
Traitements Automatiques pour la Migration de Documents Numériques vers XML

Jérôme Fuselier^{*,**} — Boris Chidlovskii^{*}

**Xerox Research Centre Europe,
6, chemin de Maupertuis, 38240 Meylan, France
jerome.fuselier@xrce.xerox.com, boris.chidlovskii@xrce.xerox.com*

***Université de Savoie - Laboratoire SysCom,
Domaine Universitaire, 73376 Le Bourget-du-Lac, France*

RÉSUMÉ. De plus en plus de sociétés migrent leur système de gestion de fond documentaire vers le format XML, le standard industriel pour l'échange de données. Afin de réduire les coûts de la migration, nous proposons une approche pour réaliser des conversions de documents orientés présentation vers des documents sémantiques. L'intérêt de notre méthode consiste à automatiser le processus de conversion en utilisant des techniques d'apprentissage supervisé pour apprendre un modèle de conversion pour une collection de documents. Nous décomposons la conversion en deux étapes pour simplifier le problème, une première étape d'annotation sémantique et une seconde étape de structuration sémantique du document qui respecte le schéma XML décrivant la classe des documents finaux.

ABSTRACT. More and more companies are migrating their legacy document management systems toward XML format, the industrial standard for data exchange. In order to reduce the migration cost we propose an approach aimed at automating the conversion of layout-oriented documents to semantic-oriented annotations. The conversion module uses supervised machine learning techniques to learn a conversion model for a collection of documents. The conversion is achieved through a semantic annotation of the document content and structuring the annotations, accordingly to a XML schema that specify the class of target documents.

MOTS-CLÉS : Apprentissage supervisé, Extraction d'informations, XML.

KEYWORDS: Machine Learning, Information Extraction, XML.

1. Introduction

Le formalisme XML est devenu le standard industriel pour l'échange de données entre services et entre entreprises. Utiliser XML comme format d'échange pour la capture et la réutilisation d'informations est devenu critique pour les entreprises. Ce formalisme offre de nouvelles possibilités dans le domaine de la gestion documentaire, de la publication ou du multimédia. Sa simplicité permet de définir un vocabulaire et une syntaxe adaptés aux données et facilite leur échange et la réutilisation du contenu. Les technologies construites autour de lui offrent de nouvelles fonctionnalités, les recherches peuvent par exemple devenir plus significatives grâce à un balisage sémantique très précis sur les parties importantes du document. Il est également possible d'intégrer des données en provenance de sources diverses et la notion de document est redéfinie, un document est ainsi vu comme une aggrégation d'informations sémantiques et non plus comme un document monolithique. Ainsi modularisés, les mises à jour de ces documents sont facilitées. L'avantage pour les entreprises est donc important mais le volume de documents à migrer vers ce nouveau formalisme crée de nombreux problèmes pour la conversion de fonds documentaires.

Les fonds documentaires dans les entreprises sont constitués d'une collection de documents très variés comme par exemple des documentations techniques, des manuels utilisateurs, des rapports internes, des publications, des factures, etc.. Ces documents sont souvent disponibles en formats électroniques, dans des formats privilégiant la présentation comme XHTML, PDF ou Microsoft Word. Ils décrivent correctement comment le document doit être présenté mais ne décrivent pas ce qui compose effectivement le document ni comment il est organisé. A l'opposé, en utilisant l'extensibilité du langage XML, il est possible d'annoter sémantiquement le contenu des documents (titres, auteurs, références, etc.), en laissant la tâche de présentation à des composants spécialisés. Le document est alors indépendant de la présentation et il est très facile d'adapter le rendu du document en fonction des périphériques utilisés pour la visualisation (assistants personnels, écrans larges, etc.). Nous nous intéressons à la découverte d'un processus automatique pour effectuer la migration du fond documentaire original vers XML.

Le processus de conversion nécessite souvent la définition d'un modèle de document cible exprimé par une grammaire XML. Cette grammaire peut être représentée par exemple sous la forme d'un XML Schema, d'une DTD ou d'un schéma RelaxNG. Elle définit les éléments structurels et sémantiques des documents propres à l'entreprise et à l'utilisation souhaitée. Il est fréquent que dans la conversion le document cible préserve une part importante du contenu du document source mais qu'elle supprime toutes les informations relatives à la présentation du document comme la pagination, le format des titres, etc.. Structurellement, les documents source et cible sont souvent très différents car ils suivent deux paradigmes opposés, nous parlerons par la suite de l'annotation orientée présentation d'un document en opposition à l'annotation orientée sémantique du même document. Par exemple, si nous considérons le fragment de contenu "*William Shakespeare*", son annotation orientée présentation sera "**<i>William Shakespeare</i>**" alors que son annotation sémantique

sera "*<auteur>William Shakespeare</auteur>*". La conversion de fonds documentaires vers le formalisme XML est référencée comme une transformation de documents semi-structurés.

Actuellement, la conversion de fonds documentaires dans un formalisme XML sémantique est réalisée par des experts du domaine et reste essentiellement manuelle et très coûteuse. Les communautés Web et XML offrent plusieurs outils pour transformer les données en XML, comme par exemple XSLT ou XQuery. Cependant, l'écriture de règles de transformation précises pour la conversion de gros volumes de données semble difficile voire impossible à cause de la taille et de la complexité des documents d'entrées et des schémas de sortie. L'état de l'art actuel dans le domaine de l'annotation sémantique ne laisse pas beaucoup d'espoir pour la création de convertisseurs entièrement automatiques. Néanmoins, nous cherchons à réduire le coût de la conversion en automatisant au maximum la transformation.

Ce papier présente un travail qui s'inscrit dans le projet "Legacy Document Conversion" (LegDoC) du Centre de Recherche Européen de Xerox (Chanod *et al.*, 2005). Il a pour objectif l'automatisation de la migration en masse de fonds documentaires vers XML. Un cas typique de conversion commence avec des documents disponibles en PDF, Postscript ou Microsoft Word, et un schéma pour les documents cibles, défini sous la forme d'une DTD ou d'un XML Schema. Le but de la conversion est de migrer les documents sources vers des fichiers XML conformes à un schéma cible. Nous nous intéressons plus particulièrement aux techniques d'apprentissage supervisées appliquées à la conversion documentaire. Nous supposons qu'il existe une collection d'apprentissage qui fournit des exemples de transformations, composée des documents sources et de leurs annotations en XML. Cela suppose la présence d'un expert qui est capable d'extraire manuellement les informations sémantiques des documents d'origine et de les porter vers un document structuré selon les contraintes imposées par la grammaire cible. Chaque paire de l'ensemble d'apprentissage (*document source, document cible*) est utilisé pour mettre au point un modèle de transformation reproductible qui pourra être appliqué par la suite sur l'ensemble de la collection à convertir. Un des buts visés est de réduire le travail de l'expert et d'automatiser la conversion en utilisant un nombre minimum d'exemples. De plus, nous travaillons avec des méthodes probabilistes pour améliorer la robustesse et les performances des modèles. Cette approche nous permet par exemple de gérer les incohérences de la collection qui peuvent être introduites par différents auteurs, ce bruit dans les données ne pourrait pas être capturé par des méthodes déterministes et fournirait des résultats incohérents.

Le reste de ce papier présente l'approche que nous avons mis au point pour réaliser cette conversion. La section 2 présente une vue d'ensemble du processus de conversion. La section 3 présente la décomposition du problème en deux sous-problèmes plus simples et notre apport dans la conversion de fonds documentaires. La section 4 décrit le processus d'évaluation de la conversion ainsi que les résultats des expériences que nous avons menés. La section 5 présente les approches alternatives et la section 6 conclut le papier.

2. La conversion documentaire

La vue générique du diagramme de conversion est présentée sur la Figure 1. Sur cette figure, nous pouvons distinguer trois grandes vues possibles pour des documents structurés en XML. La première se réfère aux annotations de présentation qui gèrent le rendu final des éléments du document (les positions x et y, la hauteur, la largeur, la fonte, etc.). La seconde permet de représenter des documents de façon plus abstraite, en définissant la structure logique du document. Elle décrit les relations spatiales entre les éléments de la page comme les colonnes, les paragraphes ou les lignes, etc.. Enfin, la dernière vue que nous considérons est la structuration sémantique du document. Elle ne considère que le sens des éléments et non la façon de les présenter sur la page.

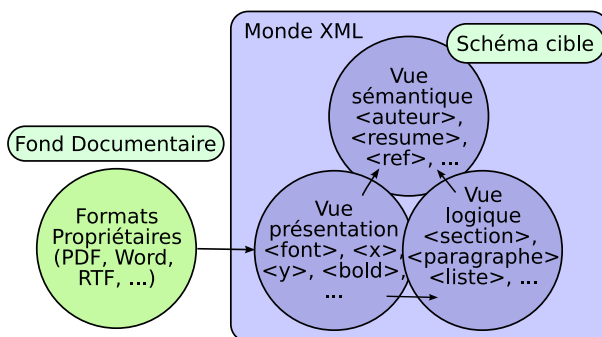


Figure 1. Les trois vues d'un document structuré.

Pour réaliser la conversion de fonds documentaires vers XML, le projet LegDoC définit un ensemble de composants qui forment une chaîne de traitement. En entrée de la chaîne se situe un document dans un format propriétaire et en sortie nous obtenons un document sémantique conforme à une grammaire. Voici les principaux composants :

– **Convertisseur bas-niveau** : La conversion commence par une réécriture du document écrit dans un format propriétaire vers un document XML "brut". Nous utilisons ici les convertisseurs disponibles pour différents formats comme Adobe PDF¹ ou Microsoft Word². Les documents produits par ces convertisseurs sont des fichiers XML qui préservent les informations de présentation du document (positions x et y, fontes, etc.). Ces logiciels sont très limités pour la reconnaissance des annotations logiques ou sémantiques et ne réalisent pas la conversion souhaitée.

– **Prétraitement** : Afin de corriger la sortie des convertisseurs précédents, nous utilisons un composant qui nettoie et indexe le fichier XML "brut". L'indexage des différents éléments du document nous garantit la persistance des informations initiales et nous permet de tracer les résultats tout au long de la chaîne de traitement.

1. <http://pdftohtml.sourceforge.net>

2. <http://www.turnkey.com.au/tksweb/products/xice.html>

– *Analyse logique* : Ce composant propose des méthodes pour améliorer la qualité du document produit par le convertisseur de bas-niveau en ajoutant des balises logiques et en assurant des propriétés spécifiques au document comme l'ordre de lecture. La qualité de ce document intermédiaire est une aide importante pour le composant d'annotation sémantique et améliore la qualité de la reconnaissance.

– *Annotation sémantique* : En utilisant le document produit par le module d'analyse logique, ce module extrait les informations sémantiques de manière automatique dans le but d'effectuer la conversion du document numérique.

La chaîne de traitement de la conversion documentaire est séquentielle. Chaque composant de la chaîne permet d'améliorer la qualité du document en cours de transformation en se rapprochant incrémentalement de son annotation finale. En particulier, le module d'analyse logique permet d'inférer de nouvelles connaissances qui sont un guide précieux pour le module d'annotation sémantique. Le reste de ce papier s'intéresse plus particulièrement à ce dernier composant.

3. L'annotation sémantique

3.1. Le problème de conversion

Le problème que nous cherchons à résoudre est l'annotation de documents sémantiques guidée par un schéma cible. Nous disposons en entrée d'un schéma cible fourni par un expert du domaine qui décrit la sémantique et la structure des documents, d'une collection de documents qui sont destinés à être visualisés et d'un sous-ensemble annoté de cette collection pour pouvoir apprendre un schéma de conversion reproductible. Ce que nous cherchons en sortie est un modèle de conversion qui pourra être appliqué à l'ensemble des documents orientés présentation pour créer des documents sémantiques appartenant au langage défini par la grammaire fournie.

Comme nous l'avons dit précédemment, le module d'apprentissage repose sur un format de document pivot, produit par un ensemble de composant du projet LegDoC. Ces documents transformés sont les documents d'entrées du module d'apprentissage. Le premier intérêt de ce format pour l'apprentissage est la remise en ordre des éléments du document. Les convertisseurs de bas-niveau traitent les éléments dans l'ordre défini dans la représentation interne du document propriétaire. Cet ordre ne correspond pas forcément à l'ordre des éléments du document cible. Nous utilisons pour le projet une approche basée sur la géométrie des éléments qui s'est montrée très performante sur nos classes de documents (Meunier, 2005). Le deuxième intérêt de ce format est la structuration logique du document d'entrée qui permet d'ajouter des informations utiles pour le module d'apprentissage. Après le premier traitement, nous avons perdu ces informations et le document est devenu une liste de feuilles, avec une profondeur très faible pour la structure. En utilisant la table des matières des documents, le module peut structurer le document automatiquement.

La Figure 2 présente un exemple de conversion avec à gauche le fichier d'origine et à droite le document sémantique qu'il faut obtenir avec le schéma XML fourni. Ce schéma est représenté sous la forme d'une grammaire BNF. Le fichier d'origine est en XHTML, le module d'annotation sémantique prend en entrée un document structuré (XML) ou semi-structuré (XHTML) sur lequel aucune hypothèse n'est faite. Pour cette raison, il peut utiliser un fichier XML provenant de la chaîne de traitement ou bien n'importe quelle autre document en XML. Le document est un fragment du CV d'un étudiant qui définit ses domaines de compétences et les études qu'il a suivies. Le contenu du document est présenté de manière à faciliter la compréhension des informations par des humains, il respecte une certaine nomenclature définie par un modèle de CV fourni par Microsoft Office. Le document cible est un fragment XML qui ne possède que les informations sémantiques du CV et qui respecte la grammaire. Toutes les informations de visualisation ont disparues pour ne conserver que les informations de contenu. Nous pouvons remarquer que certains fragments de contenu ne servent qu'à améliorer la visibilité des informations, ils ne conservent aucune information sémantique et devront être supprimés lors de la conversion.

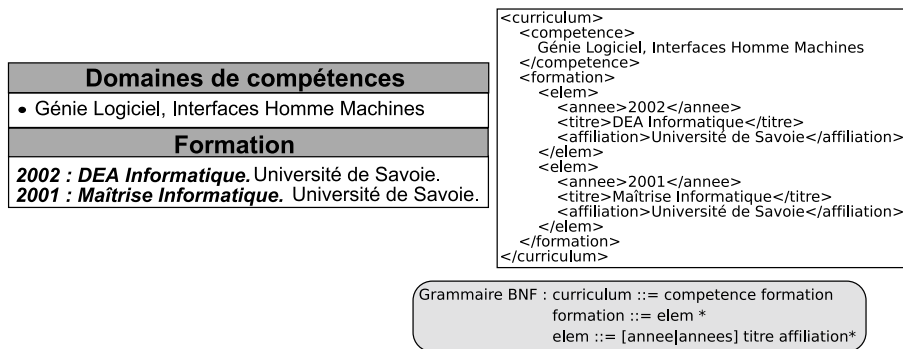


Figure 2. *Un exemple de conversion.*

La Figure 3 présente le même exemple sous un autre point de vue avec la représentation interne des documents sous forme arborescente. L'arbre du haut est constitué de balises de présentation (b, span, etc.) qui vont être interprétées par un navigateur pour afficher le contenu du document. Nous pouvons remarquer que le modèle appliqué pour générer le CV utilise des tableaux pour structurer la présentation. Cette information pourra être utilisée par des modèles d'apprentissage comme MaxEnt (Berger *et al.*, 1996) ou SVM (Schölkopf, 2000) pour cibler efficacement le positionnement des informations pertinentes dans la structure. A l'opposé, l'arbre situé en dessous est uniquement constitué des balises sémantiques propres au domaine, les informations de présentation ont disparues.

Dans ce papier, nous considérons le cas général de l'annotation arborescente d'un document semi-structuré. Contrairement aux approches existantes (Chung *et al.*, 2002, Ishitani, 2003, Altamura *et al.*, 2001), nous ne faisons pas d'hypothèses sur la

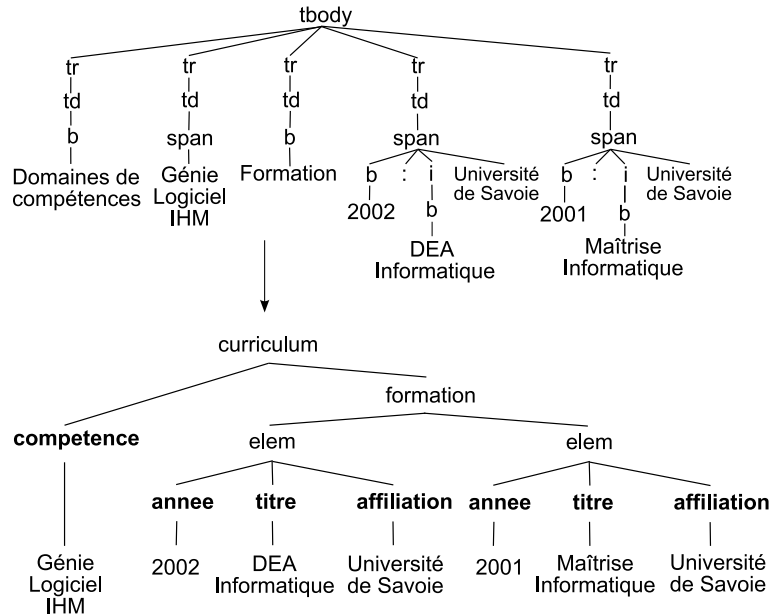


Figure 3. Représentation arborescente de l'exemple.

structure des documents cible et d'origine et nous ne recherchons pas des similarités de structure. Le contenu du document est représenté par une séquence d'observation $\mathbf{x} = (x_1, \dots, x_n)$, où chaque observation x_i est un fragment de contenu à convertir. Dans le cas de documents XHTML, les fragments sont les feuilles de l'arbre, elles sont entourées d'informations contextuelles sous la forme d'attributs, de balises, etc. Nous pouvons remarquer qu'il existe un alignement entre les feuilles des deux documents.

Notre approche consiste à diviser le problème en deux sous-problèmes plus simples. La première étape consiste à parcourir la séquence de feuilles du document d'entrée pour estimer une séquence de classes associées à ces feuilles. Les classes possibles sont choisies parmi les éléments fils des arbres de la grammaire. Pour l'exemple précédent, les classes possibles sont en gras, ce sont *competence*, *annee*, *titre* et *affiliation*. La séquence de classes estimée $\mathbf{y}_{\text{est}} = (y_1, \dots, y_n)$ représente les estimations y_i pour chaque observation x_i . Elle est utilisée comme point d'entrée pour le deuxième traitement, il a pour but la reconstruction d'un arbre de dérivation d associé à la séquence \mathbf{y}_{est} en utilisant les contraintes grammaticales fournies par le schéma ou bien dérivées de la collection. Cette arbre de dérivation correspond à l'arbre sémantique recherché. La réalisation de ces deux étapes correspond à la conversion d'un document orienté présentation vers un document sémantique.

3.2. Classification probabiliste

La première étape définie par notre décomposition du problème est une étape de classification. A partir d'une séquence de feuilles \mathbf{x} , l'annotation consiste à estimer la séquence de classes \mathbf{y} qui est la plus probable. Cette estimation est basée sur un modèle d'apprentissage entraîné avec un ensemble d'apprentissage, $\{(\mathbf{x}, \mathbf{y})\}$. En reprenant notre exemple, nous avons $\mathbf{x} = (\text{"Domaines de compétences", "Génie Logiciel, IHM", "Formation", "2002", ":", "DEA Informatique", "Université de Savoie", "2001", ":", "Maîtrise Informatique", "Université de Savoie"})$ et la séquence la plus probable recherchée $\mathbf{y} = (\text{REMOVE, compétence, REMOVE, année, REMOVE, titre, affiliation, année, REMOVE, titre, affiliation})$. Nous pouvons remarquer l'introduction d'une nouvelle classe *REMOVE* qui correspond à l'annotation des feuilles qui ne sont pas présentes sous cette forme dans l'annotation cible. Ce sont des feuilles qui conservent des informations utiles pour la présentation du document ("*:*") ou pour la sémantique des feuilles voisines ("*Formation*"). Ces informations sont en général présentes de façon implicite dans le document sémantique et peuvent être régénérées pour produire une visualisation du document.

Un classifieur probabiliste crée un modèle d'apprentissage cohérent avec les exemples annotés par un expert qui est le plus général possible, c'est-à-dire performant sur des données non vues en apprentissage. Le but recherché est de déterminer des valeurs de caractéristiques propres à chaque classe pour déterminer, à partir d'un x_i , les probabilités de chacune des classes. Formellement, un classifieur probabiliste cherche à estimer la probabilité conditionnelle $p(y|x_i)$ qui définit la probabilité que la classe de l'observation x_i (la feuille) soit y . Si le classifieur probabiliste est utilisé seul, la classe estimée est la classe pour laquelle cette probabilité est maximale.

Pour être utilisées avec des méthodes d'apprentissage existantes, les instances à classer (les observations x_i) doivent être projetées dans un modèle des données consistant. Dans ce modèle, une instance va être vue comme un vecteur de caractéristiques permettant de décrire le plus fidèlement possible les spécificités des instances d'une même classe. Nous classons ces caractéristiques suivants trois catégories différentes qui récupèrent des informations utiles pour la discrimination :

1) *Les attributs de contenu*

La première source d'attributs que nous pouvons utiliser concerne les fragments textuels du document d'origine, les feuilles de l'arbre. Ces attributs permettent de décrire précisément les caractéristiques spécifiques aux chaînes de caractères contenues dans les feuilles. Nous pouvons penser par exemple au nombre de caractères de la chaîne, à la présence de caractères spéciaux ou encore au caractère numérique de la chaîne. Dans l'exemple précédent, le fait de savoir que la chaîne "2002" est numérique peut aider le classifieur à proposer une plus forte probabilité pour la classe nommée "année".

2) *Les attributs de structure*

La deuxième source d'attributs qui est à notre disposition concerne la structure de l'arbre XHTML à convertir. Il peut être judicieux de connaître les balises proches d'une feuille dont nous cherchons à estimer la classe pour trouver des motifs structurels propres à chaque classe. Dans notre cas, les attributs sont à valeurs discrètes et permettent seulement de simuler les structures. Cependant, cette approche permet d'utiliser simplement la majorité des méthodes existantes et de ne pas avoir à mettre au point des méthodes plus spécifiques. Comme nous l'avons dit précédemment, la structure de tableau de l'exemple est une source d'informations structurelles pertinente. Les feuilles situées dans la première colonne par exemple peuvent être spécifiées par la description suivante : le père de la feuille est l'élément "td", il n'a pas de frère gauche et le grand père de la feuille est l'élément "tr".

3) *Les attributs de contenu XML*

Enfin, la dernière source d'attributs est un mélange de structure et de contenu. Il s'agit des valeurs des attributs présents dans les éléments XML qui entourent la feuille. Par exemple, il peut être intéressant de savoir que la fonte du père de la feuille est "times" ou encore que le tableau est dessiné avec une bordure de deux pixels d'épaisseur.

En utilisant cette représentation, une feuille peut être projetée dans ce modèle et le classifieur probabiliste travaille alors sur cette représentation simplifiée. Les attributs extraits dont nous disposons sont de types hétérogènes (discrets, numériques ou booléens) mais principalement à valeurs discrètes. Nous avons donc porté notre attention sur un classifieur basé sur le principe du maximum d'entropie (Berger *et al.*, 1996), appelé aussi MaxEnt. Il est très performant dans le domaine du traitement du langage pour résoudre des problèmes similaires et il s'est également montré le plus performant lors de nos comparaisons de classifieurs. Il permet notamment de gérer l'hétérogénéité des attributs, un grand nombre de classes et l'apprentissage du modèle d'apprentissage est très rapide. Ce classifieur cherche à maximiser la probabilité conditionnelle $P(y|x)$, il fait l'hypothèse qu'elle suit une loi exponentielle

$$P(y|x) = \frac{1}{Z_\alpha(x)} \exp\left(\sum_\alpha \lambda_\alpha \cdot f_\alpha(x, y)\right) \quad [1]$$

où $Z_\alpha(x)$ est un facteur de normalisation qui permet d'assurer que la valeur obtenue est une probabilité

$$Z_\alpha(x) = \sum_y \exp\left(\sum_\alpha \lambda_\alpha \cdot f_\alpha(x, y)\right). \quad [2]$$

La variable α permet d'effectuer une somme sur l'ensemble des attributs choisis pour représenter le contexte d'une feuille x et la fonction $f_\alpha(x, y)$ représente la valeur

de cet attribut α , pour le couple d'apprentissage (x, y) . Les valeurs λ_α représentent une pondération des attributs et permettent de déterminer un modèle pour lequel la distribution définie soit la plus exacte possible pour les données de l'ensemble d'apprentissage. Pour chaque choix de $\lambda = (\lambda_{\alpha_1}, \dots, \lambda_{\alpha_m})$ que nous pouvons faire, nous définissons donc un modèle différent, le classifieur MaxEnt va déterminer parmi toutes ces possibilités le modèle optimal, en utilisant le principe de maximum d'entropie. Ce principe privilégie les modèles les plus uniformes et permet de trouver un maximum local. Pour l'estimation itérative des paramètres λ_α du modèle, nous utilisons la méthode quasi Newton (Malouf, 2002).

3.3. Grammaires hors-contextes probabilistes

Pour utiliser les contraintes grammaticales fournies par la grammaire XML qui définit la sémantique métier de la collection, nous utilisons des grammaires probabilistes. La partie des schémas XML W3C (ou DTD) qui nous intéresse concerne uniquement les déclarations qui définissent la structure des arbres recherchés. Cette partie peut être transformée de manière équivalente vers le formalisme des grammaires hors-contextes (Papakonstantinou *et al.*, 2000). Il est également possible d'inférer une DTD probabiliste à partir d'une collection de documents XML annotés (Winkler *et al.*, 2002).

Définition : Une grammaire hors-contexte probabiliste G est définie par un 5-uplet $\langle N, T, R, S, P \rangle$ où :

- N est l'ensemble des symboles non terminaux,
- T est l'ensemble des symboles terminaux,
- R est l'ensemble des règles r_i de la forme : $A \rightarrow \alpha$, $A \in N$, $\alpha \in (N \cup T)^*$,
- S est l'axiome de départ,
- P est l'ensemble des probabilités p_i associées aux règles r_i telles que :

$$\sum_{\alpha} p(A \rightarrow \alpha) = 1, \forall A \in N. \quad [3]$$

Définition : On dit que $A \in N$ domine une chaîne $\mathbf{y} = (y_1, \dots, y_n)$ si $A \xrightarrow{*} \mathbf{y}$. Nous l'écrivons A_i^j .

Dans notre cas, nous cherchons à trouver une séquence \mathbf{y} qui soit dominée par S , l'axiome de départ qui est aussi la racine de l'arbre. La suite de règles de production $S \xrightarrow{*} \mathbf{y}$ utilisée pour produire la séquence définit un arbre de dérivation d qui est équivalent à un arbre XML. La Figure 4 schématise la domination de la racine par rapport à une séquence de classes $\mathbf{y} = (y_1, \dots, y_n)$.

Les règles de la grammaire hors-contexte probabiliste peuvent être écrites manuellement en se référant à la grammaire XML. Elles peuvent également être inférées

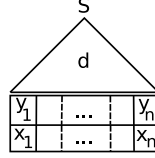


Figure 4. Dérivation de l'arbre.

automatiquement à partir des documents cible de la collection d'apprentissage. Les probabilités de chaque règles peuvent être calculées automatiquement en les dénombrant, en utilisant la formule suivante :

$$P(A \rightarrow \alpha) = \frac{\text{nombre}(A \rightarrow \alpha)}{\sum_{A \rightarrow \beta \in R} \text{nombre}(A \rightarrow \beta)}. \quad [4]$$

L'utilisation de grammaires probabilistes permet de proposer plusieurs arbres de dérivations pour une séquence d'éléments terminaux et donc d'introduire la notion d'arbre le plus probable qui correspond à l'arbre qui possède la plus grande probabilité. La probabilité d'un arbre de dérivation est calculée en effectuant le produit des probabilités des règles de production qui ont été utilisées pour le créer.

3.4. Combinaison de méthodes

La spécificité de nos travaux consiste à combiner les deux technologies précédentes, un classifieur probabiliste et des grammaires probabilistes, pour réussir à convertir automatiquement des documents semi-structurés (Chidlovskii *et al.*, 2005). Formellement, nous cherchons à trouver le couple (y, d) qui soit le plus probable étant donné une grammaire probabiliste G et un document d'origine projeté sous une forme vectorielle d'attributs x . Cela peut s'écrire :

$$(\mathbf{y}, d)_{max} = \underset{(\mathbf{y}, d)}{\operatorname{argmax}} P(\mathbf{y}, d | \mathbf{x}, G). \quad [5]$$

L'avantage de cette approche est que nous cherchons à maximiser une probabilité jointe entre y et d . En utilisant le théorème de Bayes et des hypothèses d'indépendances entre x et d et entre y et G , il est possible de reformuler l'équation 5 en :

$$(\mathbf{y}, d)_{max} = \underset{(\mathbf{y}, d)}{\operatorname{argmax}} P(d | \mathbf{y}, G) \cdot P(\mathbf{y} | \mathbf{x}). \quad [6]$$

Dans l'équation 6, la première partie correspond à la partie grammaticale et à la recherche de l'arbre de dérivation le plus probable pour une séquence d'éléments terminaux (des classes) fixée. La deuxième partie correspond à la classification probabiliste d'une séquence de feuilles pour estimer une séquence de classes (les terminaux).

La formule indique qu'il faut calculer la probabilité jointe pour tous les couples (y, d) et prendre le couple dont la valeur est maximale. La théorie impose d'effectuer le test pour tous les couples de valeurs possibles afin de trouver un maximum global. Ce n'est malheureusement pas réalisable en pratique. Afin de pallier à ce problème, nous avons mis en place une modification de l'algorithme inside-outside pour les grammaires hors-contextes probabilistes. Il permet de calculer efficacement la probabilité d'un arbre de dérivation à partir d'une séquence donnée (Lari *et al.*, 1990). Plus spécifiquement, nous modifions la partie inside de l'algorithme en injectant la distribution de probabilité estimée par le classifieur probabiliste. La modification que nous avons apporté nous permet de conserver la complexité de l'algorithme initial en $O(n^3)$ avec n la longueur de la séquence. Les détails de cet algorithme ainsi qu'un exemple peuvent être trouvés dans (Chidlovskii *et al.*, 2005).

4. Résultats

Nous avons testé notre méthode pour l'annotation XML sur deux collections. La première est une collection de 39 pièces de Shakespeare, disponibles dans les formats HTML et XML sur le web³. Nous avons extrait aléatoirement 60 scènes de ces pièces pour l'évaluation, elles possèdent de 17 à 189 feuilles. Le fragment de la DTD correspondant aux scènes est composé de 4 terminaux et de 6 non-terminaux.

La seconde collection, appelée TechDoc, est constituée de 60 documents techniques décrivant des opérations de maintenance. Les documents cibles ont une granularité sémantique bien plus fine que pour la collection des pièces de Shakespeare et ont une profondeur plus importante. Le plus long document possède 218 feuilles. Le schéma cible est donné par une DTD complexe qui possède 27 terminaux et 35 non-terminaux.

Pour évaluer la précision de notre annotation, nous utilisons deux métriques différentes. Le *Pourcentage d'Erreurs Terminales* (PET) est similaire au pourcentage d'erreurs sur les mots en traitement du langage naturel, il calcule le pourcentage d'éléments terminaux (classes) qui ont été correctement annotés dans les documents de test (Lehnert *et al.*, 1991). La deuxième métrique est le *Pourcentage d'Erreurs des Non-terminaux* (PEN) qui calcule le pourcentage de sous-arbres correctement annotés. Nous considérons un sous-arbre bien annoté si l'estimation du symbole N_i^j dominant la sous-séquence (y_i, \dots, y_j) correspond effectivement au symbole dominant la même séquence dans le document à obtenir. La précision PEN correspond donc au rapport du nombre de nœuds corrects sur le nombre de nœuds total.

3. Les fichiers HTML sont disponibles ici : <http://www-tech.mit.edu/Shakespeare>, les fichiers XML sont présents ici : <http://www.ibiblio.org/xml/examples/shakespeare>.

Method	TechDoc		Shakespeare	
	PET	PEN	PET	PEN
MaxEnt	92.68	–	100.0	–
NB - contenu	71.84	–	81.90	–
NB - structure	76.37	–	99.95	–
MaxEnt + G	93.39	80.00	99.97	99.81

Tableau 1. Résultats de l'évaluation.

Pour le modèle d'apprentissage de MaxEnt, nous extrayons 38 attributs de contenu pour chaque observation comme le nombre de mots, sa longueur, etc. Ensuite, nous extrayons 14 attributs de structure et de présentation qui incluent les balises entourant la feuille ainsi que les attributs XML associés.

Pour chaque test, nous effectuons une validation croisée. Nous avons testé la classification de séquence de terminaux seuls avec différents classifieurs, MaxEnt en utilisant ensemble les 52 attributs et deux classifieurs basé sur Naive Bayes (NB) qui utilisent respectivement les attributs de contenu et les attributs de structure et de présentation. Ces classifieurs servent de référence pour la comparaison avec notre approche qui rajoute des contraintes grammaticales pour guider la classification de MaxEnt, ils fournissent les résultats pour PET. Le classifieur basé sur MaxEnt en combinaison avec les grammaires (MaxEnt + G) permet de calculer PET et PEN.

Les résultats des différents tests sont collectés dans le Tableau 1. La méthode combinatoire permet de montrer une amélioration du classifieur simple et nous montre que les contraintes grammaticales permettent de récupérer certaines erreurs de classification de MaxEnt. Cependant, seule la dernière méthode permet d'effectuer la conversion complète et produit un arbre sémantique.

5. Autres approches

La transformation de documents définis dans un schéma source (basé sur la présentation dans notre cas) vers des documents définis dans un schéma cible (fourni par un utilisateur dans notre cas) a fait l'objet de plusieurs langages de transformations d'arbres comme XPath ou XSLT par exemple. Ils fournissent tous des outils de programmation très puissants qui permettent de réaliser un grand nombre de tâches liées à la transformation de documents.

Ces approches sont déclaratives et nécessitent une écriture manuelle des règles de transformation. Des méthodes d'apprentissages comme (Curran *et al.*, 1999) peuvent apprendre des règles simples de transformation. Elles supposent que des documents sources peuvent être transformés dans des documents XML grâce à une série d'opérations de transformations élémentaires comme l'insertion, le remplacement, la su-

pression et l'échange. Le modèle de traduction apprend un ensemble d'opérations qui minimisent une erreur donnée par une fonction d'évaluation.

Dans le domaine de l'analyse de la présentation des documents, l'utilisation de balises XML ou HTML peuvent faciliter la récupération de documents sur le web. Des systèmes comme (Altamura *et al.*, 2001) (Wang *et al.*, 1999) sont ainsi capables de transformer des documents scannés sous la forme de documents bien structurés. Cependant, le résultat de ces systèmes restent orientés présentation et contiennent très peu d'informations sémantiques. L'objectif principal est de préserver une visualisation qui soit la plus proche possible du document original dans un navigateur web.

Une autre catégorie de système adresse le problème de conversion de documents. Ces méthodes, comme (Chung *et al.*, 2002), traite plus particulièrement de la conversion de documents HTML vers des documents XML. En analysant les collections et en utilisant des techniques d'apprentissage non supervisées, l'auteur définit des méthodes manuelles d'extraction et des règles de composition qui sont capable de trouver des motifs structurels représentatifs dans l'arbre d'entrée, de définir un label à affecter à un élément extrait et de finalement restructurer les éléments pour former un arbre converti. Enfin, (Ishitani, 2003) est allé un peu plus loin dans la conversion documentaire, la sortie hiérarchique de l'analyse logique permet de générer un document dans un format XML pivot qui est ensuite converti manuellement vers un schéma utilisateur en XML.

6. Conclusion

Nous proposons une méthode probabiliste pour l'annotation XML de documents semi-structurés. Cette méthode s'inscrit dans le projet LegDoC qui a pour objectif la conversion en masse de documents vers XML. Le problème de l'annotation d'arbre est réduit à la dérivation hors-contexte probabiliste d'une séquence d'observation. Nous déterminons l'arbre d'annotation le plus probable en maximisant la probabilité jointe d'estimer une séquence de symboles terminaux et de dériver un arbre pour cette séquence.

Les résultats obtenus valident notre approche, la performance des algorithmes nous permet d'affirmer qu'il est déjà possible avec cette approche d'effectuer automatiquement une partie importante de la conversion. Dans le futur, nous envisageons d'adresser de nouveaux challenges dans l'automatisation de la conversion de documents HTML vers XML. Nous sommes plus particulièrement intéressés dans la prise en compte des structures d'arbres d'entrée dans le modèle d'apprentissage. Nous envisageons également de rendre les algorithmes actifs pour minimiser la tâche de l'annotation des documents pour l'apprentissage supervisé et pour améliorer les résultats.

7. Bibliographie

- Altamura O., Esposito F., Malerba D., « Transforming Paper Documents into XML Format with WISDOM++ », *IJDAR*, vol. 4, n° 1, p. 2-17, 2001.
- Berger A. L., Pietra S. A. D., Pietra V. J. D., « A Maximum Entropy Approach to Natural Language Processing », *Computational Linguistics*, vol. 22, n° 1, p. 39-71, March, 1996.
- Chanod J. P., Chidlovskii B., Déjean H., Fambon O., Fuselier J., Jacquin T., Meunier J. L., « From Legacy Documents to XML : A Conversion Framework », *9th European Conference on Research and Advanced Technology for Digital Libraries (ECDL'05)*, Vienna, Austria, September, 2005.
- Chidlovskii B., Fuselier J., « A Probabilistic Learning Method for XML Annotation of Documents », *Nineteenth International Joint Conference on Artificial Intelligence (IJCAI'05)*, Edimbourg, Scotland, August, 2005.
- Chung C. Y., Gertz M., Sundaresan N., « Reverse Engineering for Web Data : From Visual to Semantic Structures », *Proceedings of the 18th International Conference on Data Engineering (ICDE'02)*, IEEE Computer Society, San Jose, CA, p. 53-63, February, 2002.
- Curran J. R., Wong R. K., « Transformation-Based Learning for Automatic Translation from HTML to XML », *Proceedings of the Fourth Australasian Document Computing Symposium (ADCS99)*, Coffs Harbour, Australia, December, 1999.
- Ishitani Y., « Document Transformation System from Papers to XML Data Based on Pivot XML Document Method », *Proceedings of the 7th International Conference on Document Analysis and Recognition (ICDAR2003)*, Edinburgh, Scotland, p. 250-255, August, 2003.
- Lari K., Young S. J., « The Estimation of Stochastic Context-free Grammars using the Inside-Outside Algorithm », *Computer Speech and Language*, vol. 4, p. 35-56, 1990.
- Lehnert W. G., Sundheim B., « A Performance Evaluation of Text-Analysis Technologies », *AI Magazine*, vol. 12, n° 3, p. 81-94, 1991.
- Malouf R., « A Comparison of Algorithms for Maximum Entropy Parameter Estimation », *Proceedings of the 6th Conference on Natural Language Learning (CoNLL-2002)*, Taipei, Taiwan, p. 49-55, August, 2002.
- Meunier J. L., « Optimized XY-Cut for Text Ordering », *Proceedings of the 8th International Conference on Document Analysis and Recognition*, Seoul, Korea, September, 2005.
- Papakonstantinou Y., Vianu V., « DTD Inference for Views of XML Data », *Proceedings of the Nineteenth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, Dallas, Texas, p. 35-46, May, 2000.
- Schölkopf B., *Statistical Learning and Kernel Methods*, Technical report, Microsoft Research, 2000.
- Wang Y., Phillips I., Haralick R., « From Image to SGML/XML Representation : One Method », *International Workshop on Document Layout Interpretation and its Applications (DLIAP'99)*, Bangalore, India, September, 1999.
- Winkler K., Spiliopoulou M., « Structuring Domain-Specific Text Archives by Deriving a Probabilistic XML DTD », *Proceedings of the 6th European Conference on Principles of Data Mining and Knowledge Discovery*, Helsinki, Finland, p. 461-474, August, 2002.

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

1. ARTICLE POUR LA REVUE :

Document Numérique

2. AUTEURS :

Jérôme Fuselier^{,**} — Boris Chidlovskii^{*}*

3. TITRE DE L'ARTICLE :

*Traitements Automatiques pour la Migration de Documents Numériques
vers XML*

4. TITRE ABRÉGÉ POUR LE HAUT DE PAGE MOINS DE 40 SIGNES :

Vers un XML sémantique

5. DATE DE CETTE VERSION :

11 juillet 2005

6. COORDONNÉES DES AUTEURS :

– adresse postale :

*Xerox Research Centre Europe,
6, chemin de Maupertuis, 38240 Meylan, France
jerome.fuselier@xrce.xerox.com, boris.chidlovskii@xrce.xerox.com

**Université de Savoie - Laboratoire SysCom,
Domaine Universitaire, 73376 Le Bourget-du-Lac, France

– téléphone : 00 00 00 00 00

– télécopie : 00 00 00 00 00

– e-mail : Roger.Rousseau@unice.fr

7. LOGICIEL UTILISÉ POUR LA PRÉPARATION DE CET ARTICLE :

L^AT_EX, avec le fichier de style `article-hermes.cls`,
version 1.2 du 03/03/2005.

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>