

Chapitre 1

Analyse automatique de corpus judiciaires

L'étude bibliographique de ce chapitre est focalisée sur l'application de techniques d'analyse de données textuelles judiciaires. Une synthèse bibliographique plus technique sur les algorithmes de fouille de texte est détaillée dans les chapitres qui traitent, dans la suite, des méthodes que nous avons mises en œuvre. Plus précisément, suivant la structure du présent chapitre, il s'agit des chapitres 2 et ?? pour l'extraction d'information, du chapitre ?? pour la classification des documents, et du chapitre ?? pour la similarité entre documents.

1.1 Introduction

Les deux grands paradigmes de jugement se distinguent par l'importance qu'ils accordent aux règles juridiques [Tumonis, 2012]. D'une part, les adeptes du Formalisme Juridique, plus pertinent dans le droit civil, considèrent que toutes les considérations normatives ont été incorporées dans les lois par leurs auteurs. D'autre part, l'école du Réalisme Juridique, plus proche du « *Common Law* », permet un pouvoir discrétionnaire entre les jugements en raisonnant selon le cas. Les premières tentatives d'anticipation des comportements judiciaires s'appuyaient sur une formalisation des lois. Il en est né le « droit computationnel », qui est une sous-discipline de l'« informatique juridique¹ ». Il s'intéresse, en effet, au raisonnement juridique automatique axé sur la représentation sémantique riche et plus formelle de la loi, des régulations, et modalités de contrat [Love & Genereth, 2005]. Il vise à réduire la taille et la complexité de la loi pour la

1. Application des techniques modernes de l'informatique à l'environnement juridique, et par conséquent aux organisations liées au droit.

rendre plus accessible. Plus précisément, le « droit computationnel » propose des systèmes répondant à différentes questions, comme « Quel montant de taxe dois-je payer cette année ? » (planification juridique), « Cette réglementation contient-elle des règles en contradiction » (analyse réglementaire), « L'entreprise respecte-t-elle la loi ? » (vérification de la conformité) [Genesereth, 2015]. Les techniques pro Formalisme Juridique étaient déjà critiquées au début des années 60, parce qu'excessivement focalisées sur les règles juridiques qui ne représentent qu'une partie de l'institution juridique [Llewellyn, 1962]. Pour analyser le comportement judiciaire, plusieurs variables plus ou moins contrôlables, comme le temps, le lieu et les circonstances, doivent aussi être prises en compte [Ulmer, 1963]. Etant donné que les juristes s'appuient sur la recherche de précédents, Ulmer [1963] conseille de se concentrer sur les motifs réguliers que comprennent les données pour réaliser des analyses quantitatives. Il est possible d'exploiter la masse de décisions pour identifier de telles régularités car une collection suffisante d'une certaine forme de données révèle des motifs qui une fois observés sont projetables dans le futur [Ulmer, 1963]. Il s'agit de raisonnements à base de cas qui se distinguent de ceux à base de règles.

Les premiers outils automatiques d'anticipation des décisions étaient généralement des systèmes experts juridiques. Ces derniers raisonnent sur de nouvelles affaires en imitant la prise de décision humaine par la logique en général et souvent par analogie. Ils s'appuient sur un raisonnement à base de règles, c'est-à-dire à partir d'une représentation formelle des connaissances des experts ou du domaine. En droit, il s'agit de la connaissance qu'a l'expert des normes juridiques, et de l'ordre des questions à traiter lors du raisonnement sur un cas (appris par expérience). Le modèle explicite de domaine nécessaire ici se trouve dans une base de connaissances où les normes juridiques sont représentées sous forme de « SI ... ALORS ... », et les faits sont généralement représentés dans la logique des prédicats. Un système expert juridique doit s'appuyer sur une base de connaissances juridiques exhaustive et disposer d'un moteur d'inférence capable de trouver les règles pertinentes et le moyen efficace, par déduction, de les appliquer afin d'obtenir la solution du cas d'étude aussi rapidement que possible. Les systèmes experts ont échoué dans leur tentative de prédire les décisions de justice [Leith, 2010]. La première raison découle de ce que Berka [2011] a appelé le « goulot d'acquisition de connaissances » c'est-à-dire le problème d'obtention des connaissances spécifiques à un domaine d'expertise sous la forme de règles suffisamment générales.

L'autre raison tient à l'interprétation ouverte du droit et à la complexité de la formalisation applicable sans tenir compte des particularités de l'affaire.

Contrairement au raisonnement à base de règles, le raisonnement à base de cas concerne une recherche de solution, une classification ou toute autre inférence pour un cas courant à partir de l'analyse d'anciens cas et de leurs solutions [Moens, 2002]. Un tel système juridique résout les nouveaux cas en rapprochant les cas déjà réglés et en adaptant leurs décisions [Berka, 2011]. Le raisonnement fondé sur des cas connaît un succès croissant dans la prédiction de l'issue d'affaires, davantage aux États-Unis qu'ailleurs. Pour exemple, Katz *et al.* [2014] entraînent des forêts aléatoires [Breiman, 2001] sur les cas de 1946-1953 pour prédire si la Cour Suprême des États-Unis infirmera ou confirmera une décision de juridiction inférieure. Leur approche parvient à prédire correctement 69,7% des décisions finales pour 7700 cas des années 1953-2013. Ils ont légèrement amélioré ce résultat plus récemment en augmentant le nombre d'arbres et la quantité de données [Katz *et al.*, 2017]. Toujours pour la prédiction des décisions de la Cour Suprême des États-Unis, Walzl *et al.* [2017b] utilisent des techniques de traitement automatique du langage naturel (TALN) pour extraire moins d'attributs caractéristiques de décision que [Katz *et al.*, 2014] à partir des décisions d'appel de la Cour Fiscale Allemande (11 contre 244). Ils obtiennent des valeurs de F_1 -mesures entre 0,53 et 0,58 (validation croisée à 10 itérations) pour la prédiction de la confirmation ou l'infirimation d'un jugement en appel avec un classifieur bayésien naïf. Par ailleurs, Ashley & Brüninghaus [2009] ont obtenu une précision de 91,8% pour la prédiction de la partie (plaignant/défendeur) qui sera favorisée à l'issue d'affaires d'appropriation illicite de secrets commerciaux.

Notre objectif est d'alimenter les analyses quantitatives de corpus jurisprudentiels en proposant des méthodes d'extraction de connaissances pertinentes telles que les références des affaires (juge, date, juridiction, etc.), les règles juridiques associées, les demandes des parties, les réponses des tribunaux, et les liens entre ces données. Les juges apportent une réponse à chaque demande. Par conséquent une partie peut voir chacune de ses demandes individuellement acceptée ou rejetée partiellement ou entièrement. Un juriste sera donc plus intéressé à formuler et défendre les demandes qui ont de meilleures chances d'être acceptées pour un type de contentieux précis plutôt que de prévoir une victoire du procès. C'est la raison pour laquelle notre analyse se situe à un niveau de granularité plus fin (la demande), contrairement aux travaux sur la prédiction qui

traitent d'un résultat global sur la décision (par ex. confirmer/infirmier ou gagner/perdre). Un des postulats considérés dans cette thèse est que l'identification de ces diverses connaissances est possible par l'analyse sémantique des textes judiciaires grâce aux méthodes du TALN. Cependant, l'application de ces techniques exigent certaines adaptations pour surmonter les divers défis décrits par Nazarenko & Wyner [2017] : textes très longs et en grande quantité, corpus régulièrement mis à jour, influence subjective de facteurs sociaux et d'opinions politiques, couverture de problématiques économiques, sociales, politiques très variées, langage complexe, etc.. Dans la suite de ce chapitre, nous passons en revue des travaux qui ont été menés dans ce sens pour traiter des problématiques proches des nôtres, en particulier celles décrites dans l'introduction (??).

1.2 Annotation et extraction d'information

L'annotation consiste à enrichir les documents pour les préparer à des analyses, faciliter la recherche d'affaires pertinentes, et faire la lumière sur des connaissances linguistiques sous-jacentes au raisonnement juridique. Les éléments annotés peuvent être de courts segments de texte mentionnant des entités juridiques [Waltl *et al.*, 2016; Wyner, 2010] comme la date, le lieu (juridiction), les noms de juges, des citations de loi. L'annotation de passages plus longs consiste à identifier des instances de concepts juridiques plus complexes comme les faits [Wyner, 2010; Wyner & Peters, 2010; Shulayeva *et al.*, 2017], les définitions [Waltl *et al.*, 2016, 2017a], des citations de principes juridiques [Shulayeva *et al.*, 2017], ou des arguments [Wyner *et al.*, 2010].

Différentes méthodes ont été expérimentées pour la reconnaissance d'information dans les documents judiciaires. La plupart reposent sur l'entraînement d'algorithmes d'apprentissage automatique supervisé sur un ensemble d'exemples annotés manuellement (résultats attendus). Parmi ces algorithmes, on retrouve par exemple les modèles probabilistes HMM (Modèles Cachés de Markov, cf. § 2.2.1.1) et CRF (Champs Aléatoires Conditionnels, cf. § 2.2.1.2) que nous étudions au chapitre 2. Ces modèles peuvent être combinés à d'autres approches dans un système global. En effet, après avoir segmenté les documents à l'aide d'un modèle CRF, Dozier *et al.* [2010] ont par exemple combiné plusieurs approches pour reconnaître des entités dans les décisions de la Cour Suprême des États-Unis. Ils ont dé-

fini manuellement des détecteurs distincts à base de règles pour identifier séparément la juridiction (zone géographique), le type de document, et les noms des juges, en plus de l'introduction d'une recherche lexicale pour détecter la cour, ainsi qu'un classifieur entraîné pour reconnaître le titre. Ces différents détecteurs ont atteint des performances prometteuses, mais avec des rappels limités entre 72% et 87%. Suivant la complexité des éléments à extraire, un système peut exploiter un lexique pour les motifs simples et non-systématiques (indicateurs de mentions de résultats ou de parties) et des règles pour des motifs plus complexes et systématiques (e.g., noms de juges, énoncés de décisions) [Waltl *et al.* , 2016, 2017a; Wyner, 2010]. Cardellino *et al.* [2017] ont par ailleurs utilisé un modèle CRF et des réseaux de neurones pour la reconnaissance d'entités nommées juridiques dans des jugements de la Cour Européenne des Droits de l'Homme. Ils définissent une hiérarchie des entités nommées distinguant au niveau 1, les entités nommées et des non-entités, spécialisées par 6 classes au niveau 2 (par exemple, Personne, Document), spécialisées par 69 classes au niveau 3 (par exemple, Rôle Juridique, Règlement), spécialisées par 358 classes au niveau 4 (par exemple Juge, Code Juridique). Les basses performances qu'ils rapportent sur le corpus juridique illustrent bien la difficulté de la détection d'entités juridiques dans les décisions judiciaires (F_1 -mesures de 0.25, 0.08, 0.03 en moyenne respectivement pour les niveaux 2, 3, 4). Plus récemment encore, Andrew & Tannier [2018] proposent une approche pour l'extraction d'entités nommées d'une transaction d'investissement² et des relations qu'elles partagent dans des décisions du Luxembourg rédigées en français. Ils combinent un modèle CRF pour les entités à une grammaire GATE JAPE [Thakker *et al.* , 2009] pour les relations, et obtiennent un faible taux d'erreur pour le CRF de 3.12%.

Pour la détection des arguments, par contre, Moens *et al.* [2007] proposent une classification binaire des phrases : *argumentative* / *non argumentative*. Ils comparent notamment le classifieur bayésien multinomial et le classifieur d'entropie maximum tout en explorant plusieurs caractéristiques textuelles. Mochales & Moens [2008] proposent, pour la même tâche, une méthode d'extraction basée sur une formalisation de la structure des arguments dans les jugements par une grammaire sans contexte.

2. Entités : Personne, Nom, Adresse, Société Principale, Société Secondaire, Rôle, Fonction, Type Société.

1.3 Classification des jugements

La classification de texte permet d'organiser un corpus en rangeant les documents dans des catégories généralement prédéfinies par des experts. Pour la classification des décisions, le principe des propositions de la littérature est d'entraîner un modèle statistique traditionnel sur une représentation des documents généralement définie à partir des connaissances du domaine. Par exemple, par classification binaire avec une Machine à Vecteurs de Support (SVM) [Vapnik, 1995] à noyau linéaire (cf. ??), Aletras *et al.* [2016] identifient s'il y a eu une violation d'un article donné de la convention des droits de l'homme sur les jugements de la Cour Européenne des Droits de l'Hommes (CEDH)³. Les vecteurs représentant les documents sont construits sur la base des 2000 n-grammes les plus fréquents. Certaines composantes sont les fréquences normalisées des n-grammes sélectionnés (modèle sac-de-mots [Salton *et al.*, 1975; Salton & McGill, 1983]), calculées distinctement pour différentes parties du document (Procédure, Circonstances, Faits, Loi applicable, la Loi et le document entier); ce qui résulte en une matrice document-terme C . D'autres composantes sont définies par la fréquence des thématiques extraites par une catégorisation non supervisée (*clustering*) avec la similarité cosinus des n-grammes les plus fréquents représentés par leurs vecteurs dans C , i.e. le vecteur de leurs scores d'occurrence dans les différentes parties précédemment citées du document. Aletras *et al.* [2016] obtiennent une précision moyenne de 79% sur les 3 articles qu'ils ont manipulés. Notons tout de même la sélection des régions particulières (circonstances, faits, lois, ...) du document à partir desquelles sont extraits les n-grammes. Cette sélection est un ajustement de la représentation des textes qui paraît nécessaire pour obtenir de bons résultats. La structuration préalable des documents est ainsi utile pour réduire le bruit qui occupe généralement plus d'espace que les passages ou éléments d'intérêt. Medvedeva *et al.* [2018] étendent ces travaux à neuf articles de loi, tout en montrant empiriquement, entre autres, la possibilité de prédire la violation des articles sur des périodes futures à celles couvertes par les données utilisées lors des phases d'entraînement. Şulea *et al.* [2017a] traitent, d'autre part, l'identification des résultats dans des arrêts⁴ de la Cour Française de Cassation.

3. HUDOC ECHR Database : <http://hudoc.echr.coe.int>.

4. Documents de <https://www.legifrance.gouv.fr>.

Après un essai [Şulea *et al.*, 2017b] avec un SVM entraîné sur une représentation des documents par le modèle TF-IDF [Salton & Buckley, 1988], ils améliorent les résultats à l'aide d'un classifieur ensembliste de SVM à probabilité moyenne, parvenant à des F_1 -mesures de plus de 95% Şulea *et al.* [2017a]. Un classifieur SVM à probabilité moyenne combine plusieurs modèles SVM dits « faibles » (ou de base) entraînés chacun sur un sous-ensemble de la base d'apprentissage. Lors de la prédiction, chacun des SVM estime une probabilité d'appartenance du document classifié à chaque classe. La classe du document est celle dont la probabilité moyenne (robustement estimé par la médiane [Kittler *et al.*, 1998]) est maximale.

Par ailleurs, Ashley & Brüninghaus [2009] identifient les « Facteurs » (*factors* [Ashley, 1990]) qui s'appliquent à une affaire par classification de la description des faits de cette affaire. Les Facteurs sont en effet des questions juridiques qui se posent dans un domaine ou contentieux [Bench-Capon, 1997]. Par exemple, l'environnement d'enseignement CATO définit 26 Facteurs sur l'appropriation illicite de secrets commerciaux (*trade secret misappropriation*) notamment *Unique-Product* (Le produit est-il unique ?) et *Nondisclosure-Agreement* (Existait-il in accord de non-divulgaration entre le défendant et le plaignant ?). Ils sont utilisés dans des raisonnements à base de cas où ils sont définis sous forme de prédicats faisant abstraction des faits. Certains sont binaires (réponse "oui" ou "non"). Un Facteur s'applique à une affaire, si les faits de cette dernière contiennent le patron correspondant. Ils entraînent un classifieur par facteur (les plus-proches-voisins) pour identifier ceux qui s'appliquent à la décision. **(comment seb : détailler quelques exemples de facteurs, 'pour savoir s'il s'applique à la décision' tu veux dire pour savoir s'ils sont pertinents pour la prédiction de la décision ?)** La partie remportant le procès est par la suite prédite par un algorithme séquentiel qui compare les parties (plaignant et défendeur) suivant le niveau de préférence des questions juridiques dégagées par les facteurs observés dans la base d'entraînement. D'autres catégorisations sont tout aussi utiles pour faciliter la recherche d'information. Par exemple, Şulea *et al.* [2017b,a] expérimentent la classification des décisions dans les formations judiciaires (par exemple, chambre civile, chambre commerciale, chambre sociale) ou l'identification de la période⁵ de la décision. La classification peut aussi servir à évaluer d'autres problématiques comme la similarité [Ma *et al.*, 2018].

5. Intervalle d'années dans laquelle la décision a été prononcée.

1.4 Similarité entre décisions judiciaires

La similarité entre textes est indispensable pour des applications qui nécessitent de regrouper des textes traitant de sujets similaires, et séparer ceux dont les sujets sont différents. La mesure de similarité doit être définie de sorte à rapprocher ou éloigner les documents suivant l'aspect sémantique que l'on souhaite révéler. Nair & Wagh [2018] exploitent les citations de lois et précédents⁶ pour retrouver les textes juridiques qui ont une similarité. Ils analysent le réseau de 597 citations⁷ sous l'Acte 2000 des Technologies de l'Information (*Information Technology Act, 2000*⁸) dans des jugements indiens. Leur proposition est d'utiliser des règles d'association générées par l'algorithme Apriori [Agrawal *et al.*, 1994] pour regrouper les jugements susceptibles d'être cités ensemble. Cet algorithme recherche les ensembles singletons de citations suffisamment fréquentes (seuil nécessaire), puis fusionne de manière itérative les ensembles tant que la co-occurrence des citations de la fusion est suffisamment fréquente dans le réseau. Une règle d'association $\{c_1, \dots, c_n\} \rightarrow \{c'\}$ indique qu'une citation c' est observable si l'on observe une co-occurrence d'un ensemble donné de citations $\{c_1, \dots, c_n\}$. A chaque règle est associé un score de confiance calculé à partir d'une métrique appelé score de support $sc()$ qui indique, pour un ensemble $\{c_1, \dots, c_n\}$, la fréquence de co-occurrence des citations de cet ensemble. Le support d'un singleton est sa fréquence d'occurrence. La similarité (**laquelle ?**) est confirmée si le score de confiance de la règle $conf(\{c_1, \dots, c_n\} \rightarrow \{c'\}) = sc(\{c_1, \dots, c_n, c'\}) / sc(\{c'\})$ est suffisamment élevé. Nair & Wagh [2018] démontrent au travers de scénarios (aucune évaluation quantifiée de l'efficacité de l'approche n'est proposée) que les documents qui sont fréquemment cités ensemble sont similaires car traitant de thématiques proches. Cette relation permet par transitivité de retrouver les documents pertinents dans une base de données.

Les métriques traditionnelles de similarité ne sont pas toujours très efficaces sur les décisions judiciaires. La raison peut être une représentation inadéquate des textes qui ne permet pas de traduire une fois comprise la notion de similarité telle qu'entendue dans la plupart des travaux. Thenmozhi *et al.* [2017] comparent par exemple, l'utilisation de la simi-

6. Les jugements du « Common Law » citent des décisions antérieures similaires.

7. Disponibles sur <https://indiankanoon.org>.

8. <https://www.meity.gov.in/content/information-technology-act-2000>.

larité cosinus sur trois représentations différentes des jugements dans le cadre de la campagne de recherche d'affaires antérieures pertinentes IR-LeD@FIRE2017[Mandal *et al.*, 2017] : (1) TF-IDF des concepts (noms), (2) TF-IDF des concepts et relations (verbes), (3) et la moyenne des plongements lexicaux *Word2Vec* [Le & Mikolov, 2014] des concepts et relations. Au vu des résultats (0.1795, 0.178, 0.0755 de précision@10⁹ et 0.681, 0.661, 0.435 de rappel@10⁹ respectivement pour les méthodes 1, 2, et 3), la première représentation semble mieux capter la similarité contrairement à l'utilisation des verbes et de la représentation distribuée. Ma *et al.* [2018] utilisent une forme de connaissances a priori définie dans des modèles de type ontologie pour estimer la similarité entre décisions. Ils proposent notamment d'aligner le document sur une ontologie des concepts et relations d'un corpus judiciaire. L'idée est de calculer la similarité sur un résumé du texte qui regroupe des aspects pertinents. Cette méthode permet ainsi de mieux capter la sémantique des jugements, d'avoir une meilleure précision, et de réduire la complexité temporelle inhérente à l'exploitation de longs documents notamment lors de l'utilisation de la « distance du déménageur de mots » ou WMD (*Word Mover's Distance*) de Kusner *et al.* [2015] (cf. ??). L'amélioration est observée sur une tâche de classification des jugements Chinois relatifs aux crimes de la circulation routière dans quatre catégories correspondant à des sentences d'emprisonnement¹⁰ (précision de 90.3% et 92.3% pour le résumé contre 84.8% et 82.4% respectivement pour le document original).

Toujours dans l'objectif d'une représentation pertinente des textes, Kumar *et al.* [2011] proposent quatre méthodes propres aux décisions judiciaires pour l'estimation de la similarité entre deux jugements x et y de la Cour Suprême indienne :

1. *all-term cosine similarity* : le cosinus de similarité entre les représentations TF-IDF de x et y dont tous les termes présents dans les jugements sont les dimensions.
2. *legal-term cosine similarity* : le cosinus de similarité sur les réductions des dimensions précédentes uniquement aux termes apparaissant dans un dictionnaire juridique.

9. Précision@ N , rappel@ N : précision et rappel calculées sur les N premiers résultats retournés par un système de recherche d'information.

10. Détention, emprisonnement à durée déterminée de moins de 3 ans, emprisonnement de durée déterminée de 3 à 7 ans et emprisonnement de plus de 7 ans.

3. *bibliographic coupling similarity* : la similarité de couplages bibliographiques égal au nombre de citations de jugements communes à x et y .
4. *co-citation similarity* : la similarité de co-citation qui est le nombre de citations de x et y dans un même jugement.

Bien qu'ils aient interprété les résultats sur de très faibles proportions des données utilisées (5/2430 et 18/2430), il en ressort que le cosinus de similarité avec les termes juridiques et le couplage bibliographique correspondent aux valeurs de similarité des experts, contrairement à la similarité basée sur tous les termes du corpus ou sur la co-citation. **(comment seb : fournir les performances)**

En synthèse, la similarité entre documents est utilisée pour répondre à plusieurs tâches, comme par exemple, la recherche de décisions similaires [Thenmozhi *et al.* , 2017], le regroupement non-supervisé de jugements [Raghuvier, 2012] et la classification supervisée de ces derniers [Ma *et al.* , 2018]. Ces diverses applications définissent aussi la sémantique juridique liée à la notion de similarité. Parmi les questions liées à la conception d'une mesure de la similarité entre documents, on distingue : la sémantique experte qui fonde cette similarité, sa métrique de mesure, la représentation des documents, le contexte d'exploitation et les critères d'évaluation.

1.5 Conclusion

En résumé, les travaux portant sur l'analyse automatique des décisions ont donné des résultats encourageants grâce aux éléments spécifiques aux affaires. Ces éléments peuvent être extraits des décisions grâce aux techniques de TALN et de fouille de texte. L'analyse des données textuelles juridiques a pour but la structuration des documents, l'extraction d'information, et l'organisation sémantique de corpus. Le domaine est très actif depuis déjà plusieurs décennies, au point que des librairies de développement, spécifiques au domaine, commencent à voir le jour [Bommarito *et al.* , 2018]. Dans la littérature, nous remarquons que le concepteur investit un minimum d'ingénierie d'adaptation que ce soit pour la définition des caractéristiques pertinentes pour les modèles à apprentissage automatique, ou pour définir les règles pour les méthodes à base de règles ou à base de grammaire. Notons aussi l'effort d'évaluation quantitative avec la parti-

cipation d'experts pour l'annotation d'exemples de référence même pour des tâches qui peuvent paraître subjectives comme la mesure de similarité.

Chapitre 2

Annotation des sections et entités juridiques

Ce chapitre traite de la détection de sections et des mentions (occurrences) d'entités dans les décisions jurisprudentielles françaises. Ce problème est important car il vise une structuration des documents par le balisage des sections organisant le document, des méta-données de référence de l'affaire, et des citations des normes juridiques employées. L'utilité pour le métier de juriste est améliorée de plusieurs façons. L'identification des méta-données d'indexation des décisions est automatisée. La lecture des décisions et leur compréhension manuelles ou automatiques est facilitée. Aussi, en complétant le problème par la désambiguïsation des méta-données et des normes, les critères de recherche de décisions peuvent être enrichis. Les principales contributions du travail discuté ici sont : la constitution et l'annotation manuelles d'un corpus d'évaluation, l'expérimentation et la discussion de l'impact, sur le problème, de divers aspects de la conception d'un système d'extraction d'information par étiquetage de séquence (l'ingénierie manuelle et automatique des caractéristiques des objets à classer, le choix du schéma d'étiquetage, l'augmentation des données d'entraînement). Les résultats montrent l'efficacité de modèle à base de champs aléatoires conditionnels à chaîne linéaire (CRF) pour les différentes tâches.

2.1 Introduction

Bien que les décisions ne soient pas structurées, leur contenu est organisé en sections dont les principales sont : l'entête, le corps, et le dispositif. Chacune de ces sections 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.

Certaines informations spécifiques se retrouvent très souvent dans une même section, e.g. méta-données (localisation, date), prétentions des parties, décisions finales. Compte tenu de la répartition standard de certaines informations, certaines tâches d'extraction d'information peuvent être abordées comme des traitements spécifiques à appliquer à certaines sections. Ce chapitre traite dans un premier temps des modèles utilisés pour appliquer cette phase de segmentation des décisions en sections. Par la suite, les entités, et données sur les demandes et résultats, pourront plus facilement être extraites. Nous nous focaliserons en particulier ici sur la détection des mentions d'entités telles que la date à laquelle le jugement a été prononcé, le type de juridiction, sa localisation (ville), les noms des juges, des parties, et les règles de loi citées (normes). La Table 2.1 liste les différentes entités cibles et fournit des exemples illustrant leurs occurrences dans les décisions avec lesquelles nous avons travaillé.

On pourrait s'attendre à ce qu'une institution comme la justice respecte un modèle strict et commun à tous les tribunaux pour la rédaction des décisions pour permettre de facilement pouvoir les lire et les analyser. Malheureusement, même si les décisions décrivent des informations de même nature, le modèle employé semble varier entre les juridictions. C'est ce que l'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 automatiquement ces marqueurs de transition par l'utilisation d'expressions régulières. Cependant, les marqueurs retrouvés ne sont généralement pas standards. Les indicateurs de transitions sont en effet souvent différents d'une décision à l'autre ; ils peuvent correspondre à des titres ou des motifs à base de symboles (astérisques, tirets, etc.). Il arrive même parfois que la transition soit implicite et que l'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. 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

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	appellant	« 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 – Exemples d'entités et statistiques sur la base d'exemples annotés manuellement

variantes qui peuvent être très simples (par exemple « Par Ces Motifs ») ou exceptionnelles (par exemple « 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. Par ailleurs, lors 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 sont entièrement en majuscules. Cependant, certains mots communs peuvent apparaître aussi en majuscules (par ex. APPELANTS, DÉBATS, ORDONNANCE DE CLÔTURE). Les entités peuvent contenir des chiffres (identifiants, 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, on rencontre une multitude de types d'entités dans l'entête, contrairement aux autres sections où seules les normes nous intéressent. De plus, le texte est mieux structuré dans l'entête que dans les autres sections. Ces nombreuses différences entre décisions rendent inadéquates les expressions régulières.

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 séquences. L'idée est de découper un texte en segments atomiques distincts (*token*) qui peuvent être des mots, des phrases, des paragraphes, etc. Le texte est ainsi représenté sous forme de séquences 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

Parmi les approches d'extraction d'information Chau *et al.* [2002], on retrouve principalement :

- 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 fournie par 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 change 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 »).

De plus, ces approches sont sujettes aux problèmes d'ambiguïté, par exemple lorsque différentes entités comprennent les mêmes mots. Ces problèmes ont largement limité ces premiers systèmes [Palmer & Day, 1997].

- Les **systèmes à base de règles** décrivent la variété des mentions d'entités en fonction de la régularité du contexte, de la structure et du lexique. Il existe plusieurs plate-formes et langages permettant de formaliser l'écriture des règles. Par exemple, dans le formalisme JAPE de Gate, Wyner [2010] détecte les énoncés de décisions à l'aide d'une règle qui sélectionne les phrases contenant un terme de jugement (*affirm*, *grant*, etc.) et suivies d'un nom de juge :

```
Rule: DecisionStatement
Priority: 10
(
{Sentence contains JudgementTerm}
):termtemp
{JudgeName}
->
:termtemp.DecisionStatement = {rule = 'DecisionStatement'}.
```

Ces systèmes présentent l'avantage de reposer sur des expressions déclaratives qui facilitent la maintenance (erreurs faciles à tracer et à expliquer) et l'expression directe des connaissances du domaine en règles [Waltl *et al.* , 2018]. Bien que parfois suffisant pour traiter des corpus modestes et spécialisés, ces systèmes sont très souvent limités en pratique. La définition manuelle de règles exige notamment des efforts considérables, en particulier pour le traitement de grands corpus. Par ailleurs, un ensemble donné de règles est difficilement réutilisable dans d'autres domaines ou sur des données n'intégrant pas exactement les subtilités linguistiques exprimées par les règles. Quelques approches adaptatives ont néanmoins 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 basés sur l'apprentissage automatique** exécutent des classifieurs multi-classes sur des segments de texte. Par exemple, un

algorithme traditionnel de classification comme le modèle 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 séquences 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 entités 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 étiqueter, 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, dans Kríž *et al.* [2014], le modèle HMM a été comparé 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 donné de bonnes performances avec des scores F_1 de 89% et 97% pour le HMM utilisant les tri-grammes comme descripteurs de mots, et des scores F_1 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 (mot, 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 HMM¹ est une machine à états finis définie par un ensemble d'états $\{s_1, s_2, \dots, s_m\}$. Un modèle HMM a pour fonction d'affecter une probabilité jointe $P(T, L) = \prod_{i=1}^n P(l_i | l_{i-1}) P(t_i | l_i)$ à des paires de séquences d'observations $T = t_{1:n}$ et de séquences 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 d'étiquetage 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 du texte et m le nombre de labels candidats, l'algorithme de décodage Viterbi [Viterbi, 1967], basé sur de la programmation dynamique, permet d'obtenir une estimation de L^* . Cet algorithme utilise des paramètres estimés par apprentissage sur un corpus de textes annotés manuellement :

- un alphabet ou vocabulaire $\{o_1, o_2, \dots, o_k\}$;
- un ensemble d'états $\{s_1, s_2, \dots, s_m\}$;
- 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 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.

1. Rabiner [1989] fournit plus de détails sur le modèle 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).

2.2.1.2 Les champs conditionnels aléatoires à chaîne linéaire (CRF)

Même si l'algorithme Viterbi est aussi utilisé pour appliquer le modèle CRF à l'étiquetage de séquences, 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 :

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

où $Z = \sum_{l_{1:n} \in L(T)} \exp \left(\sum_{i=1}^n \sum_{j=1}^q \lambda_j f_j(l_{i-1}, l_i, t_{1:n}, i) \right)$ est le facteur de normalisation, $L(T)$ étant l'ensemble des séquences possibles de labels pour T , q le nombre de fonction $f_j(\cdot)$.

Les fonctions potentielles $f_j(\cdot)$ 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 $f_j(\cdot)$ sont définies à l'aide de fonctions à valeurs binaires ou réelles $b(T, i)$ qui combinent 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 étiqueter « 700 » dans ce contexte « ... l'article 700 du code de procédure civile ... » :

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

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

avec

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

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

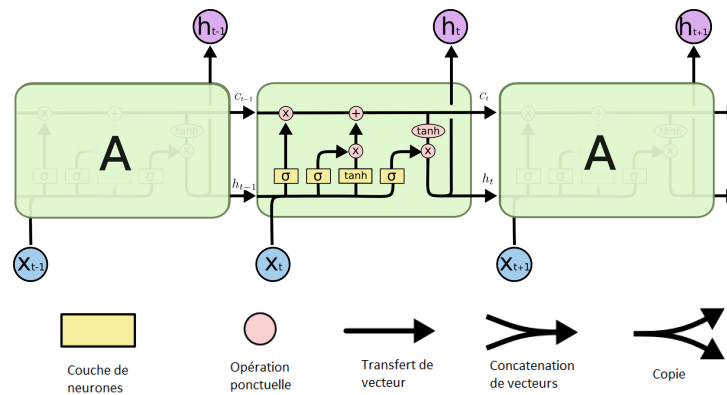
t_i étant l'observation (le mot) en position i dans T , POS_i étant le rôle grammatical de t_i (NUM = valeur numérique, NOM = nom), et $NP1_i$ et $NS1_i$ sont les lemmes des mots avant et après t_i , respectivement. Les symboles *<unknown>* et *@card@* encodent les lemmes inconnus et ceux des nombres 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 $l_i = \text{NORME}$ est renforcée par la somme $\lambda_1 + \lambda_2$ des poids affectés respectivement à f_1 et f_2 [Zhu, 2010]. Un modèle CRF active une fonction f_j lorsque ses conditions sont satisfaites (celles activant $b_j(T, \cdot)$) 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 optimale 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 & Nocedal [1989].

2.2.1.3 CRF et réseaux de neurones artificiels

Définitions Un réseau de neurones artificiels est un algorithme d'apprentissage automatique dont le fonctionnement est inspiré de celui du cerveau [McCulloch & Pitts, 1943; Rosenblatt, 1958]. Un neurone reçoit des valeurs $x = [x_1, x_2, \dots, x_l]$ en entrée, puis calcule un élément de sortie y par une fonction de la combinaison pondérée des entrées (poids $W = [w_0, w_1, \dots, w_l]$). Un réseau de neurones comprend des neurones structurés en couches. Les sorties d'une couche servent d'entrées à la couche sui-

vante et les nœuds d'une couche ne sont pas connectés. Les poids sont déterminés lors d'une phase d'entraînement qui les ajuste afin de minimiser l'erreur entre les valeurs attendues y en sorties et celles prédites \hat{y} par le modèle. Traditionnellement, les réseaux de neurones sont à « propagation avant » (*feed-forward*) i.e. l'unique sens de propagation de l'information est de la couche d'entrée successivement vers la couche de sortie. Mais comme nous l'avons vue précédemment, la prise en compte du contexte (état précédent ou suivant) est très importante dans la modélisation des séquences. Les réseaux à propagation avant ne sont pas adaptés pour prédire la sortie à l'instant t à partir de ses connaissances des instants précédents.

Les réseaux récurrents de neurones (*recurrent neural networks* - RNN) [Jordan, 1986; Elman, 1990] sont une architecture conçue pour modéliser les données séquentielles $X = X_{1:n}$. Le principe est de passer à l'instant t en entrée du réseau, la sortie ou état du réseau de l'instant précédent $t - 1$ en plus de l'observation courante X_t . Un LSTM est une variante particulière de RNN dont l'état est définie par la sortie h_t du réseau et la sortie C_t qui permet de gérer la mémoire à plus long terme (Figure 2.1 Page 21).

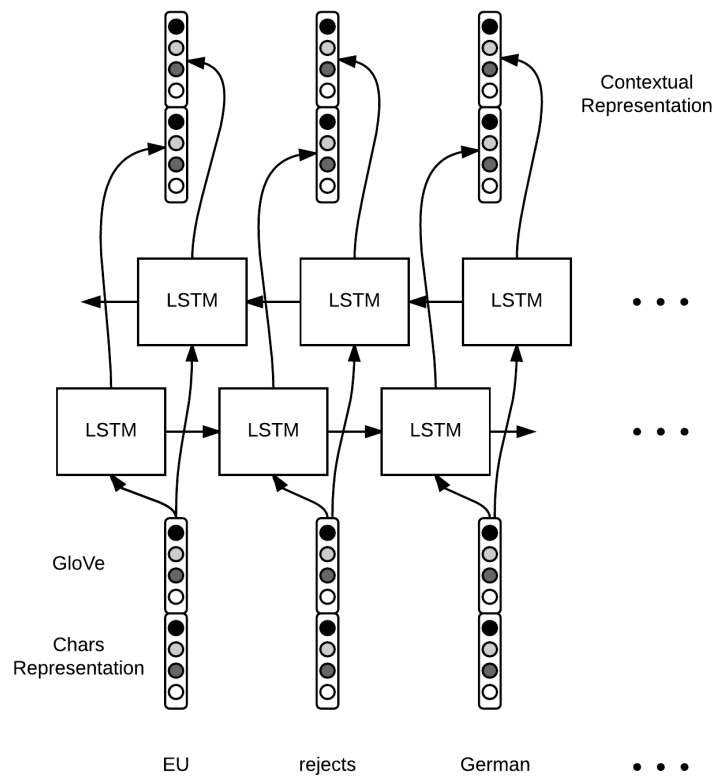


Source : <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>

Figure 2.1 – Structure interne d'un LSTM dans une couche de réseau LSTM

Les entrées et sorties d'un LSTM sont des vecteurs de nombres réels. L'entrée X_i est la représentation vectorielle indépendante de la séquence entrée de l'observation en position i , par exemple celle du mot t_i du texte $T = t_{1:n}$. La sortie h_i représente le contexte de l'observation en position i dans la séquence entrée. On peut ainsi chaîner des LSTM pour modéliser les textes entrées et y appliquer un CRF pour l'étiquetage. C'est le prin-

cipe du BiLSTM-CRF de Lample *et al.* [2016] une des architectures les plus efficaces pour la l'étiquetage d'entités nommées. En effet, le BiLSTM comprend deux couches enchaînat des LSTM (Figure 2.2 Page 22). Une couche permet d'apprendre le contexte "gauche" des mots, les états étant propager dans le sens du début vers la fin du texte. L'autre couche apprend le contexte droit en propageant les informations dans le sens inverse.



Source : <https://guillaumegenthial.github.io/sequence-tagging-with-tensorflow.html>

Figure 2.2 – Apprentissage de la représentation contextuelle avec une double couche de réseaux LSTM (BiLSTM)

Représentation Les réseaux de neurones permettent d'apprendre des caractéristiques grâce à des méthodes de plongement de mots telles que Word2Vec [Le & Mikolov, 2014] et GloVe [Pennington *et al.*, 2014].

Entraînement

Décodage

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 & 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 alors 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 & Baryamureeba, 2007]. Elle peut aussi fournir une meilleure compréhension du comportement des modèles entraînés [Klinger & 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 des performances d'évaluation qu'elles permettent d'obtenir (par exemple la F_1 -mesure obtenue sur un ensemble d'exemples). Même si les méthodes filtrantes sont plus rapides, elles sont en général moins performantes car elles ne per-

mettent 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.

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 segments. La Figure 2.3 Page 24 illustre la différence entre des schémas appelés IO, BIO, IEO et BIEO, 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>,</i>	<i>Président</i>	<i>de</i>	<i>chambre</i>	<i>,</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 2.3 – Illustration des schémas d'étiquetage IO, BIO, IEO, BIEO

Nous comparons dans cette étude quelques schémas d'étiquetage dont certains sont décrits par Konkol & Konopík [2015]. Le principe de ces schémas est d'étiqueter différemment des segments atomiques en fonction de leur position dans l'entité. Pour cela, le label associé à l'entité est préfixé par l'une des lettres suivantes :

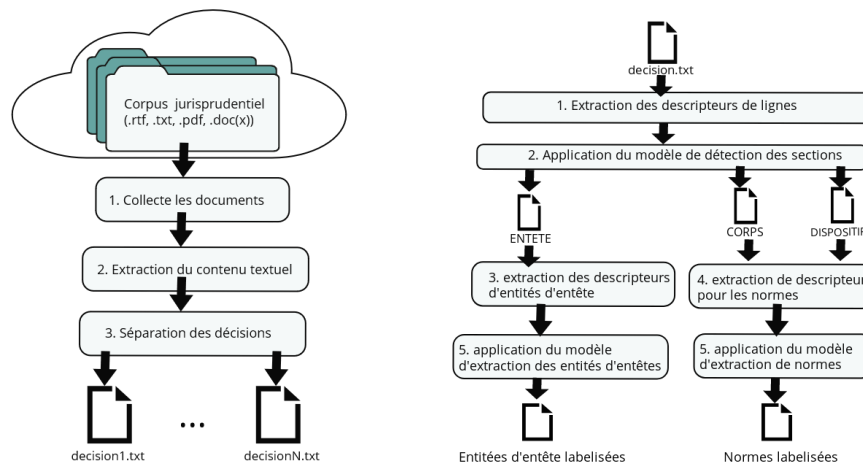
- B : début (*beginning*) ;
- I : intérieur (*inside*) ;
- E (ou L, ou M) : fin (*end* ou *last* ou *middle*) ;
- S (ou U, ou W) : singleton ou entité à segment unique (*single* ou *unit* ou *whole*) ;
- O : hors de toute entité (*outside*).

Le schéma IO 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 l'on souhaite distinguer les entités à un seul segment (par ex. ville ou numéro R.G.). 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 des schémas plus complexes que IO est de pouvoir distinguer des entités du même type qui se suivent sans être explicitement séparées (par exemple, des appelants mentionnés sur des lignes consécutives). 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 pré-traitement 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 2.4 – Application des modèles entraînés pour l'étiquetage de sections 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 cette uniformisation des documents comme point d'entrée (Figure 2.4 Page 25). 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.

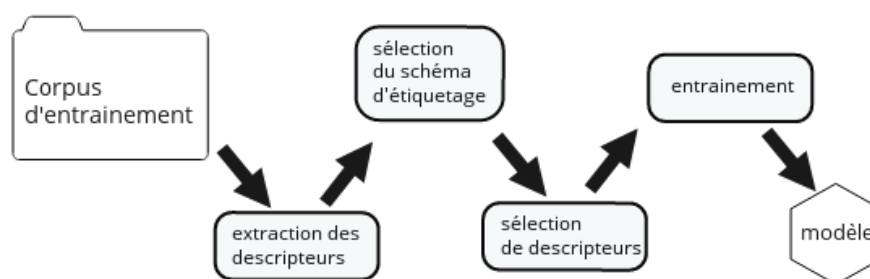


Figure 2.5 – Entraînement des modèles.

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 2.5 Page 26).

2.3.1 Définition de descripteurs candidats

2.3.1.1 Descripteurs 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 aussi été 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 d'avantage 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'entête), 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 absolu (nombre exact de mots dans la ligne), ou relatif (une catégorie de la longueur). Sur la base des quantiles de la distribution des 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 (t_0 , t_1 , t_2), 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 (p_0 , p_1) et suivantes (n_0 , n_1), ainsi que leurs longueurs absolues et relatives ($pLength$, $pRelLength$, $nLength$, $nRelLength$).

2.3.1.2 Descripteurs pour la détection d'entités

La détection d'entités consiste, dans notre cas, 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 défini 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 mot 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 étendu 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 d'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 & 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.

La première méthode, qui est la recherche bidirectionnelle (BDS) de Liu & Motoda [2012], combine la sélection séquentielle en avant (SFS) et la sélection séquentielle en arrière (SBS) en parallèle (Algorithme 1). La SFS recherche un sous-ensemble optimal, en commençant par un ensemble vide et en ajoutant le descripteur qui améliore le mieux l'efficacité du sous-ensemble sélectionné. Le critère d'efficacité dans notre cas est défini par la F_1 -mesure (Eq. 2.4.1). Contrairement à la SFS, la SBS commence par l'ensemble des candidats et supprime successivement les plus mauvais descripteurs. Une caractéristique ne peut être ajoutée dans $Y_{F_{k+1}}$ que si elle est présente dans Y_{B_k} .

La seconde méthode, qui est 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 (Algorithme

Algorithme 1 : Recherche bidirectionnelle BDS

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

Résultat : Meilleur sous-ensemble de descripteurs

```

1 Démarrer la SFS avec  $Y_{\mathcal{F}_0} = \emptyset$ ;
2 Démarrer la SBS avec  $Y_{\mathcal{B}_0} = X$ ;
3  $k = 0$ ;
4 tant que  $Y_{\mathcal{F}_k} \neq Y_{\mathcal{B}_k}$  faire
5    $x^+ = \operatorname{argmax}_{x \in Y_{\mathcal{B}_k} \setminus Y_{\mathcal{F}_k}} F_1(Y_{\mathcal{F}_k} + x); Y_{\mathcal{F}_{k+1}} = Y_{\mathcal{F}_k} + x^+ // \text{SFS};$ 
6    $x^- = \operatorname{argmax}_{x \in Y_{\mathcal{B}_k} \setminus Y_{\mathcal{F}_{k+1}}} F_1(Y_{\mathcal{F}_k} - x); Y_{\mathcal{B}_{k+1}} = Y_{\mathcal{B}_k} - x^- // \text{SBS};$ 
7    $k = k + 1$ ;
8 retourner  $Y_{\mathcal{F}_k}$ ;

```

2). En effet, le SFFS effectue des tests en arrière à chaque itération.

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 annotées est 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.

Algorithme 2 : Sélection séquentielle avant à flottement

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

Résultat : Meilleur sous-ensemble de descripteurs

```

1  $Y_0 = \emptyset;$ 
2  $k = 0;$ 
3 répéter
4    $x^+ = \operatorname{argmax}_{x \notin Y_k} F_1(Y_k + x); Y_k = Y_k + x^+;$ 
5    $x^- = \operatorname{argmax}_{x \in Y_k} F_1(Y_k - x);$ 
6   si  $F_1(Y_k - x^-) > F_1(Y_k)$  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  $Y_k;$ 

```

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 Developer². 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 260 lignes et 3900 mots environ. Les deux dernières colonnes du Tableau 2.1 Page 14 pré-

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

sentent 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 [1960]. Ces mesures d'accord inter-annotateur ont été calculées au niveau des caractères parce que certains mots peuvent être coupés par des annotations incorrectes (par ex. *<juridiction> cour d'appel </juridiction> l* contre *<juridiction> cour d'appel </juridiction>*), ou bien les annotateurs pourraient ne pas être d'accord si une apostrophe devrait être incluse ou pas dans l'annotation (par ex. *l'<norme>article 700* contre *<norme >l'article 700*). Les taux de Kappa de 0,705 et 0,974 ont été obtenus 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 F_1 -mesure comme mesures d'évaluation car elles sont généralement utilisées comme références en extraction d'information. La F_1 -mesure se calcule à l'aide de la formule suivante :

$$F_1 = 2 \times \frac{Precision \times Rappel}{Precision + Rappel} \quad (2.4.1)$$

L'évaluation peut être faite au niveau des segments atomiques ou des entités selon que l'on soit plus intéressé respectivement par l'étiquetage du maximum de segments atomiques ou par la labellisation complète d'un maximum d'entités.

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 :

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

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

Evaluation au niveau entité (*entity-level*) : cette évaluation mesure le taux d'entités parfaitement identifiées c'est-à-dire seulement celles 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 au niveau des entités comme suit :

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

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

Ces métriques sont calculées de la même façon au niveau atomique.

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*), 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 TreeTagger³ [Schmid, 1994]. L'implémentation dans Mallet du LDA [Blei *et al.*, 2003] a permis d'inférer 100 thèmes à partir d'un corpus lemmatisé d'environ 6k documents. Le Tableau 2.2 Page 34 présente des mots représentatifs trouvés dans les premiers thèmes inférés. L'extraction des autres descripteurs a été implémentée pour cette expérimentation.

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

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 F_1 -mesure ont été calculées à l'aide du script d'évaluation de la campagne CoNLL-2002⁴. Elles sont indiquées en pourcentage dans les tableaux de résultats d'évaluation des sections suivantes.

2.4.2 Sélection du schéma d'étiquetage

Dans le but d'évaluer comment la représentation de segments affecte les performances, nous avons implémenté quatre représentations (IO, IE02, BIO2, BIE0). 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 Page 35 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'entités dans l'entête avec le CRF. Il semble évident que cette durée croisse avec le nombre de labels candidats de la section et la complexité du schéma d'étiquetage. En effet, BIE0 exige beaucoup plus de temps, et IO exige le temps d'entraînement le plus bas, et le schéma IOE semble être plus rapide que BIO même s'ils ont le même nombre de labels. Nous remarquons aussi que les représentations complexes n'améliorent pas significativement les résultats par rapport au simple IO qui demande pourtant beaucoup moins de temps.

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

Tâche	Modèle	Niveau atomique ^a			Niveau entité ^a			Durée ^b	Schéma
		Précision	Rappel	F_1	Précision	Recall	F_1		
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

^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 maximum)

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

Tableau 2.3 – Comparaison des schémas d'étiquetage.

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 d'identification des sections. La méthode BDS quant à elle a duré plus de 15h pour 600 itérations d'entraînement-test. Malgré la sauvegarde des scores F_1 pour éviter d'exécuter plusieurs fois l'entraînement pour les mêmes sous-ensembles de descripteurs, le processus de sélection est toujours resté très long pour les deux algorithmes. Nous avons testé individuellement chacun des descripteurs candidats pour les modèles HMM. Les résultats sont reportés dans le Tableau 2.4 Page 36.

Le résultat le plus remarquable est la forte réduction du nombre de descripteurs par les algorithmes. En général, la moitié est éliminée par la sélection BDS, tandis que la méthode SFFS élimine beaucoup plus de candidats (par exemple en ne sélectionnant que 4 descripteurs parmi les

Tâche	Modèle	niveau atomique ^a			niveau entité ^a			Sous-ensemble sélectionné
		Précision	Rappel	F ₁	Précision	Rappel	F ₁	
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
Entités d'entête	CRF	37.59	37.59	37.59	18.81	18.81	18.81	token
		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}
	HMM	86.86	78.96	82.73	80.84	65.17	72.17	token
		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.

14 candidats définis pour l'annotation des normes).

Par ailleurs, les algorithmes de sélection forment des combinaisons in-

attendues. Par exemple, dans le cas de la détection de sections, 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 ces sous-ensembles sélectionnés, les modèles sont plus performants que lorsqu'ils exploitent seulement 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 Évaluation 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'itérations maximal a été fixé à 500 pour assurer la convergence lors de l'entraînement. Les Tableaux 2.5 et 2.6 présentent les résultats 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 aussi 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>	F_1	<i>Precision</i>	<i>Rappel</i>	F_1
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, F_1 -mesures pour chaque type d'entité et section au niveau atomique.

	HMM			CRF		
	<i>Precision</i>	<i>Rappel</i>	F_1	<i>Precision</i>	<i>Rappel</i>	F_1
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, F_1 -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 2.6 Page 40), les *intervenants* sont parfois mal classifiés comme *intimé* en majorité (17 %), mais aussi comme *appelant* (4 %) ou *avocat* (2 %) probablement parce qu'il s'agit d'entités mentionnées les unes à la suite des autres dans l'entête (les *intervenants* sont généralement 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 classifiés comme *intimés* (16 %). Ce qui signifie que la transition entre la liste des appelants et celle des intimés est difficilement identifiable avec les descripteurs de mots que nous avons définis.

La proximité crée aussi des confusions entre les sections CORPS et DISPOSITIF qui se suivent (Figure 2.7 Page 41).

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 plusieurs fois et en alternant souvent les formes abrégées et longues (par exemple, la juridiction, la date, les normes). Bien que les différentes occurrences d'une même méta-données ne soient pas toujours identiques, de telles redondances 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 annotés

Des expérimentations ont été menées pour évaluer les variations des modèles lorsque l'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'ensemble d'entraînement ont été utilisées (de 5% à 100%). A chaque session entraînement-test, le même jeu de test a été

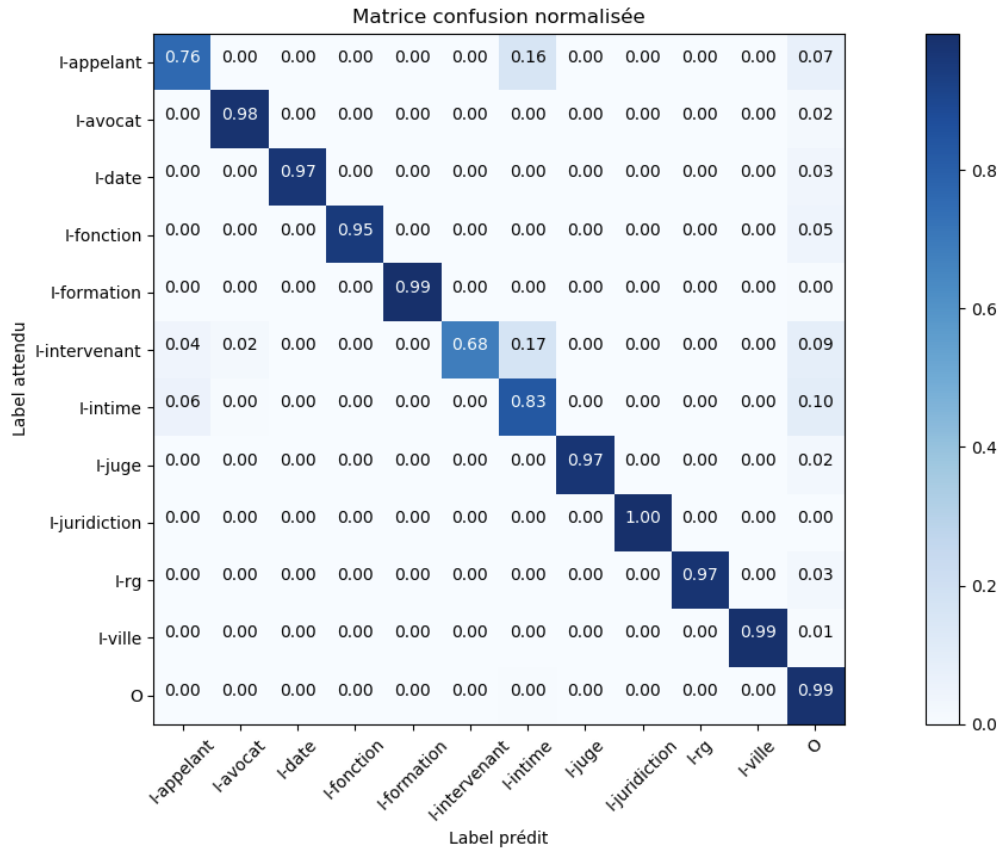


Figure 2.6 – Matrice de confusion entre méta-données d’entête avec le modèle CRF

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 2.8a (Page 42) et 2.8b (Page 42) .

Il apparaît que les scores F_1 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 quelle que soit la tâche. Il est possible que les exemples ajoutés par la suite partagent la même structure que celle de ceux qui ont été ajoutés auparavant. Ainsi, cette étude doit être étendue à la sélection des exemples les plus utiles. Raman & Ioerger [2003] ont démontré les avantages des algorithmes de sélection d’exemples combinés à celle des caractéristiques pour la classification. Les mêmes méthodes sont probablement applicables à l’étiquetage

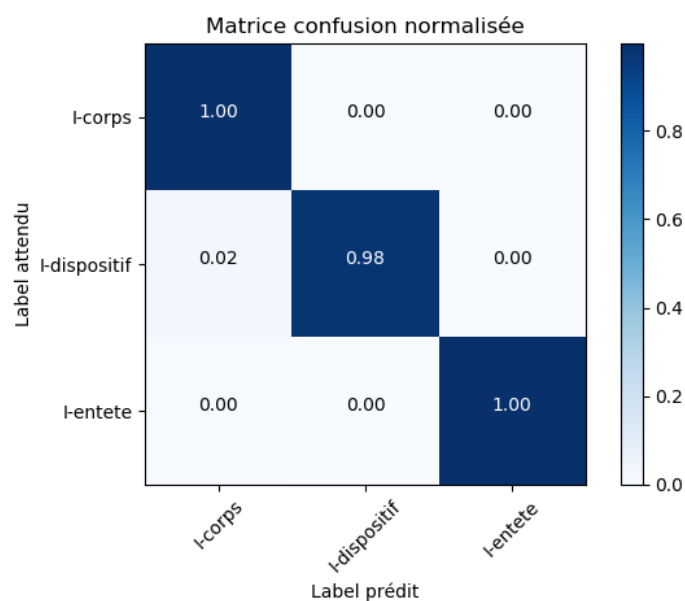


Figure 2.7 – Matrice de confusion entre lignes des sections avec le modèle CRF

de séquences.

2.4.5.4 Descripteurs manuels vs. réseau de neurones

	CRF + descripteurs manuels			BiLSTM-CRF		
	<i>Precision</i>	<i>Rappel</i>	F_1	<i>Precision</i>	<i>Rappel</i>	F_1
appelant	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
jugé	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 entre le CRF avec des descripteurs définis manuellement et le BiLSTM-CRF au niveau entité.

L'ingénierie manuelle des caractéristiques est difficile car arbitraire.

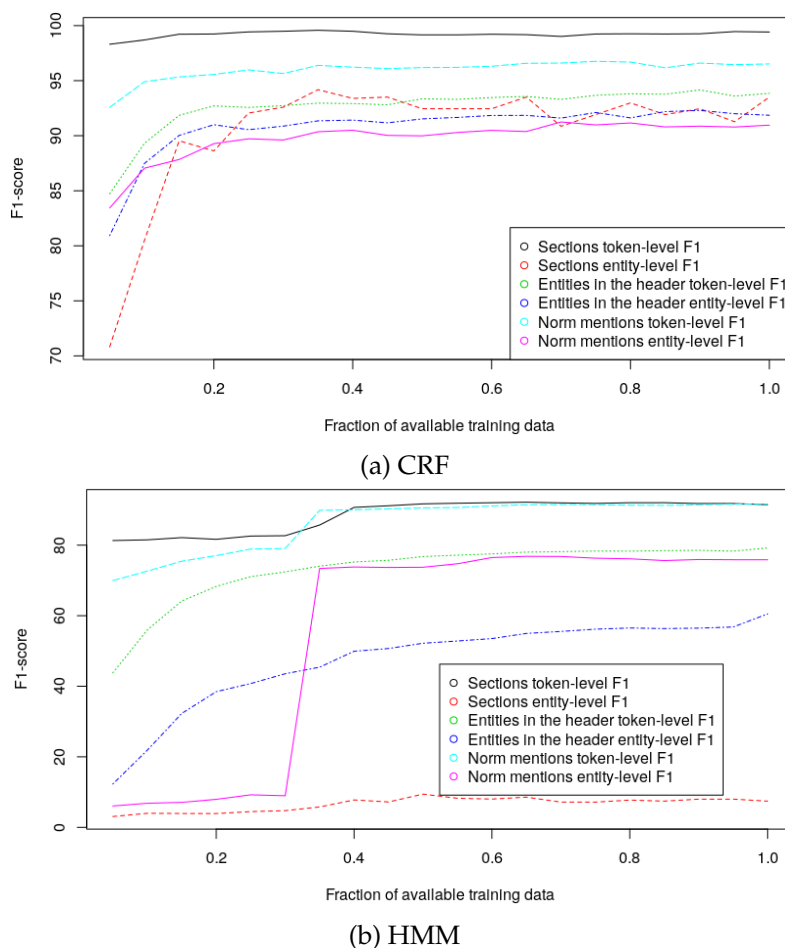


Figure 2.8 – Évolution du score F1 en fonction de l’augmentation du nombre de données d’entraînement.

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 détection 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 [Mikolov *et al.*, 2013] 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'implémentation⁵ de Le & 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 Page 41). 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.

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 l'augmentation de la taille de l'échantillon d'apprentissage, la mesure globale F_1 semble 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

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

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

Il serait intéressant de poursuivre les travaux proposés sur la tâche de reconnaissance d'entités nommées. L'étude de modèles couplant les approches à descripteurs définis manuellement (e.g., CRF), avec des approches sans définition manuelle (de type apprentissage profond e.g., BiLSTM CRF) semble particulièrement intéressante. Une étude comparative approfondie des limitations des deux approches serait alors souhaitable. Bien que les approches à base d'apprentissage profond apparaissent (légèrement) moins performantes dans nos tests, l'étude de ces approches prometteuse est bien entendu à recommander. Des travaux sur l'impact des techniques de plongement lexical sur la performance de ces systèmes méritent notamment d'être menées.

Pour l'indexation des décisions dans une base de connaissances, il est aussi important 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.

Bibliographie

- Agrawal, Rakesh, Srikant, Ramakrishnan, *et al.* . 1994. Fast algorithms for mining association rules. *Pages 487–499 of : Proc. 20th int. conf. very large data bases, vldb*, vol. 1215.
- Aletras, Nikolaos, Tsarapatsanis, Dimitrios, Preoțiuc-Pietro, Daniel, & Lampos, Vasileios. 2016. Predicting judicial decisions of the European Court of Human Rights : A Natural Language Processing perspective. *Peerj computer science*, **2**, e93.
- Andrew, Judith Jeyafreeda, & Tannier, Xavier. 2018. Automatic extraction of entities and relation from legal documents. *Pages 1–8 of : Proceedings of the seventh named entities workshop*.
- Ashley, Kevin D. 1990. *Modeling legal arguments : Reasoning with cases and hypotheticals*. MIT press.
- Ashley, Kevin D., & Brüninghaus, Stefanie. 2009. Automatically classifying case texts and predicting outcomes. *Artificial intelligence and law*, **17**(2), 125–165.
- Baldwin, Breck. 2009. *Coding chunkers as taggers : IO, BIO, BMEWO, and BMEWO+*.
- Bench-Capon, Trevor J.M. 1997. Arguing with cases. *Pages 85–100 of : Jurix*.
- Berka, Petr. 2011. Nest : A compositional approach to rule-based and case-based reasoning. *Advances in artificial intelligence*, **2011**, 15.
- Blei, David M., Ng, Andrew Y., & Jordan, Michael I. 2003. Latent Dirichlet Allocation. *the Journal of Machine Learning Research*, **3**, 993–1022.
- Bommarito, II, Michael, J, Katz, Daniel Martin, & Detterman, Eric M. 2018. Lexnlp : Natural language processing and information extraction for legal and regulatory texts. *arxiv preprint arxiv :1806.03688*.

- Breiman, Leo. 2001. Random forests. *Machine learning*, **45**(1), 5–32.
- Cardellino, Cristian, Teruel, Milagro, *et al.* . 2017. A low-cost, high-coverage legal named entity recognizer, classifier and linker. *Pages 9–18 of : Proceedings of the 16th edition of the international conference on artificial intelligence and law*. ACM.
- Chau, Michael, Xu, Jennifer J, & Chen, Hsinchun. 2002. Extracting meaningful entities from police narrative reports. *Pages 1–5 of : Proceedings of the 2002 annual national conference on digital government research*. Digital Government Society of North America.
- Chiticariu, Laura, Krishnamurthy, Rajasekar, Li, Yunyao, Reiss, Frederick, & Vaithyanathan, Shivakumar. 2010. Domain adaptation of rule-based annotators for named-entity recognition tasks. *Pages 1002–1012 of : Proceedings of the 2010 conference on empirical methods in natural language processing*. Association for Computational Linguistics.
- Cohen, Jacob. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, **20**(1), 37–46.
- Dozier, Christopher, Kondadadi, Ravikumar, Light, Marc, Vachher, Arun, Veeramachaneni, Sriharsha, & Wudali, Ramdev. 2010. Named entity recognition and resolution in legal text. *Pages 27–43 of : Semantic processing of legal texts*. Springer.
- Elman, Jeffrey L. 1990. Finding structure in time. *Cognitive science*, **14**(2), 179–211.
- Finkel, Jenny Rose, Grenager, Trond, & Manning, Christopher. 2005. Incorporating non-local information into information extraction systems by Gibbs sampling. *Pages 363–370 of : Proceedings of the 43rd annual meeting on association for computational linguistics*. Association for Computational Linguistics.
- Genesereth, Michael. 2015. Computational law : The cop in the backseat. *The standford center for legal informatics hosted the third annual futurelaw 2015 conference*.
- Hanisch, Daniel, Fundel, Katrin, *et al.* . 2005. ProMiner : rule-based protein and gene entity recognition. *BMC bioinformatics*, **6**(1), 14.

- Jordan, MI. 1986. *Serial order : a parallel distributed processing approach. technical report, june 1985-march 1986*. Tech. rept. California Univ., San Diego, La Jolla (USA). Inst. for Cognitive Science.
- Katz, Daniel Martin, Bommarito, Michael James, & Blackman, Josh. 2014. Predicting the behavior of the supreme court of the united states : A general approach. *Available at ssrn* 2463244.
- Katz, Daniel Martin, Bommarito II, Michael J, & Blackman, Josh. 2017. A general approach for predicting the behavior of the supreme court of the united states. *Plos one*, **12**(4), e0174698.
- Kitoogo, Fredrick Edward, & Baryamureeba, Venansius. 2007. A methodology for feature selection in named entity recognition. *Strengthening the role of ict in development*, 88.
- Kittler, Josef, Hatef, Mohamad, Duin, Robert PW, & Matas, Jiri. 1998. On combining classifiers. *Ieee transactions on pattern analysis and machine intelligence*, **20**(3), 226–239.
- Klinger, Roman, & Friedrich, Christoph M. 2009. Feature subset selection in conditional random fields for named entity recognition. *Pages 185–191 of : Proceedings of the international conference ranlp-2009*.
- Konkol, Michal, & Konopík, Miloslav. 2015. Segment representations in named entity recognition. *Pages 61–70 of : International conference on text, speech, and dialogue*. Springer.
- Kříž, Vincent, Hladká, Barbora, *et al.* . 2014. Statistical recognition of references in czech court decisions. *Pages 51–61 of : Gelbukh, Alexander, Espinoza, Félix Castro, & Galicia-Haro, Sofía N. (eds), Human-inspired computing and its applications : 13th mexican international conference on artificial intelligence, micai 2014, tuxtla gutiérrez, mexico, november 16-22, 2014. proceedings, part i*. Cham : Springer International Publishing.
- Kumar, Sushanta, Reddy, P Krishna, Reddy, V Balakista, & Singh, Aditya. 2011. Similarity analysis of legal judgments. *Page 17 of : Proceedings of Compute 2011 - Fourth Annual ACM Bangalore Conference*. ACM.

- Kusner, Matt, Sun, Yu, Kolkin, Nicholas, & Weinberger, Kilian. 2015. From word embeddings to document distances. *Pages 957–966 of : International conference on machine learning.*
- Lafferty, John, McCallum, Andrew, & Pereira, Fernando C. N. 2001. Conditional random fields : probabilistic models for segmenting and labeling sequence data. *International Conference on Machine Learning.*
- Lample, Guillaume, Ballesteros, Miguel, *et al.* . 2016. Neural architectures for named entity recognition. *arxiv preprint.* arXiv :1603.01360.
- Le, Quoc, & Mikolov, Tomas. 2014. Distributed representations of sentences and documents. *Pages 1188–1196 of : International conference on machine learning.*
- Leith, Philip. 2010. The rise and fall of the legal expert system. *European journal of law and technology*, **1**(1), 179–201.
- Li, Yaoyong, Zaragoza, Hugo, Herbrich, Ralf, Shawe-Taylor, John, & Kandola, Jaz. 2002. The perceptron algorithm with uneven margins. *Pages 379–386 of : Icml*, vol. 2.
- Liu, Dong C., & Nocedal, Jorge. 1989. On the limited memory BFGS method for large scale optimization. *Mathematical programming*, **45**(1), 503–528.
- Liu, Huan, & Motoda, Hiroshi. 2012. *Feature selection for knowledge discovery and data mining*. Vol. 454. Springer Science & Business Media.
- Llewellyn, Karl Nickerson. 1962. *Jurisprudence : Realism in theory and practice*. The University of Chicago Press.
- Love, Nathaniel, & Genesereth, Michael. 2005. Computational law. *Pages 205–209 of : Proceedings of the 10th international conference on artificial intelligence and law.* ACM.
- Ma, Yinglong, Zhang, Peng, & Ma, Jiangang. 2018. An Efficient Approach to Learning Chinese Judgment Document Similarity Based on Knowledge Summarization. *arXiv preprint*, 23. arXiv :1808.01843.

- Mandal, Arpan, Ghosh, Kripabandhu, Bhattacharya, Arnab, Pal, Arindam, & Ghosh, Saptarshi. 2017. Overview of the fire 2017 irded track : Information retrieval from legal documents. *Pages 63–68 of : Fire (working notes)*.
- McCallum, Andrew Kachites. 2012. *MALLET : A Machine Learning for Language Toolkit*.
- McCulloch, Warren S, & Pitts, Walter. 1943. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4), 115–133.
- Medvedeva, Masha, Vols, Michel, & Wieling, Martijn. 2018. Judicial decisions of the european court of human rights : Looking into the crystal ball. *In : Proceedings of the conference on empirical legal studies*.
- Mikheev, Andrei, Moens, Marc, & Grover, Claire. 1999. Named entity recognition without gazetteers. *Pages 1–8 of : Proceedings of the ninth conference on european chapter of the association for computational linguistics*. Association for Computational Linguistics.
- Mikolov, Tomas, Chen, Kai, Corrado, Greg, & Dean, Jeffrey. 2013. Efficient estimation of word representations in vector space. *arxiv preprint arxiv :1301.3781*.
- Mochales, Raquel, & Moens, Marie-Francine. 2008. Study on the structure of argumentation in case law. *Pages 11–20 of : Proceedings of the 2008 conference on legal knowledge and information systems*.
- Moens, Marie-Francine. 2002. What information retrieval can learn from case-based reasoning. *Pages 83–91 of : Legal knowledge and information systems*. Amsterdam : T.J.M. Bench-Capon, A. Daskalopulu and R.G.F. Winkels (eds.), for Jurix 2002 : The Fifteenth Annual Conference.
- Moens, Marie-Francine, Boiy, Erik, Palau, Raquel Mochales, & Reed, Chris. 2007. Automatic detection of arguments in legal texts. *Pages 225–230 of : Proceedings of the 11th international conference on artificial intelligence and law*. ACM.
- Nadeau, David, & Sekine, Satoshi. 2007. A survey of named entity recognition and classification. *Linguisticae investigationes*, 30(1), 3–26.

- Nair, Akhil M., & Wagh, Rupali Sunil. 2018. Similarity Analysis of Court Judgements Using Association Rule Mining on Case Citation Data - A Case Study. *International Journal of Engineering Research and Technology*, 11(3), 373–381.
- Nallapati, Ramesh, Surdeanu, Mihai, & Manning, Christopher. 2010. Blind domain transfer for named entity recognition using generative latent topic models. *Pages 281–289 of : Proceedings of the nips 2010 workshop on transfer learning via rich generative models*.
- Nazarenko, Adeline, & Wyner, Adam. 2017. Legal NLP Introduction. *Traitement automatique de la langue juridique / Legal Natural Language Processing - Revue TAL*, 58(2), 7–19.
- Palmer, David D, & Day, David S. 1997. A statistical profile of the named entity task. *Pages 190–193 of : Proceedings of the fifth conference on applied natural language processing*. Association for Computational Linguistics.
- Pennington, Jeffrey, Socher, Richard, & Manning, Christopher. 2014. Glove : Global vectors for word representation. *Pages 1532–1543 of : Proceedings of the 2014 conference on empirical methods in natural language processing (emnlp)*.
- Persson, Caroline. 2012. *Machine learning for tagging of biomedical literature*. Closing project report, Technical University of Denmark, DTU Informatics.
- Polifroni, Joe, & Mairesse, François. 2011. Using latent topic features for named entity extraction in search queries. *Pages 2129–2132 of : Interspeech*.
- Pudil, Pavel, Novovičová, Jana, & Kittler, Josef. 1994. Floating search methods in feature selection. *Pattern recognition letters*, 15(11), 1119–1125.
- Rabiner, Lawrence R. 1989. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the ieee*, 77(2), 257–286.
- Raghuveer, K. 2012. Legal documents clustering using latent dirichlet allocation. *Iaes int. j. artif. intell*, 2(1), 34–37.

- Raman, Baranidharan, & Ioerger, Thomas R. 2003. Enhancing learning using feature and example selection. *Texas a&m university, college station, tx, usa*.
- Rosenblatt, Frank. 1958. The perceptron : a probabilistic model for information storage and organization in the brain. *Psychological review*, **65**(6), 386.
- Salton, Gerard, & Buckley, Christopher. 1988. Term-weighting approaches in automatic text retrieval. *Information processing & management*, **24**(5), 513–523.
- Salton, Gerard, & McGill, Michael J. 1983. *Introduction to modern information retrieval*. mcgraw-hill.
- Salton, Gerard, Wong, Anita, & Yang, Chung-Shu. 1975. A vector space model for automatic indexing. *Communications of the acm*, **18**(11), 613–620.
- Schmid, Helmut. 1994. *Treetagger - a part-of-speech tagger for many languages*.
- Sharnagat, Rahul. 2014. *Named entity recognition : A literature survey*. Tech. rept. Center For Indian Language Technology.
- Shulayeva, Olga, Siddharthan, Advait, & Wyner, Adam. 2017. Recognizing cited facts and principles in legal judgements. *Artificial intelligence and law*, **25**(1), 107–126.
- Siniakov, Peter. 2008. *Gropus an adaptive rule-based algorithm for information extraction*. Ph.D. thesis, Freie Universität Berlin.
- Şulea, Octavia-Maria, Zampieri, Marcos, Malmasi, Shervin, Vela, Mihaela, P. Dinu, Liviu, & van Genabith, Josef. 2017a (June). Exploring the Use of Text Classification in the Legal Domain. *Page 5 of : Proceedings of 2nd Workshop on Automated Semantic Analysis of Information in Legal Texts. ASAIL'2017, London, United Kingdom*.
- Şulea, Octavia-Maria, Zampieri, Marcos, Vela, Mihaela, & van Genabith, Josef. 2017b. Predicting the Law Area and Decisions of French Supreme Court Cases. *Pages 716–722 of : Proceedings of the International Conference on Recent Advances in Natural Language Processing, RANLP 2017*.

- Thakker, Dhaval, Osman, Taha, & Lakin, Phil. 2009. *Gate jape grammar tutorial*.
- Thenmozhi, D., Kannan, Kawshik, & Aravindan, Chandrabose. 2017. A Text Similarity Approach for Precedence Retrieval from Legal Documents. *Pages 90–91 of : Proceedings of Forum for Information Retrieval Evaluation - FIRE (Working Notes)*.
- Tumonis, Vitalius. 2012. Legal realism & judicial decision-making. *Jurisprudencija*, **19**(4).
- Ulmer, S Sidney. 1963. Quantitative analysis of judicial processes : Some practical and theoretical applications. *Law and contemporary problems*, **28**(1), 164–184.
- Vapnik, Vladimir N. 1995. The nature of statistical learning. *Theory*.
- Viera, Anthony J, Garrett, Joanne M, *et al.* . 2005. Understanding interobserver agreement : the kappa statistic. *Fam med*, **37**(5), 360–363.
- Viterbi, Andrew James. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *Ieee transactions on information theory*, **13**(2), 260–269.
- Wallach, Hanna M. 2004. *Conditional Random Fields : An Introduction*. Tech. rept. University of Pennsylvania Department of Computer and Information Science.
- Waltl, Bernhard, Matthes, Florian, Waltl, Tobias, & Grass, Thomas. 2016. LEXIA - A Data Science Environment for Semantic Analysis of German Legal Texts. In : *Iris : Internationales rechtsinformatik symposium*. Salzburg, Austria.
- Waltl, BERNHARD, Landthaler, Jörg, Scepankova, Elena, Matthes, FLORIAN, Geiger, THOMAS, Stocker, CHRISTOPH, & Schneider, CHRISTIAN. 2017a. Automated extraction of semantic information from German legal documents. In : *Iris : Internationales rechtsinformatik symposium. association for computational linguistics*.
- Waltl, Bernhard, Bonczek, Georg, Scepankova, Elena, Landthaler, Jörg, & Matthes, Florian. 2017b. Predicting the outcome of appeal decisions in

- germany's tax law. *Pages 89–99 of : International conference on electronic participation*. Springer.
- Waltl, Bernhard, Bonczek, Georg, & Matthes, Florian. 2018. Rule-based information extraction : Advantages, limitations, and perspectives. *Just-letter it*, Feb.
- Welch, Lloyd R. 2003. Hidden Markov models and the Baum-Welch algorithm. *Ieee information theory society newsletter*, **53**(4), 10–13.
- Wyner, Adam, & Peters, Wim. 2010. Lexical Semantics and Expert Legal Knowledge towards the Identification of Legal Case Factors. *Pages 127–136 of : JURIX*, vol. 10.
- Wyner, Adam, Mochales-Palau, Raquel, Moens, Marie-Francine, & Milward, David. 2010. Approaches to text mining arguments from legal cases. *Pages 60–79 of : Semantic Processing of Legal Texts : where the Language of Law Meets the Law of Language*. Berlin, Heidelberg : Springer-Verlag.
- Wyner, Adam Z. 2010. Towards annotating and extracting textual legal case elements. *Informatica e diritto : special issue on legal ontologies and artificial intelligent techniques*, **19**(1-2), 9–18.
- Xiao, Richard. 2010. Corpus creation. *Chap. 7, page 146–165 of : Indurkha, Nitin, & Damerau, Fred J . (eds), Handbook of natural language processing*, second edn. Chapman and Hall.
- Zhu, Xiaojin. 2010. *Conditional random fields*. CS769 Spring 2010 Advanced Natural Language Processing.