

Chapitre 2

Annotation des sections et entités juridiques

2.1 Introduction

Ce chapitre traite de la détection de sections et d’entités dans les décisions jurisprudentielles françaises. Bien que ces dernières ne soient pas structurées, leur contenu est organisé en sections dont les principales sont : l’entête, le corps, et le dispositif. Chaque section décrit des informations spécifiques de l’affaire :

- l’entête contient de nombreuses méta-données de référence comme la date, le lieu, les participants etc.
- le corps détaille les faits, les procédures antérieures, les conclusions des parties et le raisonnement des juges ;
- le dispositif est la synthèse du résultat final c’est-à-dire qu’on y retrouve les réponses aux demandes des parties.

Compte tenu de la répartition des informations, il nous a paru plus simple d’annoter au préalable les sections en segmentant le document. Par la suite, les entités, et données sur les demandes et résultats, peuvent être plus facilement extraites en fonction des sections où elles se retrouvent généralement. Nous nous focalisons en particulier ici sur la détection d’entités telles que la date à laquelle le jugement a été prononcé, le type de juridiction, sa localisation (ville), les noms de juges, des parties, et les règles de loi citées (normes). La Table 2.1 liste les différentes entités ciblées et fournit des exemples illus-

trant leurs occurrences dans les décisions de cour d'appel avec lesquelles nous avons travaillé.



Entités	Label	Exemples	#mentions ^a	
			Médiane ^b	Total ^c
Numéro de registre général (R.G.)	rg	« 10/02324 », « 60/JAF/09 »	3	1318
Ville	ville	« NÎMES », « Agen », « Toulouse »	3	1304
Juridiction	juridiction	« COUR D'APPEL »	3	1308
Formation	formation	« 1re chambre », « Chambre économique »	2	1245
Date de prononcé	date	« 01 MARS 2012 », « 15/04/2014 »	3	1590
Appelant	appelant	« SARL K. », « Syndicat ... », « Mme X ... »	2	1336
Intimé	intime	- // -	3	1933
Intervenant	intervenant	- // -	0	51
Avocat	avocat	« Me Dominique A., avocat au barreau de Papeete »	3	2313
Juge	juge	« Monsieur André R. », « Mme BOUS-QUEL »	4	2089
Fonction de juge	fonction	« Conseiller », « Président »	4	2062
Norme	norme	« l' article 700 NCPC », « articles 901 et 903 »	12	7641
Non-entité	O	<i>mot ne faisant partie d'aucune mention d'entité</i>	-	-

^a nombre de mentions d'entités dans le corpus annoté pour les expérimentations


^b nombre médian de mentions par document dans le corpus annoté

^c nombre total d'occurrences dans le corpus annoté

* Les statistiques sur les sommes d'argent ne concernent que 100 documents annotés (max=106, min=1, moyenne=17.77), contre 500 documents pour les autres entités.



Tableau 2.1 – Entités et labels correspondant utilisés pour labelliser leurs mots.

On  tendrait à ce qu'une institution comme la justice respecte un modèle stricte et commun à tous les tribunaux pour la rédaction des décisions pour permettre de facilement les lire et les analyser. Malheureusement, même si les décisions décrivent des informations de mêmes natures, le modèle employé semble varier entre les juridictions. C'est ce qu'on remarque déjà au niveau de la transition entre sections. Au vu de leur rôle, il est évident que les sections devraient être séparées par des marqueurs bien précis. Une approche intuitive de sectionnement consisterait par conséquent à définir un algorithme capable de reconnaître ces marqueurs de transitions à travers des expressions régulières. Cependant, les marqueurs utilisés ne sont pas stan-


dards. Les indicateurs de transitions sont souvent différents d'une décision à l'autre et peuvent être des titres ou des motifs à base de symboles (astérisques, tirets, etc.). Il arrive parfois que la transition soit implicite et qu'on ne s'en rende compte que par la forme ou le contenu des lignes, au cours de la lecture. Même les marqueurs explicites sont hétérogènes. D'une part, lors de l'emploi de titres par exemple, la transition de l'entête à l'exposé du litige peut être indiquée par des titres comme « Exposé », « FAITS ET PROCÉDURES », « Exposé de l'affaire », « Exposé des faits », etc. Quant au dispositif, il est introduit généralement par l'expression « PAR CES MOTIFS » avec souvent quelques variantes qui peuvent être très simples (par ex. « Par Ces Motifs ») ou exceptionnelles (par ex. « P A R C E S M O T I F S : »). Dans certaines décisions, cette expression est remplacée par d'autres expressions comme « DECISION », « DISPOSITIF », « LA COUR », etc. D'autre part, en raison de l'utilisation de symboles, il arrive qu'un même motif sépare différentes sections et même des paragraphes dans une même section. Des différences similaires apparaissent aussi pour les entités. Les noms de parties sont généralement placés après un mot particulier comme « APPELANTS » ou « DEMANDEUR » pour les demandeurs (appelants en juridiction de 2e degré), « INTIMES » ou « DEFENDEUR » pour les défendeurs (ou intimés), et « INTERVENANTS » pour les intervenants. Les noms des individus, sociétés et lieux commencent par une lettre majuscule, et parfois sont entièrement en majuscule. Cependant, certains mots communs peuvent apparaître aussi en majuscule (par ex. APPELANTS, DÉBATS, ORDONNANCE DE CLÔTURE). Les entités peuvent contenir des chiffres (identifiant, dates, ...), des caractères spéciaux (« / », « - »), des initiales (par ex. « A. ») ou abréviations. Dans l'entête, les entités apparaissent généralement dans le même ordre (par ex. les appelants avant les intimés, les intimés avant les intervenants). Cependant, plusieurs types d'entités apparaissent dans l'entête, contrairement aux autres sections où seules les normes nous intéressent dans cette étude. L'entête est aussi mieux structurée que les

autres sections même si sa structure peut différer entre deux documents.

Notre étude consiste à analyser l'application du Modèle Caché de Markov (HMM) et des Champs Aléatoires Conditionnels (CRF) aux problèmes de sectionnement et reconnaissance d'entités juridiques. Ces deux tâches sont ainsi représentées sous la forme d'un problème d'étiquetage de séquence. L'idée est de découper un texte en des segments atomiques (*token*) qui peuvent être des mots, des phrases, des paragraphes, etc. Le texte est ainsi représenté sous forme de séquence et chaque objet d'intérêt (section ou entité) comprend un ou plusieurs segments. Un label est défini pour chaque type d'entité (par ex. PER pour les noms de personnes).

2.2 Extraction d'information par étiquetage de séquence

Chau et al. (2002) distinguent quatre catégories d'approches d'extraction d'information :

- Les **systèmes à recherche lexicale** sont conçus sur la base d'une liste d'entités préalablement connues, et leurs synonymes dans le domaine d'intérêt. Par exemple, dans le domaine juridique, un lexique pourrait contenir les identifiants de règles juridiques et les noms des juges. La liste des entités peut être **manuscrite**  des experts ou apprise à partir d'un ensemble de données annotées manuellement (phase d'apprentissage). Cependant, il s'avère très difficile de maintenir une telle liste car le domaine pourrait changer régulièrement (nouvelles lois par ex.). De plus, les mentions d'entités peuvent avoir plusieurs variantes. Par exemple, la même règle juridique « Article 700 du code de procédure civile » peut être citée seule et en entier (« article 700 du code de procédure civile »), ou abrégée (« article 700 CPC »), ou encore avec d'autres règles (« articles 700 et 699 du code de procédure civile »). Ces approches sont aussi sujets aux problèmes d'ambiguïté

par **par** exemple lorsque différentes entités comprennent les mêmes mots. Ces problèmes ont limité **les** premiers systèmes (Palmer and Day, 1997).

- Les **systèmes à base de règles** décrivent suffisamment la variété des mentions d'entités en fonction de la régularité du contexte, de la structure et du lexique. Ils sont avantageux parce que leurs erreurs sont facilement explicables. La définition manuelle de règles exige malheureusement des efforts considérables, en particulier pour les grands corpus. De plus, un ensemble donné de règles est difficilement réutilisable dans d'autres domaines. Cependant, quelques approches adaptatives ont été conçues pour surmonter ces limites tout en bénéficiant toujours de la facilité à expliquer le comportement des systèmes à base de règles (Siniakov, 2008; Chiticariu et al., 2010).
- Les **systèmes statistiques** adaptent les modèles statistiques de langage, issus typiquement des méthodes de compression de texte, pour détecter les entités. Par exemple, Witten et al. (1999) ont adapté le schéma de compression appelé « Prédiction par Correspondance Partielle ».
- Les **systèmes basés sur l'apprentissage automatique** exécutent des classifieurs multi-classes sur des segments de texte. Par exemple, un classifieur traditionnel comme le classifieur bayésien naïf peut être entraîné pour détecter les noms de gènes en classifiant les mots d'un article scientifique (Persson, 2012). Par ailleurs, les algorithmes d'étiquetage de séquence tels que le CRF classifient les mots tout en modélisant les transitions entre les labels (Finkel et al., 2005). Dans ce registre, les architectures d'apprentissage profond réalisent actuellement les meilleures performances sur de multiples tâches d'extraction d'information en général et de reconnaissance d'entités nommées en particulier (Lample et al., 2016).

Certains travaux ont combiné différentes approches pour extraire les enti-

tés à partir de documents juridiques, par exemple, par la description de l'information contextuelle en utilisant des règles pour répondre au problème d'ambiguïté des méthodes à recherche lexicale (Mikheev et al., 1999; Hanisch et al., 2005). Mais les systèmes basés sur l'apprentissage automatique sont les plus efficaces actuellement pour l'extraction d'information, en particulier les modèles graphiques probabilistes.

Trois principaux aspects doivent être traités lors de la conception des systèmes à étiquetage de séquence : la sélection du modèle d'étiquetage, l'ingénierie des caractéristiques des segments à labelliser, et le choix d'une représentation de segment (encore appelé schéma d'étiquetage).

2.2.1 Les modèles graphiques probabilistes HMM et CRF

Nous avons choisi d'analyser l'application des modèles CRF et HMM car les comparaisons avec d'autres approches démontrent bien que les modèles probabilistes obtiennent les meilleurs résultats lors de l'extraction d'information dans les documents juridiques. Par exemple, Štěpánek et al. (2014) a été comparé le modèle HMM à l'Algorithme de Perceptron à Marges Inégales (PAUM) de Li et al. (2002) pour reconnaître les institutions et références d'autres décisions de justice, ainsi que les citations d'actes juridiques (loi, contrat, etc.) dans les décisions judiciaires de la République Tchèque. Les deux modèles ont données de bonnes performances avec des scores F1 de 89% et 97% pour le HMM utilisant les trigrammes comme descripteurs de mots, et des scores F1 de 87% et 97% pour le PAUM en utilisant des 5-grammes de lemmes et les rôles grammaticaux (*Part-Of-Speech tag*) comme descripteurs.

Considérons un texte T comme étant une séquence d'observations $t_{1:n}$, avec chaque t_i étant un segment de texte (mots, ligne, phrase, etc.). En considérant une collection de labels, l'étiquetage de T consiste à affecter les labels appropriés à chaque t_i . La segmentation de T est un étiquetage particulier qui implique de découper T en des groupes qui ne se chevauchent pas (des partitions). Les tâches de sectionnement et d'annotation des entités,

prises séparément, sont des problèmes de segmentation.

2.2.1.1 Les modèles cachés de Markov (HMM)

Un modèle HMM est une machine à état défini par un ensemble d'états $\{s_1, s_2, \dots, s_m\}$. Un modèle HMM a pour fonction d'affecter une probabilité jointe $P(T, L) = \prod_i P(l_i | l_{i-1}) P(T | l_i)$ à des paires de séquences d'observations $T = t_{1:n}$ et de séquence de labels $L = l_{1:n}$. Étant donné qu'un HMM est 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 autant de labels candidats que d'états. Le processus de labellisation de T consiste à déterminer la séquence de labels L^* qui maximise la probabilité jointe ($L^* = \arg \max_L P(T, L)$). Une évaluation de toutes les séquences possibles de labels est nécessaire pour déterminer L^* . Pour éviter la complexité exponentielle $O(m^n)$ d'une telle approche, n étant la longueur de la séquence et m le nombre de labels candidats, l'algorithme de décodage Viterbi (Viterbi, 1967), basé sur une programmation dynamique, permet d'obtenir une estimation de L^* . Cette algorithme utilise des paramètres estimés par apprentissage sur un corpus de textes annotés manuellement :

- Un ensemble d'états $\{s_1, s_2, \dots, s_m\}$ et un alphabet ou vocabulaire $\{o_1, o_2, \dots, o_k\}$;
- La probabilité que s_j génère la première observation $\pi(s_j), \forall j \in [1..m]$;
- La distribution de probabilité de transition $P(s_i | s_j), \forall i, j \in [1..m]$;
- La distribution de probabilité de d'émission $P(o_i | s_j), \forall i \in [1..k], \forall j \in [1..m]$.

Les probabilités de transition et d'émission peuvent être inférées en utilisant une méthode de maximum de vraisemblance comme l'algorithme d'espérance maximale. L'algorithme Baum-Welch (Welch, 2003) en est une spécification conçue spécialement pour le HMM.

L'avantage du HMM réside dans sa simplicité et sa vitesse d'entraînement. Cependant, il est difficile de représenter les segments à l'aide de multiples

descripteurs distincts. Il est tout aussi difficile de modéliser la dépendance entre des observations distantes parce que l'hypothèse d'indépendance entre observations est très restrictive (i.e. l'état courant dépend uniquement des états précédents et de l'observation courante). Rabiner (1989) fournit plus de détails sur le modèle HMM.

2.2.1.2 Les champs conditionnels aléatoires (CRF)

Même si l'algorithme Viterbi est aussi utilisé pour appliquer le modèle CRF à l'étiquetage de séquence, la structure du CRF diffère de celle du HMM. Au lieu de maximiser la probabilité jointe $P(L, T)$ comme le HMM, un modèle CRF (Lafferty et al., 2001) cherche la séquence de labels L^* qui maximise la probabilité conditionnelle suivante : $P(L|T) = \frac{1}{Z} \exp \left(\sum_{i=1}^n \sum_{j=1}^F f_j(l_{i-1}, l_i, t_{1:n}, i) \right)$ où Z est le facteur de normalisation. Les fonctions potentielles $f(\cdot)$ sont les caractéristiques utilisées par les modèles CRF. Deux types de fonctions caractéristiques sont définies : les caractéristiques de transition qui dépendent des labels aux positions courantes et précédentes (l_{i-1} et l_i resp.) et de T ; et les caractéristiques d'état qui sont des fonctions de l'état courant l_i et de la séquence T . Ces fonctions $f(\cdot)$ sont définies à l'aide soit par des fonctions à valeur binaire ou réel $f_j(T, i)$ qui combine les descripteurs d'une position i dans T (Wallach, 2004). Pour labelliser les références aux règles de loi par exemple, un CRF pourrait inclure par exemple les fonctions potentielles pour labelliser « 700 » dans ce contexte « ... l'article 700 du code de procédure civile ... » :

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

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

avec

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


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

t_i étant une observation dans T , POS étant le rôle grammatical de t_i (NUM = valeur numérique, NOM = nom), et NP1 et NS1 sont les lemmes des mots avant et après t_i , respectivement. Les symboles *<unknown>* et *@card@* encode les lemmes inconnus et ceux des nombres respectivement. Pouvant être activées au même moment, les fonctions f_1 et f_2 définissent des descripteurs se chevauchant. Avec plusieurs fonctions activées, la croyance dans le fait que $l_i = NORME$ est renforcée par la somme $\lambda_1 + \lambda_2$ des poids des fonctions activées (Zhu, 2010). Un modèle CRF emploie une fonction f_j lorsque ses conditions sont satisfaites et $\lambda_j > 0$. Les diverses fonctions pondérées f_j sont définies par des descripteurs caractérisant les segments, et les labels des données d'entraînement. La phase d'apprentissage consiste principalement à estimer le vecteur de paramètres $\lambda = (\lambda_1, \dots, \lambda_F)$ à partir de textes annotés manuellement $\{(T_1, L_1), \dots, (T_M, L_M)\}$, T_k étant un texte et L_k la séquence de labels correspondants. La valeur optimal de λ est celle qui maximise la fonction objectif $\sum_{k=1}^M \log P(L_k | T_k)$ sur les données d'entraînement. En général, outre le maximum de vraisemblance, cette optimisation est résolue à l'aide de l'algorithme de descente de gradient dont l'exécution peut-être accélérée à l'aide de l'algorithme L-BFGS de Liu and Nocedal (1989).

2.2.2 Représentation des segments atomiques

La représentation des segments à labelliser occupe une place importante pour l'obtention de bons résultats avec les modèles décrits précédemment.

Elle consiste généralement à décrire la forme et le contexte de chaque segment en lui assignant des attributs (Nadeau and Sekine, 2007; Sharnagat, 2014). Ils peuvent être booléens (« le mot est-il en majuscule ? »), numériques (nombre de caractères du mot), nominaux (par ex. le rôle grammatical d'un mot), ou définis par des expressions régulières (par ex. pour les numéros R.G. on peut avoir `dd/dddd` où `d` désigne un chiffre). Ces descripteurs mettent en évidence des régularités relatives à l'occurrence des entités. Par exemple, préciser qu'un mot débute par une lettre majuscule permet d'indiquer les noms propres. La définition de tels descripteurs consiste ainsi à fournir au modèle des indices l'aidant à mieux distinguer les différents types d'entités.

Étant donné que les descripteurs dépendent généralement de l'intuition du concepteur du système d'étiquetage, il est difficile mais nécessaire d'identifier des descripteurs appropriés. Après avoir défini des candidats, il n'est pas sûr qu'en les combinant tous ensemble, on obtienne les meilleures performances. Une sélection de caractéristiques peut s'avérer nécessaire. Cette sélection peut améliorer les performances d'étiquetage, et accélérer l'extraction des descripteurs, l'entraînement du modèle ainsi que son application à de nouveaux textes (Kitoogo and Baryamureeba, 2007). Elle peut aussi fournir une meilleure compréhension du comportement des modèles entraînés (Klinger and Friedrich, 2009). Deux principales approches se distinguent. D'une part, les méthodes « filtrantes » (*filters*), comme l'information mutuelle, comparent individuellement les descripteurs à l'aide de scores qui ne sont pas nécessairement basés sur la performance. D'autre part, les méthodes « enveloppantes » (*wrappers*) comparent des sous-ensembles de descripteurs sur la base de leur performance.  me si les méthodes filtrantes sont plus rapides, elles sont en général moins performantes car elles ne permettent pas d'éviter les redondances, et ne prennent pas en compte l'effet de la combinaison de caractéristiques.

La définition manuelle des caractéristiques suivie de la sélection est souvent qualifiée de méthode forcée car elle dépend fortement de la capacité

du concepteur du système à identifier les descripteurs appropriés. Les réseaux de neurones permettent d'apprendre des caractéristiques grâce à des méthodes de plongement sémantique telles que Word2Vec (Le and Mikolov, 2014) et Glove (Pennington et al., 2014). Deux architectures de réseaux de neurones réalisent actuellement les meilleures performances en matière de détection d'entités nommées. Il s'agit du modèle BiLSTM-CRF de Lample et al. (2016) et du LSTM-CNN-CRF de Ma and Hovy (2016). On pourrait résumer ces architectures en trois phases. Dans un premier temps, les segments de textes (mots) ont une représentation vectorielle concaténant 2 vecteurs de plongement sémantique : l'un issu de l'apprentissage morphologique du mot à partir de ses caractères, et l'autre issu de l'apprentissage du contexte général d'occurrence du mot. Lors de la seconde phase, deux couches de cellules LSTM enchaînées permettent de modéliser le contexte à droite et à gauche de chaque mot du texte labellisé. La dernière phase détermine la séquence de labels la plus probable pour le texte à l'aide d'une implémentation neuronale du modèle CRF. Le CRF reçoit en entrée la concaténation des contextes à droite et à gauche des mots.

2.2.3 Schéma d'étiquetage

Nous traitons d'entités dont les occurrences comprennent un ou plusieurs éléments atomiques. Pour améliorer les résultats d'un modèle d'étiquetage, certaines parties des entités peuvent être mises en évidence à travers une représentation appropriée de segment. Nous comparons dans cette étude quelques schémas d'étiquetage dont certains sont décrits par Konkol and Konopík (2015). Le schéma IOE utilisé par défaut ne met l'accent sur aucune partie et affecte le même label à tous les segments d'une même entité. D'autres schémas distinguent soit le premier élément (BIO), soit le dernier (IEO), soit les deux (BIEO). Les schémas IEO et BIO ont des variantes IEO1, BIO1, IOE2, et BIO2. Les modèles IOE2, et BIO2 utilisent resp. les préfixes E- et B- pour étiqueter les entités à mot unique, contrairement à IEO1 et

BIO1 qui utilisent plutôt le préfixe I- dans ce cas. Le modèle BIEO est souvent étendu sous la forme BIESO (ou BILOU) dans le cas où on souhaite distinguer les entités à un seul segment (par ex. ville ou numéro R.G.). Les lettres des sigles de ces modèles servent de préfixes aux labels et portent la signification suivante :

- B : début (*beginning*) ;
- I : intérieur (*inside*) ;
- E (ou L, ou M) : fin (*end* ou *last* ou *middle*) ;
- S (ou U, ou W) : singleton ou entité à segment unique (*single* ou *unit* ou *whole*) ;
- O : hors de toute entité (*outside*).

La figure 2.1 illustre l'utilisation de ces différents modèles sur un extrait de décision de justice pour l'annotation du nom d'un juge et de sa fonction :

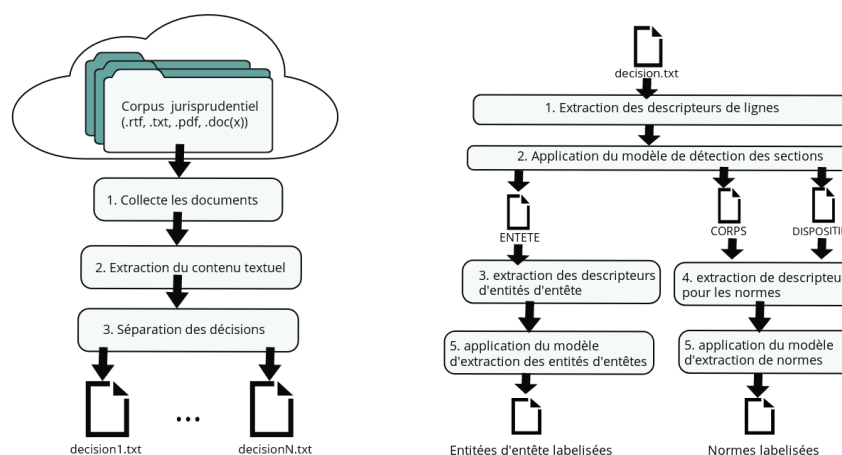
	<i>composée</i>	<i>de</i>	<i>Madame</i>	<i>Martine</i>	<i>JEAN</i>	,	<i>Président</i>	<i>de</i>	<i>chambre</i>	,	<i>de</i>
IO	O	O	I-JUGE	I-JUGE	I-JUGE	O	I-FONCTION	I-FONCTION	I-FONCTION	O	O
BIO	O	O	B-JUGE	I-JUGE	I-JUGE	O	B-FONCTION	I-FONCTION	I-FONCTION	O	O
IEO	O	O	I-JUGE	I-JUGE	E-JUGE	O	I-FONCTION	I-FONCTION	E-FONCTION	O	O
BIEO	O	O	B-JUGE	I-JUGE	E-JUGE	O	B-FONCTION	I-FONCTION	E-FONCTION	O	O

Figure 2.1 – Illustration des schémas d'étiquetage IO, BIO, IEO, BIEO

Il est possible d'aller plus loin en mettant l'accent sur les mots avant (O-JUGE) et après (JUGE-O) l'entité (JUGE par exemple) et en indiquant le début (BOS-O, *beginning of sentence*) et la fin (O-EOS, *end of sentence*) du texte ou de la phrase. Le format ainsi obtenu est appelé BMEWO+ (Baldwin, 2009).

Un autre intérêt très important de modèles plus complexes que IO est de pouvoir distinguer des entités qui se suivent sans être explicitement séparées. Cet aspect est notamment important dans les décisions de justice par exemple lorsque des noms de parties sont listés dans la section ENTETE en n'étant séparés que d'un simple retour à la ligne.

2.3 Architecture proposée



Après la collecte et le prétraitement des documents, l'étiqueteur de ligne est d'abord appliqué pour détecter les sections, puis les étiqueteurs d'entités peuvent être appliqués simultanément dans les sections.

Figure 2.2 – Application des modèles entraînés d'étiquetage de section et entités.

Nous proposons de travailler uniquement avec le contenu textuel des documents. Ce contenu est extrait des documents téléchargés en éliminant les éléments inutiles, principalement des espaces vides. Ces éléments sont typiques des documents formatés (.rtf, .doc(x), .pdf). Ils ne fournissent pas une indication standard sur le début des sections. Le choix de ne pas exploiter le formatage des documents permet d'avoir à gérer un nombre plus faible de diversités entre les textes tout en appliquant le même processus de traitement à tout document indépendamment de son format d'origine. Une simple architecture d'étiquetage de sections et d'entités juridiques a été conçue avec cet uniformisation des documents comme point d'entrée (Figure 2.2). Ainsi, les documents sont collectés puis pré-traités suivant leur format d'origine (extraction du texte et séparation des décisions apparaissant dans le même document). Ensuite, après le sectionnement des décisions, les entités sont identifiées dans les différentes sections. Par ailleurs, Comme segment

atomique à étiqueter nous avons choisi les lignes pour la détection des sections, et les mots pour les entités.

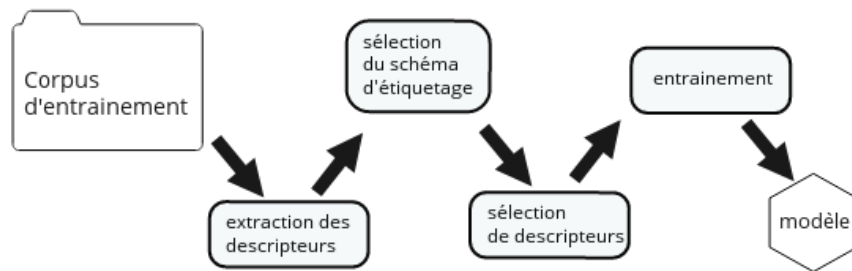


Figure 2.3 – Entraînement des modèles.

Les modèles HMM et CRF étant tous les deux supervisés, ils doivent être entraînés sur des exemples manuellement annotés pour estimer leurs paramètres. Nous proposons de sélectionner le schéma d'étiquetage et les sous-ensembles minimaux de caractéristiques manuellement définies, avant d'entraîner les modèles HMM et CRF (Figure 2.3).

2.3.1 Définition de descripteurs candidats

2.3.1.1 Descripteurs candidats pour la détection des sections

Nous considérons donc la ligne comme élément à étiqueter lors du sectionnement. Nous n'avons pas travaillé au niveau des mots afin d'éviter que des mots de la même ligne ne soient classés dans des sections différentes. L'étiquetage des phrases a été aussi évité car en découpant les documents en phrases telles qu'elles sont entendues en français, on a généralement des segments qui s'étendent d'une section à une autre (absence de ponctuation). de plus, l'entête en particulier a plus l'apparence d'un formulaire.

Plusieurs critères peuvent être utilisés pour différencier les sections, à savoir : la longueur des lignes (plus longues dans le corps, plus courtes dans

l'en-tête), les premiers termes de certaines lignes (typiques de chaque section) et le nombre total de lignes. Un HMM n'adapte qu'un descripteur assimilé à l'élément à étiqueter. D'autres descripteurs peuvent être la position de l'élément à étiqueter (numéro de ligne) ou le début de la ligne. Le descripteur capturant la longueur de ligne peut être absolue (nombre exact de mots dans la ligne) ou relative, en fonction de la catégorisation de la longueur de ligne. Sur la base des quantiles de distribution de longueurs de lignes sur un ensemble de décisions, nous avons défini trois catégories : LQ1 ($longueur \leq 5$), LQ2 ($5 < longueur \leq 12$) et LQ3 ($12 < longueur \leq 14$). Nous avons également catégorisé les parties de documents afin de capturer une position de ligne relative.

Lors de l'extraction des caractéristiques, le document est considéré comme divisé en N parties (10 dans nos expériences). La position relative d'une ligne est donc le numéro de la partie contenant la ligne particulière. En résumé, les caractéristiques sont décrites comme suit (avec leurs étiquettes entre parenthèses) :

- forme de la ligne : la ligne entière, ses premiers mots ($t0$, $t1$, $t2$), sa longueur absolue ($absLength$) et sa longueur relative ($relLength$) ;
- contexte de ligne : le numéro de ligne ($absNum$) et le numéro de la partie de document contenant la ligne ($relNum$), les deux premiers mots des lignes précédente ($p0$, $p1$) et suivantes ($n0$, $n1$), ainsi que leurs valeurs absolues respectives. longueurs relatives ($pLength$, $pRelLength$, $nLength$, $nRelLength$).

2.3.1.2 Descripteurs candidats pour la détections d'entités

La détection d'entités consiste à entraîner soit un modèle CRF, soit un modèle HMM pour étiqueter les différents segments de texte (mot, ponctuation, numéro, identifiant) suivant qu'ils appartiennent ou non à la mention d'une entité. Les deux modèles nécessitent des caractéristiques, dont certaines peuvent être définies sur la base de régularités directement observables dans

les textes. Il est également possible d'obtenir des descripteurs à partir du résultat d'autres tâches d'analyse de texte.

Sur la base des observations de décision, nous avons **dé** la morphologie des mots pour les normes et méta-données d'entête :

- forme du mot : le mot (**token**), son lemme (**lemma_W0**), « commence-t-il par une lettre majuscule ? » (**startsWithCAP**), « est-il entièrement en majuscule ? » (**isAllCAP**), « est-ce une initiale solitaire ? » comme par exemple « B. » (**isLONELYINITIAL**), « contient-il un caractère de ponctuation ? » (**PUN-IN**), « n'est-ce qu'une ponctuation ? » (**isALLPUN**), « contient-il un caractère numérique ? » (**DIGIT-IN**), « ne contient-il que des chiffres ? » (**isALLDIGIT**) ;
- contexte de mot : les mots précédents (**w-2**, **w-1**) et suivants (**w1**, **w2**) et leurs lemmes (**lemmaW_i**). La lemmatisation homogénéise les variantes du même mot. Les mots adjacents sont choisis pour indiquer les termes couramment utilisés pour introduire des entités.

Plus particulièrement pour les méta-données d'entête, nous avons défini des descripteurs supplémentaires pour capter le contexte du mot : numéro de ligne (**lineNum**), position de l'élément dans la ligne (**numInLine**), « le document contient-il le mot clé *intervenant* ? » (**intervenantInText**), le texte vient-il après le mot clé « APPELANT » (**isAfterAPPELANT**), « INTIME » (**isAfterINTIME**), « INTERVENANT » (**isAfterINTERVENANT**). Nous avons également pris en compte les dernières lignes, où le mot était précédemment rencontré dans le texte (**lastSeenAt**), ainsi que le nombre de fois où il a été trouvé (**nbTimesPrevSeen**), car les noms des parties sont souvent répétés à des emplacements différents. Nous avons également défini une caractéristique spéciale pour les normes : « le mots **s** est-il un mot clé de règles juridiques ? » (**isKEYWORD**). Pour ce dernier descripteur, nous avons établi une courte liste de mots-clés généralement utilisés pour citer des règles juridiques (*article, code, loi, contrat, décret, convention, civil, pénal*, etc.).

Nous avons étendus **s** ces caractéristiques avec les rôles grammaticaux (*Part-*

of-Speech et les modèles thématiques (*topic model*).

Rôles grammaticaux : Certaines entités ont tendance à contenir des rôles grammaticaux particuliers. Par exemple, les noms des **les** individus sont composés de noms propres (Chang et Sung, 2005). Nous avons extrait le rôle grammatical du mot courant (POS) ainsi que celui de ses voisins (POSW-2, POSW-1, POSW1, POSW2).

Modèles thématiques : comme Polifroni and Mairesse (2011) et Nallapati et al. (2010), nous utilisons des associations mot-thème pour décrire les mots. Il s'agit de modéliser un ensemble de N thèmes et d'utiliser leurs identifiants comme descripteurs. Il serait peut-être intéressant d'utiliser la probabilité déduite du modèle thématique, mais l'inférence sous-jacente au modèle LDA (Blei et al., 2003) n'est pas déterministe (la distribution de probabilité change pour le même mot entre différentes inférences). Néanmoins, l'ordre des sujets ne changeant pas de manière significative, nous avons utilisé l'identifiant du thème le plus pertinent pour le mot (`topic0`) ainsi que ceux de ses voisins (`w-2topic0`, `w-1topic0`, `w1topic0`, `w2topic0`).

2.3.2 Sélection des descripteurs

2.3.2.1 Sélection pour le modèle CRF

Nous avons étudié deux approches enveloppantes qui semblent toujours converger et qui ne nécessitent pas de définir manuellement la taille du sous-ensemble cible. Il s'agit de la recherche bidirectionnelle et de la sélection

séquentielle à flottement avant.

Algorithme 1 : Recherche bidirectionnelle BDS

Données : Données annotées, X liste de tous les descripteurs candidats

Résultat : Sous-ensemble optimal de descripteurs

```

1 Démarrer la SFS avec  $Y_{F_0} = \emptyset$ ;
2 Démarrer la SBS avec  $Y_{B_0} = X$ ;
3  $k = 0$ ;
4 tant que  $Y_{F_k} \neq Y_{B_k}$  faire
5    $x^+ = \operatorname{argmax}_{x \in Y_{B_k} \setminus Y_{F_k}} F1(Y_{F_k} + x); Y_{F_{k+1}} = Y_{F_k} + x^+;$ 
6    $x^- = \operatorname{argmax}_{x \in Y_{B_k} \setminus Y_{F_{k+1}}} F1(Y_{F_k} - x); Y_{B_{k+1}} = Y_{B_k} - x^-;$ 
7    $k = k + 1$ ;
8 retourner  $Y_{F_k}$ ;
```

La recherche bidirectionnelle (BDS), de Liu and Motoda (2012), combine la sélection séquentielle en avant (SFS) et la sélection séquentielle en arrière (SBS) en parallèle. La SFS recherche un sous-ensemble optimal, en commençant par un ensemble vide et en ajoutant le descripteur qui améliore le mieux la performance du sous-ensemble sélectionné. Le critère de performance dans notre cas est définie par la F1-mesure (Eq. 2.1). Contrairement à la SFS, la SBS commence par l'ensemble des candidats et supprime successivement les plus mauvais descripteurs. Les caractéristiques ajoutées par la SFS ne doivent pas faire partie de celles que la SBS a déjà supprimées. Le principe

de la recherche bidirectionnelle BDS est décrite par l'Algorithme 1.

Algorithme 2 : Sélection séquentielle avant à flottement

Données : Données annotées, X liste de tous les descripteurs candidats

Résultat : Sous-ensemble optimal de descripteurs

```

1  $Y_0 = \emptyset$ ;
2  $k = 0$ ;
3 répéter
4    $x^+ = \operatorname{argmax}_{x \notin Y_k} F1(Y_k + x); Y_k = Y_k + x^+$ ;
5    $x^- = \operatorname{argmax}_{x \in Y_k} F1(Y_k - x)$ ;
6   si  $F1(Y_k - x^-) > F1(Y_k)$  alors
7      $Y_{k+1} = Y_k - x^-$ ;
8      $X = X - x^-$ ;
9      $k = k + 1$ ;
10    Rentrer à 5;
11  sinon
12    Rentrer à 4;
13 jusqu'à  $X = \emptyset$  ou  $X = Y_k$ ;
14 retourner  $Y_k$ ;
```

L'algorithme de sélection séquentielle avant à flottement SFFS de Pudil et al. (1994) étend la SFS en surmontant son incapacité à réévaluer l'utilité d'un descripteur après son rejet. En effet, le SFFS effectue des tests en arrière à chaque itération comme décrit par l'Algorithme 2.

2.3.2.2 Sélection pour le modèle HMM

Pour sélectionner les meilleurs descripteurs pour les modèles HMM, nous avons testé individuellement les différents candidats. La caractéristique donnant le meilleur résultat sur l'ensemble de données annoté est ainsi sélectionnée.

2.4 Expérimentations et discussions

L'objectif de cette section est de discuter des différents aspects liés à la performance des modèles CRF et HMM. Il est question de discuter l'effet des descripteurs candidats définis, de comparer des algorithmes de sélection de caractéristiques et des schémas d'étiquetage. Nous discutons par la suite l'origine des erreurs (confusion, nombre d'exemples d'entraînement), et comparons les descripteurs définis manuellement par rapport à l'utilisation de réseaux de neurones.

2.4.1 Conditions d'expérimentations

2.4.1.1 Annotation des données de référence

Pour évaluer les méthodes de TAL, Xiao (2010) suggère de choisir un jeu d'exemples suffisant en assurant au mieux l'équilibre dans la variété des données et la représentativité du langage. Nous avons essayé de suivre cette recommandation en sélectionnant aléatoirement des décisions à annoter. Au total, 503 documents ont été rassemblés et annotés manuellement à l'aide de la plateforme GATE Developer¹. Cet outil permet de marquer les passages à annoter en les surlignant à l'aide du pointeur de la souris; ce qui allège l'annotation manuelle. Des balises XML sont rajoutées autour des passages sélectionnés, en arrière plan dans le document.

Chaque document annoté comprend en moyenne 262,257 lignes et 3955,215 mots. Les deux dernières colonnes du Tableau 2.1 présentent la distribution des entités labellisées dans le jeu de données. En se basant sur un sous-ensemble de 13 documents labellisés par 2 annotateurs différents, nous avons calculé des taux d'accord inter-annotateur en utilisant la statistique Kappa de Cohen. Ces mesures d'accord inter-annotateur ont été calculées au niveau des caractères parce que certains mots peuvent être coupés par des annotations

1. <https://gate.ac.uk/family/developer.html>

incorrectes (par ex. $\langle \textit{juridiction} \rangle \textit{ cour d'appe} \langle / \textit{juridiction} \rangle \textit{ l}$ contre $\langle \textit{juridiction} \rangle \textit{ cour d'appel} \langle / \textit{juridiction} \rangle$), ou bien les annotateurs pourraient ne pas être d'accord si un apostrophe doit être inclu ou pas dans l'annotation (par ex. $\textit{l}' \langle \textit{norme} \rangle \textit{article 700}$ contre $\langle \textit{norme} \rangle \textit{ l'article 700}$). Les taux de Kappa de 0,705 et 0,974 ont été obtenu pour l'annotation des entités et des sections respectivement. D'après la catégorisation de Viera et al. (2005), le niveau d'accord observé est *substantiel* pour les entités (0,61 – 0,80) et *presque parfait* pour les sections (0,81 – 0,99).

2.4.1.2 Mesures d'évaluation

Nous avons utilisé la précision, le rappel et la F1-mesure comme mesures d'évaluation car elles sont généralement utilisées comme référence en extraction d'information. Nous comparons aux niveaux atomique et entité que nous décrivons dans les paragraphes qui suivent. Nous présentons aussi des performances au niveau micro c'est-à-dire en général sans distinction des classes. A tous les niveau d'évaluation, la F1-mesure se calcule à l'aide de la formule 2.1.

$$F1 = 2 \times \frac{Precision \times Rappel}{Precision + Rappel} \quad (2.1)$$

Evaluation au niveau atomique (*token-level*) : Cette évaluation mesure la capacité d'un modèle à labelliser les segments atomiques des entités. Les valeurs de précision et rappel sont calculées sur les données de test pour chaque label l comme suit :

$$Precision_l = \frac{\text{nombre de segments correctement labélisés par le modèle avec } l}{\text{nombre de segments labélisés par le modèle avec } l}$$

$$Rappel_l = \frac{\text{nombre de segments correctement labélisés par le modèle avec } l}{\text{nombre de segments manuellement labélisés avec } l}$$

Evaluation au niveau entité (*entity-level*) : Cette évaluation mesure le

taux d'entités parfaitement identifiées c'est-à-dire seulement ceux dont les segments atomiques ont été tous correctement labellisés. Les valeurs de précision et rappel sont calculées sur les données de test pour chaque classe d'entité e comme suit :

$$Precision_e = \frac{\text{nombre d'entités de type } e \text{ parfaitement détectées par le modèle}}{\text{nombre d'entités détectées et classifiées } e \text{ par le modèle}}$$

$$Rappel_e = \frac{\text{nombre d'entités de type } e \text{ parfaitement détectées par le modèle}}{\text{nombre d'entités manuellement classifiées } e}$$

Evaluation globale (*overall-level*) : L'évaluation globale donne les performances générales d'un modèle sans distinction des classes ou labels. Elle est réalisée aux deux niveaux décrits précédemment mais indépendamment du label d'élément ou du type d'entité. La précision et le rappel sont calculées comme pour au niveau des entités (resp. au niveau des segments) suit :

$$Precision = \frac{\text{nombre d'entités correctement labellisées par le modèle}}{\text{nombre d'entités labellisées par le modèle}}$$

$$Rappel = \frac{\text{nombre d'entités correctement labellisées par le modèle}}{\text{nombre d'entités manuellement labellisées}}$$

2.4.1.3 Outils logiciels

Nous avons utilisé les modèles HMM et CRF tels qu'implémentés dans la librairie Mallet (McCallum, 2012). Les modèles étudiés ont été entraînés par la méthode d'espérance maximale pour ceux basés sur le HMM, et par la méthode L-BFGS pour ceux basés sur le CRF. Le découpage des textes en mots (*tokenisation*), ainsi que la lemmatisation et l'annotation des rôles grammaticaux (*Part-of-Speech tagging*) ont été effectués à l'aide de la fonctionnalité d'annotation de textes français de TreeTagger² (Schmid, 1994). L'implémentation dans Mallet du LDA (Blei et al., 2003) a permis d'inférer

2. <http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger>

100 thèmes à partir d'un corpus lemmatisé d'environ 6k documents. Le tableau 2.2 présente des mots représentatifs trouvés dans les premiers thèmes inférés. L'extraction des autres descripteurs a été implémentée pour cette expérimentation.

Id thème	Mots représentatifs
0	préjudice dommage somme subir réparation titre faute payer intérêt responsabilité
1	société salarié groupe mirabeau pouvoir demande article licenciement cour titre
2	harcèlement travail salarié moral employeur fait attestation faire santé agissements
3	vente acte prix vendeur acquéreur notaire condition clause vendre immeuble
4	travail poste reclassement employeur médecin licenciement salarié inaptitude visite
5	monsieur nîmes avocat appel barreau arrêt madame disposition prononcer président
6	mademoiselle madame non mesure décision tutelle surendettement comparant
7	transport marchandise jeune sed éducateur bateau navire transporteur responsabilité
8	congé salarié conversion emploi plan convention employeur sauvegarde reclassement
9	marque site contrefaçon sous droit auteur joseph produit propriété photographie
10	pierre patrick bordeaux bruno catherine civil article corinne cour avocat

Tableau 2.2 – Mots représentatifs des 10 premiers thèmes sur les 100 inférés

Les valeurs de précision, rappel, et F1-mesure ont été calculées à l'aide du script d'évaluation de la campagne CoNLL-2002³. Elles sont indiquées en pourcentage.

2.4.2 Sélection du schémas d'étiquetage

Dans le but d'évaluer comment la représentation de segment affecte les performances, nous avons implémenté quatre représentations (IO, IEO2, BIO2, BIEO). Nous avons réalisé un simple découpage des données en deux ensembles : 25% pour l'entraînement et 75% pour les tests. Les performances reportées dans le Tableau 2.3 sont les performances globales sur la base de test. Seul l'élément (mot/ligne) est utilisé comme descripteur. La durée d'entraînement est très longue, particulièrement pour la détection d'entité dans l'entête avec le CRF. Il semble évident que cette durée croît proportionnellement avec le nombre de labels candidats de la section et la complexité du schéma d'étiquetage. En effet, BIEO exige beaucoup plus de temps, et IO

3. <http://www.cnts.ua.ac.be/conll2002/ner/bin/conlleval.txt>

exige le temps d'entraînement le plus bas, et le schéma IOE semble être plus rapide à que BIO même s'ils ont le même nombre de labels. Nous remarquons aussi que les représentations complexes n'améliorent pas trop les résultats par rapport au simple IO qui demande pourtant beaucoup moins de temps.

Tâche	Modèle	Niveau atomique ^a			Niveau entité ^a			Durée ^b	Schéma
		Précision	Rappel	F1	Précision	Recall	F1		
Sections	CRF	91.75	91.75	91.75	64.49	56.55	60.26	4.685	IO
		88.95	88.95	88.95	48.12	38.26	42.63	11.877	IEO2
		87.09	87.09	87.09	46.79	37.20	41.45	12.256	BIO2
		86.00	86.00	86.00	58.98	41.86	48.97	35.981	BIEO
	HMM	32.64	32.64	32.64	22.16	18.91	20.41	6.564	IO
		32.92	32.92	32.92	17.73	16.09	16.87	7.827	IEO2
		32.39	32.39	32.39	31.93	26.65	29.05	8.391	BIO2
		33.06	33.06	33.06	32.47	27.53	29.80	8.7	BIEO
Entités d'entête	CRF	86.86	78.96	82.73	80.84	65.17	72.17	70.525	IO
		87.77	79.65	83.51	82.46	65.19	72.82	228.751	IEO2
		87.41	78.14	82.51	81.66	66.80	73.49	230.865	BIO2
		87.72	79.55	83.44	84.38	68.35	75.53	475.249	BIEO
	HMM	79.12	67.75	73.00	61.48	35.05	44.64	6.345	IO
		78.82	68.69	73.40	66.63	40.16	50.11	8.298	IEO2
		80.68	67.48	73.49	70.37	45.32	55.14	7.908	BIO2
		80.05	69.01	74.12	74.73	50.77	60.46	9.973	BIEO
Normes	CRF	95.60	92.96	94.26	88.06	83.50	85.72	28	IO
		95.40	93.18	94.27	88.75	85.65	87.17	32.136	IEO2
		95.20	93.30	94.24	85.65	83.13	84.37	50.769	BIO2
		95.46	91.57	93.47	88.83	84.71	86.72	50.566	BIEO
	HMM	89.83	88.78	89.30	73.74	75.02	74.37	41.389	IO
		88.20	89.23	88.71	78.01	81.27	79.61	44.086	IEO2
		89.25	87.83	88.53	73.89	76.63	75.24	46.634	BIO2
		87.39	88.10	87.74	77.76	82.35	79.99	45.52	BIEO

Tableau 2.3 – Comparaison des schémas d'étiquetage.

^a Résultats sur une simple division du jeu de données en 25% pour l'entraînement et 75% pour les tests (entraînement limité à 100 itérations au max)

^b Durée d'entraînement en secondes avant l'arrêt de l'entraînement

2.4.3 Sélection des descripteurs

Pour comparer les méthodes BDS et SFFS, nous exploitons le schéma IO. Durant nos expérimentations, la méthode SFFS a exécuté 185 entraînements pour le modèle CRF de d'identification des sections. La méthode BDS quant à elle à durée plus de 15h pour 600 itérations d'entraînement-test. Malgré la sauvegarde des scores F1 pour éviter d'exécuter plusieurs fois l'entraînement

pour les mêmes sous-ensembles de descripteurs, le processus de sélection est resté toujours très long pour les deux algorithmes. Nous avons testé individuellement chacun de descripteur candidat sur les modèles HMM. Les résultats sont reportés dans le Tableau 2.4.

Le plus remarquable dans ces résultats est la forte réduction du nombre de descripteurs par les algorithmes. En général, La moitié est éliminée par la sélection BDS, lorsque méthode SFFS élimine beaucoup plus (par exemple en ne sélectionnant que 4 descripteurs parmi les 14 candidats définis pour l'annotation des normes).

Par ailleurs, Les algorithmes de sélections forment des combinaisons inattendus. Par exemple, dans le cas de la détection de section, la ligne suivante semble être beaucoup plus indicatrice que la première. Il est aussi intéressant de noter que les descripteurs basés sur notre observation apparaissent dans les sous-ensembles sélectionnés (par ex. `isAfterIntervenant`, `isKEYWORD`). Remarquons aussi que la longueur absolue des lignes (`absLength`) joue un rôle important dans l'identification des sections vu qu'il a été sélectionné à la fois pour le CRF et le HMM (sélection BDS). Avec ses sous-ensembles sélectionnés, les modèles sont plus performants que lorsqu'ils ne doivent exploiter qu'uniquement le segment ou l'ensemble tout entier des candidats. Cette amélioration des résultats n'est pas très importante au regard de la longue durée d'exécution des algorithmes. Ainsi, un algorithme plus rapide et plus efficace devrait être utilisé.

2.4.4 Evaluation détaillée pour chaque classe

Nous discutons ici la capacité des modèles à identifier individuellement chaque type d'entité et de section. Les expérimentations ont été réalisées avec tous les descripteurs pour les modèles CRF. Seuls `absLength` et `token` ont été utilisés comme descripteurs dans les modèles HMM pour l'identification des sections et des entités respectivement. Le schéma d'étiquetage est IO. Le nombre d'itération maximal a été fixé à 500 pour assurer la conver-

Tâche	Modèle	niveau atomique ^a			niveau entité ^a			Sous-ensemble sélectionné
		Précision	Rappel	F1	Précision	Rappel	F1	
Sections	CRF	99.31	99.31	99.31	90.28	90.68	90.48	BDS ^{b1}
		99.55	99.55	99.55	85.69	85.84	85.76	SFFS ^{b2}
		99.36	99.36	99.36	88.16	88.39	88.27	TOUS ^{b0}
		91.75	91.75	91.75	64.49	56.55	60.26	token
	HMM	90.99	90.99	90.99	4.18	3.63	3.89	absLength
		86.97	86.97	86.97	4.08	3.30	3.65	relLength
		37.59	37.59	37.59	18.81	18.81	18.81	token
Entités d'entête	CRF	94.00	91.42	92.69	92.26	88.76	90.47	BDS ^{c1}
		94.10	91.93	93.00	92.64	88.96	90.76	SFFS ^{c2}
		94.20	91.86	93.02	93.05	89.59	91.28	TOUS ^{c0}
		86.86	78.96	82.73	80.84	65.17	72.17	token
	HMM	76.90	80.41	78.61	62.66	52.16	56.93	token
		66.48	69.67	68.04	39.34	28.36	32.96	lemma_W0
		39.63	37.50	38.54	15.49	5.35	7.95	POS
Normes	CRF	95.91	96.72	96.31	91.14	90.45	90.80	BDS ^{d1}
		95.68	95.45	95.57	90.34	88.27	89.29	SFFS ^{d2}
		95.07	96.69	95.87	90.87	90.64	90.76	TOUS ^{d0}
		95.60	92.96	94.26	88.06	83.50	85.72	token
	HMM	89.21	94.25	91.66	72.67	77.28	74.90	token
		90.31	92.81	91.54	69.24	69.46	69.35	lemma_W0

^a Résultats sur un simple découpage des données de 25% pour l'entraînement, 75% pour le test avec 100 itérations d'entraînement au maximum pour le CRF, et 80% pour l'entraînement et 20% pour le test avec 50 itérations au maximum pour l'entraînement du HMM

^{b0} Tous les candidats définis pour les sections (16 descripteurs) : { relNum, relLength, pRelLength, absLength, t0, t1, t2, absNum, pLength, nRelLength, n0, nLength, p0, p1, n1, token }

^{b1} Sélection par BDS pour les sections (07 descripteurs) : { p0, n0, relNum, absLength, t0, t1, t2 }

^{b2} Sélection par SFFS pour les sections (06 descripteurs) : { n0, nRelLength, relNum, t0, t1, t2 }

^{c0} Tous les candidats définis pour les méta-données d'entête (34 descripteurs) : { isLONELYINITIAL, isALLCAP, isALLDIGIT, DIGIT-IN, intervenantInText, lineNum, lastSeenAt, nbTimesPrevSeen, isAfterAPPELANT, isAfterINTIME, isAfterINTERVENANT, startsWithCAP, PUN-IN, isALLPUN, POSW2, w2topic0, numInLine, POSW-1, lemmaW2, lemmaW-2, POSW-2, w-2topic0, POSW1, w1topic0, token, POS, lemma_W0, topic0, w2, w-1topic0, lemmaW-1, w-1, w1, lemmaW1 }

^{c1} Sélection par BDS pour les méta-données d'entête (17 descripteurs) : { POSW1, isAfterAPPELANT, numInLine, w-2topic0, POSW2, isAfterINTERVENANT, isAfterINTIME, POSW-2, isLONELYINITIAL, token, lemma_W0, lemmaW-2, isALLPUN, w-1, w1, w2, isALLCAP }

^{c2} Sélection par SFFS pour les entités d'entête (10 descripteurs) : { numInLine, w-2topic0, lemmaW-2, isAfterINTERVENANT, isAfterINTIME, w-1, w1, w2, isALLCAP, token }

^{d0} Tous les candidats définis pour les normes (28 descripteurs) : { isALLPUN, isALLDIGIT, DIGIT-IN, isKEYWORD, POSW2, w2topic0, PUN-IN, POSW-1, isLONELYINITIAL, startsWithCAP, isALLCAP, lemmaW-2, POSW-2, w-2topic0, POS, topic0, POSW1, w1topic0, w2, lemmaW2, token, lemma_W0, w-2, w-1topic0, w-1, lemmaW-1, w1, lemmaW1 }

^{d1} Sélection par BDS pour les normes (14 descripteurs) : { POSW1, w-2topic0, isKEYWORD, lemmaW2, DIGIT-IN, token, lemmaW1, lemmaW-2, POS, isALLPUN, w-1, w2, PUN-IN, w-2 }

^{d2} Sélection par SFFS pour les normes (04 descripteurs) : { POSW1, lemmaW-2, w-1, DIGIT-IN }

Tableau 2.4 – Performances des sous-ensembles sélectionnés de descripteurs.

gence lors de l'entraînement même si les modèles HMM ne convergeaient jamais après 500 itérations. Les Tableaux 2.5 et 2.6 présentent les résul-

tats d'une validation croisée à 5 itérations, respectivement aux niveaux atomique et entité. D'un point de vue général (évaluation globale), les modèles HMM se comportent assez bien au niveau élément avec un seul descripteur, particulièrement pour l'identification des sections et des normes. Le modèle HMM est capable de labelliser les normes car plusieurs d'entre elles sont répétées entre les décisions. De plus, la citation des normes est quasi standard (`article [IDENTIFIANT] [TEXTE D'ORIGINE]`). Le modèle HMM n'est cependant pas **autant efficace** pour détecter entièrement les mots des entités d'où le faible score enregistré au niveau entité. Quant aux modèles CRF, leurs résultats sont très bons sur toutes les tâches et à tous les niveaux d'évaluation malgré quelques limites observées sur l'identification des parties.

	HMM			CRF		
	<i>Precision</i>	<i>Rappel</i>	<i>F1</i>	<i>Precision</i>	<i>Rappel</i>	<i>F1</i>
I-corps	92.46	95.25	93.83	99.57	99.69	99.63
I-dispositif	53.44	48.46	50.83	98.63	97.59	98.11
I-entete	97.91	91.93	94.83	99.51	99.55	99.53
Evaluation globale	90.63	90.63	90.63	99.48	99.48	99.48
I-appelant	34.46	16.87	22.65	84.34	76.27	80.1
I-avocat	85.17	98.75	91.46	98.02	98.15	98.09
I-date	75.67	72.45	74.02	98	96.6	97.3
I-fonction	88.81	64.46	74.7	95.23	95.13	95.18
I-formation	79.38	94.38	86.23	98.8	99.45	99.12
I-intervenant	82.07	38.04	51.98	83.38	68.26	75.07
I-intime	50.4	68.09	57.93	82.54	83.33	82.93
I-juge	73.4	88.73	80.34	97.55	97.23	97.39
I-juridiction	85.15	98.37	91.28	98.91	99.69	99.3
I-rg	68.53	22.14	33.47	97.81	97.44	97.62
I-ville	91.5	82.41	86.72	98.94	99.15	99.04
Evaluation globale	76.21	82.26	79.12	95.13	94.51	94.82
I-norme	88.23	93.7	90.89	97.14	96.09	96.62

Tableau 2.5 – Précision, Rappel, F1-mesures pour chaque type d'entité et section au niveau atomique.

	HMM			CRF		
	<i>Precision</i>	<i>Rappel</i>	<i>F1</i>	<i>Precision</i>	<i>Rappel</i>	<i>F1</i>
corps	0.99	0.99	0.99	89.57	90.1	89.83
dispositif	12.05	7.33	9.11	98.02	97.82	97.92
entete	10.47	10.5	10.48	92.11	92.48	92.29
Evaluation globale	7.22	6.27	6.71	93.22	93.47	93.34
appelant	17.84	5.6	8.52	84.05	77.29	80.53
avocat	44.29	39.15	41.56	90.97	90.3	90.63
date	66.87	62.15	64.43	97.96	96.6	97.27
fonction	89.84	64.13	74.84	96.89	96.94	96.92
formation	61.5	65.86	63.61	98.4	98.95	98.68
intervenant	14.29	4	6.25	62.5	40	48.78
intime	30.28	27.47	28.8	79.31	78.93	79.12
juge	73.54	83.21	78.07	96.58	96.35	96.47
juridiction	81.31	87.66	84.37	98.86	99.54	99.2
rg	68.53	22.41	33.77	97.57	98.02	97.79
ville	89.52	84.7	87.05	98.85	99.15	99
Evaluation globale	64.59	54.56	59.15	93.77	92.93	93.35
norme	71.94	78.45	75.05	92.66	91.38	92.01

Tableau 2.6 – Précision, Rappel, F1-mesures pour chaque type d’entité et section au niveau entité.

2.4.5 Discussions

2.4.5.1 Confusion de classes

Certaines erreurs sont probablement dues à la proximité des entités de types différents. D’après la matrice de confusion des méta-données d’entête (Figure 2.4), les *intervenants* sont parfois classifiés comme *appelant*, *intimé* ou *avocat* probablement parce qu’il s’agit d’entités mentionnés les uns à la suite des autres dans l’entête (les *intervenants* sont mentionnés juste après les *avocats* des *intimés*). De plus, les intervenants apparaissent dans une très faible proportion de documents annotés. Par ailleurs, une quantité considérable d’appelants sont aussi classifiées comme *intimés*.

La proximité crée aussi des confusions entre les sections CORPS et DISPOSITIF qui se suivent (Figure 2.5).

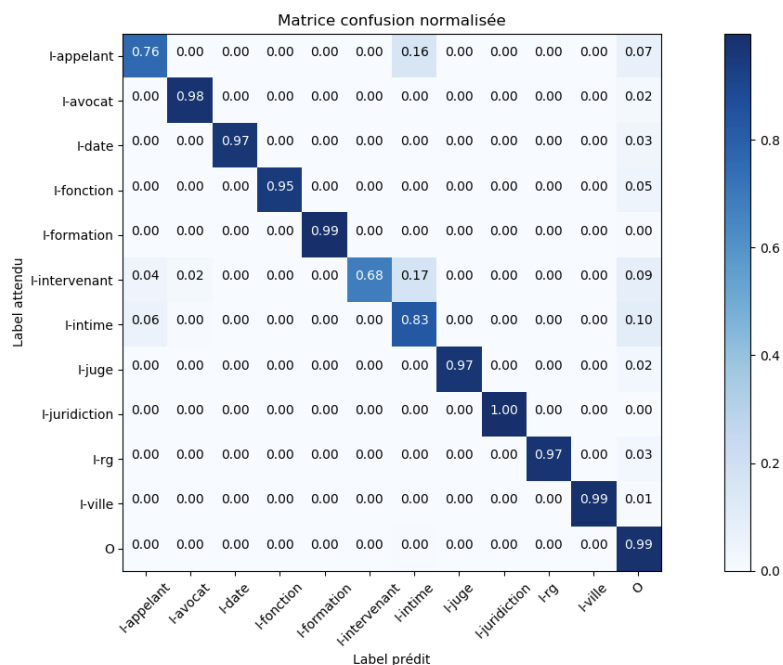


Figure 2.4 – Matrice de confusion entre méta-données d’entête avec le modèle CRF

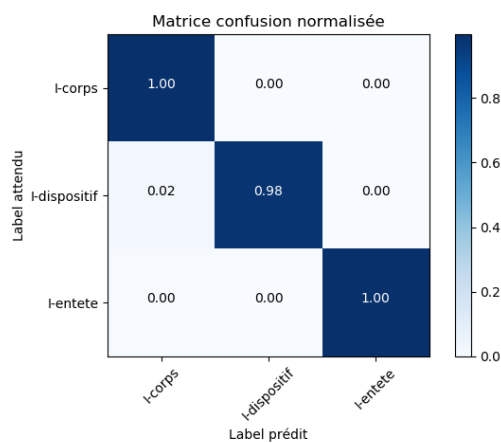


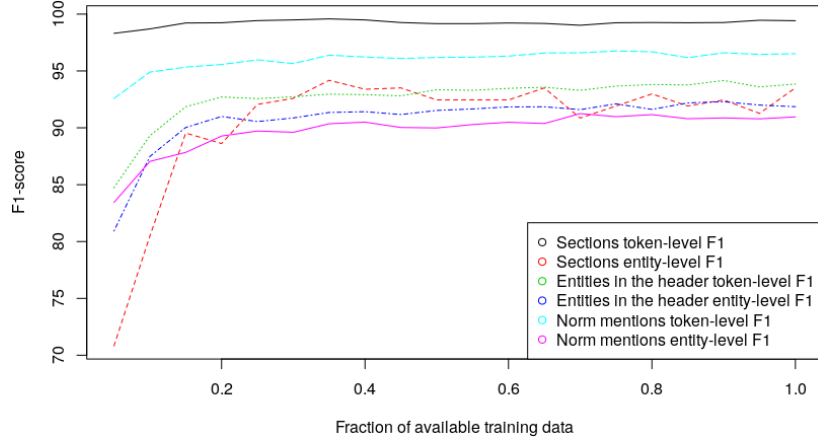
Figure 2.5 – Matrice de confusion entre lignes des sections avec le modèle CRF

2.4.5.2 Redondance des mentions d'entités

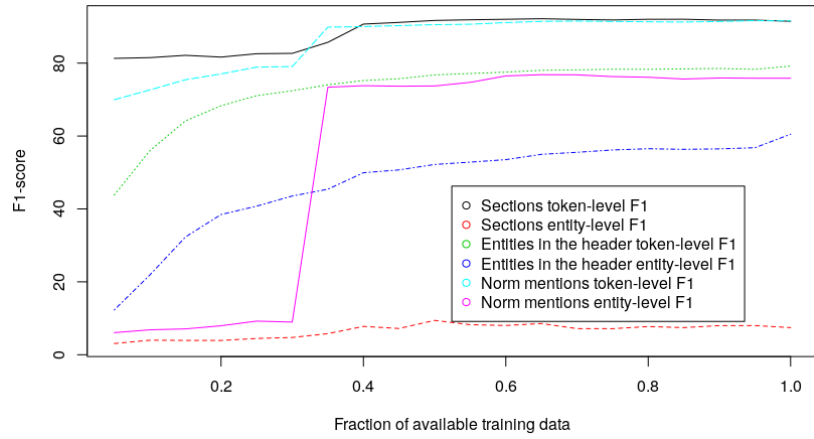
Il est aussi intéressant de remarquer que certaines entités sont répétées dans le document. Par exemple, les noms des parties apparaissent précédemment à une mention qui donne plus de détails. Certaines normes sont aussi citées de plusieurs fois et en alternant souvent les formes allongées et des longues (par exemple, la juridiction, la date, les normes). Malgré le fait que les mentions répétées ne sont pas identiques, de telles redondance aident à réduire le risque de manquer une entité. Cet aspect peut être exploité afin de combler l'imperfection des modèles.

2.4.5.3 Impact de la quantité d'exemples annotées

Des expérimentations ont été menées pour évaluer comment les modèles s'améliorent lorsqu'on augmente le nombre de données d'entraînement. Pour cela, nous avons évalué différentes tailles de la base d'entraînement. Les données ont été divisées en 75% – 25% pour resp. l'entraînement et le test. 20 fractions de l'ensembles d'entraînement ont été utilisées (de 5% à 100%). A chaque session entraînement-test, le même jeu de test a été employé pour les différentes fractions de l'ensemble d'entraînement. Les courbes d'apprentissage des modèles CRF et HMM sont représentées resp. sur les Figures 2.6a et 2.6b. Il est évident que les scores F1 croissent avec le nombre de données d'entraînement pour les CRF et HMM, mais cette amélioration devient très faible au-delà de 60% de données d'entraînement quelque soit la tâche. Il est possible que les exemples rajoutés à partir de là partagent la même structure qu'une majorité d'autres. Ainsi, cette étude doit être étendue à la sélection des exemples les plus utiles. Raman and Ioerger (2003) ont démontré les avantages des algorithmes de sélection d'exemples combinés à la sélection de caractéristique pour la classification. Les mêmes méthodes sont probablement applicables à l'étiquetage de séquence.



(a) CRF



(b) HMM

Figure 2.6 – Courbes d'apprentissages aux niveaux élément et entité

2.4.5.4 Descripteurs manuels vs. réseau de neurones

L'ingénierie manuelle des caractéristiques est difficile car arbitraire. Nous avons comparé les performances de nos descripteurs avec celles des réseaux de neurones qui apprennent une représentation des segments. Pour cela nous avons choisi le BiLSTM-CRF de Lample et al. (2016) qui fait partie des meilleures approches récentes. La comparaison a été effectuée pour la détec-

	CRF + descripteurs manuels			BiLSTM-CRF		
	<i>Precision</i>	<i>Rappel</i>	<i>F1</i>	<i>Precision</i>	<i>Rappel</i>	<i>F1</i>
appellant	82.49	69.42	74.72	80.26	71.53	75.04
avocat	90.15	89.02	89.56	84.93	87.88	86.36
date	95.34	91.46	93.12	95.04	90.79	92.63
fonction	95.87	95.08	95.44	92.69	93.48	93.03
formation	96.91	91.31	93.7	91.05	89.47	89.84
intervenant	51.42	32.71	36.8	31.48	20	23.11
intime	76.01	79.15	77.22	67.7	75.43	70.83
juge	95.67	94.07	94.84	95.44	95.56	95.46
juridiction	98.55	98.25	98.33	97.95	99.22	98.57
rg	95.46	95.29	95.27	91.13	97.26	93.92
ville	98.33	93.01	94.71	91.43	95.34	93.3
norme	91.08	90.27	90.67	91.43	92.65	92.03
Evaluation globale	92.2	90.09	91.12	89.21	90.43	89.81

Tableau 2.7 – Comparaison du CRF avec descripteurs manuellement défini et le BiLSTM-CRF au niveau entité.

tion des entités avec le schéma d’étiquetage BIEO et une validation croisée à 9 itérations. Le BiLSTM-CRF prend en entrée les plongements sémantiques Word2Vec des mots. Pour cela, nous avons entraîné des vecteurs de mots à partir d’un corpus jurisprudentiel de plus de 800K documents provenant de www.legifrance.gouv.fr avec l’implémentation⁴ de Le and Mikolov (2014). Les vecteurs obtenus ont une dimension de 300. Etant donné que les décisions sont des documents particulièrement longs, leur contenu a été découpé en des morceaux de texte dont la taille n’excède pas 300 mots. Les résultats obtenus par le BiLSTM-CRF sont assez proches de ceux que nous observons avec les descripteurs manuellement définis (Tableau 2.7) . Etant donné que ces derniers permettent de mieux détecter certaines entités comme les *intervenants*, les *avocats* ou les numéro *R.G.*, et vice-versa pour les *normes* ou les *appelants* chez le BiLSTM-CRF, une combinaison des deux types de descripteurs pourrait améliorer les résultats actuels.

4. <https://code.google.com/archive/p/word2vec/>

2.5 Conclusion

L'application des modèles HMM et CRF dans le but de détecter des sections et des entités dans les décisions de justice est une tâche difficile. Ce chapitre a examiné les effets de divers aspects de la conception sur la qualité des résultats. En résumé, malgré une importante réduction du nombre de descripteurs, l'amélioration des résultats semble être insignifiante lorsque l'on sélectionne séparément la représentation du segment et le sous-ensemble de caractéristiques. Cependant, opter pour la bonne configuration en évaluant les approches de sélection combinées avec diverses représentations de segment pourrait peut-être offrir de meilleurs résultats. En raison de la longue durée de recherche du sous-ensemble optimal de descripteurs, il serait préférable d'utiliser un algorithme de sélection beaucoup plus rapide que les méthodes BDS et SFFS que nous avons expérimentées. De plus, même si les résultats s'améliorent avec la croissance de l'échantillon d'apprentissage, la mesure globale F1 semble néanmoins atteindre une limite très rapidement. Étant donné que certaines entités ne sont pas très bien détectées, il peut être avantageux d'ajouter des exemples appropriés afin de traiter ces problèmes spécifiques.

L'application des modèles pose deux difficultés majeures : l'annotation d'un nombre suffisant d'exemples et la définition de caractéristiques discriminantes. Les efforts d'annotation peuvent être réduits avec un système automatique à faible performance d'étiquetage. Il suffirait alors de vérifier manuellement ces annotations afin de corriger les erreurs commises par le système sur de nouvelles décisions à l'aide d'un outil d'aide à l'annotation. En ce qui concerne la définition des caractéristiques, dans la mesure où notre approche actuelle est réalisée manuellement par l'analyse de quelques documents, il est possible que de tels descripteurs ne s'adaptent pas parfaitement à un nouvel ensemble de données (différents pays, différentes langues, différentes juridictions). Pour éviter les énormes efforts requis pour définir les fonctionnalités manuellement, il serait préférable d'utiliser des descripteurs

automatiquement apprises à partir de corpus étiquetés ou non, comme des mots incorporés.

Dans les travaux futurs, Il serait intéressant d'achever la tâche de reconnaissance d'entités nommées. Pour l'indexation des décisions dans une base de connaissances, il est en effet essentiel de définir des méthodes de désambiguïsation et de résolution pour les entités à occurrences multiples, en plus de la correspondance des entités extraites avec des entités de référence, comme l'ont expérimenté Dozier et al. (2010) et Cardellino et al. (2017). Ces travaux peuvent être poursuivis par d'autres applications telles que l'anonymisation automatique qui aiderait à publier plus rapidement l'énorme volume de décisions prononcées régulièrement.