabilité $P(d|\theta_{c_i})$.

Étape M : Lors de l'étape M, les paramètres des différentes classes vont être mis à jour en fonction du résultat de la classification obtenue lors de l'étape précédente. Cette mise à jour est faite en maximisant la log-vraissemblance

$$\forall i \in [1...k], \theta_{c_i}^{t+1} = \underset{\theta_{c_i}}{\text{argmax}}\, L_i = \underset{\theta_{c_i}}{\text{argmax}} \left(\sum_{d \in D_i} \log P(d|\theta_{c_i}) \right) \tag{6}$$

Cette équation est résolue par une procédure d'apprentissage qui sera détaillée au paragraphe 4.3.
Durant cette étape, les paramètres α_{c_i} sont aussi mis à jour à l'aide de la relation : $\alpha_{c_i}^{t+1} = \frac{|D_i|}{|D|}$ où $|D|$ correspond au nombre de document dans le corpus et $|D_i|$ au nombre de documents dans la classe i à l'instant t.

4.3. *Paramètre des modèles*

Nous allons détailler ici l'apprentissage des modèles *parent* et *grammaire*. Les équations données dans le cas du modèle *parent* sont facilement généralisables à l'ensemble des modèles de structure de type 1. Le modèle *grammaire* nécessite une explication plus précise de l'estimation de ses paramètres et des coefficients de lissage que nous utilisons.

4.3.1. *Modèle parent*

L'apprentissage du modèle nécessite que l'on puisse estimer, pour l'ensemble des nœuds s_i du corpus d'apprentissage, la probabilité que son père soit $parent(s_i)$. Pour cela nous allons maximiser L_D le logarithme de la vraisemblance du modèle sur le corpus d'apprentissage D ($L_D = \log\left(\prod_{d\in D} P(d|\theta)\right)$), ce qui revient à résoudre l'équation :

$$\nabla_\theta L_D = \frac{\delta L}{\delta \theta} = 0 \tag{7}$$

sous les contraintes assurant que les sommes des différentes probabilités estimées soient égales à 1. En utilisant les multiplicateurs de Lagrange et un modèle de lissage, on obtient :

$$\forall n, m \in \Lambda \times \Lambda, P(m|n,\theta) = \frac{\sum_{d\in D} NS^d_{m,d}}{\sum_{d\in D}\sum_{m'\in\Lambda} NS^d_{m',n}} \tag{8}$$

où $NS^d_{m,n}$ correspond au nombre de fois où, dans le document d, un nœud de label n possède un enfant de label m.

4.3.2. *Modèle grammaire*

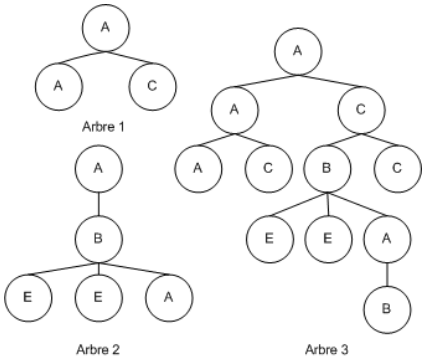
Pour le modèle *grammaire*, il est nécessaire d'estimer la probabilité de trouver un ensemble de nœuds ordonnés sous une étiquette donnée : $\forall n, m \in \Lambda \times \Lambda^*, P(m|n)$. Il est important de noter que m décrit a priori l'ensemble des séquences d'étiquettes possibles. Comme pour le modèle *père*, nous allons réaliser cette estimation par une maximisation de la log-vraisemblance L_D sur un ensemble d'apprentissage D. Notre corpus d'apprentissage étant de taille finie, nous ne pouvons considérer que les séquences d'étiquettes apparaissant dans celui-ci. On obtient alors :

$$\forall n, m \in \Lambda \times \Lambda^*, P(m|n,\theta) = \frac{\sum_{d\in D} NS^d_{m,n}}{\sum_{d\in D}\sum_{m'\in\Lambda^*} NS^d_{n',m}} \tag{9}$$

où $NS^d_{m,n}$ est le nombre de fois où un nœud de label n possède un ensemble d'enfants dont la séquence des étiquettes est m. Λ^* est un ensemble infini, mais nous nous limitons à l'ensemble des séquences apparaissant dans l'ensemble d'apprentissage. Comme dans tous les problèmes d'estimation, il est alors nécessaire de lisser ces estimations, pour éviter que des probabilités nulles ne faussent les algorithmes. Nous attribuerons donc de manière abusive une probabilité ϵ faible mais non nulle à toutes les règles qui n'ont pas été rencontrées dans l'ensemble d'apprentissage.

4.4. *Modèle grammaire et représentant d'une classe*

Le modèle *grammaire* trouve sa justification dans l'interprétation des paramètres qu'il permet d'apprendre. La figure 5 illustre un corpus fictif de documents arborescents et les paramètres appris par ce modèle.



$$P(A, C|A) = \frac{3}{5}$$
$$P(B|A) = \frac{2}{5}$$
$$P(E, E, A|B) = \frac{2}{2} = 1$$
$$P(B, C|C) = \frac{1}{1} = 1$$

Figure 5. *Exemple de paramètres appris par le modèle grammaire. La partie de gauche représente le corpus d'apprentissage, la partie droite les probabilités estimées par ce modèle.*

Lors d'une classification automatique, les paramètres de chacune des composantes du mélange constituent un « résumé » des documents attribués à cette composante. Dans le cas du modèle *grammaire*, l'interprétation des paramètres est relativement naturelle pour un utilisateur du format XML : ceux-ci permettent la reconstitution d'une DTD possible du groupe de documents. La DTD est un langage permettant de décrire les contraintes (hiérarchie des champs, paramètres, type de données...) imposées à la structure des documents XML. La figure 6 illustre le passage d'un corpus de documents à une DTD représentative de ce groupe de document.

Ainsi, le modèle *grammaire* permet d'obtenir une DTD représentative d'un corpus. Cette DTD peut même être vue comme une DTD probabiliste, puisqu'une probabilité est associée à chaque règle de la DTD. L'étude de ces probabilités permet de faire apparaître des régularités, qui peuvent, par exemple, caractériser la source d'information ayant crée les documents. Cette utilisation du modèle *grammaire* correspond aujourd'hui à une utilisation prospective : le domaine de l'inférence automatique de DTD

Paramètres du modèle	ECFG probabiliste	ECFG non probabiliste	DTD
$P(A, C\|A) = \frac{3}{5}$	$A \rightarrow AC[\frac{3}{5}]$	$A \rightarrow AC$	`<!DOCTYPE A [` `  <!ELEMENT A (A, C)>`
$P(B\|A) = \frac{2}{5}$	$A \rightarrow B[\frac{2}{5}]$	$A \rightarrow B$	`  <!ELEMENT A (B)>`
$P(E, E, A\|B) = 1$	$B \rightarrow EEA[1]$	$B \rightarrow EEA$	`  <!ELEMENT B (E, E, A)>`
$P(B, C\|C) = 1$	$C \rightarrow BC[1]$	$C \rightarrow BC$	`  <!ELEMENT C (B, C)>` `]>`

Figure 6. *Le processus permettant d'obtenir une DTD à partir de l'ensemble des documents de la figure 5. À partir des documents, on estime les paramètres du modèle grammaire, on en déduit une PECFG puis un ECFG et on peut alors reconstruire la DTD.*

est un domaine à part entière et notre modèle devrait être testé sur un corpus réel pour être vraiment validé. Nous souhaitons, ici, juste montrer que le formalisme général que nous proposons permet, selon les dépendances prises en compte, de construire naturellement une représentation de la structure de chacune des classes obtenues.

5. Expériences

Nous allons décrire dans ce paragraphe les différentes expériences que nous avons effectuées pour évaluer l'efficacité de notre méthode de classification automatique et, plus particulièrement, l'efficacité des différentes modélisations des structures de documents que nous avons proposées. Dans un premier temps, nous allons voir quelles sont les dépendances qui permettent une bonne classification automatique à l'aide d'expériences sur le corpus INEX. Puis nous verrons, à l'aide d'un corpus simulé, que le modèle *grammaire* est capable de détecter différentes sources d'informations.

5.1. *Qualité de la classification automatique*

De nombreuses manières d'évaluer un système de classification ont été proposées dans la littérature. [ZHA 01] présente une synthèse des différentes méthodes utilisées. Nous avons choisi d'utiliser comme mesure d'évaluation l'*entropie croisée* mesurée entre un étiquetage *a priori* du corpus et les classes trouvées. Cette mesure permet de caractériser la répartition des différentes étiquettes à l'intérieur des classes. Plus l'entropie est faible, meilleur est le système : une entropie nulle indique que tous les documents d'une classe ont la même étiquette. Nous utiliserons parallèlement la mesure de *pureté* qui mesure la proportion de documents d'une classe ayant l'étiquette

la plus fréquente. De manière générale, l'évaluation d'un système de classification automatique est un problème difficile et les mesures proposées ne permettent qu'une appréciation de la qualité de la classification.

Le corpus INEX a été rassemblé dans le cadre d'une campagne d'évaluation des moteurs de recherche XML et regroupe près de 12 000 articles scientifiques au format XML ce qui représente plus de 7 000 000 de nœuds. Ces documents proviennent de deux types de journaux différents : les journaux « IEEE Transaction on... » et les autres. Nous allons regarder comment notre méthode de classification de structures permet de séparer les documents provenant de ces deux sources.

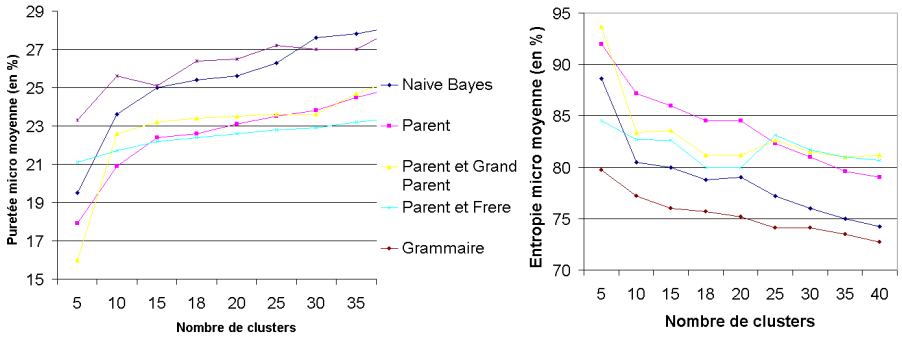


Figure 7. *Entropie et pureté des différents modèles de document sur le corpus INEX*

La figure 7 détaille les résultats obtenus sur le corpus INEX. On s'aperçoit que la prise en compte de relations relativement simples comme la relation d'inclusion (modèle *parent*) ou la relation de séquentialité (modèle *père-frère*) ne permettent pas d'obtenir, sur ce corpus, de meilleurs résultats que le modèle simple *Naïve Bayes*. Ce résultat contre intuitif provient notamment du fait que le corpus utilisé n'est pas adapté à la classification automatique. Cependant, les expériences proposées sont, à notre connaissance, les premières effectuées sur INEX, le seul corpus réel de grande taille de documents XML existant à l'heure actuelle. Par contre, le modèle *grammaire* montre de meilleures performances, notamment lorsque le nombre de classes est faible.

5.2. *Détection de sources d'information et reconstitution de DTD*

Nous avons effectué des expériences sur des données simulées. Ces expériences nous permettent, à l'aide de documents plus simples que ceux d'INEX, de mieux comprendre le comportement du modèle *grammaire* et, plus particulièrement, de mesurer sa capacité à construire une DTD représentative de chaque classe. Ce corpus comporte 3000 documents issus de trois DTD artificielles. Ces DTD ont été crées à partir des règles de dérivations de la figure 8.

Les résultats d'un point de vue de la qualité de la classification (figure 9) confirment la supériorité du modèle *grammaire* sur les autres modèles.

DTD 1	DTD 2	DTD 3
$a \rightarrow bc$	$a \rightarrow bcd$	$a \rightarrow aa$
$a \rightarrow cd$	$b \rightarrow cde$	
$c \rightarrow d$	$c \rightarrow de$	
$d \rightarrow e$	$d \rightarrow ab$	
$d \rightarrow a$		
	+	+
	DTD1	**DTD2**

Figure 8. *Les 3 DTD utilisées en simulation*

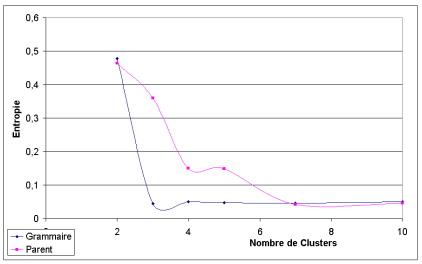


Figure 9. *Entropie des modèles grammmaire et père sur le corpus de données simulées*

Nous avons voulu connaître la qualité de reconstitutions des DTD sur la base de documents simulés. Nous avons donc « extrait » les paramètres du modèle *grammaire* les DTD probabilistes en suivant la procédure expliquée précédemment. Dans le cas à 3 classes, on constate que le modèle est capable de retrouver les 3 DTD originales bien que la première DTD soit incluse dans les deux autres. Ce modèle est donc capable de retrouver une source d'information spécifique parmi un flot de documents. Dans le cas à 5 classes (figure 10), les deux premières DTD retrouvées correspondent aux DTD 1 et 2 utilisées pour la génération de document. Les 3 dernières correspondent toutes à la DTD 3. Cependant, la mesure probabiliste associée à chaque règle de dérivation permet de distinguer des régularités structurelles spécifiques à certains groupes de documents. Par exemple, les documents de la troisième classe utilisent plus souvent la règle $c \rightarrow de$ que les documents de la classe 4.

DTD classe 1	DTD classe 2	DTD classe 3	DTD classe 4	DTD classe 5
$a \rightarrow a[0.21]$	$a \rightarrow bc[0.34]$	$a \rightarrow aa[0.20]$	$a \rightarrow aa[0.23]$	$a \rightarrow aa[0.20]$
$a \rightarrow bc[0.78]$	$a \rightarrow bcd[0.34]$	$a \rightarrow bc[0.24]$	$a \rightarrow bc[0.21]$	$a \rightarrow bc[0.20]$
$b \rightarrow cd[0.82]$	$b \rightarrow cd[0.33]$	$a \rightarrow bcd[0.21]$	$a \rightarrow bcd[0.23]$	$a \rightarrow bcd[0.21]$
$c \rightarrow d[0.84]$	$b \rightarrow cde[0.33]$	$b \rightarrow cd[0.34]$	$b \rightarrow cd[0.30]$	$b \rightarrow cd[0.32]$
$d \rightarrow e[0.43]$	$c \rightarrow d[0.33]$	$b \rightarrow cde[0.33]$	$b \rightarrow cde[0.31]$	$b \rightarrow cde[0.31]$
$d \rightarrow a[0.43]$	$c \rightarrow d[0.33]$	$c \rightarrow d[0.35]$	$c \rightarrow d[0.30]$	$c \rightarrow d[0.33]$
	$d \rightarrow e[0.23]$	$c \rightarrow de[0.35]$	$c \rightarrow de[0.30]$	$c \rightarrow de[0.31]$
	$d \rightarrow a[0.22]$	$d \rightarrow e[0.29]$	$d \rightarrow e[0.20]$	$d \rightarrow e[0.18]$
	$d \rightarrow ab[0.23]$	$d \rightarrow a[0.22]$	$d \rightarrow a[0.19]$	$d \rightarrow a[0.23]$
		$d \rightarrow ab[0.19]$	$d \rightarrow ab[0.22]$	$d \rightarrow ab[0.22]$

Figure 10. *DTD reconstruite par le modèle grammaire pour la classification à cinq classes.*

Conclusion

Nous avons proposé un modèle génératif de documents structurés qui a trouvé précédemment son application dans les domaines de la discrimination et de la restructuration de documents arborescents. Dans cet article, nous proposons un formalisme pour le calcul de la probabilité de la structure des documents arborescents. Ce formalisme basé sur les réseaux bayésiens est flexible et permet de spécifier différentes dépendances entre les unités structurelles des documents. Le modèle *grammaire* qui modélise les dépendances entre un nœud et l'ensemble de ces fils se révèle être le plus performant en classification automatique sur le corpus INEX. Des expériences prospectives sur un corpus simulé montrent comment ce modèle est capable de retrouver les différentes DTD apparaissant dans un corpus, sans toutefois réaliser une inférence de DTD La tâche de classification de structure est une tâche émergente dans la communauté de la Recherche d'Information Structurée et ce travail propose un ensemble d'expériences sur la modélisation statistique et l'utilisation d'un tel système sur une base réelle et de grande taille : la base INEX. Cependant, il reste à tester l'utilisabilité d'un tel système notamment s'il est couplé à un système de Recherche Documentaire afin d'en augmenter la précision.

6. Bibliographie

[CAR 02] CARRASCO R. C., RICO-JUAN J. R., « A similarity between probabilistic tree languages : application to XML document families », *Pattern Recognition*, , 2002.

[DEN 04a] DENOYER L., GALLINARI P., « Bayesian Network Model for Semi-Structured Document Classification », *Information Processing and Management*, , 2004.

[DEN 04b] DENOYER L., WISNIEWSKI G., GALLINARI P., « Document Structure Matching for heterogeneous corpora », *SIGIR 2004*, Workshop on IR and XML, Sheffield, 2004.

[DOU 02] DOUCET A., AHONEN-MYKA H., « Naïve clustering of a large XML document collection », *Proceedings of the First INEX Workshop.*, 2002, p. 81–87.

[FUH 02] FUHR N., GOVERT N., KAZAI G., LALMAS M., « INEX : Initiative for the Evaluation of XML Retrieval », *Proceedings ACM SIGIR 2002 Workshop on XML and Information Retrieval*, 2002.

[LEE 02] LEE M. L., YANG L. H., HSU W., YANG X., « XClust : Clustering XML Schemas for Effective Integration », *Proceedings of the eleventh international conference on Information and knowledge management*, 2002, p. 292–299.

[NIE 02] NIERMAN A., JAGADISH H. V., « Evaluating Structural Similarity in XML Documents », *Proceedings of WebDB 2002*, 2002.

[PAP 00] PAPAKONSTANTINOU Y., VIANU V., « DTD inference for views of XML data », *Proceedings of PODS'00*, 2000, p. 35–46.

[TER 02] TERMIER A., ROUSSET M.-C., SEBAG M., « TreeFinder : a First Step towards XML Data Mining », *Proceedings of ICDM'02*, 2002.

[ZHA 01] ZHAO Y., KARYPIS G., « Criterion functions for document clustering : Experiments and analysis », rapport, 2001, Department of Computer Science, University of Minnesota, Minneapolis.