Reconnaissance de sections et d'entités dans les décisions de justice : application des modèles probabilistes HMM et CRF

Gildas Tagny Ngompé[*,**], Sébastien Harispe[*], Guillaume Zambrano[**], Jacky Montmain[*], Stéphane Mussard[**]

[*]Laboratoire LGI2P , École des mines d'Alès
{gildas.tagny-ngompe, sebastien.harispe}@mines-ales.fr
[**]Equipe CHROME , Université de Nîmes

Résumé. Une décision de justice est un document textuel rapportant le dénouement d'une affaire judiciaire. Les juristes s'en servent régulièrement comme source d'interprétation de la loi et de compréhension de l'opinion des juges. La masse disponible de décisions exige des solutions automatiques pour aider les acteurs du droit. Nous proposons d'adresser certains des défis liés à la recherche et l'analyse du volume croissant de décisions de justice en France dans un projet plus global. La première phase de ce projet porte sur l'extraction d'information des décisions dans l'objectif de construire une base de connaissances jurisprudentielles structurant et organisant les décisions. Une telle base facilite l'analyse descriptive et prédictive de corpus de décisions. Cet article présente une application des modèles probabilistes pour la segmentation des décisions et la reconnaissance d'entités dans leur contenu (lieu, date, participants, règles de loi, ...). Nos tests montrent l'avantage d'approches basées sur les champs aléatoires conditionnels (CRF) par rapport à des modèles plus simples et rapides basés sur les modèles cachés de Markov (HMM). Nous présentons ici les aspects techniques de la sélection et l'annotation du corpus d'apprentissage, et la définition de descripteurs discriminants. La spécificité des textes est importante et doit être prise en compte lors de l'application de méthodes d'extraction d'information dans un domaine spécifique.

1 Introduction

Une décision de justice est soit le résultat rendu par des juges à l'issue d'un procès, soit un document contenant la description de l'affaire, le résultat des juges et les motifs qui ont conduit à ce résultat. Cet article présente une approche de reconnaissance de sections (entête, exposé de l'affaire, et dispositif) et d'entités (date, ville, nom des juges, ...) dans ces documents. Plus précisément, nous évaluons l'application de deux approches de reconnaissance d'information à base de deux modèles markoviens HMM (*Hidden Markov Model*) et CRF (*Conditional Random Fields*). Les décisions jurisprudentielles sont essentielles pour les juristes parce qu'elles sont des sources d'interprétation de la loi. Les juristes doivent rassembler et analyser des décisions pertinentes pour résoudre les problèmes auxquels ils s'intéressent afin de mieux

anticiper les décisions des juges. Généralement manuelle, cette analyse rencontre quelques limites. D'abord, l'accès à un corpus exhaustif de décisions est difficile vu l'énorme volume de décisions réparti dans les juridictions (plus de 4 millions de décisions en France par an [1]). Malgré la disponibilité d'un nombre important de décisions en ligne, les moteurs de recherche juridiques proposent essentiellement des critères à mots-clés. L'extraction d'information aiderait à mieux décrire et organiser les décisions tout en enrichissant les critères de recherche avec notamment les noms des juges ou les articles de loi. D'autre part, l'analyse manuelle de décisions peut devenir pénible lorsque les documents sont longs et nombreux. Par ailleurs, la justice est complexe et son langage difficilement compréhensible (Cretin, 2014) pour permettre à un non-juriste d'estimer les conclusions d'une décision sans l'aide d'un initié en droit. Les technologies actuelles de traitement du langage naturel et de fouille de textes peuvent permettre une analyse automatisée de documents afin d'atténuer ces obstacles. Par exemple, la reconnaissance d'entités et la classification de textes ont aidé à structurer une large collection d'articles scientifiques pour faciliter leur recherche (McCallum et al., 2000b). D'une part, une analyse automatisée des décisions jurisprudentielles peut aider des avocats et chercheurs en droit à comprendre l'opinion des juges sur certaines questions. D'autre part, elle constitue potentiellement une aide précieuse pour les particuliers et entreprises soucieux de connaitre les chances que leurs requêtes aboutissent en justice.

Comment exploiter un corpus de décisions pour analyser, voire prédire, les décisions des juges sachant que l'interprétation subjective des règles juridiques rend l'application de la loi non déterministe ? Cette question intéresse de nombreuses entreprises telles que LexisNexis avec son système LexMachina [2], et de jeunes startups françaises telles que Predictice [3] et CASE LAW ANALYTICS [4]. Afin d'y répondre, nous développons actuellement une approche automatisée permettant une analyse exhaustive, descriptive et prédictive de la jurisprudence. Cette analyse nécessite tout d'abord de structurer le corpus de décisions à analyser à partir d'informations les caractérisant : numéro d'inscription au répertoire général (R.G.), juridiction, ville, date, juges, normes utilisées, demandes et quanta demandés, résultats des juges et quanta accordés... Cette formalisation des informations et de leurs relations (ex. demande fondée sur une norme) permet une description et une organisation des décisions en une base de connaissances. L'objectif premier de notre projet vise ainsi à extraire des informations des contenus textuels d'un corpus de décisions. Par la suite, ces informations doivent être normalisées afin de construire une base de connaissances de la jurisprudence française. Les cas d'application pouvant bénéficier d'une telle base sont nombreux, par ex. : mieux comprendre l'application de règles juridiques, anticiper les résultats des juridictions, rechercher des décisions similaires, analyser et comparer le risque judiciaire entre des périodes ou des lieux, ou encore identifier les facteurs qui influencent les résultats des juges. La construction d'une telle base de connaissances nécessite une description des décisions. Ces dernières sont des textes libres mais avec une structure standard. Elles comprennent plusieurs informations nécessaires à la compréhension de l'affaire (le lieu, les parties, les juges, la date, les requêtes des parties, les résultats des juges, ...). Les natures différentes de ces informations imposent différentes tâches d'analyse de texte. Par exemple, l'extraction du lieu, de la date, des noms des juges, et des règles juridiques (normes) s'assimile à de la reconnaissance d'entités nommées ; problématique largement étu-

1. http ://www.justice.gouv.fr/budget-et-statistiques-10054/chiffres-cles-de-la-justice-10303/
2. https ://lexmachina.com
3. http ://predictice.com
4. http ://caselawanalytics.com

diée en traitement automatique du langage naturel (Marrero et al., 2013). Cependant, pour l'extraction d'information concernant les demandes des parties et les résultats des juges, des approches novatrices doivent être définies.

Cet article se restreint à la segmentation des décisions et à la reconnaissance des entités (tableau 1) à l'aide de modèles probabilistes. On peut distinguer quatre approches de reconnaissance d'entités (Chau et al., 2002) : à base de lexique, à base de règles, à base de statistiques, à base d'apprentissage automatique. Ces approches ont déjà été appliquées pour l'extraction d'entités dans des textes juridiques. Après une segmentation des documents avec un CRF, Dozier et al. (2010) combinent ces approches pour reconnaître des entités dans les décisions de la cour suprême des Etats-Unis. Ils définissent séparément entre autres des détecteurs à base de règles respectivement pour identifier la juridiction (zone géographique), le type de document, et les noms de juges ; un détecteur à lexique pour la cour, et un classificateur entrainé pour le titre. Ces détecteurs ont des performances prometteuses mais avec des rappels limités entre 72% et 87%. Par ailleurs, sur des décisions tchèques, Kríž et al. (2014) comparent l'application du HMM et d'un algorithme de perceptron à marges inégales (PAUM) pour reconnaitre des institutions et des références à d'autres décisions et aux actes (loi, contrat, ...). Ces deux modèles présentent de bonnes performances avec des mesures-F1 comprises entre 89% et 97% pour le HMM avec des trigrammes et entre 87% et 97% pour le PAUM avec les 5-grammes des lemmes et les rôles grammaticaux des termes.

2 L'étiquetage de texte à base des modèles HMM et CRF

Considérons un texte T comme étant la séquence d'observations $t_{1:n}$. Chaque t_i est un segment de texte (mot, ligne, phrase, ...). Une tâche de segmentation de T consiste à découper T en des groupes ne se chevauchant pas de telle sorte que les éléments liés soient dans le même groupe. Tandis que l'étiquetage de T consiste à assigner les labels appropriés à chaque t_i. La segmentation de T passe par un étiquetage où les t_i consécutifs ayant le même label font partie du même groupe. Le HMM et le CRF ont démontré leur efficacité pour diverses tâches comme la distinction des questions et des réponses dans des foires aux questions ou FAQs (McCallum et al., 2000a), ou l'extraction d'entités dans les entêtes et références d'articles scientifiques (Peng et McCallum, 2006). Nous décrivons dans cette section leur principe de fonctionnement.

2.1 Les modèles cachés de Markov (HMM)

Un HMM est une machine à états finis $\{s_1, s_2, ..., s_m\}$ dont l'objectif est d'affecter une probabilité jointe $P(T|L)$ à des séquences couplées d'observations $T = t_{1:n}$ et de labels $L = l_{1:n}$. Le HMM étant un modèle génératif, chaque label l_i correspond à l'état s_j dans lequel la machine a généré l'observation t_i. Il y a donc autant de type de labels que d'états. Le processus d'étiquetage de T consiste à déterminer L^* tel que $L^* = \underset{L}{\operatorname{argmax}} P(T, L)$. Une évaluation de toutes les séquences possibles de labels serait nécessaire pour déterminer le L^* qui, globalement, correspond le mieux à T. Pour éviter la complexité exponentielle $O(n^m)$ de cette approche, le processus d'étiquetage utilise généralement l'algorithme de décodage Viterbi (Viterbi, 1967) basé sur une programmation dynamique. Son principe général est de parcourir le texte de t_1 à t_n tout en recherchant le chemin d'états (ou de labels) qui a le meilleur score

à chaque position i de T (probabilité $P(t_{1:i}, l_{1:i})$ la plus élevée). Rabiner (1989) donne plus de détails dans son tutoriel. Cet algorithme exploite des paramètres qui sont estimés à partir d'exemples de textes annotés :

— Un ensemble d'états $\{s_1, s_2, ..., s_m\}$ et un alphabet d'observations $\{o_1, o_2, ..., o_k\}$
— La probabilité que s_j génère la première observation $\pi(s_j), 1 \leq j \leq m$
— La distribution de probabilité de transition $P(s_i|s_j), 1 \leq i, j \leq m$
— La distribution de probabilité d'émission $P(o_i|s_j), 1 \leq i \leq k, 1 \leq j \leq m$

Les probabilités de transition et d'émission peuvent être inférées à l'aide d'une méthode d'estimation du maximum de vraisemblance (MLE) comme l'algorithme espérance maximisation (EM) dont l'algorithme Baum-Welch (Welch, 2003) est une spécification particulièrement conçue pour les HMM. L'avantage du HMM est sa simplicité et sa rapidité d'entrainement. Par contre, il est difficile de représenter plusieurs caractéristiques interactives, ou de modéliser la dépendance entre observations éloignées car l'hypothèse d'indépendance entre observations est très stricte (l'état courant ne dépend que des états précédents et de l'observation courante).

2.2 Les champs aléatoires conditionnels (CRF)

Même si l'algorithme Viterbi est aussi utilisé pour l'application d'un modèle CRF à l'étiquetage d'un texte $T = t_{1:n}$, la structure du CRF est différente de celle du HMM. Contrairement à la maximisation de probabilité jointe $P(L, T)$ par le HMM, le CRF (Lafferty et al., 2001) (linéaire dans notre cas) cherche la séquence de labels $L*$ qui maximise la probabilité conditionnelle

$$P(L|T) = \frac{1}{Z} \exp\left(\sum_{i=1}^{n}\sum_{j=1}^{F} \lambda_j f_j(l_{i-1}, l_i, t_{1:n}, i)\right)$$

où Z est le facteur de normalisation. Les fonctions potentielles $f(\cdot)$ sont les caractéristiques que manipulent le CRF. Elles sont de deux types : les caractéristiques de transition qui dépendent des labels aux positions précédente (l_{i-1}) et courante (l_i), et de T entièrement, et les caractéristiques d'état qui sont fonction uniquement de l_i et de T. Les $f(\cdot)$ sont définies à base de fonctions à valeur réelle ou binaire $b(T, i)$ (Wallach, 2004) permettant d'exprimer une combinaison de descripteurs à une position i dans T que nous trouvons discriminants. Pour l'étiquetage de normes, le CRF peut avoir, par exemple, les fonctions potentielles suivantes pour l'étiquetage de "*700*" dans le contexte "*... l'article 700 du code de procédure ...*" :

$$f_1(l_{i-1}, l_i, t_{1:n}, i) = \begin{cases} b_1(T, i) & \text{si } l_{i-1} = \text{NORME} \wedge l_i = \text{NORME} \\ 0 & \text{sinon} \end{cases}$$

$$f_2(l_{i-1}, l_i, t_{1:n}, i) = \begin{cases} b_2(T, i) & \text{si } l_i = \text{NORME} \\ 0 & \text{sinon} \end{cases}$$

avec

$$b_1(T, i) = \begin{cases} 1 & \text{si } (t_{i-1} = \text{article}) \wedge (POS_{i-1} = \text{NOM}) \\ & \wedge (NP1_{i-1} = \text{<unknown>}) \wedge (NS1_{i-1} = \text{@card@}) \\ 0 & \text{sinon} \end{cases}$$

$$b_2(T, i) = \begin{cases} 1 & \text{si } (t_i = 700) \wedge (POS_i = \text{NUM}) \wedge (NP1_i = \text{article}) \wedge (NS1_i = \text{code}) \\ 0 & \text{sinon} \end{cases}$$

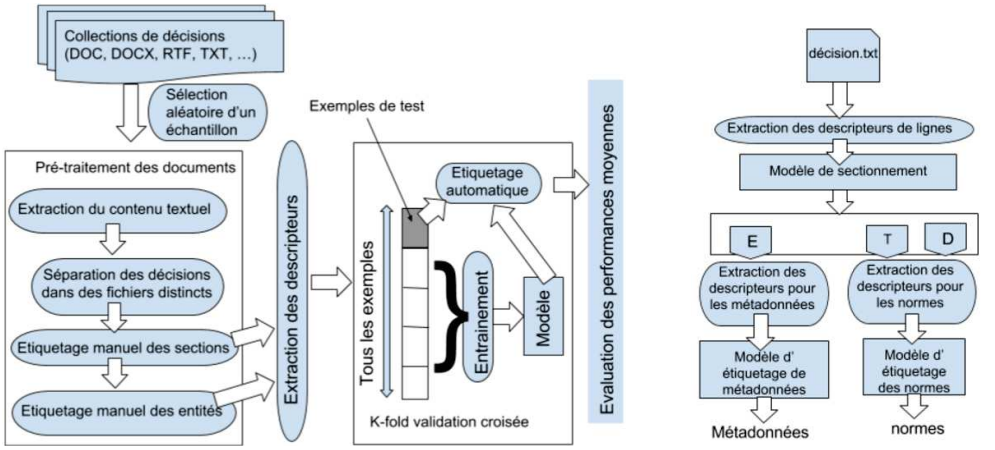
où t_i désigne la position actuelle dans T, POS est le rôle grammatical de t_i (NUM = valeur numérique), NP1 et NS1 désignent respectivement le lemme des noms précédant et suivant les plus proches de t_i. Les symboles <*unknown*> et @*card*@ représentent les lemmes inconnus et les lemmes de nombres. Les deux fonctions $f_1 et f_2$ pouvant être actives au même moment, elles définissent des caractéristiques se chevauchant. Avec plusieurs fonctions activées, la croyance en l_i = NORME est boostée à la somme des poids des fonctions activées ($\lambda_1 + \lambda_2$)(Zhu, 2010).Le CRF utilise une fonction $f_j(\cdot)$ lorsque ses conditions sont remplies et $\lambda_j > 0$. Les différentes caractéristiques pondérées $f(\cdot)$ sont définies par les descripteurs que nous définissons sur le texte (t_i) et l'étiquetage du jeu d'entrainement. L'entrainement consiste essentiellement à estimer les paramètres λ à partir de textes préalablement annotés $\{(T_1, L_1), ..., (T_M, L_M)\}$ où T_k est un texte et L_k la séquence de labels correspondante. Il s'agit de maximiser la vraisemblance conditionnelle des données d'entrainement $\sum_{k=1}^{M} \log P(L_k|T_k)$ (fonction objectif). L'approche d'apprentissage consiste généralement à calculer le gradient de la fonction objectif et de l'utiliser dans un algorithme d'optimisation comme le L-BFGS.

La suite de l'article présente (i) comment nous avons pris en compte les particularités des documents dans la définition de notre approche, et (ii) les tests que nous avons menés.

3 Application du HMM et du CRF pour la reconnaissance de sections et d'entités dans les décisions françaises

L'observation des décisions fait remarquer la répartition des informations sur trois sections dans cet ordre : les métadonnées en entête (E), les demandes et leurs fondements ou normes juridiques dans l'exposé de l'affaire et des motifs (T) qu'on appellera ici le corps, et les conclusions et leurs fondements dans le dispositif (D). Une segmentation des décisions en 3 sections contribuerait potentiellement à mieux organiser les tâches d'extraction d'information. Une approche intuitive consisterait à définir un algorithme capable de reconnaître les transitions entre les sections à partir de motifs. Mais les marqueurs de transitions sont parfois soit des titres, soit des symboles (astérisques, tirets, ...), soit absents. Même les transitions explicites restent très variées. Par exemple, le passage de l'entête au corps peut être défini par les titres *« Exposé », « FAITS ET PROCÉDURES », « Exposé de l'affaire », « Exposé des faits », ...* Quant au dispositif, il démarre généralement par l'expression clé *« PAR CES MOTIFS »* avec des variantes simples (*« Par Ces Motifs »*, ...) ou exceptionnelles (*« P A R C E S M O T I F S : »*). Certains greffiers préfèrent d'autres expressions telles que *« DÉCISION »*, *« DISPOSITIF »*, *« LA COUR, »*. Il arrive souvent que le même marqueur soit utilisé aussi bien pour le sectionnement que pour un sous-sectionnement. Notre première tentative de sectionnement à base de règles s'est ainsi montrée infructueuse. Elle consistait à formaliser, à l'aide d'expressions régulières, les schémas de transitions observés dans un ensemble de décisions. Puis, l'ensemble des schémas est représenté sous forme d'un graphe à 3 couches de sommets où chaque couche correspond à une section et les sommets correspondent chacun à une variante du début de la section. Enfin, le graphe est parcouru en profondeur simultanément avec la décision à segmenter afin de trouver le chemin connu qui correspond le mieux au schéma de cette décision. Après une expérimentation sur 2688 décisions d'apprentissage des schémas et 1002 décisions de test, l'approche a montré ses désavantages avec un nombre très important de schémas multiples proposés (46.9%) et la difficulté de définir manuellement les expressions régulières surtout

pour des transitions sans marqueur. Nous avons donc choisi de définir un modèle basé sur un CRF ou un HMM. Après avoir segmenté une décision, les entitées sont identifiées en fonction de la structure interne aux sections comme décrit dans les sous-sections suivantes (figure 1).



(a) Evaluation des modèles. (b) Application des modèles.

FIG. 1: Architecture de l'approche.

3.1 Extraction des descripteurs de ligne pour reconnaître les sections

Quelque soit le document, les sections s'enchainent dans le même ordre : $E \rightarrow T \rightarrow D$. Plusieurs critères différentient les sections : la longueur des lignes (plus longues dans le corps, plus courtes dans l'entête), les premiers termes de certaines lignes (typiques à chaque section), et le nombre de lignes. Le HMM ne supporte qu'un seul descripteur qui est généralement assimilé à l'élément à étiqueter. D'autres descripteurs peuvent être la position de l'élément à étiqueter (numéro de ligne), les premiers mots de la ligne, etc. Nous avons choisi le numéro de ligne parce qu'il donne un meilleur résultat. Pour le CRF, nous avons choisi de capturer la forme de la ligne : toute la ligne (*ligne*), les premiers termes (*t0, t1, t2*), le nombre de termes (*long*) ; et son contexte : le numéro de ligne (*num*), la longueur de la ligne précédente (*p-long*), les premiers termes des 2 lignes précédentes (*p0, p1*) et des 2 lignes suivantes (*n0, n1*). Plus précisément, considérons par exemple les trois lignes suivantes, à la transition entre T et D :

```
application de l'article 700 du Code de procédure civile ;
</T>
<D>PAR CES MOTIFS
La COUR ;
```

Les descripteurs des lignes sont extraits sous une forme nominale pour le CRF comme suit :

```
ligne=<application de l'article 700 du Code de procédure civile ;> p0=bardaille p1=au
num=275 t0=application t1=de t2=l'article long=10 p-long=13 n0=PAR n1=CES label=T

ligne=<PAR CES MOTIFS> p0=application p1=de num=276 t0=PAR t1=CES t2=motifs long=3
p-long=10 n0=La n1=COUR label=D
```

```
ligne=<La COUR ;> p0=par p1=CES num=277 t0=La t1=COUR t2=; long=3 p-long=3 n0=Statuant
n1=par label=D
```

Par contre, les descripteurs de ces lignes pour le HMM sont réduits au numéro de ligne :

```
num=275 label=T
num=276 label=D
num=277 label=D
```

3.2 Extraction des descripteurs pour reconnaître les entités

Dans l'entête, se retrouvent de nombreux types d'entités contrairement aux deux autres sections qui ne contiennent que les normes. Par ailleurs, l'entête est mieux structurée que les autres sections mais avec de nombreuses variantes différentes selon le greffier ou la juridiction.

Entités	**Labels**	**Exemples**
Numéro R.G.	**RG**	"10/02324", "60/JAF/09"
Ville	**VL**	"NÎMES", "Agen", "Toulouse"
Type de juridiction	**JR**	"COUR D'APPEL"
Formation	**FM**	"1re chambre", "Chambre économique"
Date	**DT**	"01 MARS 2012", "15/04/2014"
Partie appelante	**AP**	"SARL K.", "Syndicat ...", "Mme X ..."
Partie intimée	**IM**	- // -
Partie intervenante	**IV**	- // -
Avocat	**AV**	"Me Dominique A., avocat au barreau de Papeete"
Juge	**JG**	"Monsieur André R.", "Mme BOUSQUEL"
fonction du juge	**FT**	"Conseiller", "Président"
Norme	**NO**	"l' article 700 NCPC", "articles 901 et 903"
Element à éviter	**O**	*tout élément ne faisant partie d'aucune entité ciblée*

TAB. 1: Labels utilisés lors de l'étiquetage des entités dans les sections.

Pour la reconnaissance d'entité dans les entêtes, un jeu d'exemples est constitué en marquant les entités ciblées dans les sections avec les labels correspondants (tableau 1). Notre approche consiste à entrainer notre modèle CRF ou HMM à étiqueter les différents éléments constituant les entités (mot, ponctuation, nombre, identifiant). Les parties et avocats se trouvent très souvent après des mots clés, par exemple, « *APPELANTS* » ou « *DEMANDEUR* » pour les *appelants*, « *INTIMES* » pour les *intimés*, et « *INTERVENANTS* » pour les *intervenants*. Les noms de personnes commencent par une majuscule ou sont entièrement en majuscule. Les numéros R.G. et les dates contiennent des éléments qui sont des nombres (rôle grammatical). Ils contiennent souvent des caractères de ponctuation (ex. « / »), tout comme certaines initiales et abréviations. On observe couramment dans le même ordre les lignes contenant ces entités. Nous avons ainsi considéré des descripteurs de formes (l'élément, son rôle grammatical, son lemme, "commence-t-il par une lettre majuscule ?", "est-il un mot entièrement en majuscule ?", "est-ce une lettre initiale ?" (ex. « B. »), "contient-il un caractère de ponctuation ?", les 2 éléments précédents et les 2 suivants ainsi que leur lemme). Nous avons considéré aussi des descripteurs de contexte (numéro de ligne, position de l'élément dans la ligne, nombre d'éléments dans la ligne, "le texte contient-il la chaine « *intervenant* » ?"). Dans le cas où l'élément

est un nom propre, une abréviation, ou un nombre, nous considérons aussi les numéros des lignes précédente et suivante où il a été détecté, et son numéro d'occurrence, parce que les noms des parties sont très souvent rappelés à plusieurs emplacements. Pour le HMM, nous n'avons considéré que l'élément tel qu'il apparait dans le texte.

L'approche est similaire pour les normes, mais nous avons défini un jeu différent de descripteurs : l'élément, son lemme, son rôle grammatical, les lemmes des 2 éléments noms ou adjectifs précédents et suivants, "l'élément est-il un terme clé des normes ?". Pour ce dernier descripteur, nous avons défini un court lexique de quelques termes comme *article*, *code*, *loi*, *contrat*, *règlement*, *convention*, *décret*. Le lemme homogénéise des variantes d'un même terme. Les éléments voisins ont été choisis pour indiquer au modèle la proximité de l'élément avec des termes couramment utilisés pour référencer les normes.

3.3 Architecture

Les phases de notre approche applicative sont résumées comme suit (Fig. 1) :

Pré-traitement : Les décisions sont téléchargeables sous divers formats (RTF, DOC(X), TXT,...). Les documents téléchargés contiennent une ou plusieurs décisions. Leur contenu textuel doit être extrait en le nettoyant d'éléments inutiles comme des caractères invisibles continus et les lignes vides. Ces éléments apparaissent généralement dans les documents RTF, DOCX, ou DOC, pour la mise en forme du texte. Elles ne donnent aucune indication sur le début des sections ou d'autres informations. Par la suite, les décisions sont séparées dans des fichiers plein texte distincts. On peut dès lors marquer leurs sections à l'aide de balises XML (<E>, <T>, <D>) pour obtenir un corpus d'exemples pour la segmentation. Partant de ces documents XML, nous constituons le jeu d'exemples pour la détection d'entités en marquant le début et la fin des entités ciblées dans les sections (figure 2).

```
@NO l' article 276 du code de procédure civile #NO pour solliciter une contre-expertise ;
qu' en se fondant ainsi sur un moyen tiré d' une fin de non-recevoir qu' elle
soulevait d' office , sans que les parties aient été invitées au préalable à
présenter leurs observations ' , la Cour d' appel avait violé @NO l' article 16 du code
de procédure civile #NO .
```

FIG. 2: Exemple d'annotation manuelle des normes dans le corps (*début : @NO, fin : #NO*).

Extraction de descripteurs : Elle commence par le découpage du texte en éléments. L'extracteur calcule ensuite les descripteurs de chaque élément. Le tout est stocké dans un fichier pour chaque décision. Pour rappel, les éléments ne sont pas les entités entières mais les éléments issus du découpage des textes : lignes pour le sectionnement, et mots / nombres / identifiants / ponctuation pour la reconnaissance d'entités. Une entité contient des éléments. Les éléments ne faisant partie d'aucune entité sont étiquetés avec le label par défaut "O".

k-fold validation croisée : Elle permet de randomiser le jeu d'exemples et d'effectuer au moins k tests afin d'avoir une meilleure appréciation de la performance des modèles.

Application du modèle : Le modèle de segmentation est appliqué en premier pour organiser l'extraction des entités. L'application des modèles d'extraction d'entités peut être parallélisée en 3 processus par la suite. Le même modèle est entrainé pour la détection des normes aussi bien dans les corps (T) que dans les dispositifs (D), vu que les normes y sont citées pareillement.

4 Expérimentations et résultats

Constitution d'un jeu d'apprentissage : Pour les tâches de traitement du langage naturel, Xiao (2010) suggère le choix d'un échantillon suffisant en volume, équilibré sur la variété des données, et représentatif du langage. Nous avons annoté manuellement un jeu de 505 décisions de cours d'appel. Pour simuler la représentativité du corpus, les décisions ont été choisies en variant aléatoirement leur ville et leur année d'origine.

Conditions de tests : Nous avons utilisé l'implémentation du premier ordre de Markov du HMM et du CRF de la librairie Mallet (McCallum, 2002). Les modèles HMM ont été entrainés par la méthode du maximum de vraisemblance, et les CRF par la méthode L-BFGS parce qu'elle s'éxécute plus rapidement avec plusieurs processus en parallèle. Pour l'extraction des entités, le découpage du texte des sections en mots, et l'extraction de leur lemme et rôle grammatical ont été effectués à l'aide de la fonctionnalité française d'extraction de rôles grammaticaux de TreeTagger [5] (Schmid, 2013). Nous avons implémenté l'extraction des autres descripteurs pour cette expérimentation. La k-fold validation croisée pour la reconnaissance des normes est effectuée avec les exemples annotés des sections T et D.

Résultats : Dans la suite, nous appelons HMM notre modèle basé sur le modèle caché de Markov, CRF- et CRF+ notre modèle basé sur les champs aléatoires conditionnels respectivement sans et avec nos descripteurs. Nous avons effectué une 5-fold validation croisée pour évaluer chacune des tâches. Les performances sont estimées en calculant la moyenne des précisions (P), rappels (R) et mesures-F1 (F1) sur le nombre total de tests (5 dans notre cas). Ces derniers sont calculés pour chaque label l comme suit :

$$P_l = \frac{\text{nombre d'éléments correctement étiquetés par le modèle avec } l}{\text{nombre d'éléments étiquetés par le modèle avec } l}$$

$$R_l = \frac{\text{nombre d'éléments correctement étiquetés par le modèle avec } l}{\text{nombre d'éléments manuellement étiquetés avec } l}$$

$$F1_l = 2 \times \frac{P_l \times R_l}{P_l + R_l}$$

Les résultats, les moyennes y comprises, ont été tronqués à un 10^{-1} près.

	HMM			CRF-			CRF+		
	P	R	F1	P	R	F1	P	R	F1
E	84.2	91.8	87.8	93.8	85.4	89.3	99.3	99.6	99.5
T	88.4	63.9	74.1	86.3	98.2	91.8	99.8	99.5	99.7
D	15.4	47.0	23.0	100.0	8.5	15.6	98.0	100.0	98.9
Moyenne	62.7	67.6	67.6	93.3	64.0	64.0	99.7	99.8	99.8

TAB. 2: Précision (P), rappel (R), F1-mesure (F1) au niveau des lignes (%).

Les résultats du sectionnement sont résumés dans le tableau 2. Il est à noter à quel point les descripteurs améliorent les performances du CRF (le CRF- a une mesure-F1 inférieure au HMM). Avec une mesure-F1 moyenne presque parfaite, le CRF+ assure un sectionnement des décisions avec de très rares cas de confusion. Il s'agit de quelques lignes généralement

5. http ://www.cis.uni-muenchen.de/ schmid/tools/TreeTagger

situées près des transitions entre sections. D'autre part, les lignes présentant les demandes des parties sont parfois similaires aux lignes des conclusions des juges. Ceci peut potentiellement pousser le modèle à les étiqueter comme étant des lignes de dispositifs. On pourrait cependant ne conserver que le dispositif détecté en fin de document. La raison de la faiblesse du HMM et du CRF- peut peut-être se justifier par l'utilisation du numéro absolu des lignes comme descripteur. La position relative des lignes dans le document peut mieux remplacer le numéro ou y être associée. Par exemple, en considérant le document découpé en des parties d'égale longueur, le descripteur de la ligne serait la partie où elle se trouve.

Les résultats de la détection d'entités sont résumés dans le tableau 3. Nos descripteurs permettent au CRF+ d'atteindre des mesures-F1 en général supérieures à 85% sauf pour les *parties intervenantes* IV (46.4%). Ces entités sont en général situées juste après la liste des *parties intimées* IM et ne sont pas toujours présentes dans les décisions. Ce qui justifie probablement leur difficile apprentissage et leur confusion en majorité avec les *parties intimées*. Une détection préalable de la région de chaque type de partie pourrait améliorer les performance même si cela nécessite plus d'effort d'annotation manuelle.

	HMM			CRF-			CRF+		
	P	R	F1	P	R	F1	P	R	F1
Section Entête (E)									
AP	35.3	14.1	20.1	64.9	48.8	55.6	92.0	86.7	89.3
AV	83.8	98.3	90.5	96.4	97.5	96.9	97.6	98.1	97.9
DT	70.9	72.6	71.7	94.4	86.8	90.4	98.8	97.7	98.2
FM	87.6	93.7	90.5	98.8	98.4	98.6	98.9	99.3	99.1
FT	88.8	59.8	71.3	94.2	92.3	93.3	97.1	95.5	96.3
IM	53.1	57.4	55.1	67.2	64.6	65.8	89.3	88.1	88.7
IV	-	2.2	-	25.9	26.5	26.2	67.3	41.4	46.4
JG	68.0	85.7	75.7	96.2	95.7	96.0	98.1	97.7	97.9
JR	75.8	99.5	86.0	98.6	99.4	99.0	99.3	99.4	99.4
RG	-	0	-	83.7	46.1	59.4	98.6	97.4	98.0
VL	93.1	27.9	42.6	98.2	98.4	98.3	99.0	99.0	99.0
Sections inférieures (T & D)									
NO	92.9	90.9	91.9	96.0	93.8	94.9	97.9	96.5	97.2

TAB. 3: Précision (P), rappel (R), F1-mesure (F1) au niveau des éléments étiquetés dans les sections ($\%$).

Par ailleurs, le HMM et le CRF- réussissent à bien détecter les normes uniquement à l'aide des éléments originaux des textes. Vu que les règles juridiques existent en nombre limité, celles qui sont référencées dans notre jeu d'exemples sont probablement en majorité très courantes dans les décisions. Ces deux modèles semblent aussi être aidés par le fait que toutes les références aux normes respectent une syntaxe standard (`article [NUMERO] [ORIGINE]`). Néanmoins, les descripteurs définis améliorent ces performances chez le CRF+.

La dernière remarque pourrait être l'occurrence multiple des entités dans les sections. Par exemple, les parties sont citées avant les détails les concernant qui sont plus bas dans l'entête, et certaines normes sont citées à plusieurs reprises et souvent de manière abrégée. Bien que ces occurrences multiples ne soient pas en tout point identiques, elles aident à réduire le risque de manquer une entité. Ce qui pourrait être exploité pour combler l'imperfection des modèles.

5 Conclusion

Cet article présente une application des modèles markoviens pour la reconnaissance de sections et d'entités dans les décisions judiciaires. Nos expérimentations actuelles montrent l'avantage d'approches à base de CRF avec des descripteurs capturant la forme et le contexte des lignes et des mots à étiqueter malgré les bonnes performances du HMM et du CRF sans les descripteurs pour la reconnaissance des normes. Le CRF avec descripteurs est donc adéquat pour notre projet. Cependant, plus d'exemples d'entêtes annotées comprenant des parties intervenantes et un zonage préalable des types de parties amélioreraient probablement la détection des *intervenants* qui restent les seules entités difficilement détectables actuellement. La difficulté majeure reste la constitution d'un jeu suffisant d'exemples et la prise en compte de motifs réguliers du langage des documents pour définir des descripteurs plus caractéristiques des entités à détecter. L'effort d'annotation peut être réduit avec un système aux performances actuelles qui peut correctement étiqueter la majorité des entités. Il suffit ensuite de vérifier manuellement cet annotation pour corriger les éventuelles erreurs du modèle dans de nouvelles décisions. Au cours de nos futurs travaux, nous envisageons d'étendre l'étude à d'autres types de juridictions (tribunaux du premier degré, cour de cassation, ordre administratif, ...). Pour constituer notre base de connaissance, il est indispensable de définir une approche de désambiguïsation pour les entités aux multiples occurrences, et une autre de résolution d'entités pour faire correspondre les entités extraites à des référentiels tout comme Dozier et al. (2010). Ces entités seront exploitées lors de l'extraction des informations plus complexes comme les demandes des parties et le sens des résultats des juges. Nous mettons ainsi progressivement en place une chaîne de traitements amenée à faciliter la conception d'approches d'analyse statistique d'un large corpus jurisprudentiel.

Références

Chau, M., J. J. Xu, et H. Chen (2002). Extracting meaningful entities from police narrative reports. In *Proceedings of the 2002 annual national conference on Digital government research*, pp. 1–5. Digital Government Society of North America.

Cretin, L. (2014). L'opinion des français sur la justice. *INFOSTAT JUSTICE 125*. `http://www.justice.gouv.fr/art_pix/1_infostat125_20140122.pdf`.

Dozier, C., R. Kondadadi, M. Light, A. Vachher, S. Veeramachaneni, et R. Wudali (2010). Named entity recognition and resolution in legal text. In *Semantic Processing of Legal Texts*, pp. 27–43. Springer.

Kríž, V., B. Hladká, J. Dědek, et M. Nečaský (2014). *Statistical Recognition of References in Czech Court Decisions*, pp. 51–61. Cham : Springer International Publishing.

Lafferty, J., A. McCallum, et F. C. Pereira (2001). Conditional random fields : Probabilistic models for segmenting and labeling sequence data. *International Conference on Machine Learning*.

Marrero, M., J. Urbano, S. Sánchez-Cuadrado, J. Morato, et J. M. Gómez-Berbís (2013). Named entity recognition : fallacies, challenges and opportunities. *Computer Standards & Interfaces 35*(5), 482–489.

McCallum, A., D. Freitag, et F. C. Pereira (2000a). Maximum entropy markov models for information extraction and segmentation. In *ICML*, Volume 17, pp. 591–598.

McCallum, A. K. (2002). *MALLET : A Machine Learning for Language Toolkit*. `http://mallet.cs.umass.edu/`.

McCallum, A. K., K. Nigam, J. Rennie, et K. Seymore (2000b). Automating the construction of internet portals with machine learning. *Information Retrieval 3*(2), 127–163.

Peng, F. et A. McCallum (2006). Information extraction from research papers using conditional random fields. *Information processing & management 42*(4), 963–979.

Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE 77*(2), 257–286.

Schmid, H. (2013). Probabilistic part-ofispeech tagging using decision trees. In *New methods in language processing*, pp. 154. Routledge.

Viterbi, A. J. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory 13*(2), 260–269.

Wallach, H. M. (2004). Conditional random fields : An introduction. *University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-04-21*.

Welch, L. R. (2003). Hidden markov models and the baum-welch algorithm. *IEEE Information Theory Society Newsletter 53*(4), 10–13.

Xiao, R. (2010). *Handbook of Natural Language Processing* (Second Edition ed.)., Chapter 7 - Corpus Creation, pp. 146–165. Chapman and Hall.

Zhu, X. (2010). *Conditional Random Fields*. CS769 Spring 2010 Advanced Natural Language Processing. `http://pages.cs.wisc.edu/~jerryzhu/cs769/CRF.pdf`.

Summary

A court decision is a text document, which is a synthesis of the outcome of a court case. Lawyers regularly use them as a source of interpretation of the law and also in order to understand the opinion of judges. The available huge quantity of decisions requires automated solutions to help the actors of law. We propose to address some of the challenges related to the search and the analysis of the growing set of court decisions in France in a larger project. The first phase of this project focuses on extracting information from decisions in order to build a jurisprudential knowledge base structuring and organizing decisions. Such a base facilitates the descriptive and predictive analysis of decisions corpora. This paper presents an application of probabilistic models for the zoning of decisions and the recognition of entities in their content (location, date, participants, rules of law, ...). Our tests show the advantage of the approaches based on Conditional Random Fields (CRF) compared to simpler and faster models based on Hidden Markov Models (HMM). We present the technical aspects of the selection and annotation of the training corpus, and the definition of discriminating descriptors. The specificity of the texts is important and should be taken into account when applying information extracting methods in a specific domain.