

Résumé

Mots clés : Analyse de données textuelles, extraction de connaissances jurisprudentielles, analyse descriptive de corpus

Abstract

In this dissertation, we investigate the application of some text mining methods to support descriptive and predictive analysis of the jurisprudence.

Keywords : textual data analysis, jurisprudential knowledge extraction, descriptive analytics of corpora

Table des matières

Résumé	i
Abstract	ii
Table des matières	iii
Liste des figures	ix
Liste des tableaux	xi
Introduction générale	1
i Contexte et motivations	1
ii Objectifs	4
ii.a Collecte, gestion et pré-traitement des documents	8
ii.b Extraction de connaissances	10
ii.c Analyse descriptive	11
iii Méthodologie	12
iv Résultats	12
v Structure de la thèse	14
Chapitre 1 Analyse automatique de corpus judiciaires	15
1.1 Introduction	15
1.2 Annotation et extraction d'information	19
1.3 Classification des jugements	20
1.4 Similarité entre décisions judiciaires	21

1.5 Conclusion	24
Chapitre 2 Annotation des sections et entités juridiques	25
2.1 Introduction	25
2.2 Extraction d'information par étiquetage de séquence	28
2.2.1 Les modèles graphiques probabilistes HMM et CRF	30
2.2.1.1 Les modèles cachés de Markov (HMM)	31
2.2.1.2 Les champs conditionnels aléatoires (CRF)	32
2.2.2 Représentation des segments atomiques	34
2.2.3 Schéma d'étiquetage	36
2.3 Architecture proposée	37
2.3.1 Définition de descripteurs candidats	39
2.3.1.1 Descripteurs pour la détection des sections	39
2.3.1.2 Descripteurs pour la détection d'entités	40
2.3.2 Sélection des descripteurs	42
2.3.2.1 Sélection pour le modèle CRF	42
2.3.2.2 Sélection pour le modèle HMM	44
2.4 Expérimentations et discussions	44
2.4.1 Conditions d'expérimentations	44
2.4.1.1 Annotation des données de référence	44
2.4.1.2 Mesures d'évaluation	45
2.4.1.3 Outils logiciels	47
2.4.2 Sélection du schéma d'étiquetage	48
2.4.3 Sélection des descripteurs	48
2.4.4 Evaluation détaillée pour chaque classe	51
2.4.5 Discussions	51
2.4.5.1 Confusion de classes	51
2.4.5.2 Redondance des mentions d'entités	53
2.4.5.3 Impact de la quantité d'exemples annotés	54
2.4.5.4 Descripteurs manuels vs. réseau de neurones	56

2.5 Conclusion	57
Chapitre 3 Extraction des données concernant les demandes et leurs résultats correspondants	59
3.1 Introduction	59
3.1.1 Données cibles à extraire	60
3.1.1.1 Catégorie de demande	60
3.1.1.2 Quantum demandé	60
3.1.1.3 Sens du résultat	61
3.1.1.4 Quantum obtenu ou résultat	61
3.1.2 Expression, défis et indicateurs d'extraction	62
3.1.3 Formulation du problème	63
3.2 Travaux connexes	64
3.2.1 Problèmes analogues : extraction d'éléments structurés	64
3.2.2 Approches d'extraction d'éléments structurés	65
3.2.3 Extraction de la terminologie d'un domaine	67
3.2.3.1 Métriques non-supervisées	68
3.2.3.2 Métriques supervisées	69
3.2.3.3 Discussions	72
3.3 Méthode	72
3.3.1 Détection des catégories par classification des documents	72
3.3.2 Extraction basée sur la proximité entre sommes d'ar- gent et les termes-clés	74
3.3.2.1 Pré-traitement	75
3.3.2.2 Apprentissage des termes-clés d'une catégorie	75
3.3.3 Application de l'extraction à de nouveaux documents .	76
3.4 Résultats expérimentaux	77
3.4.1 Données d'évaluation	77
3.4.2 Métriques d'évaluation	78
3.4.3 Détection des catégories par classification	80

3.4.4	Extraction de données des paires demandes-résultats .	81
3.4.5	Analyse des erreurs	83
3.5	Conclusion	85
Chapitre 4 Identification du sens du résultat par classifica-		
tion des documents		86
4.1	Introduction	86
4.2	Classification de documents	88
4.2.1	Algorithmes traditionnels de classification de données .	88
4.2.1.1	Le Bayésien naïf (NB)	88
4.2.1.2	Machine à vecteurs de support (SVM)	90
4.2.1.3	k -plus-proches-voisins (kNN)	90
4.2.1.4	Arbre de décision	92
4.2.1.5	Analyses discriminantes linéaires et quadra- tiques	95
4.2.2	Algorithmes dédiés aux textes	97
4.2.2.1	NBSVM	97
4.2.2.2	FastText	97
4.2.3	Discussion : Techniques d'amélioration de la précision .	98
4.3	La Regression PLS et ses extensions	98
4.3.1	Gini-PLS	99
4.3.2	Logit-PLS	100
4.3.3	Gini-Logit-PLS	100
4.4	Méthode	100
4.5	Expérimentations et résultats	100
4.5.1	Protocole d'évaluation	100
4.5.2	Classification de l'ensemble du document	101
4.5.3	Réduction du document aux régions comprenant le vo- cabulaire de la catégorie	103

4.6 Conclusion	103
Chapitre 5 identification des Circonstances Factuelles	104
5.1 Introduction	104
5.2 Formulation du Problème	105
5.3 Regroupement non-supervisé de documents	105
5.3.1 Choix de l'algorithme de clustering	105
5.3.2 Métrique de similarité ou de dissimilarité (Comment mesurer la similarité ?)	105
5.3.3 Déterminer le nombre approprié de clusters	106
5.3.4 Définir une représentation appropriée pour les textes .	107
5.3.5 Labeliser les clusters	108
5.3.6 Evaluation du clustering généré	108
5.3.6.1 Évaluation supervisée	108
5.3.6.2 Évaluation non-supervisée	108
5.4 Méthodes proposées	108
5.4.1 K-médoïdes et « Word Mover's Distance »	108
5.4.2 Apprentissage d'une métrique fondée la modification du document	109
5.5 Expérimentations et interprétation des résultats	113
5.5.1 Annotations de données d'évaluation	113
5.5.2 Apprentissage de la métrique	113
5.5.3 Comparaison d'approches	113
5.6 Conclusion	114
Chapitre 6 Démonstrateur d'analyse descriptive	115
6.1 Objectif et Cas d'Utilisation	115
6.2 Description du Pipeline	115
6.3 Illustration d'analyses descriptives	115

6.3.1	Implémentation du système	115
6.3.2	Données	116
6.3.2.1	Distribution de la base dans l'espace et dans le temps	116
6.3.3	Analyse du sens du résultat	116
6.3.3.1	Evolution dans le temps	116
6.3.3.2	Différence dans l'espace	116
6.3.4	Analyse des quanta	116
6.3.4.1	Evolution dans le temps	116
6.3.4.2	Différence dans l'espace	116
6.3.4.3	Quantum demandé vs. quantum accordé . . .	116
6.4	Conclusion	116
Conclusion		117
F.5	Contributions	117
F.6	Critique du travail	117
F.7	Travaux futurs de recherche	117
F.8	Perspectives du domaine	117
Bibliographie		118

Liste des figures

1	Exemples de critères des moteurs de recherche juridique	2
2	Organisation des institutions judiciaires françaises	4
3	La demande au centre de la compréhension des décisions . . .	6
4	Chaine d’analyse du corpus jurisprudentiel à mettre en œuvre	8
2.1	Illustration des schémas d’étiquetage IO, BIO, IEO, BIEO . .	36
2.2	Application des modèles entraînés pour l’étiquetage de sections et entités.	38
2.3	Entraînement des modèles.	39
2.4	Matrice de confusion entre méta-données d’entête avec le modèle CRF	53
2.5	Matrice de confusion entre lignes des sections avec le modèle CRF	54
2.6	Courbes d’apprentissages aux niveaux élément et entité	55
3.1	Enoncés simples, ou comprenant des références et des agrégations (extraits de la décision 14/01082 de la cour d’appel de Saint-Denis (Réunion))	60
3.2	Illustration de la proximité des quantas et termes-clés	74
3.3	Répartitions des demandes dans les documents annotées. . . .	78
4.1	Répartition des sens de résultat dans les données annotées. . .	87
4.2	Illustration de l’hyperplan d’un SVM.	91
4.3	Répartition des documents à une demande de la catégorie considérée.	101

Liste des tableaux

1	Nombre de décisions prononcées en France par an de 2013 à 2017	9
2.1	Exemples d'entités et statistiques sur la base d'exemples annotées manuellement	26
2.2	Mots représentatifs des 10 premiers thèmes sur les 100 inférés	47
2.3	Comparaison des schémas d'étiquetage.	49
2.4	Performances des sous-ensembles sélectionnés de descripteurs.	50
2.5	Précision, Rappel, F1-mesures pour chaque type d'entité et section au niveau atomique.	52
2.6	Précision, Rappel, F1-mesures pour chaque type d'entité et section au niveau entité.	52
2.7	Comparaison entre le CRF avec des descripteurs définis manuellement et le BiLSTM-CRF au niveau entité.	56
3.1	Exemples de catégories de demandes	61
3.2	Exemples d'analogie entre relations, évènements et demandes .	65
3.3	Notation utilisée pour formuler les métriques	68
3.4	Métriques locales	73
3.5	Mots introduisant les énoncés de demandes et de résultats . .	75
3.6	Extrait du tableau d'annotations manuelles des demandes. . .	78
3.7	Résultats d'une 5-fold validation croisée pour la détection de catégories	80
3.8	$F1_{c,(q_d,s_r,q_r),D_c}$ moyenne pour une 5-fold validation croisée pour chaque métrique de sélection de termes pour un seuil égal à 0.5	81

3.9	Résultats détaillés pour l'extraction des données avec sélection automatique de la méthode d'extraction des termes-clés	82
3.10	Types et taux d'erreurs (pourcentage en moyenne sur les 6 catégories de demandes)	83
3.11	Taux de quanta demandés (q_d) mentionnés dans les documents annotés	83
3.12	Taux de quanta accordés (q_r) mentionnés dans les documents annotés	84
3.13	Premiers termes sélectionnés lors de la première itération de la validation croisée	84
4.1	Comparaison des algorithmes sur une représentation globale des documents pour la détection du sens du résultat.	102
4.2	Détails des résultats de FastText et NBSVM.	102
4.3	Détection du sens du résultat : Comparaison des réductions du document.	103

Introduction générale

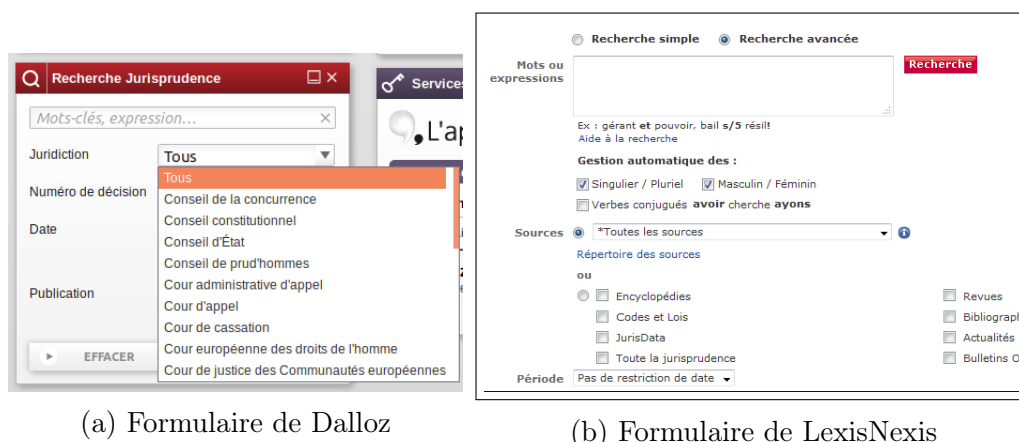
i Contexte et motivations

Une décision jurisprudentielle peut être définie soit comme le résultat rendu par les juges à l'issue d'un procès, soit comme un document décrivant une affaire judiciaire. Un tel document rapporte, notamment, les faits, les procédures judiciaires antérieures, le verdict des juges, et les raisons qui y les ont conduits. Dans cette thèse, nous désignons par « décision » le document, et par « résultat » la conclusion, ou réponse finale des juges. Une jurisprudence¹ est un ensemble de décisions rendues par les tribunaux et qui représente la manière dont ces derniers interprètent les lois pour résoudre un problème juridique donné (type de contentieux). Les juristes doivent collecter ces documents, les sélectionner, les analyser afin de mieux la prise de décision des juridictions pour mener, par exemple, des recherches empiriques en droit [Ancel, 2003, Jeandidier and Ray, 2006]. Les avocats exploitent aussi les décisions passées pour anticiper les résultats des juges. Ils peuvent ainsi mieux conseiller leurs clients sur le risque judiciaire que ces derniers encourent, et sur la stratégie à adopter pour un type de contentieux. Cette activité de collecte et d'analyse est manuelle en général, et par conséquent, sujette à plusieurs difficultés liées à l'accès et à l'exhaustivité des documents.

Les documents sont dispersés entre les tribunaux. Les procédures administratives ne facilitent pas toujours leur accès du fait de la nécessité de préserver la confidentialité des parties. Les décisions n'étant pas anonymisées ne peuvent être rendues aux juristes qui en font la demande. Un

1. <http://www.toupie.org/Dictionnaire/Jurisprudence.htm>

certain nombre de documents sont néanmoins accessibles sur internet grâce à des sites de publication de données ouvertes gouvernementales, comme <http://data.gouv.fr> en France, <https://www.judiciary.uk> en Grande Bretagne, <http://www.scotusblog.com/> aux Etats-Unis, et <https://www.scc-csc.ca/> au Canada. Ces derniers publient régulièrement des décisions récemment prononcées. Il existe aussi des moteurs de recherche juridiques qui permettent de retrouver des décisions intéressantes. Cependant, qu'ils soient payants (LexisNexis², Dalloz³, Lamyline⁴,...) ou gratuits (CanLII⁵, Légifrance⁶, ...), leurs critères de recherche limitent la pertinence des résultats. En effet, il ne s'agit en général que de combinaisons de mots-clés et autres métadonnées (date, type de juridiction, ...), ou d'expressions régulières, comme l'illustre la Figure 1. Ces critères n'appréhendent pas la sémantique juridique qui ramènerait, aux juristes, des échantillons plus pertinents.



(a) Formulaire de Dalloz

(b) Formulaire de LexisNexis

Figure 1 – Exemples de critères des moteurs de recherche juridique

L'exhaustivité de l'analyse, ou tout au moins sa représentativité, rencontre un frein face à l'énorme volume de documents existants. En effet, plus

2. <https://www.lexisnexus.fr/>

3. <http://www.dalloz.fr>

4. <http://lamyline.lamy.fr>

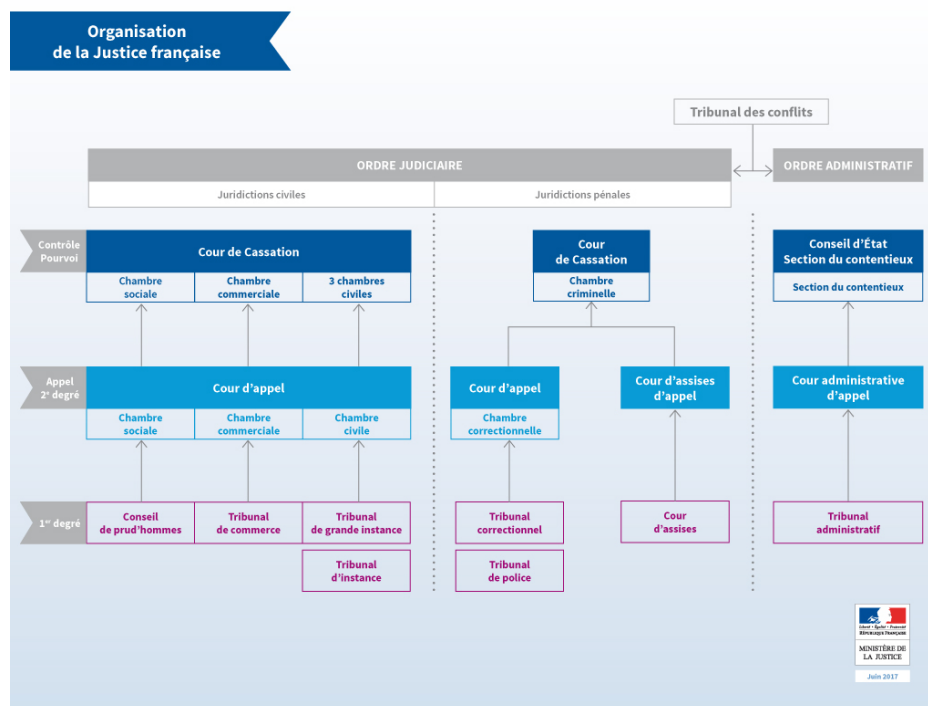
5. <https://www.canlii.org>

6. <https://www.legifrance.gouv.fr>

de 4 millions de décisions sont prononcées en France par an d'après les chiffres du ministère français de la justice (Tableau 1).

Au regard de la croissance rapide de la quantité de décisions, on imagine facilement que même pour une étude sur une question très précise, le corpus utile reste large. Par ailleurs, il peut s'avérer très pénible de lire les décisions. Au regard de la croissance rapide de la quantité de décisions, on imagine facilement que même pour une étude sur une question très précise, le corpus utile reste large. Par ailleurs, il peut s'avérer très pénible de lire les décisions pour en identifier les données d'intérêt. Les documents sont très souvent longs. Ils sont aussi complexes dans leur style de rédaction. Par exemple, les phrases sont longues et comprennent plusieurs clauses discutant parfois d'aspects différents. On y retrouve aussi des références à des jugements antérieurs, et des omissions. Les documents sont très souvent longs. Ils sont aussi complexes dans leur style de rédaction. Par exemple, les phrases sont longues et comprennent plusieurs clauses discutant parfois d'aspects différents. On y retrouve aussi des références à des jugements antérieurs, et des omissions. Il est évident qu'une automatisation du traitement des corpus de décisions s'impose pour répondre aux diverses difficultés d'accès, de volumétrie, et de complexité liées à la compréhension des décisions. L'automatisation ferait gagner du temps aux juristes sur des tâches de traitement préalables à leur raisonnement d'experts, tout en leur fournissant une vue pertinente de la jurisprudence. D'autre part, Cretin [2014] fait remarquer que la justice est complexe dans son organisation (Figure 2) et son fonctionnement, et son langage est pratiquement incompréhensible. Il est donc presque impossible pour les "profanes" d'estimer leurs droits et le risque judiciaire qu'ils encourent dans leur quotidien sans consulter un initié en droit. L'automatisation pourrait ainsi améliorer l'accessibilité du droit dans ce cas. L'exigence pour le profane étant l'exacte pertinence des ressources, leur accessibilité, et l'intuitivité du processus de leur exploitation [Nazarenko and Wyner, 2017]. Le traitement automatique constitue, en

résumé, une aide précieuse non seulement pour les professionnels du droit, mais aussi pour les particuliers et entreprises soucieux de voir l'issue de leur affaire leur être favorable. Par exemple, en comparant le montant qu'on peut espérer d'une juridiction et le coût d'un procès, on peut plus aisément se décider entre un arrangement à l'amiable et la poursuite du litige en justice [Langlais and Chappe, 2009].



Source : <http://www.justice.gouv.fr/organisation-de-la-justice-10031/>

Figure 2 – Organisation des institutions judiciaires françaises

ii Objectifs

Ce mémoire propose des stratégies et modèles visant à automatiser l'extraction d'information à partir des décisions françaises. Le but est de faciliter

la classification, l'analyse descriptive, et l'analyse prédictive sur une masse de documents. L'approche traditionnelle d'analyse d'un contentieux [Ancel, 2003] consiste à :

1. **Choisir un échantillon représentatif** : collection des décisions suivant des contraintes définies : période précise et d'une couverture géographique, types d'affaires, etc.
2. **Sélectionner les décisions** : élimination des décisions qui ne correspondent pas au type de demande d'intérêt.
3. **Elaborer la grille d'analyse** : création d'un modèle de grille (tableau) qui permettra d'enregistrer les informations potentiellement importantes. Chaque ligne correspond à une demande et les colonnes sont les différents types d'informations qu'il est possible d'extraire sur une demande. Ces variables vont de la procédure suivie, aux solutions proposées en passant par la nature de l'affaire. Les champs à remplir ne sont pas connus à l'avance ; c'est au cours de la lecture des décisions qu'on retrouve les informations qui paraissent intéressantes.
4. **L'analyse des décisions et l'interprétation des informations** : saisie des décisions et calculs statistiques dans un logiciel tableur.

Ancel [2003] évoque principalement le problème de la différence entre l'état capté de la jurisprudence et son état présent. D'une part en effet, les longs délais de travail sont caractéristiques de ces études. Nous avons pour exemple, l'étude menée par l'équipe de Jeandidier and Ray [2006] pour l'analyse empirique des déterminants de la fixation de pensions alimentaires pour enfant lors de divorce. Cette analyse a duré 9 mois pour l'extraction manuelle des informations et la modélisation par régression de la relation entre les déterminants extraits et les pensions alimentaires accordées. D'autre part, il est impossible d'observer l'évolution des pratiques judiciaires dans le temps et dans l'espace du fait de la faible taille de l'échantillon choisi. Notre principal objectif est donc de proposer des solutions pour un traitement rapide et efficace d'une grande masse de décisions.

La problématique de notre étude de « **capter automatiquement la sémantique d'un corpus jurisprudentiel pour comprendre la prise de décision des juges sachant que l'interprétation subjective des règles juridiques rend l'application de la loi non déterministe ?** ». Cette question intéresse des entreprises telles que LexisNexis, et plusieurs startups à l'exemple de Predictice⁷ et CASE LAW ANALYTICS⁸. Afin d'y répondre, nous nous intéressons aux concepts manipulés par les experts, au centre desquels on retrouve les demandes des parties (prétentions) et qui feront l'objet d'une décision. Tout autour de la demande, gravitent d'autres concepts importants qui enrichissent la compréhension de la décision (Figure 3).

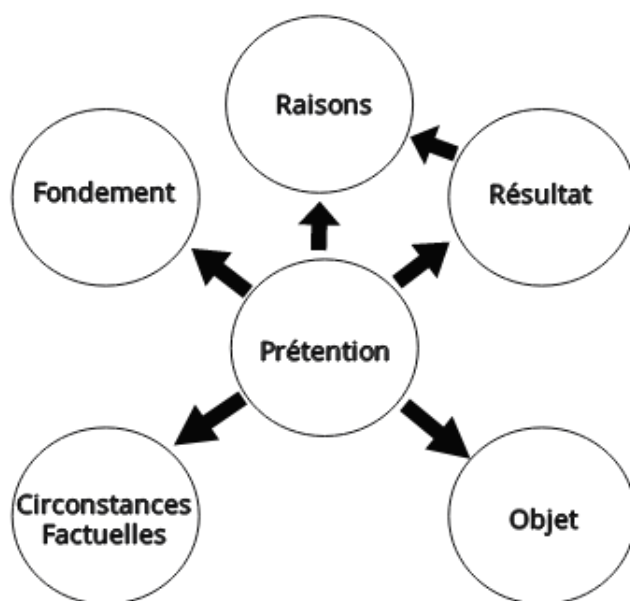


Figure 3 – La demande au centre de la compréhension des décisions

Ces autres aspects d'une décision sont :

- le résultat associé qui est décrit par une polarité (« accepte » ou « rejette »), souvent lié à un quantum accordé, par exemple 5000 euros de

7. <http://predictice.com>

8. <http://caselawanalytics.com>

- dommages-intérêts ou 2 mois d'emprisonnement ;
- le fondement ou la norme juridique qui est la règle qui est associé et qui légitime la prétention ou le résultat ;
- l'objet qui représente ce qui a été demandé (par ex. dommages et intérêts) ;
- les circonstances factuelles dans lesquelles sont formulées les demandes ; elles décrivent les types de faits caractérisant ainsi les types de contentieux ou d'affaires ;
- les divers arguments apportés par les parties (resp. les juges) pour justifier leurs requêtes (resp. leurs solutions).

En fait, Ces concepts descriptifs d'une demande, et qui gravitent autour de celle-ci, couvre l'essentiel de l'information pertinente pour les experts. L'analyse sémantique vise donc à identifier les connaissances sur les nombreuses demandes présentent dans les décisions.

Les travaux de cette thèse s'inscrivent dans un projet qui vise, entre autres, à automatiser l'extraction de l'ensemble de ces informations, de les structurer afin de concevoir une base de connaissances par catégories demande dont le but est de fournir une estimation des chances d'obtenir un résultat positif suivant des critères comme la juridiction, le type de demande, ou les circonstances du litiges, et d'identifier les facteurs influençant le résultat. Le projet comprend deux phases principales : une phase d'indexation des connaissances de la masse des décisions, suivie d'une phase d'analyse prédictive. La phase d'indexation doit déjà permettre de réaliser automatiquement, de manière exhaustive, des analyses descriptives. Ces dernières consistent, par exemple, à comparer le nombre d'acceptations à la fréquence des rejets. Par conséquent, le système doit apprendre à reconnaître dans les décisions, les informations pertinentes sur les prétention et résultats associés. La phase d'analyse consiste à regrouper des paquets de décisions similaires (même résultat sur la même prétention dans les circonstances similaires), pour découvrir les facteurs influençant le sens du résultat (par ex. le fait que

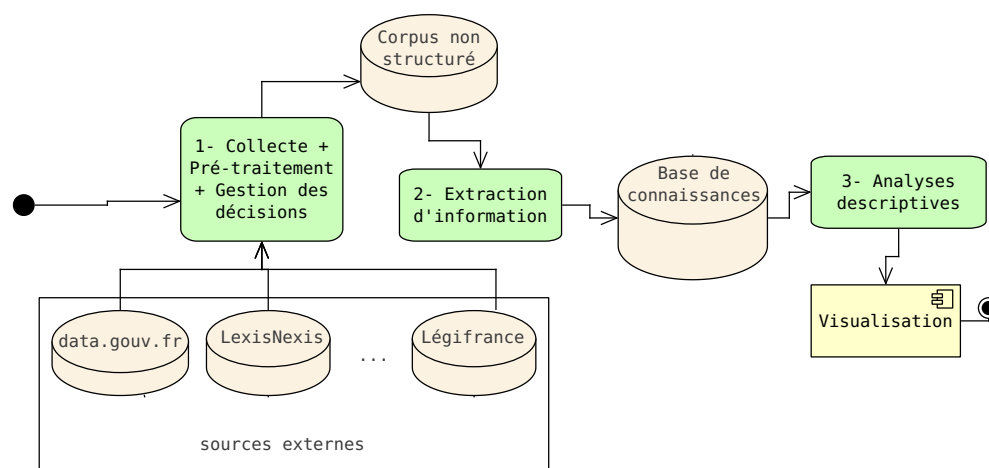


Figure 4 – Chaîne d’analyse du corpus jurisprudentiel à mettre en œuvre

« le revenu de l’époux soit le plus élevé du foyer » encourage les juges à accorder la pension alimentaire à l’épouse). En effet, c’est la connaissance de ces factuels circonstanciels démunis de toute teneur juridique qui permet à l’expert de pouvoir anticiper les décisions judiciaires.

La chaîne de traitement à mettre en œuvre consiste en quatre étapes principales qui s’enchaînent comme le présente la figure 4. Ce document n’aborde pas l’aspect prédictif du projet. Notre étude s’est limitée aux problématiques liées à l’analyse descriptive, et qui sont décrites dans la suite.

ii.a Collecte, gestion et pré-traitement des documents

Le volume de décisions prononcées croît très rapidement (Tableau 1).

Il est donc nécessaire de trouver des moyens pour collecter le maximum de documents bruts non-structurés, les pré-traiter, et organiser leur gestion afin de les indexer en local pour faciliter leur traitement. Les décisions de cours d’appel de justice civile sont les plus accessibles à partir des moteurs de recherche juridique (LexisNexis, Dalloz, LamyLine, Legifrance, etc.) et de la grande base de données JuriCa de la Cour de cassation. Cependant, l’accès

Justice	2013	2014	2015	2016	2017
civile	2 761 554	2 618 374	2 674 878	2 630 085	2 609 394
pénale	1 303 469	1 203 339	1 206 477	1 200 575	1 180 949
administrative	221 882	230 477	228 876	231 909	242 882

Source : <http://www.justice.gouv.fr/statistiques-10054/chiffres-cles-de-la-justice-10303/>

Tableau 1 – Nombre de décisions prononcées en France par an de 2013 à 2017

à ces décisions est généralement payant et le nombre de documents simultanément téléchargeables est très faible sur les sites payants (généralement 10 à 20 décisions au maximum à la fois). De plus, le nombre de téléchargements par jour est limité. La base JuriCa est la plus grosse base de décisions de cours d’appel en France. Elle est gérée par la Cour de cassation. L’accès à cette base est offert par Le service de documentation, des études et du rapport⁹ (SDER). Cet accès est payant pour les professionnels et gratuit pour les universités et centres de recherche en partenariat avec le SDER. Légitime, le moteur de recherche du ministère de la justice, fournit quant à lui un accès public et gratuit à un nombre considérable de documents. Les décisions y sont identifiées à l’aide de numéros consécutifs et accessibles à partir d’un service web par une requête GET du protocole HTTP. Ainsi, il est possible de programmer un client web capable de télécharger l’ensemble des documents de Legifrance. Ce dernier a l’avantage de proposer des décisions de tous les ordres et de tous les degrés. Cependant, les décisions des juridictions du premier degré (appelées jugements) restent plus rares sur internet et principalement disponibles auprès des tribunaux. La disponibilité des décisions du second degré ou d’appel (appelées arrêts) en justice civile est l’une des raisons pour lesquelles notre étude s’est portée sur celles-ci.

Les décisions existent sous divers formats PDF, DOC, DOCX, RTF, TXT, XML, etc. Il arrive parfois qu’un fichier téléchargé comprenne plusieurs décisions (sur LexisNexis par exemple). Nous avons par conséquent préféré

9. https://www.courdecassation.fr/institution_1/composition_56/etudes_rapport_28.html

convertir tous les documents au format plein texte pour homogénéiser les traitements. Par ailleurs, les décisions sont collectées à partir de diverses sources pouvant contenir des documents identiques. Il se pose donc un problème d'identification unique des décisions pour éviter des redondances. Pour cela, nous avons défini un schéma de nomination unique des fichiers. Ce dernier repose sur 3 informations : le type de juridiction (tribunal, cour d'appel, ...), la ville, et le numéro R.G. (registre général) qui est l'identifiant unique de la décision au sein de la juridiction. Par exemple, le numéro « CAREN1606137 » identifie la décision de numéro R.G. « 16/06137 » de la cour d'appel (« CA ») de la ville de Rennes (« REN »). Ces 3 informations sont présentes dans les premières lignes de la décision, et sont facilement identifiables à l'aide d'une routine à base de règles simples. D'autre part, certains moteurs de recherche ne fournissent souvent qu'un résumé au lieu du contenu original des décisions. Il est important de supprimer ces fichiers du corpus.

ii.b Extraction de connaissances

Les problématiques d'extraction de connaissances constitue la pierre angulaire de cette thèse car les informations sur les demandes, les parties, les juges, les juridictions et les faits conditionnent la qualité des prévisions du sens du résultat pour un type de demande considéré. La difficulté découle de l'état non-structuré des documents, et de la complexité et la spécificité du langage employé. L'extraction des connaissances nécessite de mettre en œuvre des techniques de fouille de textes adaptées à la nature des éléments à identifier. Nous avons ainsi abordé l'annotation des références de l'affaire (juridiction, ville, participants, juges, date, numéro R.G., normes citées, ...), l'extraction des demandes et résultats correspondants, et l'identification des circonstances factuelles.

Les métadonnées de référence sont des segments de texte qu'on peut directement localiser dans le document. Leur reconnaissance est donc semblable à celle des entités nommées. C'est une problématique intensivement étudiée

en traitement automatique du langage naturel [Yadav and Bethard, 2018] dans plusieurs travaux et compétitions, aussi bien pour des entités communes [Tjong Kim Sang and De Meulder, 2003, Grishman and Sundheim, 1996], que pour des entités spécifiques à un domaine [Kim et al., 2004, Persson, 2012, Hanisch et al., 2005], et dans diverses langues [Li et al., 2018, Alfred et al., 2014, Amarappa and Sathyanarayana, 2015].

Le problème d'extraction des demandes et de la réponse correspondante des juges consiste à reconnaître pour chaque prétention : son objet, son fondement, le quantum demandé, le sens du résultat, et le quantum accordé. La paire demande-résultat s'apparente donc à des entités structurées comme les événements ACE [2005] qui sont décrits par un type, un terme-clé, des participants, un temps, une polarité.

Le problème d'identification des circonstances factuelles consiste à constituer des regroupements de décisions mentionnant une certaine catégorie de demande (objet+fondement). Le but est, comme indiqué précédemment, de repérer les différentes situations dans lesquelles cette catégorie de demande est formulée. Chacun des groupes représente donc une situation particulière partagée par les membres du groupe mais bien distinctes de celles reflétées par les autres groupes. Ce problème évoque des problématiques de similarité entre texte, de regroupement non supervisé (*clustering*), et de « modélisation thématique » (*topic modeling*). La similarité pourra faire l'objet, dans un travail futur, d'identification des raisons

A l'issue du processus d'extraction, les données extraites sont destinées à enrichir progressivement une base de connaissances. La structuration des données au sein d'une base facilite les diverses analyses automatiques applicables aux décisions et demandes judiciaires.

ii.c Analyse descriptive

L'analyse descriptive exploite l'ensemble des connaissances extraites et organisées pour répondre aux diverses questions que l'on pourrait se poser

sur l'application de la loi. Il est intéressant par exemple de comparer les fréquences de résultats positifs et négatifs pour une catégorie de prétention donnée dans une situation précise. Les quanta extraits servent à visualiser les différences entre les montants accordés et réclamés. D'autres analyses plus complexes permettraient d'étudier l'évolution dans le temps et les différences dans l'espace de l'opinion des juges.

iii Méthodologie

Comme illustrées précédemment (§ ii.b), les problématiques propres aux textes juridiques trouvent généralement des analogies avec les problèmes d'analyse de données textuelles. Ainsi, les méthodes issues de l'énorme progrès réalisé dans ce domaine sont applicables aux textes juridiques. Cependant, quelques adaptations sont généralement nécessaires pour obtenir des résultats de bonne qualité hors des domaines pour lesquels ces approches ont été développées [Waltl et al., 2016]. De plus, la recherche en fouille de textes est souvent réalisée sur des échantillons qui ne reflètent pas toujours la complexité des données réelles. Effectuant l'une des premières études d'analyse sémantique des décisions françaises, nous avons axé notre travail sur le rapprochement des problèmes liés à l'analyse des décisions jurisprudentielles à celles qui sont généralement traitées en analyse de données textuelles. Il s'agit ensuite d'établir des protocoles d'évaluation et d'annotation manuelle de données. Selon les problématiques identifiées et les protocoles d'évaluations définis, des méthodes adaptées sont été proposées et expérimentés sur les données réelles annotées par des experts.

iv Résultats

Une chaîne de traitement pour le sectionnement et l'annotation des métadonnées est proposée. L'applicabilité de deux modèles probabilistes, les

champs aléatoires conditionnels ou CRF (*conditional random fields*) et les modèles cachés de Markov ou HMM (*hidden Markov Model*), est étudiée en considérant plusieurs aspects de la conception des systèmes d'extraction d'entités nommées. Le sectionnement a pour but d'organiser l'extraction des informations qui sont réparties dans des sections selon leur nature.

Par la suite, nous proposons une méthode d'extraction des demandes et résultats en fonction des catégories présentes dans la décision. L'approche consiste en effet à identifier dans un premier temps les catégories présentes (objet+fondement) par classification supervisée. Un vocabulaire d'expression des demandes et résultats est exploité pour identifier les passages. Puis à l'aide de termes propres à chacune des catégories identifiées, les trois attributs (quantum demandé, sens du résultat, quantum accordé) des paires demande-résultat sont reconnus.

Par ailleurs, nous analysons l'extraction particulière du sens du résultat par classification binaire des documents. L'objectif est de s'affranchir de l'identification préalable de l'expression des demandes et résultats. En effet, les décisions comprenant des demandes d'une catégorie donnée semblent en contenir, dans une forte proportion, une seule demande. Nous pensons qu'il n'est par conséquent pas nécessaire d'identifier l'expression de cette dernière pour en déterminer le sens. A partir d'une représentation adéquate du contenu de la décision, il est possible classer cette dernière à l'aide d'un modèle de classification supervisée de documents.

L'identification des circonstances factuelles, quant à elle, est modélisée comme une tâche de regroupement non supervisée des décisions. Nous proposons dans ce cas une méthode d'apprentissage d'une métrique de dissimilarité sémantique entre textes, à l'aide d'un modèle adéquat de régression. Nous analysons différents modèles de régression. La métrique apprise a été comparée à d'autres distances établies en recherche d'information.

v Structure de la thèse

La thèse est organisée en 6 chapitres. Le chapitre 1 positionne nos travaux par rapport à ceux qui ont été réalisés précédemment sur des problématiques proches. Le chapitre 2 présente l'architecture de structuration et reconnaissance des entités juridiques, et discute des différents résultats empiriques obtenus par application des modèles CRF et HMM. Ensuite, le chapitre 3 détaille le problème d'extraction des paires demande-résultat, puis présente notre méthode et les résultats obtenus. Le chapitre 4 discute de l'extraction particulière du sens du résultat par classification directe des documents en comparant différents algorithmes et méthodes de représentation des textes. Le chapitre 5 présente notre approche d'apprentissage de la métrique de dissimilarité textuelle, et la compare à des distances établies en recherche d'information sur le problème d'identification des circonstances factuelles. Enfin, le chapitre 6 présente les résultats de scénarios d'analyses descriptives pour illustrer l'exploitation potentielle de nos propositions sur des corpus de grande taille.

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. L'état de l'art plus technique est décrit dans les chapitres qui traitent, dans la suite, des méthodes que nous avons mises en œuvre.

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 and Genesereth, 2005]. Il vise à réduire la taille et la complexité de la loi pour la rendre plus accessible. Plus

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

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 donc de raisonnements à base de cas qui se distinguent du raisonnement à base de règles.

Les premiers outils automatiques d'anticipation des décisions étaient généralement des systèmes experts juridiques. Ces derniers résonnent 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 d'avantage aux États-Unis qu'ailleurs. Pour exemple, Katz et al. [2014] entraînent des forêts aléatoires sur les cas de 1946-1953 pour prédire si la Cour Suprême des États-Unis infirmera ou confirmera une décision de juridiction inférieure. Ils ont réussi à atteindre 69,7% des décisions finales pour 7 700 cas des années 1953-2013 ; des résultats qu'ils ont légèrement améliorés plus récemment en augmentant le nombre d'arbres et la quantité de données [Katz et al., 2017]. PToujours pour la prédiction des décisions de la Cour Suprême des États-Unis, Walzl et al. [2017a] utilisent des techniques de traitement automatique du langage naturel (TALN) et extraient automatiquement moins de caractéristiques que [Katz et al., 2014] à partir des décisions d'appel de la Cour Fiscale Allemande (11 contre 244). Ils obtiennent des valeurs de f1-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. D'autre part, Ashley and Brüninghaus [2009] ont obtenu une précision de 91,8% en tentant de prédire

la partie (plaignant/défendeur) qui sera favorisée à l'issue d'affaires d'appropriation illicite de secrets commerciaux. Contrairement à [Katz et al., 2014] qui catégorisent les caractéristiques de valeurs prédéfinies pour caractériser la décision débattue, les tribunaux et les juges (opinions politiques, origine de l'affaire, identifiant du juge, raison et sens du dispositif de la cour inférieure), Ashley and Brüninghaus [2009] identifient, par classification, des facteurs pouvant influencer la décision. Les valeurs des caractéristiques de ces différents travaux sont prédéfinies et très limitées, et ne reflètent pas, par conséquent, la grande diversité de catégories qu'on peut retrouver dans les décisions.

Notre objectif est d'alimenter les analyses quantitatives de corpus jurisprudentiels en proposant des méthode d'extraction de connaissances pertinentes telles que les références des affaires (juge, date, juridiction, etc.), les règles juridiques associées, les demandes de parties, les réponses des tribunaux, et les liens entre ces données. Les juges apportent une réponse à chaque demande, et par conséquent une partie peut voir ses demandes soit toutes acceptées ou rejetées, soit une partiellement accordées. 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). L'identification de ces diverses connaissances est possible par l'analyse sémantique des textes judiciaires grâce aux méthodes du TALN et de l'analyse (ou fouille) de données textuelles. Cependant, l'application de ces techniques exigent certaines adaptations pour surmonter les divers défis décrits par Nazarenko and Wyner [2017] : textes très longs et en grande quantité, corpus régulièrement mis à jour, influence subjective de facteurs sociaux et d'opinions politiques, couvertures de problématiques économiques, sociales, politiques très variées,

langage complexe, etc. . Dans la suite, nous passons en revue des travaux qui ont été menés dans ce sens pour traiter de problématiques proches des nôtres, en particulier celles décrites précédemment dans l'introduction (Section § ii.b).

1.2 Annotation et extraction d'information

L'annotation consiste à enrichir les documents pour les préparer à d'autres 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 très 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 and Peters, 2010, Shulayeva et al., 2017], les définitions [Waltl et al., 2016, 2017b], 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. C'est le cas des modèles probabilistes HMM et CRF que nous étudions dans le chapitre 2. Ils peuvent être combinés à d'autres approches dans un système global. Après avoir segmenté les documents à l'aide d'un modèle CRF, Dozier et al. [2010] ont combiné plusieurs approches pour reconnaître des entités dans les décisions de la Cour Suprême des États-Unis. Ils ont définis 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 élé-

ments à extraire, un système peut exploité un lexique pour les motifs simples et non-systématiques (indicateurs de mentions de résultats ou de parties) et les règles pour des motifs plus complexes et systématiques (par ex. noms de juges, énoncés de décisions) [Waltl et al., 2016, 2017b, Wyner, 2010]. Cardellino et al. [2017] ont plutôt utilisé un modèle CRF et les réseaux de neurones sur des jugements de la Cour Européenne des Droits de l’Homme. Les basses performances, qu’ils rapportent, illustrent bien la difficulté de la détection d’entités juridiques. Plus récemment encore, Andrew and Tannier [2018] obtiennent de bons résultats en combinant l’extraction d’entités non-juridiques par CRF à celle des relations entre ces dernières par une grammaire GATE JAPE [Thakker et al., 2009] sur des décisions du Luxembourg rédigées en français.

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 and Moens [2008] proposent, pour la même tâche, une approche d’extraction basée sur une formalisation de la structure des arguments dans les jugements par une grammaire sans contexte.

1.3 Classification des jugements

La classification permet d’organiser un corpus en rangeant les documents dans des catégories prédéfinies généralement par des experts. 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 jugements² de la Court Européenne des Droits de l’Hommes (ECHR). Avec un SVM (Machine à Vecteurs de Support) et une représentation vectorielle basée sur les plus fréquents n-grammes et le cluster de leur vecteur de plongement sémantique (word2vec), ils obtiennent

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

une précision moyenne de 79% sur les 3 articles qu'ils ont manipulés. Notons tout de même la sélection particulière des régions du documents où sont extraits les n-grammes (circonstances, faits, lois, ...). Cette sélection est un ajustement de la représentation des textes qui paraît nécessaire pour obtenir de bons résultats. La structuration préalable des documents est 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 tout démontrant empiriquement, entre autres, la possibilité de prédire la violation des articles sur des périodes futures à celles des données d'entraînement. Şulea et al. [2017a] traitent, d'autre part, l'identification des résultats dans des arrêts³ de la Court Française de Cassation. Après un essai avec un SVM [Şulea et al., 2017b], ils améliorent les résultats à l'aide d'un classifieur ensembliste de SVM à moyenne de probabilités, parvenant ainsi à des f1-mesures de plus de 95%. Par ailleurs, Ashley and Brüninghaus [2009] entraînent un classifieur (les plus-proches-voisins) pour chacun des 27 facteurs prédéfinis pour savoir s'il s'applique à la décision (phase SMILE). La partie remportant le procès est prédite par un algorithme séquentiel qui compare les parties (plaignant et défendeur) suivant le niveau de préférence des questions juridiques dégagées par les facteurs tel qu'observé dans la base d'entraînement (phase IBP). D'autres catégorisations, comme la formation judiciaire ou la période du prononcé [Şulea et al., 2017b,a], sont toutes aussi utiles pour faciliter la recherche d'information. La classification peut aussi être utilisée à des fins d'évaluation sur d'autres problématiques comme la similarité [Ma et al., 2018].

1.4 Similarité entre décisions judiciaires

La similarité entre textes est indispensable pour des applications qui nécessitent de rapprocher quantitativement des textes traitant de sujets simi-

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

lares, et resp. éloigner 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 qu’on veut révéler. Nair and Wagh [2018] arrivent à exploiter les citations de lois et précédents, car les jugements du « Common Law » citent des décisions d’affaires similaires antérieures. Ils analysent les réseaux de citations d’un corpus de 597 documents, à l’aide de règles d’association générées par l’algorithme Apriori pour regrouper les jugements susceptibles d’être cités ensemble. Ils démontrent au travers de scénarios (pas d’évaluation statistique) que les documents qui sont cités fréquemment ensemble sont similaires, et cette relation permet par une transitivité de retrouver les documents pertinents dans une base de données. Certaines métriques traditionnels, comme la distance cosinus [Thenmozhi et al., 2017], ont été utilisées sur les décisions judiciaires mais pas toujours avec succès. La raison peut venir notamment de la représentation des textes qui doit accentuer l’aspect sous-jacent de la similarité. Ma et al. [2018] proposent donc d’aligner le document sur une ontologie des concepts et relations du corpus judiciaire. L’idée est de calculer la similarité sur un résumé du texte qui compacte le texte uniquement sur les aspects pertinents. Cette méthode permet ainsi de mieux capter la sémantique pure des jugements, d’avoir une meilleure précision, et réduire la complexité temporelle inhérente à l’exploitation de long document notamment lors de l’utilisation de la « distance du déménageur de mot » ou WMD (*Word Mover’s Distance*) de Kusner et al. [2015]. L’amélioration a été observée sur une tâche de classification avec des jugements Chinois relatifs aux crimes de la circulation routière dans quatre catégories correspondantes à des sentences d’emprisonnement (précision de 90.3% et 92.3% pour le résumé contre 84.8% et 82.4% resp. 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 d’Inde : (i) le cosinus de similarité entre les représentations TF-IDF

de x et y (*term frequency - inverse document frequency*) dont tous les termes présents dans les jugements sont les dimensions (*all-term cosine similarity*), (ii) le cosinus de similarité des termes juridiques réduit les dimensions précédentes uniquement qu'aux termes apparaissant dans un dictionnaire juridique (*legal-term cosine similarity*), (iii) la similarité de couplage bibliographique égal au nombre de citations de jugements qu'ont en commun x et y (*bibliographic coupling similarity*), (iv) la similarité de co-citation qui est le nombre de citations de x et y dans un même jugement (*co-citation similarity*). La similarité étudiée ici est basée sur trois critères : la similarité sur la question discutée, la similarité sur les faits sous-jacents, et l'utilité du document pour les avocats cherchant des documents similaires à une décisions données. Malgré 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. Thenmozhi et al. [2017] compare aussi la similarité cosinus sur trois représentations différentes des affaires dans le cadre de la campagne de recherche d'affaires antérieures pertinentes IRLed@FIRE2017 : (i) TF-IDF des concepts (noms), (ii) TF-IDF des concepts et relations (verbes), (iii) et la moyenne des *Word2Vec* [Le and Mikolov, 2014] des concepts et relations. Vue les performances qu'ils ont obtenues (précision@10 et rappel@10), la première représentation capte mieux la similarité par rapport à l'utilisation des verbes et de la représentation distribuée.

En bref, la similarité entre documents est utilisée pour répondre à plusieurs tâches, comme par exemple, la recherche des décisions similaires [Thenmozhi et al., 2017], le regroupement non-supervisé des jugements [Raghuveer, 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 métriques d’évaluation. Les diverses études menées sur la similarité démontrent l’importance de l’abstraction des textes par les concepts soit via l’alignement du document avec une ontologie, soit via la sélection de termes clés.

1.5 Conclusion

En résumé, les travaux portant sur l’analyse automatique des décisions ont donné des résultats encourageant grâce aux éléments spécifiques aux affaires et généralement contenus dans les documents correspondants. 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 où des librairies de développement, spécifiques au domaine, commencent à voir le jour [Bommarito et al., 2018]. La revue littéraire fait remarquer 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, soit 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 participation 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

2.1 Introduction

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

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

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

trant leurs occurrences dans les décisions avec lesquelles nous avons travaillé.

Entités	Label	Exemples	#mentions ^a	
			Médiane ^b	Total ^c
Numéro de registre général (R.G.)	rg	« 10/02324 », « 60/JAF/09 »	3	1318
Ville	ville	« NÎMES », « Agen », « Toulouse »	3	1304
Juridiction	juridiction	« COUR D'APPEL »	3	1308
Formation	formation	« 1re chambre », « Chambre économique »	2	1245
Date de prononcé	date	« 01 MARS 2012 », « 15/04/2014 »	3	1590
Appelant	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 BOUSQUEL »	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 annotées manuellement

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 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 qu'on remarque déjà au niveau de la transition entre sections. Au vu de leur rôle, il est évident que les sections devraient être séparées par des marqueurs bien précis. Une approche intuitive de sectionnement consisterait par conséquent à définir un algorithme capable de reconnaître ces marqueurs de transition à travers des expressions régulières. Cependant, les marqueurs utilisés ne sont pas standards. Les indicateurs de transitions sont souvent différents d'une décision

à l'autre et peuvent être des titres ou des motifs à base de symboles (astérisques, tirets, etc.). Il arrive parfois que la transition soit implicite et qu'on 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 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 parfois, ils sont entièrement en majuscule. Cependant, certains mots communs peuvent apparaître aussi en majuscule (par ex. APPELANTS, DÉBATS, ORDONNANCE DE CLÔTURE). Les entités peuvent contenir des chiffres (identifiant, dates, ...), des caractères spéciaux (« / », « - »), des initiales (par ex. « A. ») ou abréviations. Dans l'entête, les entités apparaissent généralement dans le même ordre (par ex. les appelants avant les intimés, les intimés avant les intervenants). Cependant, 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.

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 des segments atomiques (*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

Chau et al. [2002] distinguent quatre catégories d'approches d'extraction d'information :

- Les **systèmes à recherche lexicale** sont conçus sur la base d'une liste d'entités préalablement connues, et leurs synonymes dans le domaine d'intérêt. Par exemple, dans le domaine juridique, un lexique pourrait contenir les identifiants de règles juridiques et les noms des juges. La liste des entités peut être 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 pourrait changer régulièrement (nouvelles lois par ex.). De plus, les mentions d'entités peuvent avoir plusieurs variantes. Par exemple, la même règle juridique « Article 700 du code de procédure civile » peut être citée seule et en entier (« article 700 du code de procédure civile »), ou abrégée (« article 700 CPC »), ou encore avec d'autres règles (« articles 700 et 699 du code de procédure civile »). De plus, ces approches sont sujettes aux problèmes d'ambiguïté par exemple lorsque différentes entités comprennent les mêmes mots. Ces

problèmes ont limité ces premiers systèmes [Palmer and 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 plateformes 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"}.
```

Ils sont avantageux grâce à leur implémentation déclarative qui facilite la maintenance (erreurs facile à tracer et à expliquer) et l'expression directe des connaissances du domaine en règles [Waltl et al., 2018]. La définition manuelle de règles exige malheureusement des efforts considérables, en particulier pour les grands corpus. Cependant, ces efforts sont parfois rapidement récompensés par de bonnes performances d'extraction. 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 statistiques** adaptent les modèles statistiques de lan-

gage, issus typiquement des méthodes de compression de texte, pour détecter les entités. Par exemple, Witten et al. [1999] ont adapté le schéma de compression appelé « Prédiction par Correspondance Partielle ».

- Les **systèmes basés sur l'apprentissage automatique** exécutent des classifieurs multi-classes sur des segments de texte. Par exemple, un 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 labelliser, et le choix d'une représentation de segment (encore appelé schéma d'étiquetage).

2.2.1 Les modèles graphiques probabilistes HMM et CRF

Nous avons choisi d'analyser l'application des modèles CRF et HMM car les comparaisons avec d'autres approches démontrent bien que les modèles

probabilistes obtiennent les meilleurs résultats lors de l'extraction d'information dans les documents juridiques. Par exemple, 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 F1 de 89% et 97% pour le HMM utilisant les trigrammes comme descripteurs de mots, et des scores F1 de 87% et 97% pour le PAUM en utilisant des 5-grammes de lemmes et les rôles grammaticaux (*Part-Of-Speech tag*) comme descripteurs.

Considérons un texte T comme étant une séquence d'observations $t_{1:n}$, avec chaque t_i étant un segment de texte (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 $P(T, L) = \prod_i P(l_i | l_{i-1}) P(t_i | l_i)$ 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 ap-

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

proche, n étant la longueur de la séquence et m le nombre de labels candidats, l'algorithme de décodage Viterbi [Viterbi, 1967], basé sur une programmation dynamique, permet d'obtenir une estimation de L^* . Cette algorithme utilise des paramètres estimés par apprentissage sur un corpus de textes annotés manuellement :

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

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

L'avantage du HMM réside dans sa simplicité et sa vitesse d'entraînement. Cependant, il est difficile de représenter les segments à l'aide de multiples descripteurs distincts. Il est tout aussi difficile de modéliser la dépendance entre des observations distantes parce que l'hypothèse d'indépendance entre observations est très restrictive (i.e. l'état courant dépend uniquement des état précédents et de l'observation courante).

2.2.1.2 Les champs conditionnels aléatoires (CRF)

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

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

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

Les fonctions potentielles $f(\cdot)$ sont les caractéristiques utilisées par les modèles CRF. Deux types de fonctions caractéristiques sont définies : les caractéristiques de transition qui dépendent des labels aux positions courantes et précédentes (l_{i-1} et l_i resp.) et de T ; et les caractéristiques d'état qui sont des fonctions de l'état courant l_i et de la séquence T . Ces fonctions $f(\cdot)$ sont définies à l'aide 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

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

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

t_i étant une observation dans T , POS étant le rôle grammatical de t_i (NUM

= valeur numérique, $NOM = \text{nom}$), et **NP1** et **NS1** sont les lemmes des mots avant et après t_i , respectivement. Les symboles $\langle unknown \rangle$ 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 $l_i = NORME$ 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 and Nocedal [1989].

2.2.2 Représentation des segments atomiques

La représentation des segments à labelliser occupe une place importante pour l'obtention de bons résultats avec les modèles décrits précédemment. Elle consiste généralement à décrire la forme et le contexte de chaque segment en lui assignant des attributs [Nadeau and Sekine, 2007, Sharnagat, 2014]. Ils peuvent être booléens (« le mot est-il en majuscule ? »), numériques (nombre de caractères du mot), nominaux (par ex. le rôle grammatical d'un mot), ou définis par des expressions régulières (par ex. pour les numéros R.G. on peut avoir $dd/dddd$ où d désigne un chiffre). Ces descripteurs mettent en évidence des régularités relatives à l'occurrence des entités. Par exemple,

préciser qu'un mot débute par une lettre majuscule permet d'indiquer les noms propres. La définition de tels descripteurs consiste ainsi à fournir au modèle des indices l'aidant à mieux distinguer les différents types d'entités.

Etant donné que les descripteurs dépendent généralement de l'intuition du concepteur du système d'étiquetage, il est difficile mais nécessaire d'identifier des descripteurs appropriés. Après avoir défini des candidats, il n'est pas sûr qu'en les combinant tous ensemble, on obtienne les meilleures performances. Une sélection de caractéristiques peut s'avérer nécessaire. Cette sélection peut améliorer les performances d'étiquetage, et accélérer l'extraction des descripteurs, l'entraînement du modèle ainsi que son application à de nouveaux textes [Kitoogo and Baryamureeba, 2007]. Elle peut aussi fournir une meilleure compréhension du comportement des modèles entraînés [Klinger and Friedrich, 2009]. Deux principales approches se distinguent. D'une part, les méthodes « filtrantes » (*filters*), comme l'information mutuelle, comparent individuellement les descripteurs à l'aide de scores qui ne sont pas nécessairement basés sur la performance. D'autre part, les méthodes « enveloppantes » (*wrappers*) comparent des sous-ensembles de descripteurs sur la base des performances d'évaluation qu'elles permettent d'obtenir (par exemple la F1-mesure obtenue sur un ensemble d'exemples). Même si les méthodes filtrantes sont plus rapides, elles sont en général moins performantes car elles ne permettent pas d'éviter les redondances, et ne prennent pas en compte l'effet de la combinaison de caractéristiques.

La définition manuelle des caractéristiques suivie de la sélection est souvent qualifiée de méthode forcée car elle dépend fortement de la capacité du concepteur du système à identifier les descripteurs appropriés. Les réseaux de neurones permettent d'apprendre des caractéristiques grâce à des méthodes de plongement sémantique telles que Word2Vec [Le and Mikolov, 2014] et Glove [Pennington et al., 2014]. Deux architectures de réseaux de neurones réalisent actuellement les meilleures performances en matière de détection d'entités nommées. Il s'agit du modèle BiLSTM-CRF de Lample et al. [2016]

et du LSTM-CNN-CRF de Ma and Hovy [2016]. On pourrait résumer ces architectures en trois phases. Dans un premier temps, les segments de textes (mots) ont une représentation vectorielle concaténant 2 vecteurs de plongement sémantique : l'un issu de l'apprentissage morphologique du mot à partir de ses caractères, et l'autre issu de l'apprentissage du contexte général d'occurrence du mot. Lors de la seconde phase, deux couches de cellules LSTM enchaînées permettent de modéliser le contexte à droite et à gauche de chaque mot du texte labellisé. La dernière phase détermine la séquence de labels la plus probable pour le texte à l'aide d'une implémentation neuronale du modèle CRF. Le CRF reçoit en entrée la concaténation des contextes à droite et à gauche des mots. **schémas du biLSTM**

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 2.1 illustre l'utilisation 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>Président</i>	<i>de</i>	<i>chambre</i>	,	<i>de</i>
IO	O	O	I-JUGE	I-JUGE	I-JUGE	O	I-FONCTION	I-FONCTION	I-FONCTION	O	O
BIO	O	O	B-JUGE	I-JUGE	I-JUGE	O	B-FONCTION	I-FONCTION	I-FONCTION	O	O
IEO	O	O	I-JUGE	I-JUGE	E-JUGE	O	I-FONCTION	I-FONCTION	E-FONCTION	O	O
BIEO	O	O	B-JUGE	I-JUGE	E-JUGE	O	B-FONCTION	I-FONCTION	E-FONCTION	O	O

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

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

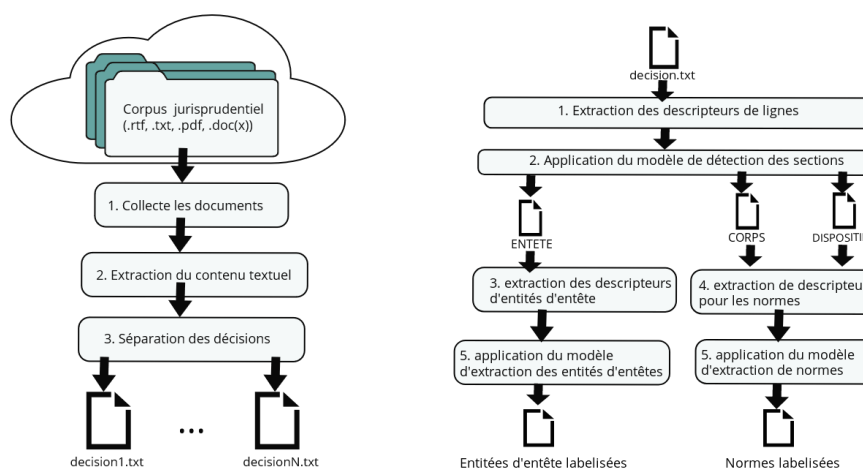
- B : début (*beginning*);
- I : intérieur (*inside*);
- E (ou L, ou M) : fin (*end* ou *last* ou *middle*);
- S (ou U, ou W) : singleton ou entité à segment unique (*single* ou *unit* ou *whole*);
- O : hors de toute entité (*outside*).

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 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

Nous proposons de travailler uniquement avec le contenu textuel des documents. Ce contenu est extrait des documents téléchargés en éliminant les



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

Figure 2.2 – Application des modèles entraînés pour l'étiquetage de sections et entités.

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 2.2). Ainsi, les documents sont collectés puis pré-traités suivant leur format d'origine (extraction du texte et séparation des décisions apparaissant dans le même document). Ensuite, après le sectionnement des décisions, les entités sont identifiées dans les différentes sections. Par ailleurs, comme segment atomique à étiqueter nous avons choisi les lignes pour la détection des sections, et les mots pour les entités.

Les modèles HMM et CRF étant tous les deux supervisés, ils doivent

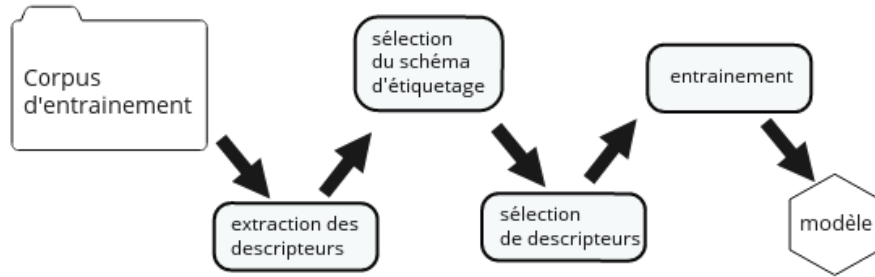


Figure 2.3 – Entraînement des modèles.

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

2.3.1 Définition de descripteurs candidats

2.3.1.1 Descripteurs pour la détection des sections

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

Plusieurs critères peuvent être utilisés pour différencier les sections, à savoir : la longueur des lignes (plus longues dans le corps, plus courtes dans l’en-tête), les premiers termes de certaines lignes (typiques de chaque section) et le nombre total de lignes. Un HMM n’adapte qu’un descripteur assimilé à l’élément à étiqueter. D’autres descripteurs peuvent être la position de

l'élément à étiqueter (numéro de ligne) ou le début de la ligne. Le descripteur capturant la longueur de ligne peut être 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 (**t0**, **t1**, **t2**), sa longueur absolue (**absLength**) et sa longueur relative (**relLength**) ;
- contexte de ligne : le numéro de ligne (**absNum**) et le numéro de la partie de document contenant la ligne (**relNum**), les deux premiers mots des lignes précédente (**p0**, **p1**) et suivantes (**n0**, **n1**), ainsi que leurs longueurs absolues et relatives (**pLength**, **pRelLength**, **nLength**, **nRelLength**).

2.3.1.2 Descripteurs pour la détection d'entités

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

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

2.3.2 Sélection des descripteurs

2.3.2.1 Sélection pour le modèle CRF

Nous avons étudié deux approches enveloppantes qui semblent toujours converger et qui ne nécessitent pas de définir manuellement la taille du sous-ensemble cible.

Algorithme 1 : Recherche bidirectionnelle BDS

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

Résultat : Sous-ensemble optimal de descripteurs

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

La première méthode, qui est la recherche bidirectionnelle (BDS) de Liu and Motoda [2012], combine la sélection séquentielle en avant (SFS) et la sélection séquentielle en arrière (SBS) en parallèle (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 F1-mesure (Eq. ??). 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} .

Algorithme 2 : Sélection séquentielle avant à flottement

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

Résultat : Sous-ensemble optimal de descripteurs

```

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

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

2.3.2.2 Sélection pour le modèle HMM

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

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

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

l’annotation manuelle. Des balises XML sont rajoutées autour des passages sélectionnés, en arrière plan dans le document.

Chaque document annoté comprend en moyenne 262,257 lignes et 3955,215 mots. Les deux dernières colonnes du Tableau 2.1 présentent la distribution des entités labellisées dans le jeu de données. En se basant sur un sous-ensemble de 13 documents labellisés par 2 annotateurs différents, nous avons calculé des taux d’accord inter-annotateur en utilisant la statistique Kappa de Cohen. Ces mesures d’accord inter-annotateur ont été calculées au niveau des caractères parce que certains mots peuvent être coupés par des annotations incorrectes (par ex. *<juridiction> cour d’appe </juridiction> l* contre *<juridiction> cour d’appel </juridiction>*), ou bien les annotateurs pourraient ne pas être d’accord si une apostrophe doit 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 F1-mesure comme mesures d’évaluation car elles sont généralement utilisées comme références en extraction d’information. La F1-mesure se calcule à l’aide de la formule suivante :

$$F1 = 2 \times \frac{Precision \times Rappel}{Precision + Rappel}.$$

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 calculés 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 présente des mots représentatifs trouvés dans les premiers thèmes inférés. L’extraction des autres descripteurs a été implémentée pour cette expérimentation.

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

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

Les valeurs de précision, rappel, et F1-mesure ont été calculées à l’aide du script d’évaluation de la campagne CoNLL-2002⁴. Elles sont indiquées en pourcentage dans les tableaux de résultats d’évaluation des sections suivantes.

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

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

2.4.2 Sélection du schéma d'étiquetage

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

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 F1 pour éviter d'exécuter plusieurs fois l'entraînement pour les mêmes sous-ensembles de descripteurs, le processus de sélection est resté toujours très long pour les deux algorithmes. Nous avons testé individuellement chacun des descripteurs candidats pour les modèles HMM. Les résultats sont reportés dans le Tableau 2.4.

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.

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

Tableau 2.3 – Comparaison des schémas d'étiquetage.

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

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

tion BDS, tandis que la méthode SFFS élimine beaucoup plus de candidats (par exemple en ne sélectionnant que 4 descripteurs parmi les 14 candidats définis pour l'annotation des normes).

Par ailleurs, les algorithmes de sélection forment des combinaisons inattendues. Par exemple, dans le cas de la détection de section, la ligne suivante semble être beaucoup plus indicatrice que la première. Il est aussi intéressant de noter que les descripteurs basés sur notre observation apparaissent dans les sous-ensembles sélectionnés (par ex. `isAfterIntervenant`, `isKEYWORD`). Remarquons aussi que la longueur absolue des lignes (`absLength`) joue un rôle important dans l'identification des sections vu qu'il a été sélectionné à la fois pour le CRF et le HMM (sélection BDS). Avec ces sous-ensembles sélec-

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

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

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

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

^{b2} **Selection 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} **Selection 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} **Selection 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} **Selection 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} **Selection par SFFS pour les normes (04 descripteurs)** : { POSW1, lemmaW-2, w-1, DIGIT-IN }

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

tionnés, 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 Evaluation détaillée pour chaque classe

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

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

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

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

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

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

(Figure 2.4), les *intervenants* sont parfois classifiés comme *appelant*, *intimé* ou *avocat* probablement parce qu’il s’agit d’entités mentionnées les unes à la suite des autres dans l’entête (les *intervenants* sont mentionnés juste après

les *avocats* des *intimés*). De plus, les intervenants apparaissent dans une très faible proportion de documents annotés. Par ailleurs, une quantité considérable d'*appelants* sont aussi classifiés comme *intimés*.

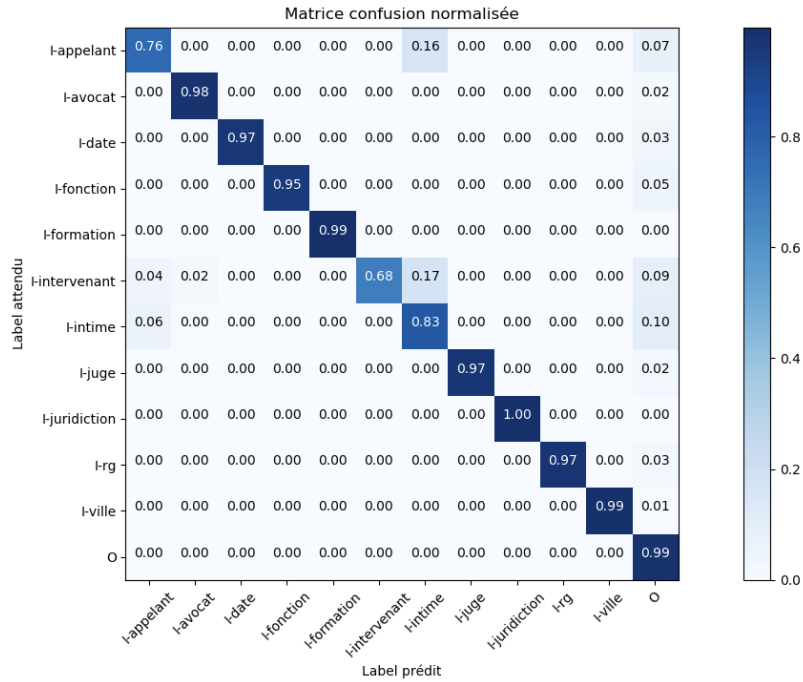


Figure 2.4 – Matrice de confusion entre méta-données d’entête avec le modèle CRF

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

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 mentions ré-

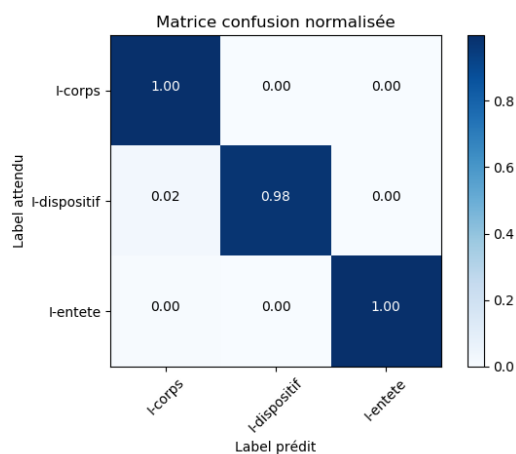
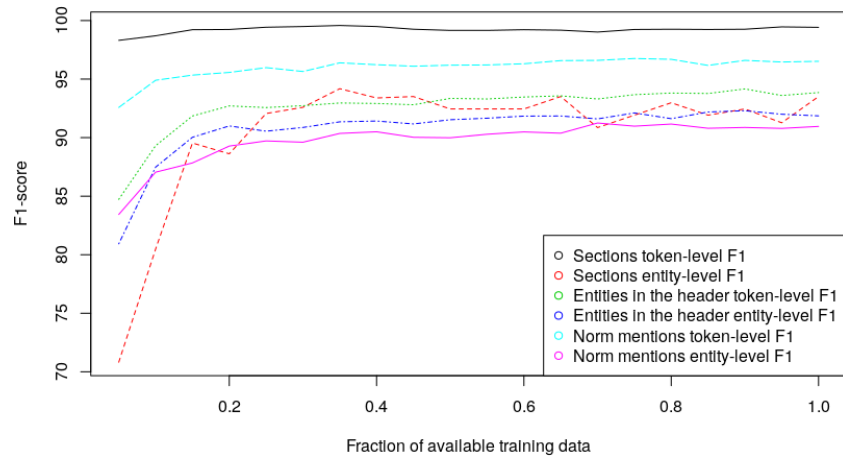


Figure 2.5 – Matrice de confusion entre lignes des sections avec le modèle CRF

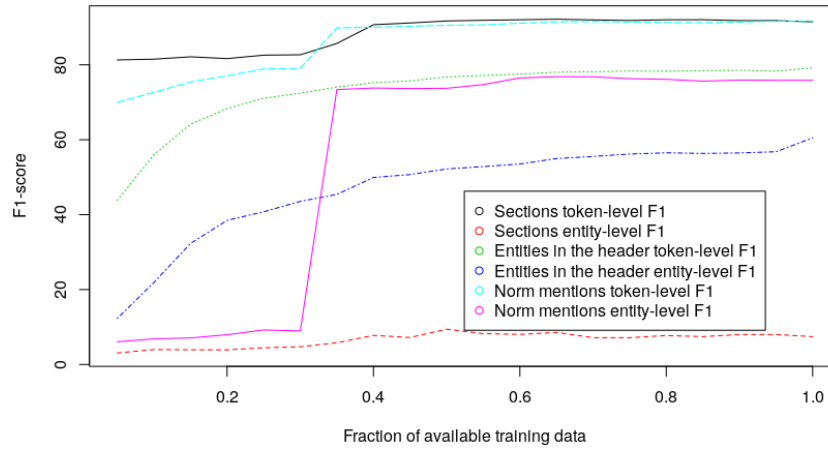
pétées ne soient pas identiques, de telles redondances aident à réduire le risque de manquer une entité. Cet aspect peut être exploité afin de combler l'imperfection des modèles.

2.4.5.3 Impact de la quantité d'exemples annotés

Des expérimentations ont été menées pour évaluer la manière dont les modèles s'améliorent lorsqu'on augmente le nombre de données d'entraînement. Pour cela, nous avons évalué différentes tailles de la base d'entraînement. Les données ont été divisées en 75% – 25% pour resp. l'entraînement et le test. 20 fractions de l'ensemble d'entraînement ont été utilisées (de 5% à 100%). A chaque session entraînement-test, le même jeu de test a été employé pour les différentes fractions de l'ensemble d'entraînement. Les courbes d'apprentissage des modèles CRF et HMM sont représentées resp. sur les Figures 2.6a et 2.6b. Il est évident que les scores F1 croissent avec le nombre de données d'entraînement pour les CRF et HMM, mais cette amélioration devient très faible au-delà de 60% de données d'entraînement quelle que soit la tâche. Il est possible que les exemples ajoutés à partir de là partagent la même structure que celle de ceux qui ont été ajoutés auparavant. Ainsi, cette étude doit être



(a) CRF



(b) HMM

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

étendue à la sélection des exemples les plus utiles. Raman and Ioerger [2003] ont démontré les avantages des algorithmes de sélection d'exemples combinés à celle des caractéristiques pour la classification. Les mêmes méthodes sont probablement applicables à l'étiquetage de séquences.

	CRF + descripteurs manuels			BiLSTM-CRF		
	<i>Precision</i>	<i>Rappel</i>	<i>F1</i>	<i>Precision</i>	<i>Rappel</i>	<i>F1</i>
appellant	82.49	69.42	74.72	80.26	71.53	75.04
avocat	90.15	89.02	89.56	84.93	87.88	86.36
date	95.34	91.46	93.12	95.04	90.79	92.63
fonction	95.87	95.08	95.44	92.69	93.48	93.03
formation	96.91	91.31	93.7	91.05	89.47	89.84
intervenant	51.42	32.71	36.8	31.48	20	23.11
intime	76.01	79.15	77.22	67.7	75.43	70.83
jug	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.

2.4.5.4 Descripteurs manuels vs. réseau de neurones

L'ingénierie manuelle des caractéristiques est difficile car arbitraire. Nous avons comparé les performances de nos descripteurs avec celles des réseaux de neurones qui apprennent une représentation des segments. Pour cela nous avons choisi le BiLSTM-CRF de Lample et al. [2016] qui fait partie des meilleures approches récentes. La comparaison a été effectuée pour la 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 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 and Mikolov [2014]. Les vecteurs obtenus ont une dimension de 300. Etant donné que les décisions sont des documents particulièrement longs, leur contenu a été découpé en des morceaux de texte dont la taille n'excède pas 300 mots. Les résultats obtenus par le BiLSTM-CRF sont assez proches de ceux que nous observons avec les descripteurs manuellement définis (Tableau 2.7). Etant donné que ces derniers permettent de mieux détecter certaines entités comme les *in-*

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

tervenants, 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 combinés avec diverses représentations de segment pourrait peut-être offrir de meilleurs résultats. En raison de la longue durée de recherche du sous-ensemble optimal de descripteurs, il serait préférable d'utiliser un algorithme de sélection beaucoup plus rapide que les méthodes BDS et SFFS que nous avons expérimentées. De plus, même si les résultats s'améliorent avec la croissance de l'échantillon d'apprentissage, la mesure globale F1 semble néanmoins atteindre une limite très rapidement. Étant donné que certaines entités ne sont pas très bien détectées, il peut être avantageux d'ajouter des exemples appropriés afin de traiter ces problèmes spécifiques.

L'application des modèles pose deux difficultés majeures : l'annotation d'un nombre suffisant d'exemples et la définition de caractéristiques discriminantes. Les efforts d'annotation peuvent être réduits avec un système automatique à faible performance d'étiquetage. Il suffirait alors de vérifier manuellement ces annotations afin de corriger les erreurs commises par le système sur de nouvelles décisions à l'aide d'un outil d'aide à l'annotation. En ce qui concerne la définition des caractéristiques, dans la mesure où notre approche actuelle est réalisée manuellement par l'analyse de quelques docu-

ments, 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.

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

Chapitre 3

Extraction des données concernant les demandes et leurs résultats correspondants

3.1 Introduction

Au cœur de l'analyse des décisions de justice se trouve le concept de demande. Il s'agit d'une réclamation ou requête effectuée par une ou plusieurs parties aux juges. Une partie peut demander des dommages-intérêts en réparation d'un préjudice subi ou à l'issue d'un divorce, des indemnités auxquelles elle pense avoir droit, ou encore une étude d'expert, etc. Les demandes sont fondamentales car l'argumentation au cours d'une affaire a deux buts : faire accepter ses demandes, et faire rejeter celle de la partie adverse. L'extraction des demandes et des résultats correspondants, dans un corpus, permet ainsi de récolter des données informant de la manière dont sont jugés des types de demandes d'intérêt. Les informations qui nous intéressent sont la catégorie de la demande, le quantum (montant) demandé, le sens du résultat (par ex. la demande a-t-elle été acceptée ou rejetée?), et le quantum obtenu (décidé par les juges). Pour pouvoir extraire les demandes et les résultats, il est nécessaire de comprendre comment ils sont exprimés et co-référencés dans les décisions jurisprudentielles. Leur énoncé peut comporter des expressions plus ou moins complexes, dont souvent des références à des jugements antérieurs, des agrégations ou des restrictions (Figure 3.1).

Jennifer M., Catherine M. et Sandra M. ... demandent à la Cour de :
 - les recevoir régulièrement appelantes incidentes du **jugement du 23/05/2014** ;
 - infirmer **le dit jugement** en **toutes ses dispositions** ; ...
 Statuant à nouveau ...
 - les condamner au paiement d'une somme de 3 000,00 € pour procédure abusive et aux entiers dépens ;

(a) Exemples d'énoncés de demandes

La cour, ...
 CONFIRME **le jugement entreprise** en **toutes ses dispositions**.
 Y ajoutant
 CONSTATE que Amélanie Gitane P. épouse M. est défaillante à rapporter la preuve d'une occupation trentenaire lui permettant d'invoquer la prescription acquisitive de la parcelle BH 377 située [...].
 DEBOUTE Amélanie Gitane P. épouse M. de sa demande en dommages et intérêts.
 CONDAMNE Amélanie Gitane P. épouse M. aux dépens d'appel.
 DIT n'y avoir lieu à l'application de l'article 700 du Code de Procédure Civile.

(b) Exemple d'énoncés de résultats

Figure 3.1 – Enoncés simples, ou comprenant des **références** et des **agréations** (extraits de la décision 14/01082 de la cour d'appel de Saint-Denis (Réunion))

3.1.1 Données cibles à extraire

3.1.1.1 Catégorie de demande

Une catégorie c de demande regroupe les prétentions qui sont de même nature par le fait qu'elles partagent deux aspects : l'objet demandé (par ex. dommages-intérêts, amende civile, déclaration de créance) et le fondement c'est-à-dire les règles ou normes ou principes juridiques qui fondent la demande (par ex. article 700 du code de procédure civile). Des noms particuliers sont utilisés pour identifier les catégories (Tableau 3.1).

3.1.1.2 Quantum demandé

Le quantum demandé quantifie l'objet de la demande. Nous le notons q_d . Par exemple, dans l'exemple de la Figure 3.1a, "3000 €" est le quantum demandé au titre des dommages-intérêts pour procédure abusive. Bien que cette étude ne porte que sur des sommes d'argent, le quantum peut être d'une autre nature comme par exemple une période dans le temps (garde d'enfant, ou emprisonnement, etc.). Toutes les catégories demandes n'ont

Label	Expression nominative	Objet	Fondement
acpa	amende civile pour abus de procédure	amende civile	Articles 32-1 code de procédure civile + 559 code de procédure civile
concdel	dommages-intérêts pour concurrence déloyale	dommages-intérêts	Article 1382 du code civil
danais	dommages-intérêts pour abus de procédure	dommages-intérêts	Articles 32-1 code de procédure civile + 1382 code de procédure civile
dcppc	déclaration de créance au passif de la procédure collective	déclaration de créance	L622-24 code de commerce
doris	dommages-intérêts pour trouble de voisinage	dommages-intérêts	principe de responsabilité pour trouble anormal de voisinage
styx	frais irrépétibles	dommages-intérêts	Article 700 du code de procédure civile

Les labels ont été définis particulièrement pour cette étude, et par conséquent, ils n'existent pas dans le langage juridique.

Tableau 3.1 – Exemples de catégories de demandes

pas de quantum (par ex. une demande de divorce) et seul le sens du résultat sera la donnée à extraire dans ce cas.

3.1.1.3 Sens du résultat

Le sens du résultat est l'interprétation de la décision des juges sur une demande. Nous le notons s_r . En général, le sens peut être positif si la demande a été acceptée, et négatif si elle a été rejetée. Il arrive aussi que le résultat soit reporté à un jugement futur ; il s'agit dans ce cas d'un sursis à statuer.

3.1.1.4 Quantum obtenu ou résultat

Le quantum obtenu quantifie le résultat ou la décision des juges. Nous le notons q_r . Il est en général inférieur ou égal au quantum demandé. Si la demande est rejetée, q_r est évidemment nul même si cela n'est pas explicitement mentionné dans le document. Il doit être de la même nature que le quantum demandé (somme d'argent ou durée).

3.1.2 Expression, défis et indicateurs d'extraction

Les demandes sont, en général, décrites à la fin de la section d'exposé des faits, procédures, moyens et prétentions des parties (section Litige). Elles rentrent donc dans les "moyens et prétentions des parties" qui regroupent les demandes et les arguments des parties. Quant aux résultats, ils sont décrits dans la section Dispositif et dans la section Motifs (raisonnement des juges). Les demandes sont exprimées en paragraphe où chaque paragraphe correspond soit à une partie, soit à un groupe de partie partageant les mêmes demandes (par ex. des époux). Le paragraphe est parfois organisé en liste dont chaque élément exprime une ou plusieurs demandes, ou fait référence à un jugement antérieur. Les résultats ont aussi la forme de liste dans la section Dispositif. Par contre, dans les motifs de la décision, les raisonnements sont organisés en paragraphes, et ordonnés catégorie après catégorie. Le résultat est donné à la fin du groupe de paragraphes associé à la catégorie.

Cette pseudo-structure n'est pas standard et elle impose de nombreux défis à relever. En effet, une décision jurisprudentielle porte sur plusieurs demandes de catégories différentes ou similaires. Il est important de faire correspondre un quantum demandé extrait au sens et quantum du résultat qui font référence à la même demande. La séparation des demandes et des résultats rend difficile cette mise en correspondance. Ce problème peut aussi être causé par la redondance des quanta ; par exemple, les résultats exprimés dans les Motifs sont résumés dans le Dispositif. D'autre part, les références aux jugements antérieurs exigent de résoudre des références aux résultats de jugements antérieurs qui sont, généralement, rappelés dans le même document. Notons aussi que les difficultés liées aux agrégations (par ex. "*infirmer ... en toutes ces dispositions*") et aux restrictions/sélections (par ex. "*infirme le jugement ... sauf en ce qu'il a condamné M. A. ...*") devraient être résolues. Par ailleurs, les catégories de demandes sont nombreuses¹ mais ne sont pas toutes présentes à la fois dans les décisions. Tous ces aspects rendent dif-

1. plus de 500 selon la nomenclature des affaires civiles NAC+

ficile l'annotation manuelle des données de référence et la modélisation d'une approche d'extraction adéquate. Cependant, nous avons remarqué quelques indicateurs qui pourraient être utiles.

On pourrait au préalable annoter les candidats potentiels de quanta. Nous nous sommes intéressés aux demandes dont les quanta sont des sommes d'argent. Les mentions de somme d'argent sont généralement de la forme « [valeur] [monnaie] » (par ex. 3000 €, 15 503 676 francs, un euro, 339.000 XPF). Des centimes apparaissent parfois (par ex. dix huit euros et soixante quatorze centimes, 26'977 € 19). Ainsi, il est possible d'annoter les sommes d'argent à l'aide d'une expression régulière. Même s'il est difficile de reconnaître des sommes d'argent écrites en lettre, il faut remarquer que l'équivalent en chiffre est généralement mentionné tout près (par ex. neuf mille cinq cent soixante six euros et quatre vingt sept centimes (9566,87 €)).

La terminologie utilisée est aussi un bon indicateur pour reconnaître des demandes et des résultats. En effet, le vocabulaire utilisé est très souvent propre aux catégories de demandes. Par exemple le dernier élément de la Figure 3.1a comprend le terme "*pour procédure abusive*" qui est près d'une somme d'argent (3000 €); il est donc probable que ce type de terme assez particulier soit un bon indicateur de la position des quanta. Par ailleurs, des verbes particuliers sont utilisés pour exprimer les demandes et résultats : infirmer, confirmer, constater, débouter, dire, ...

3.1.3 Formulation du problème

Nous avons tenu compte de deux principaux aspects du problème :

1. Une décision comprend plusieurs demandes de catégories similaires ou différentes ;
2. Il existe un grand nombre de catégories (500+); ce qui rend difficile l'annotation d'exemples de référence pour couvrir toutes ces catégories.

Nous avons par conséquent opter pour une extraction par catégorie. L'idée est de pouvoir ajouter progressivement de nouvelles catégories. Une exécution du système d'extraction permet ainsi d'extraire les demandes d'une seule catégorie. Le problème est décomposé en deux principales tâches :

Tâche 1 : Détecter les catégories présentes dans le document pour appliquer l'extraction uniquement à ces catégories ;

Tâche 2 : Pour chaque catégorie c identifiée, extraire les demandes :

1. identification des valeurs d'attributs : quanta demandés (q_d), quanta obtenus (q_r), et sens du résultat (s_r) ;
2. mise en correspondance des attributs pour former les triplets (q_d, s_r, q_r) correspondants aux paires demande-résultat d'une catégorie c .

3.2 Travaux connexes

Chacune des tâches précédentes se rapproche d'une tâche couramment traité en fouille de texte. En effet, la détection de catégories dans les décisions peut être modélisée comme un problème de classification de document. La tâche d'extraction se rapproche plus des problématiques comme l'extraction d'évènements, le remplissage de champs, ou encore l'extraction de relations et la résolution de référencement.

3.2.1 Problèmes analogues : extraction d'éléments structurés

Les demandes ressemblent aux structures telles que les relations ou les évènements. En effet, les champs définis par ACE [2008], pour les relations, et ACE [2005] pour les évènements, se rapprochent de ceux visés lors de l'extraction des demandes comme l'illustre le Tableau 3.2. Plus précisément, une catégorie de demandes correspond à un type d'évènement ou de relation entre deux entités. Les arguments qui participent à l'évènement « demande »

ou à la relation « demande-résultat » sont le quantum demandé et le quantum résultat. Le sens du résultat représente la classe de la structure « demande ».

	Relation [ACE, 2008]	Événement [ACE, 2005]	Analogie chez les demandes
Type	Org-Aff.Student-Alum	Die	Catégorie="Dommages-intérêts pour procédure abusive"
Passage (<i>extend</i>)	<i>Card graduated from the University of South Carolina</i>	"Il est mort hier d'une insuffisance rénale."	(Figure 3.1)
Déclencheur (<i>trigger</i>)	-	"mort"	"procédure abusive"
Participants ou Arguments (<i>arguments</i>)	Arg1="Card" Arg2="the University of South Carolina"	Victim-Arg="il" Time-Arg="hier"	Quantum-demandé="3000€" Quantum-obtenu="0 €"
Classes (<i>attributes, classes</i>)	Asserted	Polarity=POSITIVE, Tense=PAST	Sens-résultat="Rejeté"

Tableau 3.2 – Exemples d’analogie entre relations, événements et demandes

3.2.2 Approches d’extraction d’éléments structurés

L’extraction d’éléments structurés a généralement une formulation modulaire du problème en tâches plus simples. D’une part, on dispose de l’identification des déclencheurs² et des arguments. D’autre part, une mise en correspondance relie les arguments et déclencheurs qui participent à la même relation ou au même événement. Les classes peuvent être déterminées par classification du passage associé. Cette décomposition a permis à de nombreuses méthodes de voir le jour.

L’approche traditionnelle consiste en une chaîne de traitement enchaînant des modules adaptés chacun à une tâche simple. La sortie d’une étape est l’entrée de la suivante. C’est ainsi que Ahn [2006] définit un enchaînement de modèles de classification (k-plus-proches-voisins [Cover and Hart,

2. terme-clé indiquant la présence d’un événement [ACE, 2005].

1967] vs. classificateur d'entropie maximum [Nigam et al., 1999]), pour extraire des champs d'évènements dans le corpus d'ACE [ACE, 2005]. Bien que les différents modules soient plus faciles à résoudre, ce type d'architecture souffre de l'accumulation et la propagation d'erreurs d'une étape à la suivante, ainsi que de la non exploitation de l'interdépendance entre les tâches. Par conséquent, l'inférence jointe des champs est préconisée. Celle-ci peut être réalisée par une modélisation graphique probabiliste ou neuronale. Par exemple, pour l'extraction d'évènement, Yang and Mitchell [2016] estiment la probabilité conditionnelle jointe du type d'entité t_i , les rôles des arguments r_i et les types d'entités qui remplissent ces rôles a : $p_{\theta}(t_i, r_i, a | i, N_i, x)$, i étant un déclencheur candidat, N_i l'ensemble des entités candidates qui sont des potentiels arguments pour i , et x est le document. Par ailleurs, Nguyen et al. [2016] illustrent l'utilisation des réseaux de neurones profonds avec une couche pour la prédiction du déclencheur, une autre pour le rôle des arguments, et la dernière encode la dépendance entre les labels de déclencheurs et les rôles d'arguments. **[PERFORMANCE DE LEUR METHODE]**

L'annotation d'ACE [2005] est un marquage des champs dans le texte, et par conséquent, la position ou l'occurrence des champs est indiquée (« annotation au niveau du segment de mot »). Comme dans notre cas, les données peuvent être annotées dans un tableau, hors des textes d'où elles sont issues, il est donc nécessaire de retrouver leur position sans supervision. Palm et al. [2017] proposent dans cette logique une architecture de réseaux de neurones point-à-point qu'ils ont expérimentés sur des corpus de requêtes de recherche de restaurant et films [Liu et al., 2013] ou de réservation de billets d'avion [Price, 1990]. Ils se sont intéressés au problème de remplissage de champs en apprenant la correspondance entre les textes et les valeurs de sorties. Leur modèle est basé sur les réseaux de pointeurs [Vinyals et al., 2015] qui sont des modèles séquence-à-séquence avec attention, dans lesquels la sortie est une position de la séquence d'entrée. Le modèle proposé consiste en un encodeur de la phrase et des contextes, plusieurs décodeurs (un pour chaque

champ). L'application de cette architecture à l'extraction des demandes serait confrontée à deux obstacles majeurs auxquels il faut répondre au préalable. Premièrement, les décisions judiciaires ont des contenus de plusieurs centaines à plusieurs milliers de lignes contrairement aux requêtes manipulées par Palm et al. [2017] dont la plus longue ne comprend que quelques dizaines de mots. La complexité des architectures neuronales de TALN augmente rapidement en espace et par conséquent en temps, avec la longueur des documents manipulés en entier. Deuxièmement, nous disposons de très peu de données annotées ; entre 23 et 198 documents annotés dans notre cas contre plusieurs milliers pour les expérimentations de Palm et al. [2017].

L'avantage de l'utilisation des réseaux de neurones vient de leur capacité à apprendre automatiquement des caractéristiques pertinentes contrairement aux modèles probabilistes qui exigent très souvent une ingénierie manuelle des caractéristiques. Par contre, il est beaucoup plus facile d'utiliser les modèles probabilistes sur des corpus de faible taille et de longs textes comme c'est le cas pour notre problème.

3.2.3 Extraction de la terminologie d'un domaine

L'identification des attributs peut être facilitée grâce à leur proximité avec des termes-clés caractéristiques des catégories de demandes au même titre que les « déclencheurs » aident à identifier les événements. Ne disposant pas au préalable de la liste des termes pertinents pour l'extraction des demandes, il est possible de les apprendre. Il existe à cet effet plusieurs métriques statistiques de pondération de termes généralement employées en recherche d'information et en classification de texte comme méthodes de sélection de caractéristiques. Ces métriques sont qualifiées de poids globaux car calculés à partir des occurrences dans un corpus, à la différence des poids locaux (Tableau 3.4) calculés à partir des occurrences dans un document. Quelques métriques sont formulées ici en utilisant les notations du Tableau 3.3 définies pour une base d'apprentissage.

Notation	Description
t	un terme
d	un document
$ t $	longueur de t (nombre de mots)
c	la catégorie (domaine ciblé)
\bar{c}	la classe complémentaire ou négative
D	ensemble global des documents de taille $ D $
D_c	ensemble des documents de c de taille $ D_c $
$D_{\bar{c}}$	ensemble des documents de \bar{c} de taille $ D_{\bar{c}} $
N_t	nombre de documents contenant t
$N_{\bar{t}}$	nombre de documents ne contenant pas t
$N_{t,c}$	nombre de documents de c contenant $t = a$
$N_{\bar{t},c}$	nombre de documents de c ne contenant pas t
$N_{t,\bar{c}}$	nombre de documents de \bar{c} contenant t
$N_{\bar{t},\bar{c}}$	nombre de documents de \bar{c} ne contenant pas t
$DF_{t c}$	proportion de documents contenant t dans le corpus de c ($DF_{t c} = \frac{N_{t,c}}{ D_c }$)
$DF_{c t}$	proportion de documents appartenant à c dans l'ensemble de ceux qui contiennent t

Tableau 3.3 – Notation utilisée pour formuler les métriques

3.2.3.1 Métriques non-supervisées

Les métriques non-supervisées affectent un score à un terme en rapport avec l'importance de ce dernier dans le corpus global D . Parmi ces métriques, on retrouve par exemple la fréquence inverse de document (*inverse document frequency*) *idf* [Sparck Jones, 1972] et ses variantes *pidf* [Wu and Salton, 1981] et *bidf* [Jones et al., 2000] accordent plus d'importance aux termes rares. Elles considèrent en fait qu'un terme rare est plus efficace pour la distinction entre des documents. Par conséquent, elles sont efficaces en recherche d'information mais moins indiquées en classification de texte où le but est plutôt de séparer

des catégories [Wu et al., 2017]. Elles se formulent comme suit :

$$idf(t) = \log_2 \left(\frac{N}{N_t} \right), pidf(t) = \log_2 \left(\frac{N}{N_t} - 1 \right), bidf(t) = \log_2 \left(\frac{N_t + 0.5}{N_t + 0.5} \right)$$

Il est possible de prendre explicitement en compte le fait que les termes peuvent comprendre plusieurs mots (n-grammes) et avoir des tailles différentes (nombre de mots). La C-value [Frantzi et al., 2000], par exemple, distingue la fréquence du terme et de ses sous-termes (termes imbriqués) par la formule :

$$C\text{-value}(t) = \begin{cases} \log_2(|t|) \cdot (N_t - \frac{1}{|T_t|} \cdot \sum_{b \in T_t} N_b), & \text{si } t \text{ est imbriqué} \\ \log_2(|t|) \cdot N_t, & \text{sinon,} \end{cases}$$

T_t étant l'ensemble des termes candidats qui contiennent t .

3.2.3.2 Métriques supervisées

Les métriques supervisées mesurent l'information contenu dans les labels des documents de la base d'apprentissage. Pour un terme t , elles expriment généralement la différence de proportion qui existe entre les occurrences de t dans D_c et ses occurrences dans $D_{\bar{c}}$. Elles sont ainsi mieux adaptées à la distinction entre catégories. Parmi les nombreuses métriques existantes, nous avons expérimenté les suivantes :

La différence de fréquence Δ_{DF} consiste simplement à calculer la différence entre les proportions de documents contenant t respectivement dans c et \bar{c} :

$$\Delta_{DF}(t, c) = DF_{t|c} - DF_{t|\bar{c}}$$

Le gain d'information ig [Yang and Pedersen, 1997] estime la quantité d'information apportée par la présence ou l'absence d'un terme t sur

l'appartenance d'un document à une classe c :

$$ig(t, c) = \frac{N_{t,c}}{N} * \log_2 \left(\frac{N_{t,c}N}{N_t} \right) + \frac{N_{\bar{t},c}}{N} * \log_2 \left(\frac{N_{\bar{t},c}N}{N_{\bar{t}}|D_c|} \right) \\ + \frac{N_{t,\bar{c}}}{N} * \log_2 \left(\frac{N_{t,\bar{c}}N}{N_t|D_{\bar{c}}|} \right) + \frac{N_{\bar{t},\bar{c}}}{N} * \log_2 \left(\frac{N_{\bar{t},\bar{c}}N}{N_{\bar{t}}|D_{\bar{c}}|} \right)$$

La fréquence de pertinence rf [Lan et al., 2009] a comme intuition de considérer que plus la fréquence d'un terme t est élevée dans D_c relativement à sa fréquence dans $D_{\bar{c}}$, plus il contribue à distinguer les documents de c de ceux de \bar{c} . Elle est calculée par la formule :

$$rf(t, c) = \log \left(2 + \frac{N_{t,c}}{\max(1, N_{t,\bar{c}})} \right)$$

Le coefficient du χ^2 [Schütze et al., 1995] estime le manque d'indépendance entre t et c . Par conséquent, une grande valeur de $\chi^2(t, c)$ indique une relation étroite entre t et c . Elle est calculée par la formule :

$$\chi^2(t, c) = \frac{N((N_{t,c}N_{\bar{t},\bar{c}}) - (N_{t,\bar{c}}N_{\bar{t},c}))^2}{N_t N_{\bar{t}} |D_c| |D_{\bar{c}}|}$$

Le coefficient de corrélation ngl de Ng, Goh et Low [Ng et al., 1997] est la racine carré positive du χ^2 [Schütze et al., 1995] :

$$ngl(t, c) = \frac{\sqrt{N}((N_{t,c}N_{\bar{t},\bar{c}}) - (N_{t,\bar{c}}N_{\bar{t},c}))}{\sqrt{N_t N_{\bar{t}} |D_c| |D_{\bar{c}}|}}.$$

L'intuition est de ne regarder que les termes qui proviennent de D_c et qui indiquent l'appartenance à c .

Le coefficient gss de Galavotti, Sebastiani, et Simi [Galavotti et al., 2000] est une fonction simplifiée du ngl [Ng et al., 1997] :

$$gss(t, c) = (N_{t,c}N_{\bar{t},\bar{c}}) - (N_{t,\bar{c}}N_{\bar{t},c}).$$

Le facteur N a été éliminé car il est le même pour tous les termes. Le facteur $\sqrt{N_t N_{\bar{t}}}$ est supprimé car il accentue les termes extrêmement rares qui ne sont pas efficaces pour la classification de textes. Le facteur $\sqrt{|D_c| |D_{\bar{c}}|}$ est éliminé car il accentue les catégories extrêmement rares, ce qui tend à réduire l'efficacité micro-moyennée (efficacité calculée globalement sur le corpus de test sans distinction à priori du label des éléments).

Le test de Marascuilo (mar) qui se calcule par la formule :

$$mar(t, c) = \frac{\begin{pmatrix} (N_{t,c} - N_t N_{t,c}/N)^2 \\ + (N_{t,\bar{c}} - N_t |D_{\bar{c}}|/N)^2 \\ + (N_{\bar{t},c} - |D_c| N_{\bar{t}}/N)^2 \\ + (N_{\bar{t}} - N_{\bar{t}} |D_{\bar{c}}|/N)^2 \end{pmatrix}}{N}$$

Le test de Marascuilo est un test de proportion multivariée. Nous proposons de l'utiliser pour tester la présence d'un terme t dans différents corpus. Autrement dit, il s'agit de tester l'homogénéité des textes du corpus contenant c . Lorsque $mar(t, c) \geq 3.84$ on accepte l'hypothèse selon laquelle la proportion de textes pour lesquels t prédit c est significative pour un risque d'erreur de 5%.

Le « delta lissé d'idf », $dsidf$ [Paltoglou and Thelwall, 2010], est une version lissée du delta idf ($didf$) de Martineau et al. [2009] ($didf(t, c) = \log_2 \left(\frac{|D_{\bar{c}}| N_{t,c}}{|D_c| N_{t,\bar{c}}} \right)$). $dsidf$ se formule comme suit :

$$dsidf(t, c) = \log_2 \left(\frac{|D_{\bar{c}}| (N_{t,c} + 0.5)}{|D_c| (N_{t,\bar{c}} + 0.5)} \right)$$

Le delta BM25 d'idf, $dbidf$ [Paltoglou and Thelwall, 2010], est une

autre variante plus sophistiquée du *didf* qui se calcule comme suit :

$$dbidf(t, c) = \log_2 \left(\frac{(|D_{\bar{c}}| - N_{t,\bar{c}} + 0.5)(N_{t,c} + 0.5)}{(|D_c| - N_{t,c} + 0.5)(N_{t,\bar{c}} + 0.5)} \right)$$

3.2.3.3 Discussions

A l'exception de la C-value, ces métriques ne tiennent pas explicitement compte de la taille des termes dans les situations où on souhaiterait manipuler des termes de tailles différentes. [Brown, 2013] propose que soit affecté à un n-gramme t le poids $\left(\frac{N_t}{N}\right)^{0.27} * |t|^{0.09}$, une formule obtenue empiriquement pour l'identification du langage d'un document. Par ailleurs, la méthode C-value [Frantzi et al., 2000] propose un produit similaire avec le logarithme de la longueur à la place des puissances. Il est par conséquent évident que le produit lissé de la longueur du terme (puissance ou logarithme) avec les métriques décrites précédemment, permet de booster les longs termes qui, bien que rares, sont très souvent plus pertinents que certains termes plus courts. Aussi, le temps pour calculer ces différentes métriques devient rapidement long, surtout pour des n-grammes de mots de n variés. Pour compter rapidement les occurrences des n-grammes des corpus, nous avons utilisé la librairie SML³ Harispe et al. [2013] lors des expérimentations.

3.3 Méthode

3.3.1 Détection des catégories par classification des documents

Étant donné l'ensemble $D_{\bar{c}}$ des documents ne comprenant aucune demande de la catégorie d'intérêt c , nous proposons de modéliser la tâche de détection des catégories en une tâche de classification de documents. Pour

3. <http://www.semantic-measures-library.org/sml/index.php?q=lib>

chaque catégorie c , un modèle de classification binaire est entraîné pour déterminer si un document d contient une demande de la catégorie c . Nous avons particulièrement expérimenté quatre algorithmes traditionnellement utilisés comme approches de base. Il s'agit du Bayésien Naïf, de l'arbre de décision, des k -plus-proches-voisins [Cover and Hart, 1967], de la machine à vecteurs de support (SVM). Les labels utilisés correspondent aux catégories d'intérêt. Par exemple, un document sera labellisé *danais* s'il contient des demandes de dommages-intérêts pour abus de procédure, et *nodanais* sinon. Chaque document d est représenté sous une forme vectorielle du type TF-IDF (*term frequency - inverse document frequency*) proposé par Salton and Buckley [1988] dont chaque dimension k est identifiée par un terme t_k . Le poids $w(t_k, d)$ affecté à ce dernier est le produit normalisé d'un poids global $g(t_k)$ au corpus du mot et d'un poids local $l(t_k, d)$ de t_k dans le document d : $w(t_k, d) = l(t, d) \times g(t) \times nf(d)$, où nf est un facteur de normalisation tel que la norme cosinus $cos(d) = \sqrt{\sum_k (w(t_k, d))^2}$ qui est généralement utilisée.

Description	Formule
Décompte brute du terme [Salton and Buckley, 1988]	$tf(t, d) = \text{nombre d'occurrences de } t \text{ dans } d$
Présence du terme [Salton and Buckley, 1988]	$tp(t, d) = \begin{cases} 1 & , \text{ si } tf(t, d) > 0 \\ 0 & , \text{ sinon} \end{cases}$
Normalisation logarithmique	$logtf(t, d) = 1 + \log(tf(t, d))$
Fréquence augmentée et normalisée du terme [Salton and Buckley, 1988]	$atf(t, d) = k + (1 - k) \frac{tf(t, d)}{\max_{t \in T} tf(t, d)}$
Normalisation basée sur la fréquence moyenne du terme [Manning et al., 2008] (avg représente la moyenne)	$logave(t, d) = \frac{1 + \log tf(t, d)}{1 + \log \text{avg}_{t \in T} tf(t, d)}$

Tableau 3.4 – Métriques locales

Etant donné le grand nombre de métriques de pondération existantes, la métrique choisie est celle qui fournit la meilleure performance sur les données d'apprentissage.

3.3.2 Extraction basée sur la proximité entre sommes d'argent et les termes-clés

Diverses approches d'extractions d'informations existent (section § 3.2.2). Il paraît important de proposer dans un premier temps une approche basique explorant la solvabilité du problème du fait de ses multiples spécificités dont l'annotation d'une seule catégorie dans un document qui en contient plusieurs, l'annotation dans un tableau et donc à l'extérieur du document, la très faible quantité des données annotées, la multiplicité des demandes et des catégories dans un même document. Par conséquent, nous proposons ici une chaîne d'extraction à base de termes-clés, applicable pour chaque catégorie de demande. Il s'agit d'une approche qui tente de reproduire une lecture naïve du document en se basant sur des expressions couramment employées pour énoncer les demandes et résultats. La méthode consiste en deux phases dont une phase d'apprentissage des termes-clés de la catégorie, à proximité desquels seront identifiés les attributs durant la phase d'application comme l'illustre la Figure 3.2. On remarque en effet que, naïvement, le seul fait que 1500 euros soit aussi proche des termes-clés *amende civile* et *pour procédure abusive* signifie bien qu'il s'agit du quantum demandé comme amende civile pour procédure abusive.

```
" ...
- débouter M. S. de ...
- le condamner à payer une amende civile de 1.500 euros pour procédure abusive ...
- le condamner à payer la somme ..."
```

(a) Extrait original d'un énoncé de demande avant marquage

```
" ...
- débouter M. S. de ...
- le <demande categorie="acpa">condamner à payer une <terme-clef catego-
rie="acpa">amende civile</terme-clef> de <argent> 1.500 euros </argent> <terme-clef
categorie="acpa"> pour procédure abusive</terme-clef> ...
- le</demande> condamner à payer la somme ..."
```

(b) Énoncé, sommes d'argent, et termes-clés marqués

Figure 3.2 – Illustration de la proximité des quantas et termes-clés

3.3.2.1 Pré-traitement

Le pré-traitement est nécessaire pour :

1. sectionner le document comme décrit au chapitre 2 en sections Entête, Litige, Motifs, Dispositif ;
2. annoter les sommes d'argent (en chiffre) à l'aide de l'expression régulière « `[0-9] ([0-9] | [' , .] | \s)*\s* ([Ee]uro[s]{0,1} | franc[s]{0,1} | €|F|XPF|CFP|EUR|EUROS| [i]) (|$) » ;`
3. annoter les énoncés de demandes et de résultats respectivement dans les sections Litige et Dispositif. Pour cela, les mots introductifs du tableau 3.5 sont employés car ils indiquent le début d'un énoncé indépendamment de la catégorie.

Demande	Résultat (organisé par polarité ou sens)		
	accepte	sursis à statuer	rejette
<i>accorder, admettre, admission, allouer, condamnation, condamner, fixer, laisser, prononcer, ramener, surseoir</i>	<i>accorde, accordons, admet, admettons, alloue, allouons, condamne, condamnons, déclare, déclarons, fixe, fixons, laisse, laissons, prononce, prononçons</i>	<i>réserve, réservons, sursoit, sursoyons</i>	<i>déboute, déboutons, rejette, rejetons</i>

Tableau 3.5 – Mots introduisant les énoncés de demandes et de résultats

La recherche de passages à l'aide listes de termes est une technique souvent utilisée dans les décisions de justice, à l'exemple de Wyner [2010] qui utilise des termes similaires à ceux du Tableau 3.5 pour annoter les énoncés de résultats (toute phrase contenant un terme de jugement) : *affirm, grant, deny, reverse, overturn, remand, ...*

3.3.2.2 Apprentissage des termes-clés d'une catégorie

Les termes-clés sont identifiés à l'aide de méthodes statistiques d'extraction ou sélection de terminologie. La base d'apprentissage comprend les corpus D_c et $D_{\bar{c}}$ dont les documents ont été pré-traités. Le processus d'apprentissage des termes se déroule comme suit :

1. restreindre le contenu de chaque document de D_c à la concaténation des énoncés de demande et résultats contenant des sommes d'argent de valeur égale à celle des quanta annotés.
2. restreindre le contenu de chaque document de $D_{\bar{c}}$ à la concaténation des énoncés de demande et résultats contenant des sommes d'argent.
3. à l'aide d'une métrique global g , calculer le score des termes du corpus $D_c \cup D_{\bar{c}}$. Ce score est le produit g' de g avec le logarithme de la longueur du terme, pour booster les termes longs : $g'(t, c) = \log_2(|t|) \times g(t, c)$.
4. normaliser les scores en appliquant à chaque score original ($g'(t, c)$) la formule $g'_{norm}(t, c) = \frac{\max_{t_k}(g'(t_k, c)) - g'(t, c)}{\max_{t_k}(g'(t_k, c)) - \min_{t_k}(g'(t_k, c))}$.
5. trier par ordre décroissant des termes ;
6. sélectionner les premiers termes qui obtiennent les performances optimales sur la base d'apprentissage .

3.3.3 Application de l'extraction à de nouveaux documents

A l'aide des termes-clés appris, l'extraction des données de couples demandes-résultats se déroule comme suit :

1. reconnaître et marquer les occurrences des termes dans le document ;
2. extraire les quanta demandés (q_d) et résultats (q_r) à proximité des termes-clés respectivement dans les énoncés de demande et résultat qui contiennent des sommes d'argent et un terme-clé ;
3. le mot introductif de l'énoncé résultat indique le sens du résultat (s_r) tel que catégorisé dans le Tableau 3.5 ;
4. relier les attributs (q_d, s_r, q_r) correspondant à une même pair demande-résultat :

- (a) former les paires (énoncé de demande, énoncé de résultat) similaire (nous utilisons la métrique de « la plus longue sous-séquence commune » [Bakkelund, 2009])
- (b) pour chaque paire d'énoncés formée, relier les quanta demandés et quanta résultats par ordre d'occurrence similaire.

3.4 Résultats expérimentaux

Nous analysons ici la capacité de l'approche proposée à reconnaître efficacement les catégories de demandes présentes dans les documents, et à extraire les valeurs des attributs des différentes paires demandes-résultats qui y sont exprimées. Il y est discuté les données et métriques d'évaluation employées, ainsi que des résultats expérimentaux observés avec des exemples annotés pour les six catégories du Tableau 3.1.

3.4.1 Données d'évaluation

L'annotation manuelle d'exemples s'effectue pour une catégorie à la fois afin que la tâche soit plus facile pour les experts. Le protocole d'annotation se déroule en étapes :

1. définir une catégorie c par son objet et sa norme juridique ;
2. former un corpus D_c de documents contenant des demandes de c , et un autre $D_{\bar{c}}$ de documents n'en contenant pas ;
3. extraire toutes les demandes de catégories c mentionnées dans D_c , pour annoter les données des paires demande-résultat dans un tableau comme celui illustré par le Tableau 3.6 ;

Toutes les demandes du corpus $D_c \cup D_{\bar{c}}$ annoté manuellement, sont considérées inscrites dans le tableau des annotations manuelles. La répartition des documents d'évaluation est donnée par l'histogramme de la Figure 3.3.

	A	B	C	D	F	H	L	N
1	IDENTIFICATION DE LA DECISION			DESCRIPTION DE LA PRETENTION			DESCRIPTION DU RESULTAT	
2	Type	Ressort	RG	OBJET	NORME	QUANTUM	RESULTAT	QUANTUM RESULTAT (obtenu)
	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
441	CA	Lyon	14/06911	dommages-intérêts	700 Code de Procédure Civile	3,500.00 €	rejette	0.00 €
442	CA	Lyon	14/06911	dommages-intérêts	700 Code de Procédure Civile	2,000.00 €	accepte	1,500.00 €

Les noms des champs sont sur les 2 premières lignes et les demandes sont données en exemple pour la catégorie *dommages-intérêts sur le fondement de l'article 700 du code de procédure civile* (décision 14/06911 de la cour d'appel de Lyon).

Tableau 3.6 – Extrait du tableau d'annotations manuelles des demandes.

Répartition des demandes dans les documents annotées pour chaque catégorie

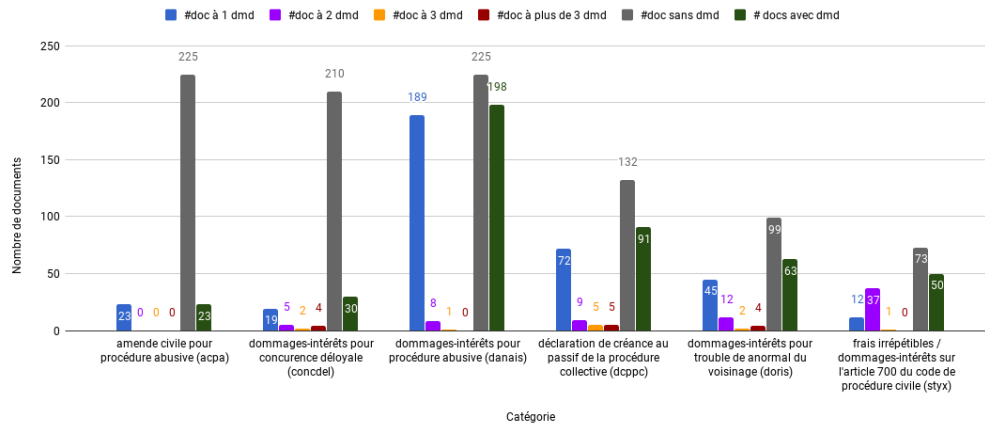


Figure 3.3 – Répartitions des demandes dans les documents annotées.

Il faut aussi noter que bien que l'annotation manuelle des demandes et résultats soit réalisée dans un tableau (annotation externe au contenu), elle reste une tâche très difficile. Le très faible nombre de documents annotés manuellement en témoigne. Le nombre maximum de documents annotés pour une catégorie est seulement de 198 (barres vertes de *danais*).

3.4.2 Métriques d'évaluation

Reconnaissance de catégories par classification La classification des documents est évaluée en utilisant les métriques précision (P), rappel (P),

f1-mesure (F1) calculées à l'aide des nombres de vrais positifs (TP), faux positifs (FP), faux négatifs (FN) comme suit :

$$P = \frac{TP}{TP + FP}; R = \frac{TP}{TP + FN}; F1 = 2 \times \frac{P \times R}{P + R}$$

Extraction des attributs des paires demande-résultat Nous évaluons les approches proposées sur l'extraction de 3 données : le quantum demandé q_d , le sens du résultat s_r et le quantum obtenu q_r . Une demande est donc un triplet (q_d, s_r, q_r) . Il est possible d'évaluer le système pour un sous-ensemble x de ce triplet sur les demandes extraites d'un corpus annotées D de test. Nous utilisons les métriques traditionnellement employées en extraction d'information : la précision (Eq. 3.1), le rappel (Eq. 3.2), et la F1-mesure (Eq. 3.3).

$$Precision_{c,x,D} = \frac{TP_{c,x,D}}{TP_{c,x,D} + FP_{c,x,D}} \quad (3.1)$$

$$Rappel_{c,x,D} = \frac{TP_{c,x,D}}{TP_{c,x,D} + FN_{c,x,D}} \quad (3.2)$$

$$F1_{c,x,D} = 2 \times \frac{Precision_{c,x,D} \times Rappel_{c,x,D}}{Precision_{c,x,D} + Rappel_{c,x,D}} \quad (3.3)$$

Ces mesures sont définies à partir des nombres de vrais positifs (TP), faux positifs (FP) et faux négatifs (FN). Au niveau d'un document d :

- le nombre de vrais positifs $TP_{c,x,d}$ est le nombre de demandes extraites de d par le système, qui sont effectivement de la catégorie c ;
- le nombre de faux positifs $FP_{c,x,d}$ est le nombre de demandes extraites de d par le système, mais qui ne sont pas des demandes de c (demandes en trop) ;
- le nombre de faux négatifs $FN_{c,x,d}$ est le nombre de demandes annotées comme étant de c mais qui n'ont pas pu être extraites par le système (demandes manquées).

Au niveau d'un corpus d'évaluation D , ces métriques sont sommées :

$$TP_{c,x,D} = \sum_{d \in D} TP_{c,x,d}; FP_{c,x,D} = \sum_{d \in D} FP_{c,x,d}; FN_{c,x,D} = \sum_{d \in D} FN_{c,x,d}$$

Une donnée observée (par exemple « 3 000 € ») est bien extraite automatiquement si sa valeur (le nombre 3000) correspond à celle du quantum annoté dans le tableau. Nous considérons que les unités monétaires, entre les quanta extraits et ceux manuellement annotés, sont égales.

3.4.3 Détection des catégories par classification

Les implémentations de la bibliothèque Weka [Frank et al., 2016] ont permis d'utiliser plusieurs modèles de classification : le modèle Bayésien naïf (NB), l'arbre de décision (J48), les k-plus-proches-voisins (KNN), et le SVM. A chaque entraînement, s'exécute une sélection de modèle par validation croisée sur les données d'entraînement. Elle a pour but de sélectionner la métrique locale et la métrique globale appropriée. Les résultats obtenus par 5-folds validation croisée sont présentés sur le tableau 3.7.

	NB			J48			KNN			SVM		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
acpa	1.0	1.0	1.0	0.996	0.955	0.972	1.0	1.0	1.0	0.996	0.955	0.972
concdel	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.995	0.967	0.979
danais	0.988	0.989	0.988	0.996	0.995	0.995	0.995	0.995	0.995	0.993	0.993	0.993
dcppc	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
doris	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
styx	1.0	1.0	1.0	0.984	0.983	0.983	1.0	1.0	1.0	1.0	1.0	1.0

(P= Précision, R=Rappel, F1 = F1-mesure)

Tableau 3.7 – Résultats d'une 5-fold validation croisée pour la détection de catégories

D'après les résultats, la tâche 1 est relativement aisée pour les algorithmes traditionnels qui détectent parfaitement la présence ou non d'une catégorie dans les documents. Par conséquent, pour toute catégorie c , les résultats de

l'extraction, dans la suite, ne sont discutés que pour les documents de c , car, grâce à l'efficacité de la phase de classification, aucun document de \bar{c} ne sera traité par la phase d'extraction.

3.4.4 Extraction de données des paires demandes-résultats

Les scores des termes-clés candidats étant normalisés, si on sélectionne les termes dont les scores sont supérieurs à un seuil fixé, on remarque que chaque métrique d'extraction a un niveau d'efficacité différent entre les catégories de demande (Tableau 3.8 avec 0.5 comme seuil fixé).

	<i>acpa</i>	<i>concdel</i>	<i>danaïs</i>	<i>dcppc</i>	<i>doris</i>	<i>styx</i>	Moyenne
<i>bidf</i>	37.33	32.73	23.96	20.46	8.08	28.43	25.17
χ^2	54.55	25.88	43.97	28.35	13.11	52.73	36.43
<i>dbidf</i>	37.58	24.63	56.25	29.06	11.58	52.73	35.31
Δ_{DF}	54.55	25.55	48.16	28.1	19.64	52.73	38.12
<i>dsidf</i>	37.58	25.25	56.42	26.05	8.72	53.46	34.58
<i>gss</i>	54.55	25.11	48.16	28.1	19.64	52.73	38.05
<i>idf</i>	38.78	32.73	22.31	20.53	8.27	25.22	24.64
<i>ig</i>	4	12.4	45.21	14.99	16.74	51.13	24.08
<i>marascuilo</i>	54.55	23.65	43.97	26.67	17.91	52.73	36.58
<i>nql</i>	42.02	23.97	52.31	27.21	13.29	53.2	35.33
<i>pidf</i>	26.19	33.71	21.83	20.46	8.76	27.68	23.11
<i>rf</i>	41.11	33.09	55.72	28.56	14.93	51.23	37.44

Tableau 3.8 – $F1_{c,(q_d,s_r,q_r),D_c}$ moyenne pour une 5-fold validation croisée pour chaque métrique de sélection de termes pour un seuil égal à 0.5

Par conséquent, la métrique et le seuil doivent être bien sélectionnés en fonction de la catégorie de demandes traitée. En choisissant, pour ces méta-paramètres, les valeurs qui donnent les meilleurs performances d'extraction sur la base d'apprentissage, les résultats suivants sont observés (Tableau 3.9).

Ces résultats détaillés font remarquer que les attributs, pris individuellement, présentent d'assez bonnes performances. Cependant, la mise en correspondance des attributs (triplet (q_d, s_r, q_r)) peine toujours à montrer des performances du même rang. On remarque néanmoins que les mesures-F1 (q_d, s_r, q_r) sont proches de celles des attributs qui présentent le plus de difficulté. L'échec de l'extraction de ces attributs est une des principales causes

<i>c</i>	Données	$ V_c $	Données d'entraînement				Données de test			
			P	R	F1	%Docs	P	R	F1	%Docs
<i>acpa</i>	q_d	1	86.4	56.37	68.13	56.37	68.33	54	58.99	46
	q_r	1	100	65.09	78.74	65.09	93.33	63	71.43	55
	s_r	1	100	65.09	78.74	65.09	93.33	63	71.43	55
	(s_r, q_r)	1	100	65.09	78.74	65.09	93.33	63	71.43	55
	(q_d, s_r, q_r)	1	86.4	56.37	68.13	56.37	68.33	54	58.99	46
<i>concdel</i>	q_d	26	49.33	44.02	45.31	24.17	73.2	29.72	33.29	26.67
	q_r	26	48.3	42.66	44.1	22.5	75.73	28.89	34.3	26.67
	s_r	26	46.52	40.89	42.36	22.5	74.93	26.39	33.09	26.67
	(s_r, q_r)	26	46.52	40.89	42.36	22.5	74.93	26.39	33.09	26.67
	(q_d, s_r, q_r)	26	42.43	37.41	38.68	20.83	68.27	23.06	28.65	23.33
<i>danaïs</i>	q_d	37	77.71	48.71	59.68	37.3	79.25	47.5	59	37.3
	q_r	37	77.68	48.71	59.67	37.03	77.78	46.46	57.79	36.22
	s_r	37	77.05	48.33	59.19	37.03	77.78	46.46	57.79	36.22
	(s_r, q_r)	37	77.05	48.33	59.19	37.03	77.78	46.46	57.79	36.22
	(q_d, s_r, q_r)	37	74.45	46.65	57.16	35.81	74.41	44.38	55.23	34.59
<i>dcppc</i>	q_d	35	45.71	36.64	40.66	34.05	44.64	40.73	41.75	31.4
	q_r	35	78.99	63.21	70.2	59.33	75.48	64.51	68.41	53.82
	s_r	35	84.73	67.85	75.33	63.24	81.21	69.14	73.51	57.43
	(s_r, q_r)	35	78.99	63.21	70.2	59.33	75.48	64.51	68.41	53.82
	(q_d, s_r, q_r)	35	34.2	27.39	30.41	28.03	31.66	28.55	29.41	25.37
<i>doris</i>	q_d	8	31.98	35.76	32.94	7.75	37.48	35.9	36.63	7.12
	q_r	8	35.73	39.72	36.69	8.63	39.43	38.47	38.89	7.12
	s_r	8	35.06	39.56	36.24	9.06	42.91	41.44	42.12	8.94
	(s_r, q_r)	8	32.61	36.16	33.45	8.2	38.14	37.04	37.54	7.12
	(q_d, s_r, q_r)	8	24.48	27.16	25.13	5.61	29.7	28.53	29.08	7.12
<i>styx</i>	q_d	4	69.34	59.55	64.04	33.5	69.3	59.49	63.61	32
	q_r	4	75.87	65.17	70.08	31.5	74.86	64.08	68.63	28
	s_r	4	75.87	65.17	70.08	31.5	74.86	64.08	68.63	28
	(s_r, q_r)	4	75.87	65.17	70.08	31.5	74.86	64.08	68.63	28
	(q_d, s_r, q_r)	4	57.61	49.44	53.19	25.5	57.24	48.36	52.08	24

P : précision, R : rappel, F1 : F1-mesure

%Docs : proportion de documents dont l'ensemble des données extraites est égale à l'attendu
(documents parfaitement traités)

$|V_c|$: nombre moyen de termes-clés identifiés pour la catégorie *c*

Tableau 3.9 – Résultats détaillés pour l'extraction des données avec sélection automatique de la méthode d'extraction des termes-clés

des performances observées pour la liaison des attributs de paires similaires demande-résultat. Par ailleurs, les données sur le résultat, s_r et q_r , sont en générale plus faciles à extraire que le quantum demandé q_d . Il est aussi bien de noter qu'une plus grande quantité d'exemples annotés de documents ne semble pas être la garantie d'une meilleure extraction. On remarque en effet que les meilleures performances sont obtenues pour la catégorie disposant du plus faible nombre d'exemples annotés (*acpa*) avec en moyenne un seul terme-clé appris.

3.4.5 Analyse des erreurs

En extraction d'éléments structurés, on retrouve trois types d'erreurs [Yang and Mitchell, 2016] : les données manquées (faux négatifs), les données en plus des attendues (faux positifs), et les mauvaises classifications (confusions). La confusion n'est pas discutée ici car les annotations ne sont faites que pour une seule classe.

Etant donné que la précision est en général supérieure au rappel, il est certain que les erreurs sont majoritairement dues aux données manquées comme le confirme le Tableau 3.10.

	Données d'entraînement		Données de test	
	%erreurs FP	%erreurs FN	%erreurs FP	%erreurs FN
q_d	36.90	63.10	36.52	63.48
q_r	32.30	67.70	34.32	65.68
s_r	31.72	68.28	34.11	65.89
(s_r, q_r)	32.32	67.68	34.39	65.61
(q_d, s_r, q_r)	37.77	62.23	37.72	62.28

Tableau 3.10 – Types et taux d'erreurs (pourcentage en moyenne sur les 6 catégories de demandes)

Trois raisons peuvent expliquer le fait que peu de données attendues soient extraites. Premièrement, certaines valeurs d'attributs ne sont pas mentionnées dans les sections Litige et Dispositif utilisées (pourcentage inférieurs à 100 dans les Tableaux 3.11 et 3.12 comme par exemple les quanta résultat de *doris* plus présents dans la section Motifs que dans le Dispositif).

	# q_d	# $q_d \neq NUL$	# dans doc.	# dans Litige	# dans Motifs	# dans Dispositif
acpa	23	16	16 (100%)	16 (100%)	9 (56.25%)	5 (31.25%)
concdel	58	56	55 (98.21%)	55 (98.21%)	7 (12.5%)	2 (3.57%)
danais	208	182	182 (100%)	179(100%)	39 (21.43%)	23 (12.64%)
dcppc	126	126	122 (96.83%)	109 (86.51%)	71 (56.35%)	65 (51.59%)
doris	94	83	83 (100%)	82 (98.80%)	21 (25.30)%	6 (7.23%)
styx	89	86	86 (100%)	86 (100%)	12 (13.95%)	9 (10.47%)

Les pourcentages ne sont calculés que pour les valeurs non nulles

Tableau 3.11 – Taux de quanta demandés (q_d) mentionnés dans les documents annotés

	# q_r	# $q_r \neq NUL$	# dans doc.	# dans Litige	# dans Motifs	# dans Dispositif
acpa	23	6	6 (100%)	3 (50%)	6 (100%)	5 (83.33%)
concdel	58	8	8 (100%)	2 (25%)	8 (100%)	6 (75%)
danais	208	23	23 (100%)	15 (65.22%)	22 (95.65%)	20 (86.96%)
dcppc	126	76	75 (98.68%)	55 (72.37%)	56 (73.68%)	64 (84.21%)
doris	94	44	44 (100%)	28 (63.64%)	40 (90.91%)	24 (54.55%)
styx	89	30	29 (96.67%)	16 (53.33%)	22 (73.33%)	29 (96.67%)

Les pourcentages ne sont calculés que pour les valeurs non nulles

Tableau 3.12 – Taux de quanta accordés (q_r) mentionnés dans les documents annotés

Deuxièmement, la sélection des termes-clés n'est pas parfaite (Tableau 3.13). D'une part, l'ensemble sélectionné ne couvre pas toutes les situations d'expression de la catégorie (par exemple, pour la catégorie *styx*, le terme « frais irrépétibles » est souvent utilisés à la place de « article 700 du code de procédure civile », mais dans très peu d'exemples annotés). D'autre part, certains termes sont trop spécifiques à la base d'apprentissage (par exemple, pour la catégorie *concdel*, des sommes d'argent et autres termes comme « condamner in solidum les sociétés » apparaissent dans la liste).

Catégorie	Termes-clés appris
<i>acpa</i>	amende civile
<i>concdel</i>	titre de la concurrence déloyale, somme de 15000euros à titre, réparation de son préjudice financier, payer la somme de 15000euros, condamner in solidum les sociétés, agissements constitutifs de concurrence déloyale
<i>danais</i>	dommages et intérêts pour procédure, 32-1 du code de procédure, intérêts pour procédure abusive, titre de dommages-intérêts pour procédure, intérêts pour procédure, article 32-1 du code, dommages-intérêts pour procédure abusive
<i>dcppc</i>	admet la créance déclarée, admet la créance, passif de la procédure collective, passif de la procédure, hauteur de la somme, créance déclarée, titre chirographaire, admission de la créance, rejette la créance,
<i>doris</i>	préjudices, abusive, condamner solidairement, solidairement, réparation du préjudice, réparation, titre de dommages et intérêts, dommages, titre de dommages, dommages et intérêts, titre de dommages-intérêts, payer aux époux, jouissance
<i>styx</i>	700 du code de procédure, article 700 du code, 700 du code, article 700, 700

Les termes candidats sont des n-grammes de taille variant d'1 à 5 mots consécutifs

Tableau 3.13 – Premiers termes sélectionnés lors de la première itération de la validation croisée

Troisièmement, les expérimentations ont été réalisées sur des décisions d'appel mais les énoncés, de demande et résultat renvoyant aux décisions de jugements antérieurs, ne sont pas encore traités dans l'approche. Ces

références aux décisions antérieures représentent une part importante des demandes discutées dans les décisions d'appel. Il est donc nécessaire de les intégrer explicitement dans le processus d'extraction, pour compléter les données extraites.

3.5 Conclusion

Ce chapitre décrit le problème d'extraction de données pertinentes relatives aux paires demande-résultat mentionnées dans les décisions de justice. Les divers défis relatifs à la tâche y sont discutés en remarquant des analogies avec d'autres tâches classiques de la fouille de données textuelles. Il a été démontré la solvabilité du problème par la proposition et l'expérimentation d'une approche d'extraction basée sur l'apprentissage de la terminologie des catégories de demande et autres connaissances du domaine judiciaire telles que les motifs d'énoncés de demandes et de résultat, ainsi que leur position conventionnelle dans les documents. Les expérimentations démontrent que l'approche permet d'extraire plus ou moins bien des demandes selon la catégorie traitée. A cause de la forte dépendance aux subtilités de rédaction des décisions judiciaires, la méthode rencontre des limites qui ne peuvent être surmontées qu'en rendant la méthode beaucoup plus complexe qu'elle ne l'est déjà. Des approches d'apprentissage automatique sont recommandées comme perspectives. Elles devront être capables d'apprendre l'emplacement des données à extraire de manière semi-supervisée à l'aide de faibles quantités de documents annotés de grande taille.

Chapitre 4

Identification du sens du résultat par classification des documents

4.1 Introduction

Comme le précédent, ce chapitre est relatif à l'extraction de données sur les demandes et résultats correspondants. Cependant, il est question ici d'extraire uniquement le sens du résultat d'une demande connaissant sa catégorie. Cette étude est intéressante parce que le problème devient plus simple. En se passant de la localisation précise de l'énoncé du résultat, l'extraction du sens du résultat peut être formulée comme une tâche de classification de documents. Nous modélisons la tâche comme un problème de classification binaire consistant à entraîner un algorithme à reconnaître si la demande a été rejetée (sens = rejette) ou acceptée (sens = accepte). Cette modélisation est proposée sur une restriction du problème définie par les postulats 4.1.1 et 4.1.2 suivants.

Postulat 4.1.1 *Pour toute catégorie de demande C , les documents ne contenant qu'une demande de catégorie C sont majoritaires.*

Ce postulat est légitime car les statistiques sur les données labellisées de la Figure 3.3 montre bien que dans chaque catégorie, les décisions contiennent en majorité une demande. On remarque néanmoins l'exception de la catégorie STYX (dommage-intérêt sur l'article 700 CPC), où dans la majorité des

Répartition des sens du résultat par catégorie

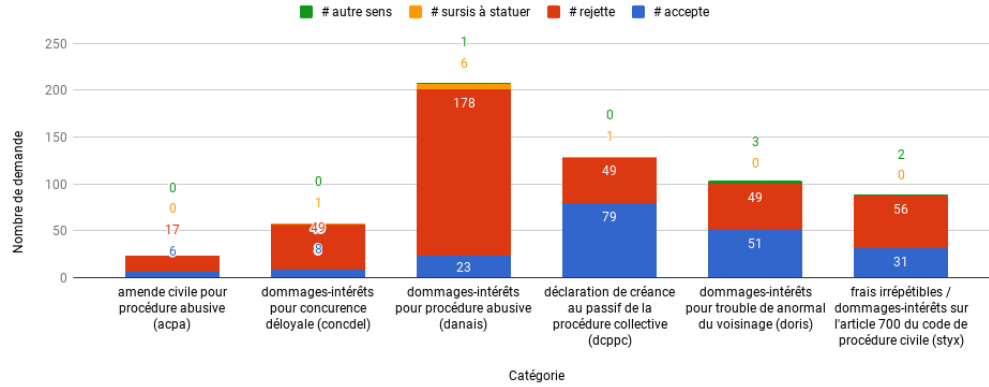


Figure 4.1 – Répartition des sens de résultat dans les données annotées.

documents, on a plutôt 2 demandes. Cette exception peut se justifier par le fait que chaque partie fait généralement ce type de demande car elle porte sur le remboursement des frais de justice. Ce postulat présente cependant un inconvénient dû au fait que la majorité des demandes se trouvent dans des décisions à plus d'une demande. Il est donc possible de manquer un grand nombre de demandes.

Postulat 4.1.2 *Le sens du résultat est généralement binaire : accepte ou rejette.*

Ce postulat est justifié car le sens d'un résultat est pratiquement toujours une de ces deux valeurs (Figure 4.1). Les autres sens ne sont pas considérés car ils sont très rares.

Cette étude porte sur l'analyse de l'impact de différents aspects techniques généralement impliqués dans la classification de texte qui consistent en générale à une combinaison de représentation des documents et d'algorithme de classification. Cette analyse permettra de savoir s'il existe une certaine configuration permettant de déterminer le sens du résultat à une demande sans nécessairement l'avoir identifiée précisément dans le document.

4.2 Classification de documents

La classification de texte permet d'organiser des documents $x^{(k)}$ dans des groupes prédéfinis. Elle reçoit depuis longtemps beaucoup d'attentions. Deux choix techniques influencent principalement les performances : la représentation des textes et l'algorithme de classification.

4.2.1 Algorithmes traditionnels de classification de données

Bien que la classification de documents voit se développer récemment des algorithmes propres aux textes, un grand nombre de méthodes ont été développées précédemment autour. Ces méthodes sont généralement basées sur une représentation vectorielle des textes et délimitent une frontière entre les classes dans un espace multidimensionnel.

4.2.1.1 Le Bayésien naïf (NB)

Principe : Les classifieurs naïfs bayésiens [Duda et al., 1973] sont des modèles à densité qui estiment la probabilité qu'un texte appartienne à une classe à partir du théorème de Bayes [Raschka, 2014] :

$$\text{probabilité a posteriori} = \frac{\text{probabilité conditionnelle} \cdot \text{probabilité a priori}}{\text{évidence}} \quad (4.1)$$

La probabilité a posteriori peut être interprétée dans le cadre de la classification des décisions dans la classe DIPA en "Quelle est la probabilité qu'une décision t_i contienne une demande de type c_j étant donné que t contient les termes $\{w_1, \dots, w_K\}$?". La réponse à cette question se formalise comme suit :

$$\mathbb{P}(c_j|t_i) = \frac{\mathbb{P}(t_i|c_j)\mathbb{P}(c_j)}{\mathbb{P}(t_i)}$$

ou plus simplement $\mathbb{P}(c_j|t_i) = \mathbb{P}(c_j)\mathbb{P}(t_i|c_j)$ car $\mathbb{P}(t_i)$ ne change pas en fonction de la catégorie et peut donc être ignorée [Rish, 2001]. L'appellation "*naïf*" est due à l'hypothèse d'**indépendance mutuelle entre les caractéristiques** des données. Une hypothèse forte dont la violation, par les données réelles, n'empêche pourtant pas les NB de bien fonctionner [Rish, 2001].

Hypothèse 4.2.1 (indépendance mutuelle des caractéristiques) / *Un modèle naïf bayésien étant de type génératif, la position de chaque mot dans le texte est générée indépendamment de tout autre mot étant connue la catégorie du texte.*

L'hypothèse 4.2.1 implique, pour des catégories de demande indépendantes,

$$\mathbb{P}(d_{i1}, \dots, d_{iK}|c_j) = \prod_{k=1}^K \mathbb{P}(d_{ik}|c_j).$$

Ainsi,

$$\mathbb{P}(c_j|d_{i1}, \dots, d_{iK}) = \frac{\mathbb{P}(c_j)\mathbb{P}(d_{i1}, \dots, d_{iK}|c_j)}{\mathbb{P}(d_{i1}, \dots, d_{iK})} = \frac{\mathbb{P}(c_j) \prod_{k=1}^K \mathbb{P}(d_{ik}|c_j)}{\mathbb{P}(d_{i1}, \dots, d_{iK})}.$$

La fonction score issue de ce classifieur bayésien est construite en maximisant la probabilité $\mathbb{P}(c_j|d_{i1}, \dots, d_{iK})$. Il n'est donc pas nécessaire de connaître ni d'estimer la probabilité jointe $\mathbb{P}(d_{i1}, \dots, d_{iK})$.

Estimation des paramètres :

Grâce à l'hypothèse 4.2.1, la probabilité conditionnelle ($\mathbb{P}(t_i|c_j)$) peut-être réécrite :

$$\mathbb{P}(t_i|c_j) = \mathbb{P}(w_1|c_j) \cdot \mathbb{P}(w_2|c_j) \cdot \dots \cdot \mathbb{P}(w_d|c_j) = \prod_{k=1}^d \mathbb{P}(w_k|c_j)$$

pour une représentation vectorielle des textes dans un espace de dimension d . Les paramètres du modèle peuvent donc être estimés à partir d'un jeu

d'exemples d'entraînement. Plus précisément, $\forall k \in \{1, 2, \dots, d\} : \mathbb{P}(w_k | c_j) = \frac{N_{wc_j}}{N_{c_j}}$ et $\mathbb{P}(c_j) = \frac{N_c}{N}$.

Actuellement, les algorithmes NBSVM et FastText sont les plus populaires pour la classification de documents et ont les performances pour l'analyse de sentiment sont très bonnes. L'ensemble des labels / classes est noté C .

VARIANTE

4.2.1.2 Machine à vecteurs de support (SVM)

Le SVM [Vapnik, 1995] est un algorithme de classification binaire, qui construit, lors de la phase d'entraînement un hyperplan séparant les points, représentant les exemples d'entraînement, dans un espace à grande dimension, suivant leur classe (Figure 4.2¹). L'hyperplan est la surface située entre les droites formées par les points les plus proches des deux classes. La classification d'un nouvel objet consiste à projeter son vecteur de caractéristiques dans cet espace, et le label qui lui est prédit est celui associé à la classe du côté où il se trouve. La projection d'une entrée x dans le nouvel espace, est réalisée par une fonction non-linéaire appelé noyau.

optimisation de la fonction objectif pour l'entraînement, + calcul des hyperparamètres

4.2.1.3 k -plus-proches-voisins (kNN)

L'algorithme k -plus-proches-voisins est un algorithme très simple qui consiste à affecter à un nouvel objet la classe majoritaire y' parmi ceux des k points d'exemples d'entraînement $\{(x_i, y_i)\}_{1:k}$, les plus proches du point x' de cet objet selon la métrique d choisie. Ainsi, trois éléments clés influencent l'efficacité de la classification :

1. les données d'entraînement dont le nombre s'il est très grand peut

1. <http://www.clrc.rhul.ac.uk/svm.html>

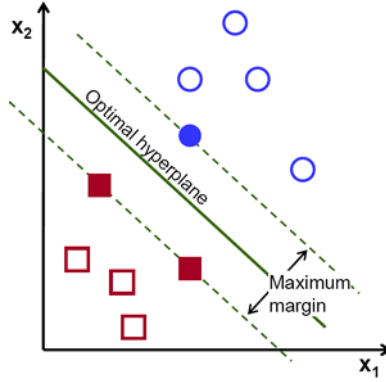


Figure 4.2 – Illustration de l'hyperplan d'un SVM.

rendre chère le processus de classification, car la distance du nouvel objet à chaque point annoté, est calculée ;

2. le nombre de voisins (c'est-à-dire la valeur de k) qui ne doit être ni très petit (sensibilité aux bruits / *outliers*), ni très grand (risque d'avoir dans le voisinage beaucoup de points d'une autre classe). La sensibilité au nombre de voisins peut être atténuée en pondérant les points par leur distance à l'objet à classer. La stratégie de choix de la classe du nouvel objet est ainsi appelé « vote pondéré par la distance » :

$$y' = \operatorname{argmax}_c \sum_{(x_i, y_i)} \lim_{d(x', x_i)^2} \frac{1}{d(x', x_i)^2} \times I(c = y_i)$$

$$\text{où } I(c = y_i) = \begin{cases} 1 & \text{si } c \text{ est égal à } y_i \\ 0 & \text{sinon} \end{cases}$$

3. la métrique de calcul de distance qui doit être adéquate pour le type de donnée et la tâche (par exemple, la distance cosinus est préférable à la distance euclidienne pour la classification de documents, la deuxième métrique se dégradant lorsque le nombre d'attributs augmente).

4.2.1.4 Arbre de décision

Un arbre de décision est structure arborescente utilisée en fouille de données pour associer un label prédéfini à des objets (classification), ou prédire la valeur d'une variable continue (régression). Il comprend en des noeuds internes qui correspondent chacun à un test sur la valeur d'un attribut (test uni-varié), des arêtes correspondant à une sortie du test, et enfin des feuilles ou noeuds terminaux qui correspondent chacune à une prédiction. L'algorithme de classification d'un nouvel objet avec l'arbre de décision est très simple (Algorithme 3).

Algorithme 3 : Classification d'un objet à l'aide d'un arbre de décision

Données : Objet x , Arbre A

Résultat : label

```

1  $n := \text{racine}(A)$  ;
2 tant que  $n$  n'est pas une feuille faire
3   | Effectuer sur  $x$  le test associé à  $n$ ;
4   |  $n :=$  noeud fils de  $n$  correspondant au résultat du test ;
5 retourner le label associé à la feuille  $n$ ;
```

La construction de l'arbre, c'est-à-dire l'apprentissage, consiste à générer une hiérarchie de tests, aussi courte que possible, qui divise successivement l'ensemble S d'exemples d'apprentissage en sous-ensembles disjoints de plus en plus pures². L'arbre est construite de la racine aux feuilles en divisant les données d'entraînement S_t à chaque étape (t) de sorte à minimiser le degré d'impureté des sous-ensemble d'exemples S_{t_i} dans les noeuds fils (t_i). Les divers algorithmes de construction diffèrent par le critère (ou la métrique) de séparation. Un critère de coupe est généralement défini à partir d'une métrique d'impureté comme par exemple :

2. homogénéité des labels

- l'entropie de la distribution des classes dans S_t :

$$h_C(S_t) = - \sum_{c \in C} [p(c|S_t) \log_2 p(c|S_t)] ;$$

- l'indice de Gini mesurant la divergence entre les distributions de probabilité des valeurs de la variable prédite :

$$g_C(S_t) = 1 - \sum_{c \in C} [p(c|S_t)]^2 ;$$

- l'erreur de classification définie par :

$$e_C(S_t) = 1 - \max_{c \in C} [p(c|S_t)]$$

Pour ces métriques, $p(c|S_t)$ représente la proportion d'exemples du nœud t appartenant à c , et S_t représente . Parmi les critères de séparation les plus populaires associés à ces critères, on retrouve :

- le gain d'information apporté par le test t portant sur l'attribut a (qui divise S_t en des sous-ensembles S_{t_i}) utilisant l'entropie comme métrique d'impureté, et est définie par la différence entre l'entropie de t et l'entropie moyenne des fils de t :

$$ig(S_t, a) = h_C(S_t) - i(S_t, t, a) = h_C(S_t) - \sum_{S_{t_i}} \frac{|S_{t_i}|}{|S_t|} \cdot h_C(S_{t_i});$$

- le rapport des gains, qui corrige le gain d'information, biaisé en faveur des tests ayant un grand nombre d'alternatives (sorties du nœud), en prenant en compte l'information intrinsèque $h_t(S_t)$ de la séparation de S_t suivant le test t en sous-ensembles S_{t_i} :

$$gr(S_t, t, a) = \frac{ig(S_t, t, a)}{h_t(S_t)} \text{ avec } h_t(S_t) = \sum_i \frac{|S_{t_i}|}{|S_t|} \log_2 \left(\frac{|S_{t_i}|}{|S_t|} \right)$$

- le critère binaire de "doublage" (*twoing criteria*) qui ne s'emploie dans les arbres binaires :

$$tc(t) = \frac{P(S_{t_R}|S_t)P(S_{t_L}|S_t)}{4} \left[\sum_{c \in C} |p(c|t_L) - p(c|t_R)| \right]^2$$

où $P(S_{t_R}|S_t)$ et $P(S_{t_L}|S_t)$ sont les proportions de S_t qui vont respectivement dans les fils t_R et t_L après séparation suivant le test t .

Les variables nominales peuvent être divisées soit en utilisant autant de partitions que de valeurs distinctes (partition multiple), soit uniquement en des partitions binaires suivant des tests booléens (partition binaire) nécessitant de rechercher la division optimale. Les variables numériques sont divisées quant à elles soit suivant par discrétisation de leur domaine les transformant en variables catégoriques ordinales, soit en recherchant la meilleure division binaire parmi toutes les séparations possibles.

La construction de l'arbre étant une division récursive de noeud qui peut continuer tant qu'il est possible d'améliorer la pureté des noeuds, ce qui peut engendrer un arbre très grand résultant en un sur-apprentissage³. Pour s'arrêter plus tôt ("pré-élagage"), plusieurs conditions sont possibles comme par exemple, l'atteinte par la taille des données ($|S_t|$) d'un seuil minimum, ou l'atteinte par l'arbre d'une profondeur maximale, ou l'amélioration du critère de division est très faible, etc. Le post-élagage est appliqué après construction de l'arbre toujours dans le but de minimiser le sur-apprentissage.

Les algorithmes de construction d'arbres diffèrent ainsi par leur critère de séparation, leur stratégie d'élagage, et leur capacité à gérer les types d'attributs, les valeurs manquantes et extrêmes. On compare ainsi les deux algorithmes CART et C4.5 que nous avons utilisés (Tableau).

3. Un modèle trop précis a un très faible taux d'erreur sur les données d'entraînement (erreur d'apprentissage) mais un fort taux d'erreur pour les données de test (erreur de test).

	Critère de séparation	Variables numériques	Valeurs manquantes	Stratégie d'élagage	Détection de valeurs extrêmes	Implémentations
CART	critère de "doublement"	supportées	supportées	élagage à coût complexe	supportées	Scikit-learn
C4.5	rapport des gains	supportées	supportées	élagage basé sur l'erreur	susceptible	J48 (Weka)

4.2.1.5 Analyses discriminantes linéaires et quadratiques

L'analyse discriminante comprend l'ensemble des méthodes déterminant les combinaisons linéaires de variables qui permettent de séparer le mieux possible K catégories ou variables qualitatives. Les analyses linéaires et quadratiques sont des méthodes probabilistes basées sur la probabilité conditionnelle d'appartenance d'un objet X à une classe y_k :

$$P(Y = y_k|X) = \frac{P(Y = y_k)P(X|Y = y_k)}{P(X)} = \frac{P(Y = y_k)P(X|Y = y_k)}{\sum_{j=1}^K P(Y = y_j)P(X|Y = y_j)}$$

. La classe de X est donc $y_{k*} = \underset{k}{\operatorname{argmax}} P(Y = y_k|X) = P(Y = y_k)P(X|Y = y_k)$ car le dénominateur est le même pour toutes les classes. Dans cette expression, $P(Y = y_k)$ est la proportion d'exemples de classes y_k dans l'ensemble des données d'apprentissage. Il ne reste donc qu'à déterminer $P(X|Y = y_k)$, pour trouver y . Deux hypothèses simplifient les calculs :

1. l'hypothèse de normalité statuant que la probabilité conditionnelle

$P(X|Y)$ suit une loi normale multidimensionnelle :

$$P(X|Y = y_k) = \frac{1}{\sqrt{2\pi \det(\sum_k)}} e^{-\frac{1}{2}(X - \mu_k) \sum_k^{-1} (X - \mu_k)'}$$

μ_k étant le centre de gravité conditionnelle, et \sum_k la matrice de variance covariance conditionnelle ;

2. l'hypothèse d'homoscédasticité statuant que les matrices de variance co-variance conditionnelles sont identiques i.e. :

$$\forall j, k \in \{1, \dots, K\}, \sum_j = \sum_k = \sum.$$

L'analyse discriminante linéaire (LDA) est définie par une simplification de $P(X|y_k)$ sous ces deux hypothèses. En effet, grâce à la proportionnalité de la probabilité conditionnelle à $\ln [P(X|y_k)] \propto -\frac{1}{2}(X - \mu_k) \sum^{-1} (X - \mu_k)'$, on déduit une fonction discriminante (ou de classement) linéaire proportionnelle à $P(y_k|X)$:

$$d(y_k, X) = \ln [P(Y = y_k)] + \mu_k \sum_{-1} X' - \frac{1}{2} \mu_k \sum_{-1} \mu_k'.$$

Ainsi $y_{k*} = \operatorname{argmax}_{k \in \{1, \dots, K\}} d(y_k, X)$.

L'analyse discriminante quadratique (QDA) considère l'hétéroscédasticité (i.e. $\exists k \neq j, \sum_k \neq \sum_j$), et donc ne s'appuie que sur la 1e hypothèse (multinormalité). Dans ce cas, on obtient une règle quadratique de classification $k* = \operatorname{argmax}_{k \in \{1, \dots, K\}} Q_k(X)$ où :

$$Q_k(X) = (x - \mu_k)' \sum_k^{-1} (x - \mu_k) - 2 \ln(\pi_k) + \ln(\det(\sum_k))$$

est la fonction quadratique de classement de la classe k .

4.2.2 Algorithmes dédiés aux textes

4.2.2.1 NBSVM

Le NBSVM [Wang and Manning, 2012] est un classifieur binaire (deux labels $\{-1; 1\}$) dont le principe consiste à transformer les poids $f^{(k)}$ caractéristiques V des textes $x^{(k)}$, réduites à leur simple présence $\widehat{f}^{(k)}$ en réalisant leur produit élément à élément ($\widetilde{f}^{(k)} = r \circ \widehat{f}^{(k)}$) avec le vecteur de poids r du classifieurs bayésien multinomial (calculé avec le vecteur présence de caractéristique) : $r = \log \left(\frac{p/\|p\|_1}{q/\|q\|_1} \right)$ avec $p = \alpha + \sum_{k:y^{(k)}=1} f^{(k)}$, $q = \alpha + \sum_{k:y^{(k)}=-1} f^{(k)}$. L'ensemble des caractéristiques V est constitué de n-grammes de mots. Le nouveau vecteur issu de ce produit représente le texte ($x^{(k)} = \widetilde{f}^{(k)}$) en entrée d'un SVM classique. La classe de $x^{(k)}$ est prédite par : $y^{(k)} = \text{sign}(\mathbf{w}^T x^{(k)} + b)$, \mathbf{w} et b étant appris lors de l'entraînement du SVM. Une interpolation entre le bayésien multinomial et le SVM est nécessaire pour assurer la robustesse du NBSVM et des performances excellentes pour toute tâche de classification de documents ; les poids \mathbf{w} sont réajustés par le model $\mathbf{w}' = (1 - \beta)\bar{\mathbf{w}} + \beta\mathbf{w}$, où $\bar{\mathbf{w}} = \|\mathbf{w}\|_1/|V|$ et $\beta \in [0; 1]$.

4.2.2.2 FastText

FastText [Grave et al., 2017], quant à lui, est un modèle de réseau de neurones dont l'architecture est semblable à celle de la variante CBOW de la méthode de plongement sémantique Word2Vec dans laquelle le mot du milieu a été remplacé par le label de la classe du texte et au dessus de laquelle la fonction softmax $f(z) = \left[\frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \right]_{\forall j \in \{1, \dots, K\}}$ est rajoutée pour réaliser la classification à partir de la représentation distribuée du texte. La phase d'entraînement consiste à minimiser la fonction objectif $-\frac{1}{N} \sum_{n=1}^N y_n \cdot \log f(B \cdot A \cdot x_n)$ qui estime la distribution de probabilité des classes.

Ces deux méthodes intègrent leur propre représentation de document,

contrairement aux algorithmes opérant sur dans des espaces vectorielles aux axes et poids paramétrables à volonté comme le SVM. Il existe un très grand nombre de schémas de représentations des documents.

NBSVM et Fasttext ont démontré une bonne robustesse et des performances excellentes dans le cas de divers tâches de classification : courtes expressions, longs documents, thème, classification subjective (genre), classification de sentiment (positif, neutre, négatif), ... Mais nous voulons déjà savoir comment les algorithmes populaires se comportent sur notre tâche d'identification de la polarité du résultat d'une demande pour une catégorie bien définie. La particularité ici est que la tâche porte sur une demande en particulier parmi les nombreuses que compte le document, des données en faible nombre annotés (23 à 189 documents), une annotation en général déséquilibrée entre les classes (risque d'ignorer une classe très faiblement représentée dans le jeu d'entraînement, par exemple 21 "accepte" contre 166 "rejette").

4.2.3 Discussion : Techniques d'amélioration de la précision

Livre Data mining

4.3 La Regression PLS et ses extensions

Justification : Pourquoi le PLS ? : La regression PLS est une méthode de regression avec laquelle l'on tente d'expliquer une ou plusieurs variables Y (dite dépendantes) par des variables $X = x_1, x_2, \dots, x_p$ (dites explicatives). Elle consiste principalement à transformer les variables explicatives en un nombre réduit de composantes principales orthogonales t_1, t_2, \dots, t_h . Les composantes t_h sont construites étapes par étapes en appliquant l'algorithme du PLS de façon récurrente sur les données mal prédites (résidus). Plus préci-

sément, à chaque itération h , la composante t_h est calculée par la formule $t_h = w_{h1}x_1 + \dots + w_{hj}x_j + \dots + w_{hp}x_p$.

Malgré quelques faiblesses comme celles liées au choix du nombre de composantes, à la complexité des sorties et la linéarité du modèle, la regression PLS présente quelques atouts qui ont notamment de l'intérêt dans notre cas de figure. Par exemple, le PLS gère assez bien la forte disproportion entre le nombre de variables explicatives et le nombre d'observations, lorsque ce dernier est faible comme on peut l'observer dans nos données (faible quantité de données d'apprentissage). Nous avons aussi la prise en compte de la multicollinéarité qui peut exister entre les variables explicatives, notamment quand celles-ci sont associées aux mots/termes souvent cooccurents de nos documents.

Il est intéressant de noter la floraison d'extensions proposées pour répondre aux différentes limites du PLS. Notamment, nous pouvons citer la "*sparse*" PLS introduite pour palier à la "*sparsité*" et la colinéarité des variables explicatives [?], la PLS non-linéaire proposée pour les cas de données non-linéairement séparables [?], ou encore la PLS discriminante combinant la régression PLS et l'analyse discriminante [?]. Nous nous sommes intéressés à deux extensions particulières : la régression Gini-PLS [Mussard and Souissi-Benrejeb, 2018] dont l'intérêt est de réduire la sensibilité aux valeurs aberrantes des variables, et la regression Logit-PLS [Tenenhaus, 2005] combinant la regression logistique et la PLS.

4.3.1 Gini-PLS

Cette méthode élimine la sensibilité du PLS aux valeurs extrêmes en remplaçant la covariance $cov(x_j, y)$ par la covariance de Gini $cog(y; x_j) := cov(y; R(x_j))$ pour l'estimation des résidus $u_{(h)j}$ et des poids w_{hj} [Mussard and Souissi-Benrejeb, 2018].

4.3.2 Logit-PLS

Dans cette approche, $\forall j > 1$, les w_{hj} sont les coefficients de la régression logistique de y sur les composantes $t_1, \dots, t_{h-1}, u_{(h-1)j}$ Tenenhaus [2005].

4.3.3 Gini-Logit-PLS

Cette approche combine la covariance Gini pour $u_{(h)j}$ et le coefficient Logit pour les w_{hj} .

4.4 Méthode

Nous raisonnons toujours suivant une seule catégorie c . Notre solution est une chaîne à 2 étapes de classification : un filtre des décisions à une demande de la catégorie c et un identificateur de la polarité du résultat. Le premier classifieur discrimine les document entre 2 classes : « une demande » et « plusieurs demandes ».

4.5 Expérimentations et résultats

Nous discutons ici les performances de divers algorithmes populaires et l'impact de la quantité et du déséquilibre des données, de la restriction à des passages en particulier, ainsi que leur capacité à faire abstraction des autres demandes du document.

4.5.1 Protocole d'évaluation

Deux métriques d'évaluation sont utilisées : la précision et la F1-mesure. Pour tenir compte du déséquilibre entre les classes, la moyenne macro est préférée (agrégation de la contribution individuelle de chaque classe : $F1_{moyenne} = \frac{1}{2}(F1(accepte) + F1(rejette))$).

Les données utilisées sont les mêmes que celles du chapitre précédent. Nous avons seulement fait une restriction sur les documents n'ayant qu'une seule demande annotée pour la catégorie considérée. Le déséquilibre entre les classes est illustrée par la figure 4.3. En effet, la demande est plus souvent rejetée qu'acceptée pour les catégories ACPA, CONCDEL, DANAIS et STYX. Le contraire est observé pour DCPPC et DORIS.

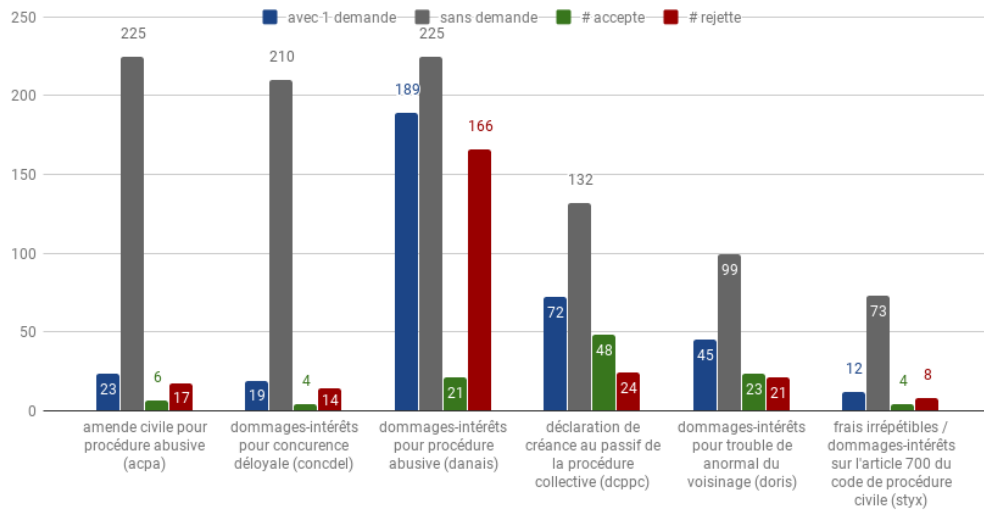


Figure 4.3 – Répartition des documents à une demande de la catégorie considérée.

4.5.2 Classification de l'ensemble du document

En représentant l'ensemble du document à l'aide de diverses représentations vectorielles, les algorithmes sont comparés avec les représentations qui leurs sont optimales. On remarque d'après les résultats du Tableau 4.1, les arbres sont en moyenne meilleurs sur l'ensemble des catégories même si en moyenne la F1-mesure moyenne est limitée à 0.668. Les résultats des extensions du PLS ne sont pas très éloignée de ceux des arbres avec des différences de F1 à moins de 0.1 (si on choisi le bon schéma de "vectorisation").

ajouter les F1 ou erreur de rejette et de accepte

Vecteur	algorithme	F1	min	Cat. min	max	Cat. max	F1 - 1erF1	max - min	rang
GSS*TF	Arbre	0.668	0.5	doris	0.92	dcppc	0	0.42	1
AVG-G*TF	LogitPLS	0.648	0.518	danais	0.781	dcppc	0.02	0.263	13
AVG-G*TF	StandardPLS	0.636	0.49	danais	0.836	dcppc	0.032	0.346	24
DELTADF*TF	GiniPLS	0.586	0.411	danais	0.837	dcppc	0.082	0.426	169
DELTADF*TF	GiniLogitPLS	0.578	0.225	styx	0.772	dcppc	0.09	0.547	220
-	NBSVM	0.494	0.4	styx	0.834	dcppc	0.174	0.434	
-	FastText	0.412	0.343	doris	0.47	danais	0.256	0.127	

Tableau 4.1 – Comparaison des algorithmes sur une représentation globale des documents pour la détection du sens du résultat.

Cat. Dmd.	Algo.	Préc.	Préc. équi.	err-0	err-1	f1-0	f1-1	f1-macro-avg
dcppc	nbsvm	0.875	0.812	0.375	0	0.752	0.916	0.834
danais	fasttext	0.888	0.5	1	0	0	0.941	0.47
danais	nbsvm	0.888	0.5	0	1	0.941	0	0.47
concdel	fasttext	0.775	0.5	1	0	0	0.873	0.437
concdel	nbsvm	0.775	0.5	0	1	0.873	0	0.437
acpa	fasttext	0.745	0.5	1	0	0	0.853	0.426
acpa	nbsvm	0.745	0.5	0	1	0.853	0	0.426
doris	nbsvm	0.5	0.492	0.85	0.167	0.174	0.63	0.402
dcppc	fasttext	0.667	0.5	0	1	0.8	0	0.4
styx	fasttext	0.667	0.5	1	0	0	0.8	0.4
styx	nbsvm	0.667	0.5	0	1	0.8	0	0.4
doris	fasttext	0.523	0.5	0	1	0.686	0	0.343

0 == accepte

1 == rejette

Tableau 4.2 – Détails des résultats de FastText et NBSVM.

Les scores F1 moyens des algorithmes NBSVM et FastText n'excèdent en général pas 0.5 malgré qu'ils soient spécialement conçus pour les textes. Soit ils sont très sensibles au déséquilibre des données entre les catégories (plus de rejets que d'acceptations), soit il est plus difficile de détecter l'acceptation des demandes. En effet, ces algorithmes classent tous les données de test avec le label (sens) majoritaire i.e. le rejet, et par conséquence, ils ne détectent peu ou pas d'acceptation de demande. Le cas des catégories DORIS et DCPPC pour le NBSVM (F1-macro moyen = 0.834) tend à démontrer la forte sensibilité aux cas négatifs de ces algorithmes puisque même avec presque autant de labels "accepte" que "rejette", la F1-mesure de rejette est toujours supérieure à celle de "accepte" (Tableau 4.2).

4.5.3 Réduction du document aux régions comprenant le vocabulaire de la catégorie

Etant donné que les décisions portent sur plusieurs catégories de demande, nous avons expérimenté la restriction du document aux passages comprenant du vocabulaire de la catégorie d'intérêt : demande, résultat, résultat antérieur (resultat_a), paragraphes dans les motifs (motifs). Les combinaisons passages-représentation vectorielle-algorithme sont comparées dans le Tableau 4.3. Les résultats s'améliorent énormément avec les réductions, sauf pour la catégorie DORIS. La meilleure restriction combine les passages comprenant le vocabulaire de la catégorie dans la section Litige (demande et résultat antérieur), dans la section Motifs (contexte), et dans la section Dispositif (Résultat).

Cat. Dmd	zone	Vecteur	classifieur	F1
acpa	demande resultat a resultat_context	DBIDF*TF	Tree	0.846
acpa	litige motifs dispositif	DELTADF*TF	StandardPLS	0.697
acpa	litige motifs dispositif	AVERAGEGlobals*TF	LogitPLS	0.683
concdel	litige motifs dispositif	GSS*TF	Tree	0.798
concdel	motifs	IDF*TF	GiniLogitPLS	0.703
concdel	contexte	DBIDF*LOGAVE	StandardPLS	0.657
danais	demande resultat a resultat_context	CHI2*AVERAGELocals	Tree	0.813
danais	demande resultat a resultat_context	AVERAGEGlobals*ATF	LogitPLS	0.721
danais	demande resultat a resultat_context	AVERAGEGlobals*ATF	StandardPLS	0.695
dcppc	demande resultat a resultat_context	CHI2*TF	Tree	0.985
dcppc	demande resultat a resultat_context	CHI2*TF	LogitPLS	0.94
dcppc	litige motifs dispositif	MARASCUILO*TP	StandardPLS	0.934
doris	litige motifs dispositif	DSIDF*TP	GiniPLS	0.806
doris	litige motifs dispositif	DSIDF*TP	GiniLogitPLS	0.806
doris	litige motifs dispositif	IG*ATF	StandardPLS	0.772
styx	motifs	DSIDF*TF	Tree	1
styx	demande resultat a resultat_context	DSIDF*LOGAVE	GiniLogitPLS	0.917
styx	litige motifs dispositif	RF*TF	GiniPLS	0.833

Tableau 4.3 – Détection du sens du résultat : Comparaison des réductions du document.

4.6 Conclusion

Chapitre 5

identification des Circonstances Factuelles

5.1 Introduction

Les circonstances factuelles sont les différents cas ou différentes situations possibles dans lesquelles une catégorie de demande peut être formulée. Il est important de les identifier car les analyses descriptives ou prédictives ne prennent sens que lorsque les différentes décisions partagent des situations similaires. Par exemple, il serait imprudent de considérer toutes les décisions portant sur l'article 700 pour prédire les chances d'acceptation d'une affaire ne portant que sur un licenciement. Il serait préférable de travailler uniquement avec des affaires de licenciement. Une méthode d'extraction des différentes circonstances factuelles devient donc indispensable. Malheureusement, les circonstances factuelles sont quasi infinies et donc pratiquement impossible à dénombrer. Il serait ainsi extrêmement difficile d'annoter manuellement des échantillons de décisions pour chaque circonstance possible afin de résoudre ce problème par classification. Il est donc plus intéressant d'adopter une approche non-supervisée qui peut extraire les différentes situations mentionnées dans un corpus donné sachant que les documents de ce corpus sont de la même catégorie de demande.

L'objectif est donc de regrouper dans l'ensemble des décisions qui traitent de problèmes similaires. La tâche peut être formulée comme une tâche de regroupement non-supervisé (clustering) ou d'extraction de thématiques (topic

modeling).

5.2 Formulation du Problème

5.3 Regroupement non-supervisé de documents

5.3.1 Choix de l'algorithme de clustering

Le clustering est une tâche d'organisation d'un ensemble d'objets qui consiste à affecter chaque objet à une catégorie inconnue au départ. Les catégories sont donc créées automatiquement au cours du même processus contrairement à la classification dite supervisée qui manipule des catégories prédéfinies par la tâche et généralement à la main. Les catégories peuvent être disjointes ou à chevauchements, et plates ou hiérarchiques. L'algorithme à utiliser dépend généralement de la forme qu'on souhaite donner à l'organisation, par ex. Si les chevauchements sont négligeables ou pas, ou bien si la structure hiérarchique permettrait de mieux expliquer et distinguer les différences inter-groupes et les ressemblances intra-groupes. Nous souhaitons organiser des décisions de justices en fonction des circonstances factuelles auxquelles ces documents sont liés. On pourrait par exemple faire une restriction des données aux cas où chaque document n'appartient qu'à une classe et proposer un système de clustering disjoint.

5.3.2 Métrique de similarité ou de dissimilarité (Comment mesurer la similarité?)

Une métrique f est une fonction qui quantifie la "distance" entre toute paire d'éléments x et y d'un ensemble, induisant ainsi une mesure de similarité¹. Elle satisfait aux propriétés suivantes :

1. http://slazebni.cs.illinois.edu/spring17/lec09_similarity.pdf

1. $f(x, y) \geq 0$ (non-négativité)
2. $f(x, y) = 0 \Leftrightarrow x = y$ (identité discernable)
3. $f(x, y) = f(y, x)$ (symétrie)
4. $f(x, z) \leq f(x, y) + f(y, z)$ (inégalité triangulaire)

On parle de **pseudo-métrie** lorsque la condition 4 n'est pas satisfaite.

Les métriques **pré-définis** sont spécifiées sans aucune connaissances à priori des données (ex. les distances de Minkowski : $f(x, y) = \|x - y\|_{L_p} = \sqrt[p]{\sum |x_i - y_i|^p}$). Par contre, les métriques **appries** sont définies à partir de connaissances des données labellisées. Ces métriques sont apprises pour répondre à la difficulté d'identifier la métrique appropriée pour un problème. Un apprentissage non-supervisé typique consiste à appliquer une transformation linéaire apprise L aux données afin d'étendre les dimensions qui contiennent plus d'information et contracter celles qui expliquent moins les données. La métrique est apprise sur les données sans aucune paire d'éléments pour laquelle la distance est connue d'avance. Par exemple, la distance de Mahalanobis pondère la distance euclidienne entre deux points par l'écart type des données : $f(x, y) = (x - y)^T M^{-1} (x - y)$ (où M est par ex. la matrice de covariance à moyenne soustraite de tous les points). Si des données labellisées sont disponibles c-à-d. les points sont catégorisés (associé à une classe prédéfinie),

5.3.3 Déterminer le nombre approprié de clusters

Au delà de l'algorithme à utiliser, le nombre k approprié de groupes (clusters) doit être déterminé mais pas prédéfini. La principale raison étant que ce nombre est censé inconnu et que le regroupement est censé inconnu et être révélé automatiquement. La technique traditionnelle part d'un faible nombre de clusters et l'incrémenté progressivement. Sur un espace à deux dimensions, l'évolution d'une fonction évaluation en fonction du nombre est observée. La technique est dite du "coude" ou "genou" (elbow/knee) et est

basé sur l'idée selon laquelle on devrait choisir un nombre de clusters tel que l'ajout d'un autre ne donne pas une meilleure modélisation des données. Le coude correspond donc au point (nb de clusters) où l'on considère insignifiante la décroissance de la valeur de la métrique d'évaluation. Cette métrique est généralement la variance intra-cluster qui est la somme des erreurs au carré :

$$J(k) = \sum_{j=1}^k \sum_{x_i \in C_j} ||x_i - \bar{x}_j||^2$$

C_j : ensemble des objets du cluster j

\bar{x}_j : échantillons moyens du cluster j

Comme on peut le remarquer, il s'agit d'une métrique non-supervisée (pas besoin de données labellisées) qu'il faut minimiser. C'est une fonction monotone décroissante ou croissante. Le choix de ce coude est visuel et peut sembler ambigu. Salvador et Chan (2004) propose d'utiliser l'intersection des deux lignes approximant la courbe. Mais plus récemment, Zhang et al. (2016) trouve que cette approche n'est pas appropriée pour les cas où le graphe d'évaluation n'est ni lisse, ni monotone. Zhang et al. (2016) propose d'exploiter la courbure du graphe i.e. la valeur dont un objet géométrique s'écarte d'être plat ou droit dans le cas d'une ligne.

5.3.4 Définir une représentation appropriée pour les textes

<https://arxiv.org/pdf/1509.01626.pdf>

http://ad-publications.informatik.uni-freiburg.de/theses/Bachelor_Jon_Ezeiza_2017_presentation.pdf

5.3.5 Labeliser les clusters

5.3.6 Evaluation du clustering généré

5.3.6.1 Évaluation supervisée

L'idée de l'évaluation supervisée est basée sur la disponibilité de données labellisées i.e. pour lesquelles on sait d'avance quelles classes sont attendues. Une première idée consiste à comparer la constitution des groupes générés aux groupes prédéfinis. Pour cela plusieurs mesures sont utilisées dans la littérature, notamment le trio populaire précision, rappel et F1-mesure . Ces mesures ont pour interprétation la mesure de la capacité de l'algorithme utilisé de mettre dans un même cluster des objets similaires (appartenant au même groupe prédéfini) et dans des clusters différents des objets dissimilaires (de groupes prédéfinis différents).

Le clustering étant une approche non supervisée aidant à la représentations d'objets pour d'autres tâches supervisés, il est possible de considérer le clustering qui contribue le plus aux performances de la tâche cible. Candillier et al. [2006] illustrent par exemple l'évaluation et la comparaison d'algorithmes de clustering en exploitant une tâche de classification. Les clusters générés enrichissant la représentation vectorielle des objets. L'idée est de comparer les performances d'un classifieur dans les deux situations : avec ou sans l'enrichissement du clustering. Le jeu de données est divisé en 2 et

5.3.6.2 Évaluation non-supervisée

5.4 Méthodes proposées

5.4.1 K-médoïdes et « Word Mover's Distance »

Les approches de clustering sont généralement appliquées à une représentation vectorielle des objets. Particulièrement la méthodes des K-moyennes

qui met à jour le centroïde en faisant la moyenne des membres de son cluster. Cependant, Kusner et al. [2015] ont proposé récemment *la distance du déplaceur de mot* (*word mover's distance* - *WMD*), une métrique non-supervisée qui permet à la méthode des *K plus proches voisins* (KNN) d'obtenir des performances sans précédents. De plus, l'algorithme de clustering K-médoïdes [Kaufman and Rousseeuw, 1987], similaire aux K-moyennes, choisi comme centroïde le membre du cluster qui minimise la distance aux restes des membres ; ce qui n'impose pas de représentation vectorielle. Ainsi, nous pouvons utiliser la métrique WMD dans l'algorithme des K-médoïdes. Tout en nous appuyant sur un algorithme établi de clustering, nous évitons aussi la recherche de la meilleure représentation vectorielle qui influence souvent les performances du clustering.

Algorithme : <http://isiarticles.com/bundles/Article/pre/pdf/79087.pdf>

Un des désavantage de l'algorithme des K-médoïdes est son long temps de calcul dû à ????. Nous avons, par conséquent, remplacé la distance euclidienne par la WMD dans la version plus rapide de Park and Jun [2009] avec nombre de clusters prédéfinis, et celle de Sabzi et al. [2011] qui intègre une optimisation du nombre de clusters.

5.4.2 Apprentissage d'une métrique fondée la modification du document

Les algorithmes de clustering et de classification s'appuient généralement sur une représentation vectorielle à partir de laquelle une valeur de similarité est calculée de manière non supervisée et avec une formule mathématique statique. Dans le cas des textes, Il n'est pas toujours évident de définir la représentation vectorielle associée à une sémantique précise et objective. De plus il existe un large éventail de schémas de représentations vectorielles. Elles vont des représentations très ad-hoc du type TF-IDF, au représentations apprises comme le doc2vec et en passant par les agrégations pondérées de

modèles distribués de mots (word2vec par ex.).

L'idée dans notre approche est de proposer une formulation de la similarité entre deux documents qui est basée sur le degré de perturbation observé entre documents. Cette fonction de similarité est apprise à partir d'une base synthétique d'entraînement de similarité entre paire de texte.

Postulat 5.4.1 *La distance entre deux documents est une fonction du degré de perturbation permettant de transformer un document en un autre.*

La similarité est définie en fonction de la notion de perturbation du contenu d'un texte (Postulat 5.4.1) : après une légère perturbation, un texte reste assez similaire à l'original ; et après une forte perturbation, le texte sera très différent de l'identique. la similarité (resp. la distance) décroît (resp. croît inversement) donc avec l'intensité de la perturbation. Nous définissons une densité de probabilité de perturbation $P \in [0; 1]$ associée à la probabilité de modifier un mot (suppression / remplacement par un mot très différent).

Considérons que pour tout texte x , W_x désigne l'ensemble des mots dans x . Nous définissons un patron de métriques :

$$\begin{aligned} d &: C \times C \rightarrow \mathbb{R} \\ x, y &\mapsto d(x, y) = f(P_{x,y}) \end{aligned} \quad (5.1)$$

C est le corpus. $P_{x,y}$ est l'ensemble des modifications de x nécessaire pour obtenir y i.e. les paires (w_x, w_y) telles que $w_x \in W_x$ a été remplacé par $w_y \in W_y$. d désigne la métrique. Un simple estimateur de d peut être de considérer le taux de mots modifiés dans x :

$$\tilde{d}(x, y) = \frac{||P_{x,y}||}{||W_x||} \quad (5.2)$$

Un tel modèle ne considère ni l'ordre des mots, ni celui des phrases, ni la différence d'importance entre les mots, la complexité de la modification (une substitution est plus complexe qu'une suppression ou une insertion), ni le degré sémantique des perturbations. ce dernier peut être estimé en lissant

le taux de perturbation à l'aide de la distance sémantique entre les mots substitués (le vecteur représentant le mot vide étant le vecteur nul par ex.) :

$$\tilde{d}(x, y) = \frac{\sum_{(w_x, w_y) \in P_{x,y}} d_w(w_x, w_y)}{\|W_x\|} \quad (5.3)$$

d_w désignant la distance sémantique entre les mots. Ainsi, les substitutions sont pondérées par la distance cosinus entre les vecteurs des mots échangés.

Il est difficile de calculer de telle distance sur un grand corpus étant donné la longueur de notre document. Nous ne pouvons que l'estimer. Pour cela, nous générons un jeu artificiel de données pour l'entraînement d'un modèle régressif d'estimation de la métrique entre deux textes x et y . En effet, nous partons d'un ensemble C de documents et pour chacun de ces documents, noté x , nous générons aléatoirement une valeur seuil de probabilité de perturbation en deça duquel un mot de x est modifié. Par la suite, le texte y , résultant de la modification de x , est généré en modifiant séquentiellement les mots de x :

Algorithme 5 : modifie

Données : un mot w , le vocabulaire W

Résultat : un mot w'

1 $w' =$ un mot différent de w choisi aléatoirement dans W ;

2 **retourner** w'

Après avoir généré l'ensemble $B = \{(d_i, d'_i, s(d_i, d'_i))\}$ de données artificielle d'entraînement, on entraîne un modèle régressif pour prédire la similarité entre 2 documents en fonction de leur représentation vectorielle. Ce modèle régressif peut être utilisé comme métrique de similarité dans un algorithme de clustering comme l'algorithme des K-moyennes.

Issues :

- les docs sont généralement de tailles différentes, ne faudrait il pas intégrer une perturbation ajout de mots ? [combiner les modifications par suppression et par substitution](#)

Algorithme 4 : Génère une perturbation de x **Données :** texte x , valeur seuil de probabilité p **Résultat :** $y, \tilde{d}(x, y)$

```

1  $y = [];$ 
2  $P_{x,y} = \emptyset;$ 
3 pour  $w_x$  in  $x$  faire
4    $v = \text{random}(0, 1);$ 
5   si  $v < p$  alors
6      $w_{y+} = \text{modifie}(w_x);$  // Algorithme 5 ;
7      $y+ = w_y;$ 
8      $P_{x,y} = P_{x,y} \cup \{(w_x, w_y)\};$ 
9   sinon
10     $y+ = w;$ 
11     $\tilde{d}(x, y) = \frac{\sum_{(w_x, w_y) \in P_{x,y}} d_w(w_x, w_y)}{\text{card}(W_x)};$ 
12 retourner  $y, \tilde{d}(x, y);$ 

```

- il faudrait intégrer la composante taille du document : [agréger sur le nombre minimal de phrases des paires de documents](#)
- comment assurer les propriétés d'une fonction similarité ? par exemple si aucune perturbation n'est opérée, alors la similarité est maximale et si tous les mots sont modifiés alors la similarité est minimale : [agrégation par soustraction des vecteurs du couple de docs. plus deux doc seront similaire, plus le vecteur de leur paire tendra vers le nul](#)
- Ne faudrait il pas prendre en compte un poids pour les mots, car peut-être la modification de certains mots ne devrait pas avoir le même impact sur la similarité ou le taux de perturbation que celle d'autres mots : [lissage par la somme des distance des vecteurs de mots substitués Eq. 5.3](#)
- ne faudrait il pas intégré une métrique proche de la tâche : la ressemblance n'est pas forcément globale à tous le corps du document mais plus à certaines régions ; donc un document auquel on rajoute quelques phrases ne devrait pas voir son sens trop changer : [peut-être](#)

agréger les distances minimales entre les paires de phrases

5.5 Expérimentations et interprétation des résultats

5.5.1 Annotations de données d'évaluation

Pour l'évaluation supervisée, nous disposons d'une base annotée sur la catégorie de demande "dommage-intérêts / action en responsabilité civile professionnelle contre les avocats" qui concerne les contentieux impliquant des avocats. L'expert annotateur a identifié quatre cas différents (a, b, c, d) décrits en annexe. En gros :

- pour le cas a) il s'agit d'un avocat qui est négligent et envoie son assignation de manière tardive (champ sémantique : retard/délai/prescription)
- pour le cas b) il s'agit d'un avocat qui n'a pas donné un conseil opportun, qui n'a pas soulevé le bon argument
- pour le cas c) un avocat qui n'a pas rédigé un acte valide ou réussi à obtenir un avantage fiscal (champ sémantique : rédacteur d'actes)
- pour le cas d) il s'agit d'un avocat attaqué par son adversaire et non pas par son propre client.

Le dataset comprend 81 documents répartis dans 4 groupes avec 6 documents appartenant chacun à 2 groupes.

Pour l'évaluation non supervisée, les 6 catégories de demande utilisée pour l'extraction de demandes sont utilisés en plus.

5.5.2 Apprentissage de la métrique

5.5.3 Comparaison d'approches

Comparer la vectorisation du document sur tout son contenu vs. sur la restriction aux énoncés de demande de la catégorie (du type "constater",

"dire et juger") vs restriction aux conclusions (le raisonnement des parties décrits les circonstances factuelles) + motifs sur la catégorie

5.6 Conclusion

jhk lk

lkjkl

Chapitre 6

Démonstrateur d'analyse descriptive

kjjkb,n

kjgkj

6.1 Objectif et Cas d'Utilisation

jkj

6.2 Description du Pipeline

jbkj

6.3 Illustration d'analyses descriptives

bkjbj

6.3.1 Implémentation du système

bkjbj;

6.3.2 Données

6.3.2.1 Distribution de la base dans l'espace et dans le temps

lkjhb kj

6.3.3 Analyse du sens du résultat

;bkjkl

6.3.3.1 Evolution dans le temps

6.3.3.2 Différence dans l'espace

6.3.4 Analyse des quanta

,bkjl hio

6.3.4.1 Evolution dans le temps

6.3.4.2 Différence dans l'espace

6.3.4.3 Quantum demandé vs. quantum accordé

6.4 Conclusion

hg fgh lkhk

Conclusions Générales

F.5 Contributions

Introduction de nouvelles tâches d'extraction d'information motivées par des applications au monde réel.

Ouverture de la réflexion sur la nécessité ou pas de définir des approches propres NLP au domaine juridique.

Annotation manuelle de données

modélisation

expérimentation

F.6 Critique du travail

Quelle représentativité ont les données utilisées dans les expérimentations

F.7 Travaux futurs de recherche

F.8 Perspectives du domaine

Le conflit entre la qualité des données et l'automatisation est important. Galgani et al. [2015] montrent par exemple qu'il est possible en un temps raisonnable d'annoter manuellement un nombre considérable de texte. Il se pose alors la question de savoir à quel point l'exhaustivité est-elle nécessaire pour contraindre les experts à supporter la marge d'erreurs infligée par les

outils d'extraction automatique.

Premier pas pour d'autres voies de recherche : legal / norm Citation network analysis, Anonymisation, analyse des arguments (raison influençant le sens d'un résultat,

Cas d'utilisation : exhaustivité, rapidité, et perspectives multiples dans l'analyse des décisions, aide à la décision, assistance à l'enseignement du droit

Critiques : fiabilité des analyses descriptives (biais des données : nombre et type de documents analysés, biais d'erreur des modèles : faux négatifs (données manquées), faux positifs (données en trop), quelles marges d'erreur tolérées)

Bibliographie

comparative study id3, cart and c4.5 decision tree algorithm : a survey.

ACE. *ACE (Automatic Content Extraction) English Annotation Guidelines for Events*. Linguistic Data Consortium, 5.4.3 edition, 2005. <https://www ldc.upenn.edu/sites/www ldc.upenn.edu/files/english-events-guidelines-v5.4.3.pdf>.

ACE. *ACE (Automatic Content Extraction) English Annotation Guidelines for Relations*. Linguistic Data Consortium, 6.2 edition, 2008. <https://www ldc.upenn.edu/sites/www ldc.upenn.edu/files/english-relations-guidelines-v6.2.pdf>.

David Ahn. The stages of event extraction. In *Proceedings of the Workshop on Annotating and Reasoning about Time and Events*, pages 1–8. Association for Computational Linguistics, 2006.

Nikolaos Aletras, Dimitrios Tsarapatsanis, Daniel Preotiuc-Pietro, and Vasilios Lamos. Predicting judicial decisions of the European Court of Human Rights : A Natural Language Processing perspective. *PeerJ Computer Science*, 2 :e93, 2016.

Rayner Alfred, Leow Chin Leong, Chin Kim On, and Patricia Anthony. Malay named entity recognition based on rule-based approach. *International Journal of Machine Learning and Computing*, 4(3) :300, 2014.

S. Amarappa and S. V. Sathyanarayana. Kannada named entity recognition and classification (NERC) based on multinomial naïve bayes (MNB) classi-

- fier. *International Journal on Natural Language Computing (IJNLC)*, 4(4), August 2015. URL <https://arxiv.org/pdf/1509.04385.pdf>. <https://arxiv.org/pdf/1509.04385.pdf>.
- Pascal Ancel. Les décisions d'expulsion d'occupants sans droit ni titre - connaissance empirique d'un contentieux hétérogène. Technical report, [Rapport de recherche] Ministère de la Justice, 2003. URL <https://halshs.archives-ouvertes.fr/halshs-00798914/document>. Texte : <https://halshs.archives-ouvertes.fr/halshs-00798914/document>, Annexes : https://halshs.archives-ouvertes.fr/halshs-00798914/file/decisions_d_expulsion-annexes_1-5.pdf.
- Judith Jeyafreeda Andrew and Xavier Tannier. Automatic extraction of entities and relation from legal documents. In *Proceedings of the Seventh Named Entities Workshop*, pages 1–8, 2018.
- Kevin D. Ashley and Stefanie Brüninghaus. Automatically classifying case texts and predicting outcomes. *Artificial Intelligence and Law*, 17(2) :125–165, 2009. ISSN 09248463. doi : 10.1007/s10506-009-9077-9. URL <http://link.springer.com/content/pdf/10.1007%2Fs10506-009-9077-9.pdf>. Available at : <http://link.springer.com/article/10.1007%2Fs10506-009-9077-9>.
- Daniel Bakkeland. An lcs-based string metric. *Olso, Norway : University of Oslo*, 2009.
- Breck Baldwin. *Coding chunkers as taggers : IO, BIO, BMEWO, and BMEWO+*, 2009. URL <http://bit.ly/2xRo8Ni>. <http://bit.ly/2xRo8Ni>.
- Petr Berka. Nest : A compositional approach to rule-based and case-based reasoning. *Advances in Artificial Intelligence*, 2011 :15, 2011. doi : 10.1155/2011/374250. Article ID 374250. <http://www.hindawi.com/journals/aai/2011/374250/>.

- David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet Allocation. *the Journal of Machine Learning Research*, 3 :993–1022, 2003.
- II Bommarito, J Michael, Daniel Martin Katz, and Eric M Detterman. Lexnlp : Natural language processing and information extraction for legal and regulatory texts. *arXiv preprint arXiv :1806.03688*, 2018.
- Ralf D Brown. Selecting and weighting n-grams to identify 1100 languages. In *International Conference on Text, Speech and Dialogue*, pages 475–483. Springer, 2013.
- Laurent Candillier, Isabelle Tellier, Fabien Torre, and Olivier Bousquet. Évaluation en cascade d’algorithmes de clustering. In *Conférence d’Apprentissage*, 2006. URL <http://www.lattice.cnrs.fr/sites/itellier/articles/CAP06CASC.pdf>.
- Cristian Cardellino, Milagro Teruel, et al. A low-cost, high-coverage legal named entity recognizer, classifier and linker. In *Proceedings of the 16th edition of the International Conference on Artificial Intelligence and Law*, pages 9–18. ACM, 2017. URL <https://hal.archives-ouvertes.fr/hal-01541446/file/main.pdf>.
- Michael Chau, Jennifer J Xu, and Hsinchun Chen. Extracting meaningful entities from police narrative reports. In *Proceedings of the 2002 annual national conference on Digital government research*, pages 1–5. Digital Government Society of North America, 2002. URL https://www.researchgate.net/profile/Michael_Chau2/publication/2544779_Extracting_Meaningful_Entities_from_Police_Narrative_Reports/links/0c96053ad7eecd1adf3000000.pdf.
- Laura Chiticariu, Rajasekar Krishnamurthy, Yunyao Li, Frederick Reiss, and Shivakumar Vaithyanathan. Domain adaptation of rule-based annotators for named-entity recognition tasks. In *Proceedings of the 2010 Conference*

- on *Empirical Methods in Natural Language Processing*, pages 1002–1012. Association for Computational Linguistics, 2010.
- Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1) :21–27, 1967.
- Laurette Cretin. L’opinion des français sur la justice. *INFOSTAT JUSTICE*, 125, Janvier 2014. URL http://www.justice.gouv.fr/art_pix/1_infostat125_20140122.pdf. http://www.justice.gouv.fr/art_pix/1_infostat125_20140122.pdf.
- Christopher Dozier, Ravikumar Kondadadi, Marc Light, Arun Vachher, Sriharsha Veeramachaneni, and Ramdev Wudali. Named entity recognition and resolution in legal text. In *Semantic Processing of Legal Texts*, pages 27–43. Springer, 2010.
- Richard O Duda, Peter E Hart, et al. *Pattern classification and scene analysis*, volume 3. Wiley New York, 1973.
- Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 363–370. Association for Computational Linguistics, 2005.
- Eibe Frank, MA Hall, and IH Witten. The weka workbench. *Data mining : Practical machine learning tools and techniques*. Burlington : Morgan Kaufmann, 2016.
- Katerina Frantzi, Sophia Ananiadou, and Hideki Mima. Automatic recognition of multi-word terms :. the c-value/nc-value method. *International journal on digital libraries*, 3(2) :115–130, 2000.
- Luigi Galavotti, Fabrizio Sebastiani, and Maria Simi. Experiments on the use of feature selection and negative evidence in automated text categorization.

- In *International Conference on Theory and Practice of Digital Libraries*, pages 59–68. Springer, 2000.
- Filippo Galgani, Paul Compton, and Achim Hoffmann. Lexa : Building knowledge bases for automatic legal citation classification. *Expert Systems with Applications*, 42(17-18) :6391–6407, 2015. URL https://link.springer.com/content/pdf/10.1007%2F978-3-642-17432-2_45.pdf.
- Michael Genesereth. Computational law : The cop in the backseat. *The standford Center for Legal Informatics hosted the third annual FutureLaw 2015 conference*, 2015. <http://logic.stanford.edu/complaw/complaw.html>.
- E Grave, T Mikolov, A Joulin, and P Bojanowski. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 427–431, 2017.
- Ralph Grishman and Beth Sundheim. Message understanding conference-6 : A brief history. In *COLING 1996 Volume 1 : The 16th International Conference on Computational Linguistics*, volume 1, 1996.
- Daniel Hanisch, Katrin Fundel, et al. ProMiner : rule-based protein and gene entity recognition. *BMC bioinformatics*, 6(1) :14, 2005.
- Sébastien Harispe, Sylvie Ranwez, Stefan Janaqi, and Jacky Montmain. The semantic measures library and toolkit : fast computation of semantic similarity and relatedness using biomedical ontologies. *Bioinformatics*, 30(5) : 740–742, 2013.
- Bruno Jeandidier and Jean-Claude Ray. Pensions alimentaires pour enfants lors du divorce - [Les juges appliquent-ils implicitement un calcul fondé sur le coût de l’enfant ?]. *Revue des politiques sociales et familiales*, 84(1) : 5–18, 2006. URL https://www.persee.fr/doc/caf_1149-1590_2006_

num_84_1_2207. https://www.persee.fr/doc/caf_1149-1590_2006_num_84_1_2207.

K. Sparck Jones, Steve Walker, and Stephen E. Robertson. A probabilistic model of information retrieval : development and comparative experiments. *Information processing & management*, 36(6) :809–840, 2000.

Daniel Martin Katz, Michael James Bommarito, and Josh Blackman. Predicting the behavior of the supreme court of the united states : A general approach. *Available at SSRN 2463244*, 2014.

Daniel Martin Katz, Michael J Bommarito II, and Josh Blackman. A general approach for predicting the behavior of the supreme court of the united states. *PloS one*, 12(4) :e0174698, 2017.

Leonard Kaufman and Peter J. Rousseeuw. Clustering by means of medoids. In Yadolah Dodge, editor, *Statistical Data Analysis Based on the L1-Norm*, page 405–416. North Holland/Elsevier, 1987. URL https://www.researchgate.net/profile/Peter_Rousseeuw/publication/243777819_Clustering_by_Means_of_Medoids/links/00b7d531493fad342c000000.pdf. Amsterdam.

Jin-Dong Kim, Tomoko Ohta, Yoshimasa Tsuruoka, Yuka Tateisi, and Nigel Collier. Introduction to the bio-entity recognition task at JNLPBA. In *Proceedings of the international joint workshop on natural language processing in biomedicine and its applications*, pages 70–75. Association for Computational Linguistics, 2004.

Fredrick Edward Kitoogo and Venansius Baryamureeba. A methodology for feature selection in named entity recognition. *Strengthening the Role of ICT in Development*, page 88, 2007.

Roman Klinger and Christoph M Friedrich. Feature subset selection in condi-

- tional random fields for named entity recognition. In *Proceedings of the International Conference RANLP-2009*, pages 185–191, 2009.
- Michal Konkol and Miloslav Konopík. Segment representations in named entity recognition. In *International Conference on Text, Speech, and Dialogue*, pages 61–70. Springer, 2015.
- Vincent Kríž, Barbora Hladká, et al. Statistical recognition of references in czech court decisions. In Alexander Gelbukh, Félix Castro Espinoza, and Sofía N. Galicia-Haro, editors, *Human-Inspired Computing and Its Applications : 13th Mexican International Conference on Artificial Intelligence, MICA I 2014, Tuxtla Gutiérrez, Mexico, November 16-22, 2014. Proceedings, Part I*, pages 51–61, Cham, 2014. Springer International Publishing. ISBN 978-3-319-13647-9. doi : 10.1007/978-3-319-13647-9_6. URL http://dx.doi.org/10.1007/978-3-319-13647-9_6.
- Sushanta Kumar, P Krishna Reddy, V Balakista Reddy, and Aditya Singh. Similarity analysis of legal judgments. In *Proceedings of Compute 2011 - Fourth Annual ACM Bangalore Conference*, page 17. ACM, 2011.
- Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. From word embeddings to document distances. In *International Conference on Machine Learning*, pages 957–966, 2015.
- John Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields : probabilistic models for segmenting and labeling sequence data. *International Conference on Machine Learning*, 2001. URL http://repository.upenn.edu/cgi/viewcontent.cgi?article=1162&context=cis_papers.
- Guillaume Lample, Miguel Ballesteros, et al. Neural architectures for named entity recognition. *arXiv preprint*, 2016. arXiv :1603.01360.

Man Lan, Chew Lim Tan, Jian Su, and Yue Lu. Supervised and traditional term weighting methods for automatic text categorization. *IEEE transactions on pattern analysis and machine intelligence*, 31(4) :721–735, 2009.

Eric Langlais and Nathalie Chappe. *Analyses économiques du droit : principes, méthodes, résultats*, chapter 4. Analyse économique de la résolution des litiges. Editions de Boeck Université, 2009. URL <http://www.sze.hu/~kallay/letolt/2011/februar/the%20doctrine%20of%20judicial%20precedent.pdf>.

Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196, 2014.

Philip Leith. The rise and fall of the legal expert system. *European Journal of Law and Technology*, 1(1) :179–201, 2010. <http://ejlt.org/article/view/14/1>.

Jianqiang Li, Shenhe Zhao, Jijiang Yang, Zhisheng Huang, Bo Liu, Shi Chen, Hui Pan, and Qing Wang. Wcp-rnn : a novel rnn-based approach for bioer in chinese emrs. *The Journal of Supercomputing*, pages 1–18, 2018.

Yaoyong Li, Hugo Zaragoza, Ralf Herbrich, John Shawe-Taylor, and Jaz Kandola. The perceptron algorithm with uneven margins. In *ICML*, volume 2, pages 379–386, 2002.

Dong C. Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical programming*, 45(1) :503–528, 1989.

Huan Liu and Hiroshi Motoda. *Feature selection for knowledge discovery and data mining*, volume 454. Springer Science & Business Media, 2012.

Jingjing Liu, Panupong Pasupat, Scott Cyphers, and Jim Glass. Asgard : A portable architecture for multilingual dialogue systems. In *Acoustics*,

- Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8386–8390. IEEE, 2013.
- Karl Nickerson Llewellyn. *Jurisprudence : Realism in Theory and Practice*. The University of Chicago Press, 1962.
- Nathaniel Love and Michael Genesereth. Computational law. In *Proceedings of the 10th international conference on Artificial intelligence and law*, pages 205–209. ACM, 2005.
- Xuezhe Ma and Eduard Hovy. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of ACL*, 2016.
- Yinglong Ma, Peng Zhang, and Jiangang Ma. An Efficient Approach to Learning Chinese Judgment Document Similarity Based on Knowledge Summarization. *arXiv preprint*, page 23, 2018. arXiv :1808.01843.
- Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. Scoring, term weighting and the vector space model. In *Introduction to information retrieval*, chapter 6, pages 109–133. Cambridge university press, Cambridge, 2008.
- Justin Martineau, Tim Finin, et al. Delta tfidf : An improved feature space for sentiment analysis. *Icwsn*, 9 :106, 2009.
- Andrew Kachites McCallum. MALLET : A Machine Learning for Language Toolkit, 2012. <http://mallet.cs.umass.edu/>.
- Masha Medvedeva, Michel Vols, and Martijn Wieling. Judicial decisions of the european court of human rights : Looking into the crystal ball. In *Proceedings of the Conference on Empirical Legal Studies*, 2018.
- Andrei Mikheev, Marc Moens, and Claire Grover. Named entity recognition without gazetteers. In *Proceedings of the ninth conference on European*

chapter of the Association for Computational Linguistics, pages 1–8. Association for Computational Linguistics, 1999.

Raquel Mochales and Marie-Francine Moens. Study on the structure of argumentation in case law. In *Proceedings of the 2008 Conference on Legal Knowledge and Information Systems*, pages 11–20, 2008.

Marie-Francine Moens. What information retrieval can learn from case-based reasoning. In *Legal Knowledge and Information Systems*, pages 83–91. Jurix 2002 : The Fifteenth Annual Conference, T.J.M. Bench-Capon, A. Daskalopulu and R.G.F. Winkels (eds.), 2002. URL https://www.researchgate.net/profile/Marie-Francine_Moens/publication/250777261_What_Information_Retrieval_Can_Learn_from_Case-Based_Reasoning/links/0a85e52ecd015d41d4000000.pdf. https://www.researchgate.net/profile/Marie-Francine_Moens/publication/250777261_What_Information_Retrieval_Can_Learn_from_Case-Based_Reasoning/links/0a85e52ecd015d41d4000000.pdf.

Marie-Francine Moens, Erik Boiy, Raquel Mochales Palau, and Chris Reed. Automatic detection of arguments in legal texts. In *Proceedings of the 11th international conference on Artificial intelligence and law*, pages 225–230. ACM, 2007.

Stéphane Mussard and Fattouma Souissi-Benrejab. Gini-pls regressions. *Journal of Quantitative Economics*, pages 1–36, April 2018. ISSN 2364-1045. doi : 10.1007/s40953-018-0132-9. URL <https://doi.org/10.1007/s40953-018-0132-9>.

David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1) :3–26, 2007.

Akhil M. Nair and Rupali Sunil Wagh. 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, 2018. ISSN 0974-3154.
- Ramesh Nallapati, Mihai Surdeanu, and Christopher Manning. Blind domain transfer for named entity recognition using generative latent topic models. In *Proceedings of the NIPS 2010 Workshop on Transfer Learning Via Rich Generative Models*, pages 281–289, 2010.
- Adeline Nazarenko and Adam Wyner. Legal NLP Introduction. *Traitement automatique de la langue juridique / Legal Natural Language Processing - Revue TAL*, 58(2) :7–19, 2017. <http://www.atala.org/content/introduction-0>.
- Hwee Tou Ng, Wei Boon Goh, and Kok Leong Low. Feature selection, perceptron learning, and a usability case study for text categorization. In *ACM SIGIR Forum*, volume 31, pages 67–73. ACM, 1997.
- Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. Joint event extraction via recurrent neural networks. In *HLT-NAACL*, pages 300–309, 2016.
- Kamal Nigam, John Lafferty, and Andrew McCallum. Using maximum entropy for text classification. In *IJCAI-99 workshop on machine learning for information filtering*, volume 1, pages 61–67, 1999.
- Rasmus Berg Palm, Dirk Hovy, Florian Laws, and Ole Winther. End-to-end information extraction without token-level supervision. In *Proceedings of the Workshop on Speech-Centric Natural Language Processing*, 2017.
- David D Palmer and David S Day. A statistical profile of the named entity task. In *Proceedings of the fifth conference on Applied natural language processing*, pages 190–193. Association for Computational Linguistics, 1997.
- Georgios Paltoglou and Mike Thelwall. A study of information retrieval weighting schemes for sentiment analysis. In *Proceedings of the 48th annual*

- meeting of the association for computational linguistics*, pages 1386–1395. Association for Computational Linguistics, 2010.
- Hae-Sang Park and Chi-Hyuck Jun. A simple and fast algorithm for K-medoids clustering. *Expert Systems with Applications*, 36(2) :3336–3341, 2009.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove : Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.
- Caroline Persson. *Machine Learning for Tagging of Biomedical Literature*, 2012. Closing project report, Technical University of Denmark, DTU Informatics.
- Joe Polifroni and François Mairesse. Using latent topic features for named entity extraction in search queries. In *INTERSPEECH*, pages 2129–2132, 2011.
- Patti J Price. Evaluation of spoken language systems : The atis domain. In *Speech and Natural Language : Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*, 1990.
- Pavel Pudil, Jana Novovičová, and Josef Kittler. Floating search methods in feature selection. *Pattern recognition letters*, 15(11) :1119–1125, 1994.
- Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2) :257–286, 1989.
- K Raghuveer. Legal documents clustering using latent dirichlet allocation. *IAES Int. J. Artif. Intell.*, 2(1) :34–37, 2012.

Baranidharan Raman and Thomas R Ioerger. Enhancing learning using feature and example selection. *Texas A&M University, College Station, TX, USA*, 2003.

Sebastian Raschka. Naive Bayes and Text Classification I : Introduction and Theory. *arXiv preprint arXiv :1410.5329*, 2014.

Irina Rish. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46. IBM New York, 2001.

Akhtar Sabzi, Yaghoub Farjami, and Morteza ZiHayat. An improved fuzzy k-medoids clustering algorithm with optimized number of clusters. In *Proceedings of the 11th International Conference on Hybrid Intelligent Systems (HIS)*, pages 206–210. IEEE, 2011.

Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5) :513–523, 1988.

Helmut Schmid. *TreeTagger - a part-of-speech tagger for many languages*, 1994. Available at <http://www.ims.uni-stuttgart.de/forschung/ressourcen/werkzeuge/treetagger.en.html>.

Hinrich Schütze, David A Hull, and Jan O Pedersen. A comparison of classifiers and document representations for the routing problem. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 229–237. ACM, 1995.

Rahul Sharnagat. Named entity recognition : A literature survey. Technical report, Center For Indian Language Technology, 2014. <https://pdfs.semanticscholar.org/83fd/67f0c9e8e909dc7b90025e64bde0385a9a3a.pdf>.

- Olga Shulayeva, Advaith Siddharthan, and Adam Wyner. Recognizing cited facts and principles in legal judgements. *Artificial Intelligence and Law*, 25(1) :107–126, 2017.
- Peter Siniakov. *GROPUS an Adaptive Rule-based Algorithm for Information Extraction*. PhD thesis, Freie Universität Berlin, 2008.
- Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1) :11–21, 1972.
- Octavia-Maria Şulea, Marcos Zampieri, Shervin Malmasi, Mihaela Vela, Liviu P. Dinu, and Josef van Genabith. Exploring the Use of Text Classification in the Legal Domain. In *Proceedings of 2nd Workshop on Automated Semantic Analysis of Information in Legal Texts*, page 5, London, United Kingdom, June 2017a. ASAIL’2017.
- Octavia-Maria Şulea, Marcos Zampieri, Mihaela Vela, and Josef van Genabith. Predicting the Law Area and Decisions of French Supreme Court Cases. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 716–722, 2017b.
- Michel Tenenhaus. La regression logistique PLS. In Jean-Jacques Droesbeke, Michel Lejeune, and Gilbert Saporta, editors, *Modèles statistiques pour données qualitatives*, chapter 12, pages 263–276. Editions Technip, 2005.
- Dhaval Thakker, Taha Osman, and Phil Lakin. *Gate jape grammar tutorial*, 2009. URL <https://gate.ac.uk/sale/thakker-jape-tutorial/GATE%20JAPE%20manual.pdf>.
- D. Thenmozhi, Kawshik Kannan, and Chandrabose Aravindan. A Text Similarity Approach for Precedence Retrieval from Legal Documents. In *Proceedings of Forum for Information Retrieval Evaluation - FIRE (Working Notes)*, pages 90–91, 2017.

- Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the conll-2003 shared task : Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, pages 142–147, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. doi : 10.3115/1119176.1119195. URL <https://doi.org/10.3115/1119176.1119195>.
- Vitalius Tumonis. Legal realism & judicial decision-making. *Jurisprudencija*, 19(4), 2012.
- S Sidney Ulmer. Quantitative analysis of judicial processes : Some practical and theoretical applications. *Law and Contemporary Problems*, 28(1) : 164–184, 1963. URL https://www.jstor.org/stable/1190728?seq=1#page_scan_tab_contents.
- Vladimir N Vapnik. The nature of statistical learning. *Theory*, 1995.
- Anthony J Viera, Joanne M Garrett, et al. Understanding interobserver agreement : the kappa statistic. *Fam Med*, 37(5) :360–363, 2005.
- Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In *Advances in Neural Information Processing Systems*, pages 2692–2700, 2015.
- Andrew James Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory*, 13(2) :260–269, 1967.
- Hanna M. Wallach. Conditional Random Fields : An Introduction. Technical report, University of Pennsylvania Department of Computer and Information Science, 2004. https://repository.upenn.edu/cgi/viewcontent.cgi?article=1011&context=cis_reports.
- Bernhard Walzl, Florian Matthes, Tobias Walzl, and Thomas Grass. LEXIA - A Data Science Environment for Semantic Analysis of German Legal Texts.

In *IRIS : Internationales Rechtsinformatik Symposium*, 2016. Salzburg, Austria.

Bernhard Walzl, Georg Bonczek, Elena Scepankova, Jörg Landthaler, and Florian Matthes. Predicting the outcome of appeal decisions in germany’s tax law. In *International Conference on Electronic Participation*, pages 89–99. Springer, 2017a.

BERNHARD Walzl, Jörg Landthaler, Elena Scepankova, FLORIAN Matthes, THOMAS Geiger, CHRISTOPH Stocker, and CHRISTIAN Schneider. Automated extraction of semantic information from German legal documents. In *IRIS : Internationales Rechtsinformatik Symposium. Association for Computational Linguistics*, 2017b.

Bernhard Walzl, Georg Bonczek, and Florian Matthes. Rule-based information extraction : Advantages, limitations, and perspectives. *Jusletter IT*, 2018.

Sida Wang and Christopher D Manning. Baselines and bigrams : Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics : Short Papers-Volume 2*, pages 90–94. Association for Computational Linguistics, 2012.

Lloyd R Welch. Hidden Markov models and the Baum-Welch algorithm. *IEEE Information Theory Society Newsletter*, 53(4) :10–13, 2003.

Ian H Witten, Zane Bray, et al. Using language models for generic entity extraction. In *Proceedings of the ICML Workshop on Text Mining*, 1999.

Haibing Wu, Xiaodong Gu, and Yiwei Gu. Balancing between over-weighting and under-weighting in supervised term weighting. *Information Processing & Management*, 53(2) :547–557, 2017.

- Harry Wu and Gerard Salton. A comparison of search term weighting : term relevance vs. inverse document frequency. In *ACM SIGIR Forum*, volume 16, pages 30–39. ACM, 1981.
- Adam Wyner and Wim Peters. Lexical Semantics and Expert Legal Knowledge towards the Identification of Legal Case Factors. In *JURIX*, volume 10, pages 127–136, 2010. <http://wyner.info/research/Papers/WynerPetersCaseFactorsJURIX2010Final.pdf>.
- Adam Wyner, Raquel Mochales-Palau, Marie-Francine Moens, and David Milward. Approaches to text mining arguments from legal cases. In *Semantic Processing of Legal Texts : where the Language of Law Meets the Law of Language*, pages 60–79. Springer-Verlag, Berlin, Heidelberg, 2010.
- Adam Z Wyner. 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, 2010. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.232.1531&rep=rep1&type=pdf#page=9>.
- Richard Xiao. Corpus creation. In Nitin Indurkha and Fred J . Damerau, editors, *Handbook of Natural Language Processing*, chapter 7, page 146–165. Chapman and Hall, second edition, 2010.
- Vikas Yadav and Steven Bethard. A survey on recent advances in named entity recognition from deep learning models. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2145–2158, 2018.
- Bishan Yang and Tom Mitchell. Joint Extraction of Events and Entities within a Document Context. In *Proceedings of NAACL-HLT*, pages 289–299, 2016.

Yiming Yang and Jan O Pedersen. A comparative study on feature selection in text categorization. In *Icml*, volume 97, pages 412–420, 1997.

Xiaojin Zhu. *Conditional Random Fields*, 2010. URL <http://pages.cs.wisc.edu/~jerryzhu/cs769/CRF.pdf>. CS769 Spring 2010 Advanced Natural Language Processing. <http://pages.cs.wisc.edu/~jerryzhu/cs769/CRF.pdf>.