THÈSE POUR OBTENIR LE GRADE DE DOCTEUR
ÉCOLE NATIONALE SUPÉRIEURE DES MINES D'ALÈS (IMT MINES ALÈS)

En Informatique

École doctorale Risques et société

Centre de recherche LGI2P de l'IMT Mines Ales
Equipe d'accueil CHROME de l'Université de Nîmes

# Méthodes D'Analyse Sémantique De Corpus De Décisions Jurisprudentielles

Présentée par Gildas TAGNY NGOMPÉ
Le xx Janvier 2020

Sous la direction de Stéphane MUSSARD
Et Jacky MONTMAIN

Devant le jury composé de

| | |
|---|---|
| Sandra BRINGAY, Professeur, Université de Montpellier | Rapporteur |
| Boughanem MOHAND, Professeur, Université Toulouse III Paul Sabatier | Rapporteur |
| Françoise SEYTE, Maître de Conférences (HDR), Université de Montpellier | Examinateur |
| Fabrice MUHLENBACH, Maître de Conférences, Université Jean Monnet de Saint-Étienne | Examinateur |
| Stéphane MUSSARD, Professeur, Université de Nîmes | Directeur de thèse |
| Jacky MONTMAIN, Professeur, IMT Mines Alès | Co-directeur de thèse |
| Guillaume ZAMBRANO, Maître de Conférences, Université de Nîmes | Encadrant de proximité |
| Sébastien HARISPE, Maître Assistant, IMT Mines Alès | Encadrant de proximité |

Collège Doctoral Languedoc-Roussillon

IMT Mines Alès
École Mines-Télécom

# Remerciements

Ces années de thèse n'ont pas toujours été faciles. Tout au long de mes travaux de recherche et de rédaction, j'ai néanmoins reçu un précieux soutient administratif, financier, et moral de plusieurs organismes et personnes. Ce mémoire est l'occasion pour moi de tous les remercier.

En premier, je remercie mes directeurs et encadrants de thèse pour leur confiance, leur disponibilité, leurs contributions, leur effort et leur patience. Je leur prie de m'excuser pour en avoir très souvent abusé. Merci Jacky et Stéphane pour avoir accepté de diriger mes travaux. Ça a été un honneur pour moi de vous avoir eu comme directeurs de thèse. Merci Guillaume pour avoir initié ce sujet très passionnant sur lequel j'espère avoir l'occasion de continuer à contribuer. Merci aussi pour les nombreuses journées de travail que nous avons passées ensemble. Elles m'ont été indispensables pour mieux comprendre les problèmes métiers des juristes, obtenir des données annotées, et valider mes résultats. Merci à Sébastien pour les nombreuses sessions de travail durant lesquelles nous avons formulé les problèmes métiers en problèmes informatiques, réfléchi sur des approches, et discuté des résultats expérimentaux.

Je remercie Madame Sandra BRINGAY, Professeur de l'Université de Montpellier, et Monsieur Boughanem MOHAND, Professeur de l'Université Toulouse III Paul Sabatier, de m'honorer en acceptant d'être les rapporteurs de ma thèse. Je remercie aussi Madame Françoise Seyte, Maître de Conférences (HDR) de l'Université de Montpellier, et Monsieur Fabrice MUHLENBACH, Maître de Conférences de l'Université Jean Monnet de Saint-Étienne, de me faire l'honneur d'être examinateurs de ma thèse.

Je suis reconnaissant envers l'IMT Mines Alès pour avoir financer mes travaux. Je remercie le LGI2P et l'équipe CHROME pour m'avoir accueilli. Ces remerciements s'adressent en particulier à Yannick VIMONT, ancien directeur du LGI2P, à Jacky MONTMAIN, actuel directeur du LGI2P, et à Benoît Roig, Président de l'université de Nîmes et ancien directeur de l'équipe d'accueil CHROME. Je remercie aussi Valérie Roman, Claude Badiou, Édith Teychene, et Corinne VINCENT, pour leur aide précieuse pour les démarches administratives.

Je remercie les chercheurs et doctorants du LGI2P et de l'équipe Chrome

pour leur gentillesse et les bons moments que j'ai partagés avec eux. C'était le quotidien avec ma collègue de bureau Valentina. C'était les séminaires organisés par Christelle. C'était des discussions scientifiques avec mes collègues doctorants Pierre-Antoine, Diadie, Jean-Christophe, et bien d'autres. C'était du foot, du basket, et du ski avec Michel, Abdelhak, Hassan, Yannick, Sébastien, Kouadio, Brahim, Baptiste, Blazo, Alexandre, Diadie, Mouhamadou, Frank, Behrang, et bien d'autres. C'était les trajets quotidiens entre Nîmes et Alès avec Alexandre, Frank, Christelle, Cécile et les autres. C'était des week-ends avec Clément et Julien.

Je remercie aussi ma famille pour m'avoir soutenu et pour avoir supporté ma longue absence auprès d'eux.

Je remercie enfin l'INRA, le CIRAD, et la société ESII pour m'avoir accordé du temps pour travailler sur mon mémoire malgré les contrats qui nous liaient.

# Résumé

**Titre :** MÉTHODES D'ANALYSE SÉMANTIQUE DE CORPUS DE DÉCISIONS JURISPRUDENTIELLES

Une jurisprudence est un corpus de décisions judiciaires représentant la manière dont sont interprétées les lois pour résoudre un contentieux. Elle est indispensable pour les juristes qui l'analysent pour comprendre et anticiper la prise de décision des juges. Son analyse exhaustive est difficile manuellement du fait de son immense volume et de la non-structuration des documents. L'estimation du risque judiciaire par des particuliers est ainsi impossible car ils sont en outre confrontés à la complexité du système et du langage judiciaire. L'automatisation permet de retrouver exhaustivement des connaissances pertinentes pour structurer la jurisprudence à des fins d'analyses descriptives et prédictives. Afin de rendre la compréhension de la jurisprudence exhaustive et plus accessible, cette thèse aborde l'automatisation de tâches d'importante d'analyse métier de la jurisprudence. En premier, est étudiée l'application de modèles probabilistes d'étiquetage de séquences pour la détection des sections, d'entités juridiques, et de citations de lois. Ensuite, l'extraction des demandes des parties est étudiée. L'approche proposée pour la reconnaissance des quanta demandés et accordés exploite la proximité entre les sommes d'argent et des termes-clés appris automatiquement. Nous montrons par ailleurs que le sens du résultat des juges est identifiable soit à partir de termes-clés prédéfinis soit par classification de documents. Enfin, les situations ou circonstances factuelles où est formulée une catégorie de demandes sont découvertes par regroupement non supervisé des décisions. A cet effet, une méthode d'apprentissage d'une distance de similarité est proposée et comparée à des distances établies. Cette thèse discute des résultats empiriques obtenus sur des données réelles annotées manuellement par un expert. Le mémoire est clôturé par une démonstration d'applications à l'analyse descriptive d'un grand corpus de décisions judiciaires françaises.

**Mots-clés :** analyse de données textuelles, décisions jurisprudentielles, extraction d'information, classification de textes, regroupement non-supervisé.

# Abstract

**Title :** METHODS FOR THE SEMANTIC ANALYSIS OF CORPORA OF JURISPRUDENTIAL DECISIONS

A case law is a corpus of judicial decisions representing the way in which laws are interpreted to resolve a dispute. It is essential for lawyers who analyze it to understand and anticipate the decision-making of judges. Its exhaustive analysis is difficult manually because of its huge size and the non-structuring state of the documents. The estimation of the judicial risk by individuals is thus impossible because they are also confronted with the complexity of the judicial system and language. Automation can enable an exhaustive extraction of relevant knowledge for structuring case law for descriptive and predictive analysis. In order to make the comprehension of the case-law exhaustive and more accessible, this thesis deals with the automation of important tasks of business analysis of jurisprudence. First, the application of probabilistic graphical sequence labeling models for the detection of sections, legal named entities, and legal rules citations in decisions is investigated. Then, the extraction of the requests of the parties is studied. The proposed approach to the recognition of quanta requested and granted exploits the proximity between sums of money and automatically learned key-phrases. We also show that the meaning of the judges' result is identifiable either from predefined keywords or by a binary classification of documents. Lastly, situations or factual circumstances in which a category of claims is formulated are discovered by clustering decisions. For this purpose, a method of learning a similarity distance is proposed and compared with established distances. This thesis discusses the empirical results obtained on real data annotated manually by an expert. The thesis is closed by a demonstration of some applications to the descriptive analysis of a large corpus of French judicial decisions.

**Keywords :** textual data analysis, case law decisions, information extraction, text classification, document clustering.

# Table des matières

# Liste des figures

# Liste des tableaux

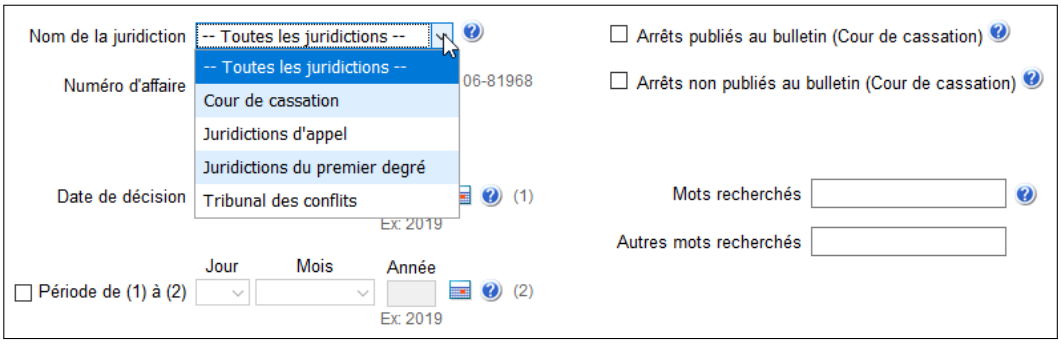
# Introduction générale

## i Contexte et motivations

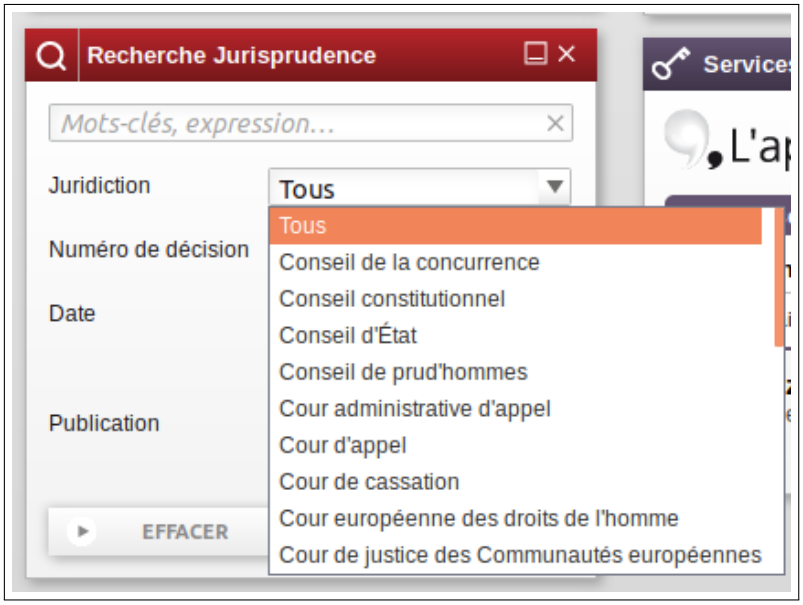
Une décision judiciaire peut être définie soit comme le résultat rendu par les juges à l'issue d'un procès, soit comme un document décrivant une affaire judiciaire. Un tel document rapporte, notamment, les faits, les procédures judiciaires antérieures, le verdict des juges, et les explications associées. Dans cette thèse, nous désignons par « décision » le document, et par « résultat » une conclusion ou réponse des juges. Un exemple (annoté) de décision est donné à l'annexe de la page 169. Une jurisprudence est un ensemble de décisions rendues par les tribunaux. Elle représente la manière dont ces derniers interprètent les lois pour résoudre un problème juridique donné (type de contentieux). Les juristes doivent alors collecter des décisions traitant de situations similaires, les sélectionner, et les analyser afin de mener, par exemple, des recherches empiriques en droit [Ancel, 2003; Jeandidier & 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 faire accepter leurs demandes et faire rejeter celles de leurs adversaires. Cette activité de collecte et d'analyse, centrale pour de nombreux métiers du droit, est généralement effectuée manuellement. Elle est par conséquent sujette à plusieurs difficultés liées à l'accès et à l'exhaustivité des documents traités même lors de l'étude d'une question spécifique. Il faut notamment souligner ici que les documents sont dispersés dans les nombreux tribunaux. Les procédures administratives ne facilitent pas toujours leur accès à cause des questions d'éthique soulevées [Muhlenbach & Sayn, 2019] comme entre autres la nécessité de préserver la confidentialité des parties. En effet, les décisions n'étant pas « anonymisées » la plupart du temps, elles restent alors inaccessibles aux juristes qui en font la demande. Un certain nombre de documents sont néanmoins accessibles sur internet grâce à des sites de publication de données ouvertes gouvernementales [1]. Ces sites publient régulièrement des décisions

1. Données ouvertes gouvernementales : data.gouv.fr en France, judiciary.uk en Grande-Bretagne, scotusblog.com aux Etats-Unis, et scc-csc.ca au Canada.

récemment prononcées.




(a) Formulaire de Légifrance.




(b) Formulaire de Dalloz.

Figure 1 – Exemples de critères des moteurs de recherche juridique.

Il existe aussi des moteurs de recherche juridiques qui permettent de retrouver des décisions intéressantes. Cependant, qu'ils soient payants (LexisNexis [2], Dalloz [3], Lamyline [4],...) ou gratuits (CanLII [5], Légifrance [6], ...), les critères de recherche offerts par leurs moteurs de recherche limitent grandement la pertinence des résultats pouvant être obtenus. En effet, il ne

2. `https://www.lexisnexis.fr/`
3. `http://www.dalloz.fr`
4. `http://lamyline.lamy.fr`
5. `https://www.canlii.org`
6. `https://www.legifrance.gouv.fr`

s'agit en général que de combinaisons de mots-clés et autres méta-données (date, type de juridiction, ...), ou d'expressions régulières, comme l'illustre la Figure 1 Page 2. La manipulation de tels critères est difficile pour constituer des échantillons pertinents suivant une sémantique souhaitée tels que l'ensemble des décisions traitant d'une catégorie de demande ou d'une circonstance factuelle donnée.

| **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.

Plus de 4 millions de décisions sont prononcées en France chaque année d'après les chiffres du ministère français de la justice (Tableau 1 Page 3). Dans ce contexte, l'analyse manuelle ne peut être limitée qu'à une infime proportion de documents disponibles. En effet, au regard de la croissance rapide du nombre de décisions, même une étude sur une question très précise nécessite la constitution d'un large corpus de décisions pertinentes. Par ailleurs, il peut s'avérer très pénible de les lire pour en identifier les données d'intérêt. Les documents sont très souvent longs et complexes dans leur style de rédaction. Par exemple, Certaines phrases comprennent plusieurs clauses discutant d'aspects différents (Figure 2 Page 4). On y retrouve aussi des références à des jugements antérieurs (Figure 3 Page 4).

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. Une telle automatisation ferait gagner du temps aux juristes lors de tâches d'analyse métier préalables à leur raisonnement d'experts, tout en leur fournissant une vue exhaustive de la jurisprudence. D'autre part, Cretin [2014] fait remarquer que la justice est complexe dans son organisation (Figure 4 Page 5) et son fonctionnement, et que son langage est peu compréhensible. Il est donc presque impossible pour les profanes en droit d'estimer leurs droits et le risque judiciaire qu'ils encourent dans leur quotidien sans consulter un initié du droit. L'exigence pour le profane étant l'exacte pertinence des ressources, leur accessibilité, et l'intuitivité du processus de leur exploitation [Nazarenko & Wyner, 2017], l'automatisation de l'analyse de la jurisprudence pourrait ainsi améliorer l'accessibilité du droit dans d'indé-

```
Exposant subir un trouble anormal de voisinage pour être privée d'une vue sur la
mer dont elle disposait auparavant, ce en raison de l'absence de taille de haies
implantées à proximité de son jardin privatif, elle a attrait devant le juge des
référés du tribunal de grande instance de Marseille, le syndicat des
copropriétaires LES CATALANS (ci-après désigné : le syndicat des copropriétaires)
, et son syndic recherché personnellement, le Cabinet L., à l'effet, au visa de
l'article 809 du code de procédure civile d'obtenir leur condamnation sous
astreinte de 200 euros par jour de retard à tailler les haies qui bouchent sa
vue et la condamnation personnelle du Cabinet L. à lui régler une provision de
2.000 euros à valoir sur dommages et intérêts, outre 1.500 euros sur le
fondement des dispositions de l'article 700 du code de procédure civile.
```

***Source** : extrait de la décision R.G. 15/10226 de la Cour d'Appel d'Aix-en-Provence du 2 Juin 2016*

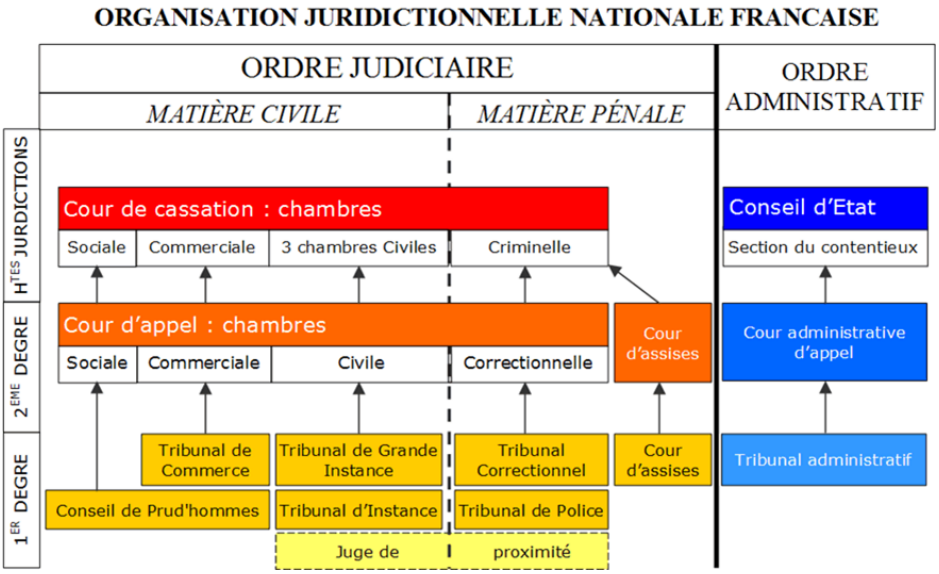
Figure 2 – Exemple de phrases composée de plusieurs clauses dont une demande de condamnation sous astreinte, une autre de dommages et intérêts pour trouble anormal de voisinage, et une dernière de dommages et intérêts sur l'article 700 du code de procédure civile.

```
  Vu le jugement du tribunal de grande instance de Versailles du 5 décembre 2013
  qui a :
  - rejeté la demande de démolition de la construction litigieuse,

  ...
  SUR CE LA COUR,

  ...
  PAR CES MOTIFS,
  ...

  Confirme le jugement en toutes ses dispositions à l'exception de celle relative
  au montant des dommages et intérêts ...
```

***Source** : extrait de la décision R.G. 14/01640 de la Cour d'Appel de Versaille du 7 Avril 2016*

Figure 3 – Exemple de référence à un jugement antérieur dans une décision d'appel.

nombrables situations. 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 & Chappe, 2009]. Le traitement automatique de la jurisprudence constituerait alors une aide précieuse non seulement pour les professionnels du droit, mais aussi pour les particuliers et entreprises tous soucieux de voir l'issue de leur affaire leur être favorable.




*Source : https://fr.wikipedia.org/wiki/Organisation_juridictionnelle_(France)*

Figure 4 – Organisation de la justice en France.

# ii Objectifs

Ce mémoire propose des approches pour automatiser l'extraction de connaissances judiciaires à partir des décisions françaises. Le but est de faciliter la structuration et l'analyse descriptive et prédictive de corpus de décisions de justice en adressant les difficultés de l'approche traditionnelle d'analyse de contentieux. L'étude de la jurisprudence pour un contentieux donnée consiste à [Ancel, 2003] :

1. **Choisir un échantillon représentatif** : Des décisions sont collectionnées suivant des contraintes définies : période précise, couverture géographique, types d'affaires, etc.
2. **Sélectionner les décisions** : élimination des décisions qui ne correspondent pas au type de demande d'intérêt.
3. **Élaborer la grille d'analyse** : Un modèle de grille est créé et permet d'enregistrer les informations potentiellement importantes. Chaque ligne de la grille correspond à une demande, et les colonnes font référence aux 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 ; ce n'est généralement qu'au cours de la lecture des décisions que l'on distingue les informations

pertinentes pour l'étude.

4. **L'analyse des décisions et l'interprétation des informations** : Les informations retrouvées dans les décisions sont saisies dans la grille, et des calculs statistiques sont effectués par la suite.

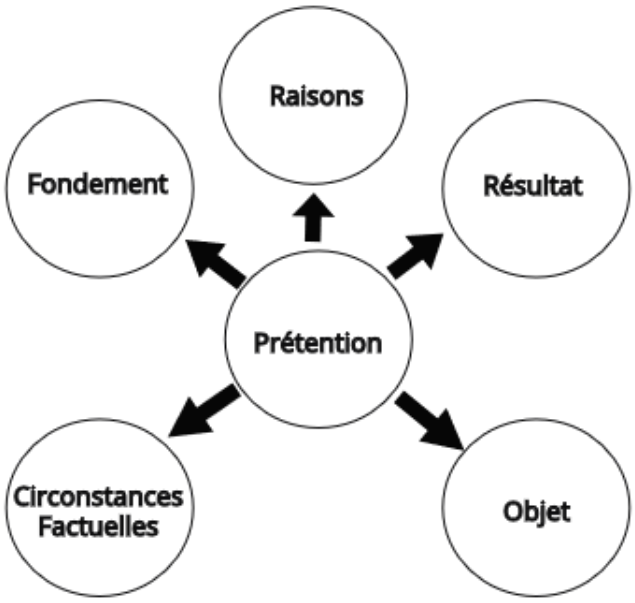
Ancel [2003] évoque principalement le problème de la différence entre l'état capté de la jurisprudence et son état présent. En effet, les longs délais de travail sont caractéristiques de ces études. L'étude de son équipe portait sur les décisions d'expulsion d'occupants sans droit ni titre. La saisie des informations à elle seule a duré 9 mois. De plus, il est difficile d'observer l'évolution des pratiques judiciaires dans le temps et leur différence entre les villes du fait de la faible taille de l'échantillon choisi. Par exemple, Jeandidier & Ray [2006] n'ont analysé que 399 dossiers d'affaires de pension alimentaire correspondant aux audiences s'étalant de fin 1999 à fin 2000 d'un seul tribunal de grande instance. L'équipe de Ancel [2003] n'a quant à elle analysé que 3865 décisions sélectionnées parmi 5656 décisions rendues du 1er juillet au 31 décembre 2001.

La problématique de notre étude est « **comment donner accès à l'analyse automatique de la sémantique d'un corpus jurisprudentiel pour comprendre la prise de décision des juges ?** ». La complexité de cette analyse s'explique notamment par l'interprétation subjective des règles juridiques, l'application non déterministe de la loi, et la technicité du langage judiciaire. Cette problématique intéresse des entreprises telles que LexisNexis [7] et Lexbase SA [8], et plusieurs startups telles que Predictice [9] et CASE LAW ANALYTICS [10]. Afin d'y répondre, nous nous intéressons aux concepts d'informations mentionnées dans les décisions, au centre desquels se trouvent les demandes des parties (prétentions) sur lesquelles portent les conclusions rendues. Ainsi, l'analyse sémantique d'un corpus jurisprudentiel vise l'identification de connaissances sur les demandes (Figure 5 Page 7).

Une demande peut être caractérisée par :

- l'objet qui a été demandé (par ex. dommages et intérêts) quantifié par un quantum ;
- le résultat associé qui est décrit par une polarité (« accepte » ou « rejette »), souvent lié à un quantum accordé, par exemple 5000 euros de dommages et intérêts ou 2 mois d'emprisonnement ;

7. `https://lexmachina.com`
8. `https://www.legalmetrics.fr`
9. `http://predictice.com`
10. `http://caselawanalytics.com`




*Sens des flèches : l'extrémité est l'information sur l'origine.*

Figure 5 – La demande au centre de la compréhension des décisions.

- le fondement ou la norme juridique qui est la règle ou l'ensemble de règles juridiques qui légitiment la prétention ou le résultat ;
- les circonstances factuelles définissent des types d'affaires et qui caractérisent les différentes situations dans lesquelles sont formulées les demandes d'une catégorie donnée ; Une catégorie de demande étant définie de manière unique par un objet et un fondement ;
- les divers arguments apportés par les parties pour justifier leurs requêtes (raisons des demandes) ;
- les motivations des solutions des juges (raisons des résultats).

Comme illustré par la Figure 6 Page 8, l'analyse sémantique identifie ou découvre différentes informations descriptives d'un corpus constitué par des décisions collectées à partir de divers moteurs de recherche juridique et des juridictions. Cette thèse s'inscrit dans un projet qui vise, entre autres, à automatiser la constitution d'une base de connaissances sur la jurisprudence française. Une telle base permettrait notamment de mener une grande variété de recherches et d'études expertes. Elle aurait aussi naturellement une importance certaine pour la définition de modèles prédictifs par exemple pour la prédiction des types de demandes à formuler et la prédiction de la solution des juges.

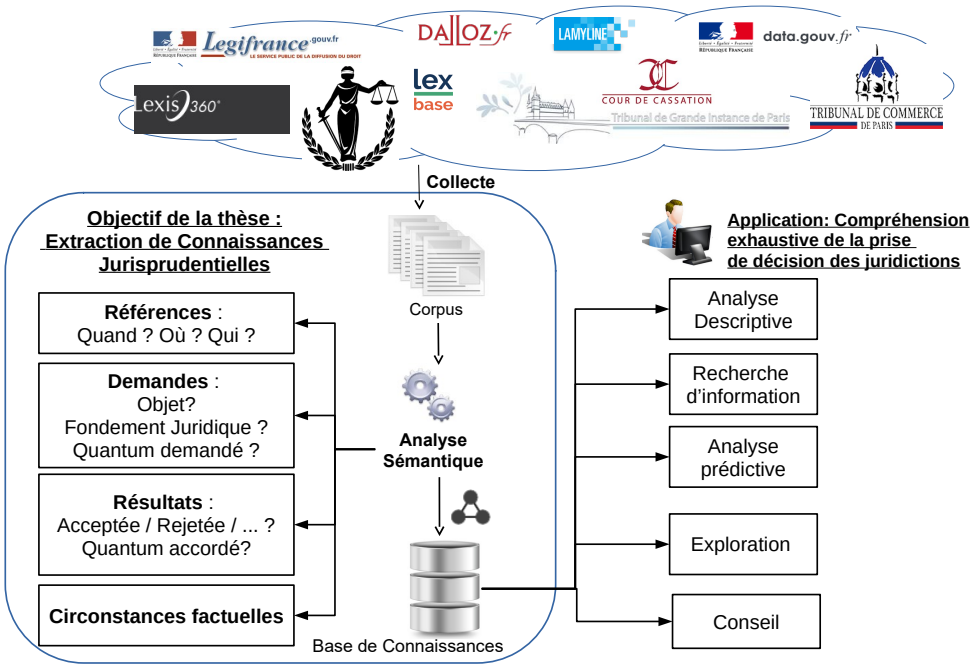



Figure 6 – Objectifs et exemples d'application de la thèse.

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

Il est 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 pour faciliter leur traitement. Les décisions de cours d'appel de justice civile sont les plus accessibles à partir des moteurs de recherche juridique (Lexis360, Dalloz, LamyLine, Lexbase, Legifrance, etc.) et de la grande base de données JuriCa alimentée d'environ 180k décisions civiles par an [Lamanda, 2010]. Cependant, l'accès 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). 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 Etudes et du Rapport [11] (SDER). L'accès est payant pour les professionnels et gratuit pour les universités et centres de recherche en partenariat avec le SDER. Lexbase dispose depuis une dizaine d'année d'une licence pour vendre les décisions de JuriCa [Emmanuel, 20 janvier 2010]. Légifrance, le

11. `https://www.courdecassation.fr/institution_1/composition_56/etudes_rapport_28.html`

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. 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. Il faut préciser que nos expérimentations se sont concentrées sur les décisions d'appel en justice civile, et ce choix a été motivé par le fait qu'elles sont les plus nombreuses sur internet.

Les décisions sont ainsi collectées à partir de diverses sources pouvant contenir des documents identiques. Il est donc important de les identifier de manière unique pour éviter des redondances de traitement. Par ailleurs, l'uniformisation préalable des documents permet d'analyser les décisions indépendamment de leur format d'origine.

## ii.b Extraction de connaissances

La difficulté d'extraire des connaissances des corpus jurisprudentielles découle de l'état non-structuré des documents et de la complexité 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. Ce mémoire aborde 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 méta-données de référence sont des segments de texte qu'on peut directement localiser dans le document. Elles sont donc semblables aux entités nommées dont la reconnaissance est une problématique intensivement étudiée en traitement automatique du langage naturel [Yadav & Bethard, 2018] dans plusieurs travaux et compétitions, aussi bien pour des entités communes [Tjong Kim Sang & De Meulder, 2003; Grishman & 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 & Sathyanarayana, 2015].

Le problème d'identification des demandes consiste à reconnaître, dans la décision analysée, l'objet, le fondement, le quantum demandé, le sens du résultat correspondant, et le quantum accordé de chaque prétention. La demande s'apparente donc aux entités structurées telles que les évènements LDC [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 au sein du corpus des décisions traitant 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 textes, de catégorisation non-supervisée (*clustering*), et de « modélisation thématique » (*topic modeling*). A l'issue du processus d'extraction, les connaissances extraites sont destinées à enrichir progressivement une base de connaissances. Cette dernière est amenée à faciliter les diverses analyses automatiques applicables aux décisions et demandes judiciaires.

### ii.c Application : 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 d'opinion entre les juges suivant leur localisation géographique (ville, département, région, etc.).

## iii Méthodologie

Les tâches sont définies par le métier. Comme illustrées précédemment (§ ii.b), les problématiques propres aux textes juridiques trouvent généralement des analogies avec les problèmes étudiés en analyse de données textuelles. Ainsi, les méthodes issues de 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 de l'analyse des décisions jurisprudentielles de ceux généralement

traités en analyse de textes. 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 ont été proposées et expérimentées sur les données réelles annotées manuellement par un expert juriste.

# iv Résultats

Les principaux résultats de cette thèse sont un ensemble de proposition d'approches d'extractions de connaissances et les discussions d'un grand nombre de résultats empiriques. Premièrement, le sectionnement a pour but d'organiser l'extraction des informations qui sont réparties dans des sections selon leur nature. Par exemple, les méta-données de références sont dans l'entête du documents, et les fondements utilisés sont cités dans le reste du document. L'application 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 pour le sectionnement et l'annotation d'entités juridiques en considérant plusieurs aspects de la conception des systèmes d'extraction d'information par étiquetage de séquence.

Par la suite, nous proposons une méthode d'extraction des demandes et des résultats en fonction des catégories présentes dans la décision. L'approche consiste en effet à identifier dans un premier temps les catégories (objet+fondement) présentes 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 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 ne contenir, dans une forte proportion, qu'une seule demande. A partir d'une représentation adéquate du contenu de la décision, il est possible d'identifier le sens du résultat de la seule demande d'intérêt par classification binaire de documents.

L'identification des circonstances factuelles, quant à elle, est modélisée comme une tâche de regroupement non supervisé de documents. Nous proposons dans ce cas une méthode d'apprentissage d'une distance entre textes, à l'aide d'un algorithme de régression. La métrique apprise est utili-

sée dans l'algorithme des « K-moyennes » (*k-means*) [Forgey, 1965] et celui des « K-medoïdes » (*k-medoids*) [Kaufman & Rousseeuw, 1987], et 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 d'analyse automatique de décisions de justice. Le chapitre 2 présente les architectures et modèles proposés pour la structuration des décisions et la reconnaissance des entités juridiques ; il discute notamment des différents résultats empiriques obtenus par application des modèles CRF et HMM. Ensuite, le chapitre 3 détaille le problème d'extraction des demandes, puis présente notre méthode et les résultats obtenus. Le chapitre 4 traite de l'identification du sens du résultat par classification directe des décisions, cela en comparant différents algorithmes de classification et de représentations des textes. Le chapitre 5 discute de l'usage de l'apprentissage proposé d'une distance qui est comparée à d'autres distances pour la découverte des circonstances factuelles. Enfin, le chapitre 6 présente des résultats de scénarios d'analyses descriptives pour illustrer l'exploitation potentielle de nos propositions sur un 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 aux décisions judiciaires. Une synthèse bibliographique plus technique sur les algorithmes de fouille de texte est détaillée dans les chapitres qui traitent, dans la suite, des méthodes que nous avons mises en œuvre. Plus précisément, suivant la structure du présent chapitre, il s'agit des chapitres 2 et 3 pour l'extraction d'information, du chapitre 4 pour la classification des documents, et du chapitre 5 pour la similarité entre documents.

## 1.1 Introduction

Les deux grands paradigmes de jugement se distinguent par l'importance qu'ils accordent aux règles juridiques [Tumonis, 2012]. D'une part, les adeptes du Formalisme Juridique, plus pertinent dans le droit civil, considèrent que toutes les considérations normatives ont été incorporées dans les lois par leurs auteurs. D'autre part, l'école du Réalisme Juridique, plus proche du « *Common Law* », permet un pouvoir discrétionnaire entre les jugements en raisonnant selon le cas. Les premières tentatives d'anticipation des comportements judiciaires s'appuyaient sur une formalisation des lois. Il en est né le « droit computationnel », qui est une sous-discipline de l'« informatique juridique [1] ». Il s'intéresse, en effet, au raisonnement juridique automatique axé sur une représentation sémantique riche et plus formelle de la loi, des régulations, et modalités de contrat [Love & Genesereth, 2005]. Il vise à réduire la taille et la complexité de la loi pour la rendre plus accessible. Plus précisément, le « droit computationnel » propose des systèmes répondant à différentes questions, comme « Quel montant de taxe dois-je payer cette année ? » (planification juridique), « Cette

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

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

Les premiers outils automatiques d'anticipation des décisions étaient généralement des systèmes experts juridiques. Ces derniers 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.

Deux paradigmes s'affirment comme de bonnes alternatives aux raisonnement à base de règles. La première est le raisonnement à base de cas qui 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 affecte à un nouveau cas en retrouvant la solution des cas les plus similaires dont sont connues les solutions [Berka, 2011]. Pour un problème de classification, l'algorithme des *k*-plus-proches-voisins est une méthode adéquate de raisonnement à base de cas [Poole & Mackworth, 2017]. L'algorithme du plus proche voisin (1-plus-proche-voisin) est utilisé notamment par Ashley & Brüninghaus [2009] pour identifier les types de faits (« Facteurs ») d'une affaire. Pour d'autres problèmes plus complexes, les différences entre les deux cas peuvent exiger une adaptation de la solution du cas le plus similaires. La seconde alternative est l'apprentissage automatique. Contrairement aux paradigmes de raisonnement précédents qui nécessitent de programmer explicitement des étapes ou instructions à exécuter, concerne le développement de programmes qui apprennent automatiquement à accomplir une tâche à partir des données auxquelles ils ont accès. L'apprentissage automatique est plus récemment utilisé pour la prédiction de l'issue d'affaires. Pour exemple, Katz *et al.* [2014] entraînent des forêts aléatoires [Breiman, 2001] sur les cas de 1946-1953 pour prédire si la Cour Suprême des États-Unis infirmera ou confirmera une décision de juridiction inférieure. Leur approche parvient à prédire correctement 69,7% des décisions finales pour 7700 cas des années 1953-2013. Ils ont amélioré ce résultat par la suite en augmentant le nombre d'arbres et la quantité de données [Katz *et al.*, 2017]. Toujours pour la prédiction des décisions de la Cour Suprême des Etats-Unis, Waltl *et al.* [2017b] utilisent des techniques de traitement automatique du langage naturel (TALN) pour extraire moins d'attributs caractéristiques de décisions que [Katz *et al.*, 2014] à partir des décisions d'appel de la Cour Fiscale allemande (11 contre 244). Ils obtiennent des valeurs de $F_1$-mesures entre 0,53 et 0,58 (validation croisée à 10 itérations) pour la prédiction de la confirmation ou l'infirmation d'un jugement en appel avec un classifieur bayésien naïf.

Notre objectif est d'alimenter les analyses quantitatives de corpus jurisprudentiels en proposant des méthodes d'extraction de connaissances pertinentes telles que les méta-données d'affaires, les règles juridiques associées, les demandes des parties, les réponses des tribunaux, et les liens entre ces données. L'un des postulats évalués empiriquement dans cette thèse est que l'identification de ces diverses connaissances est possible par l'analyse des textes judiciaires basée sur des méthodes du TALN, de la fouille de texte et de la recherche d'information. Cependant, l'application de ces méthodes exigent certaines adaptations pour surmonter les divers défis décrits par Nazarenko & Wyner [2017] : textes très longs et en grande quantité, corpus régulièrement mis à jour, influence subjective de facteurs sociaux et d'opinions politiques, couverture de problématiques

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

## 1.2 Annotation et extraction d'information

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

Différentes méthodes ont été expérimentées pour la reconnaissance d'information dans les documents judiciaires. La plupart reposent sur l'entraînement d'algorithmes d'apprentissage automatique supervisé sur un ensemble d'exemples annotés manuellement (résultats attendus). Parmi ces algorithmes, on retrouve par exemple les modèles probabilistes HMM (Modèles Cachés de Markov, cf. § 2.2.1.1) et CRF (Champs Aléatoires Conditionnels, cf. § 2.2.1.2) que dont l'application est étudiée au chapitre 2. Ces modèles peuvent être combinés à d'autres approches dans un système global. En effet, après avoir segmenté les documents à l'aide d'un modèle CRF, Dozier *et al.* [2010] ont par exemple combiné plusieurs approches pour reconnaître des entités dans les décisions de la Cour Suprême des États-Unis. Ils ont défini manuellement des détecteurs distincts à base de règles pour identifier séparément la juridiction (zone géographique), le type de document, et les noms des juges, en plus de l'introduction d'une recherche lexicale pour détecter la cour, ainsi qu'un classifieur entraîné pour reconnaître le titre. Ces différents détecteurs ont atteint des performances prometteuses, mais avec des rappels limités entre 72% et 87%. Suivant la complexité des éléments à extraire, un système peut exploiter un lexique pour les motifs simples et non-systématiques (indicateurs de mentions de résultats ou de parties) et des règles pour des motifs plus complexes et systématiques (e.g., noms de juges, énoncés de décisions) [Waltl *et al.*, 2016, 2017a; Wyner, 2010]. Cardellino *et al.* [2017] ont par ailleurs

utilisé un modèle CRF et des réseaux de neurones pour la reconnaissance d'entités nommées juridiques dans des jugements de la Cour Européenne des Droits de l'Homme. Ils définissent une hiérarchie des entités nommées distinguant au niveau 1, les entités nommées et des non-entités, spécialisées par 6 classes au niveau 2 (par exemple, Personne, Document), spécialisées par 69 classes au niveau 3 (par exemple, Rôle Juridique, Règlement), spécialisées par 358 classes au niveau 4 (par exemple Juge, Code Juridique). Les basses performances qu'ils rapportent sur le corpus juridique illustrent bien la difficulté de la détection d'entités juridiques dans les décisions judiciaires ($F_1$-mesures de 0.25, 0.08, 0.03 en moyenne respectivement pour les niveaux 2, 3, 4). Plus récemment encore, Andrew & Tannier [2018] proposent une approche pour l'extraction d'entités nommées d'une transaction d'investissement [2] et des relations qu'elles partagent dans des décisions du Luxembourg rédigées en français. Ils combinent un modèle CRF pour les entités à une grammaire GATE JAPE [Thakker *et al.*, 2009] pour les relations, et obtiennent un faible taux d'erreur pour le CRF de 3.12%.

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

## 1.3 Classification des jugements

La classification de texte permet d'organiser un corpus en rangeant les documents dans des catégories généralement prédéfinies par des experts. Pour la classification des décisions, le principe des propositions de la littérature est d'entrainer un modèle statistique traditionnel sur une représentation des documents généralement définie à partir des connaissances du domaine.Par exemple, par classification binaire avec une Machine à Vecteurs de Support (SVM) [Vapnik, 1995] à noyau linéaire (cf. § 4.2.2.2), Aletras *et al.* [2016] identifient s'il y a eu une violation d'un article donné de la convention des droits de l'homme sur les jugements de la Cour Européenne des Droits de l'Hommes (CEDH) [3]. Les vecteurs représentant

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

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

les documents sont construits sur la base des 2000 n-grammes les plus fréquents. Certaines composantes sont les fréquences normalisées des n-grammes sélectionnés (modèle sac-de-mots [Salton *et al.*, 1975; Salton & McGill, 1983]), calculées distinctement pour différentes parties du document (Procédure, Circonstances, Faits, Loi applicable, la Loi et le document entier) ; ce qui résulte en une matrice document-terme $C$. D'autres composantes sont définies par la fréquence des thématiques extraites par une catégorisation non supervisée (*clustering*) avec la similarité cosinus des n-grammes les plus fréquents représentés par leurs vecteurs dans $C$, i.e. le vecteur de leurs scores d'occurrence dans les différentes parties précédemment citées du document. Aletras *et al.* [2016] obtiennent une précision moyenne de 79% sur les 3 articles qu'ils ont manipulés. Notons tout de même la sélection des régions particulières (circonstances, faits, lois, etc.) du document à partir desquelles sont extraits les n-grammes. Cette sélection est un ajustement de la représentation des textes qui paraît nécessaire pour obtenir de bons résultats. La structuration préalable des documents est ainsi utile pour réduire le bruit qui occupe généralement plus d'espace que les passages ou éléments d'intérêt. Medvedeva *et al.* [2018] étendent ces travaux à neuf articles de loi, tout en montrant empiriquement, entre autres, la possibilité de prédire la violation des articles sur des périodes futures à celles couvertes par les données utilisées lors des phases d'entraînement. Şulea *et al.* [2017a] traitent, d'autre part, l'identification des résultats dans des arrêts [4] de la Cour Française de Cassation. Après un essai [Şulea *et al.*, 2017b] avec un SVM entrainé sur une représentation des documents par le modèle TF-IDF [Salton & Buckley, 1988], ils améliorent les résultats à l'aide d'un classifieur ensembliste de SVM à probabilité moyenne, parvenant à des $F_1$-mesures de plus de 95% Şulea *et al.* [2017a]. Un classifieur SVM à probabilité moyenne combine plusieurs modèles SVM dits « faibles » (ou de base) entrainés chacun sur un sous-ensemble de la base d'apprentissage. Lors de la prédiction, chacun des SVM estime une probabilité d'appartenance du document classifié à chaque classe. La classe du document est celle dont la probabilité moyenne (robustement estimé par la médiane [Kittler *et al.*, 1998]) est maximale.

Par ailleurs, Ashley & Brüninghaus [2009] identifient, par classification, des informations appelées « Facteurs » (*Factors* [Ashley, 1990]), indispensables à leur système *Issue-Based Prediction* [Bruninghaus & Ashley, 2003] basé sur un raisonnement à base cas pour prédire la partie qui doit être favorisée sur une question juridique. Les Facteurs sont en effet des aspects juridiques spécifiques à un domaine et importants pour la réso-

4. Documents de `https://www.legifrance.gouv.fr`.

lution d'un contentieux [Bench-Capon, 1997]. Ils font abstraction des faits dans les raisonnements à base de cas où ils sont définis sous forme de prédicats favorables soit au plaignant soit au défendeur. Sur l'appropriation illicite de secrets commerciaux (*trade secret misappropriation*), l'environnement d'enseignement CATO [5] [Aleven & Ashley, 1997; Aleven, 2003] comprend 26 Facteurs. On y retrouve par exemple les Facteurs *Unique-Product* (le produit est unique), *Agreed-Not-To-Disclose* (il existait un accord de non-divulgation entre le défendant et le plaignant), *Info-Reverse-Engineerable* (les informations du produit peuvent être apprises par ingénierie inverse), et *Disclosure-In-Negotiations* (le demandeur a divulgué des informations concernant son produit lors des négociations avec le défendeur). Les deux premiers Facteurs favorisent le plaignant, propriétaire du produit, et les deux derniers Facteurs favorise le défendeur accusé. Un Facteur s'applique à une affaire si la description de cette dernière contient un fait correspondant. Ashley & Brüninghaus [2009] définissent un classifieur (le-plus-proche-voisin) par Facteur pour identifier ceux qui s'appliquent à la décision. En effet, les phrases de faits des cas résolus sont labellisées par le Facteur auxquels il correspondent. Ensuite, la classification d'un nouveau cas consiste à comparer les différentes phrases annotées à chacune des phrases de faits du nouveau cas, et à affecter à ces derniers le Facteur de la phrases annotées la plus similaires. Au cours de leurs expérimentations, les auteurs démontrent que des adaptations de la représentation des cas par sac-de-mots sont nécessaires pour améliorer les résultats de classifications. Ils proposent deux méthodes améliorées de représentation [Brüninghaus & Ashley, 2001] : l' « abstraction des noms par les rôles » (*roles-replaced representation*) et les « schémas propositionnels » (*propositional patterns*). La représentation par abstraction des noms par les rôles consiste à remplacer les noms des parties et les informations sur le produit par leur rôle respectifs : *plaintiff* (demandeur), *defendant* (defendeur), *information* (produit). Quant à la représentation par schémas propositionnels, elle consiste à définir, à l'aide de techniques de TALN, des attributs sous forme de propositions logiques du texte qui captent la signification du Facteur. Pour le Facteur *Disclosure-In-Negotiations*, pour capter le fait que le demandeur ($\pi$) a divulgué quelque chose, les auteurs définissent, par exemple, la proposition $(\pi \textit{ disclose})$ chaque fois qu'un synonyme du verbe *disclose* (divulger) est identifié. Réalisées sur 146 affaires, les expérimentations de validation croisée *leave-one-out* [6] montrent l'impact des améliorations avec

5. CATO est un environnement intelligent d'enseignement, aux étudiants de droit, de compétences de construction d'arguments à partir de cas à travers la pratique de tâches de test de théorie et d'argumentation à base de cas [Aleven & Ashley, 1997].

6. Pour $N$ cas annotés, une validation croisée *leave-one-out* réalise $N$ expérimentations

une $F_1$-mesure moyenne de 0.211 pour les sacs-de-mots, 0.26 pour les schémas propositionnels, et 0.28 pour l'abstraction des noms par les rôles.

D'autres catégorisations sont tout aussi utiles pour faciliter la recherche d'information. Par exemple, Şulea *et al.* [2017b,a] expérimentent la classification pour identifier la formation judiciaire (chambre civile, chambre commerciale, chambre sociale, etc.) et la période (Intervalle d'années dans laquelle la décision a été prononcée) des décisions. La classification peut aussi servir à évaluer 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 regrouper des textes traitant de sujets similaires, et séparer ceux dont les sujets sont différents. La mesure de similarité doit être définie de sorte à rapprocher ou éloigner les documents suivant l'aspect sémantique que l'on souhaite révéler. Nair & Wagh [2018] exploitent les citations de lois et précédents [7] pour retrouver les textes juridiques qui ont une similarité. Ils analysent le réseau de 597 citations [8] sous l'Acte 2000 des Technologies de l'Information (*Information Technology Act, 2000* [9]) dans des jugements indiens. Leur proposition est d'utiliser des règles d'association générées par l'algorithme Apriori [Agrawal *et al.*, 1994] pour regrouper les jugements susceptibles d'être cités ensemble. Cet algorithme recherche les ensembles singletons de citations suffisamment fréquentes (seuil nécessaire), puis fusionne de manière itérative les ensembles tant que la co-occurrence des citations de la fusion est suffisamment fréquente dans le réseau. Une règle d'association $\{c_1, \ldots, c_n\} \rightarrow \{c'\}$ indique qu'une citation $c'$ est observable si l'on observe une co-occurrence d'un ensemble donné de citations $\{c_1, \ldots, c_n\}$. A chaque règle est associé un score de confiance calculé à partir d'une métrique appelé score de support $sc()$ qui indique, pour un ensemble $\{c_1, \ldots, c_n\}$, la fréquence de co-occurrence des citations de cet ensemble. Le support d'un singleton est sa fréquence d'occurrence. La similarité est confirmée si le score de confiance de la règle $conf(\{c_1, \ldots, c_n\} \rightarrow \{c'\}) = sc(\{c_1, \ldots, c_n, c'\})/sc(\{c_1, \ldots, c_n\})$ est suffisamment élevé. Nair & Wagh [2018] démontrent au travers de scénarios

qui utilisent à tour de rôle 1 cas différent pour le test et le reste $(N-1)$ comme cas d'apprentissage.

7. Les jugements du « Common Law » citent des décisions antérieures similaires.
8. Disponibles sur `https://indiankanoon.org`.
9. `https://www.meity.gov.in/content/information-technology-act-2000`.

(aucune évaluation quantifiée de l'efficacité de l'approche n'est proposée) que les documents qui sont fréquemment cités ensemble sont similaires car traitant de thématiques proches. Cette relation permet par transitivité de retrouver les documents pertinents dans une base de données.

Les métriques traditionnelles de similarité ne sont pas toujours très efficaces sur les décisions judiciaires. La raison peut être une représentation inadéquate des textes qui ne permet pas de traduire une fois comprise la notion de similarité telle qu'entendue dans la plupart des travaux. Thenmozhi *et al.* [2017] comparent par exemple, l'utilisation de la similarité cosinus sur trois représentations différentes des jugements dans le cadre de la campagne de recherche d'affaires antérieures pertinentes IRLeD@FIRE2017 [Mandal *et al.*, 2017] : (1) TF-IDF des concepts (noms), (2) TF-IDF des concepts et relations (verbes), (3) et la moyenne des plongements lexicaux *Word2Vec* [Mikolov *et al.*, 2013] des concepts et relations. Au vu des résultats (0.1795, 0.178, 0.0755 de précision@10 [10] et 0.681, 0.661, 0.435 de rappel@10[10] respectivement pour les méthodes 1, 2, et 3), la première représentation semble mieux capter la similarité contrairement à l'utilisation des verbes et de la représentation distribuée. Ma *et al.* [2018] utilisent une forme de connaissances a priori définie dans des modèles de type ontologie pour estimer la similarité entre décisions. Ils proposent notamment d'aligner le document sur une ontologie des concepts et relations d'un corpus judiciaire. L'idée est de calculer la similarité sur un résumé du texte qui regroupe des aspects pertinents. Cette méthode permet ainsi de mieux capter la sémantique des jugements, d'avoir une meilleure précision, et de réduire la complexité temporelle inhérente à l'exploitation de longs documents notamment lors de l'utilisation de la « distance du déménageur de mots » ou WMD (*Word Mover's Distance*) de Kusner *et al.* [2015] (cf. § 5.2.2). L'amélioration est observée sur une tâche de classification des jugements Chinois relatifs aux crimes de la circulation routière dans quatre catégories correspondant à des sentences d'emprisonnement [11] (précision de 90.3% et 92.3% pour le résumé contre 84.8% et 82.4% pour le document original respectivement sur les deux ensembles de données utilisés).

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

1. *all-term cosine similarity* : le cosinus de similarité entre les représen-

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

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

tations TF-IDF de $x$ et $y$ dont tous les termes présents dans les jugements sont les dimensions.

2. *legal-term cosine similarity* : le cosinus de similarité sur les réductions des dimensions précédentes uniquement aux termes apparaissant dans un dictionnaire juridique.
3. *bibliographic coupling similarity* : la similarité de couplages bibliographiques égal au nombre de citations de jugements communes à $x$ et $y$.
4. *co-citation similarity* : la similarité de co-citation qui est le nombre de citations de $x$ et $y$ dans un même jugement.

Les résultats ont été interprété sur de très faibles proportions des données utilisées (paires de jugements pris parmi 2430 cas et annotées par des experts avec des scores de similarité à valeurs réelles entre 0 pour faible et 1 pour fort).En effet, la comparaison des méthodes de calcul de la similarité est analysée sur les 18 pairs de jugements ayant un score de similarité de couplages bibliographiques supérieur ou égal à 3. Deux jugements sont considérés similaires pour un score d'expert $\geq 0.50$ (parmi les 18 paires, une seule a un score $\leq 0.50$). Il en ressort que le cosinus de similarité avec les termes juridiques (8/18 pour un seuil minimum à 0.5) et le couplage bibliographique (17/18 pour un seuil minimum à 3) correspondent plus à la notion de similarité des experts, que la similarité basée sur tous les termes (3/18 pour pour un seuil minimum à 1) du corpus ou sur la co-citation (6/18 pour pour un seuil minimum à 1).

En synthèse, la similarité entre documents est utilisée pour répondre à plusieurs tâches, comme par exemple, la recherche de décisions similaires [Thenmozhi *et al.*, 2017], le regroupement non-supervisé de jugements [Ravi Kumar & 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 critères d'évaluation.

## 1.5 Conclusion

En résumé, les travaux portant sur l'analyse automatique des décisions ont donné des résultats encourageants grâce aux éléments spécifiques aux

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

***Résumé.*** *Ce chapitre traite de la détection de sections et des mentions (occurrences) d'entités dans les décisions de justices françaises. Ce problème est important car il vise une structuration des documents par le balisage des sections organisant le document, des méta-données de référence de l'affaire, et des citations des normes juridiques employées. Cette annotation automatique facilite la lecture des décisions et fournit des méta-données pour rapidement indexer et retrouver des décisions. Le problème est formulée en des tâches d'étiquetage de séquence puisque tout document textuel est une séquence de mots, de lignes, etc. Les principales contributions discutées ici sont : l'annotation manuelle d'un corpus d'évaluation de 500 documents, une analyse de l'application des modèles probabilistes graphiques HMM et CRF, et des discussions sur l'impact de divers aspects de la conception d'un système d'annotation par étiquetage de séquence. Les expérimentations effectuées permettent de comparer l'ingénierie manuelle et l'ingénierie automatique de la représentation des objets à classer, de comparer des schéma d'étiquetage, d'analyser l'effet de l'augmentation des données d'entrainement sur la qualité des annotations. Les résultats montrent principalement l'efficacité des modèles à base de champs aléatoires conditionnels à chaîne linéaire (CRF) pour les différentes tâches.*

## 2.1 Introduction

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

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

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

| **Entités** | **Label** | **Exemples** | **#mentions**[a] | |
|---|---|---|---|---|
| | | | **Médiane**[b] | **Total**[c] |
| Numéro de registre général (R.G.) | **rg** | « 10/02324 », « 60/JAF/09 » | 3 | 1318 |
| Ville | **ville** | « NÎMES », « Agen », « Toulouse » | 3 | 1304 |
| Juridiction | **juridiction** | « COUR D'APPEL » | 3 | 1308 |
| Formation | **formation** | « 1re chambre », « Chambre économique » | 2 | 1245 |
| Date de prononcé | **date** | « 01 MARS 2012 », « 15/04/2014 » | 3 | 1590 |
| Appelant | **appelant** | « SARL K. », « Syndicat ... », « Mme X ... » | 2 | 1336 |
| Intimé | **intime** | - // - | 3 | 1933 |
| Intervenant | **intervenant** | - // - | 0 | 51 |
| Avocat | **avocat** | « Me Dominique A., avocat au barreau de Papeete » | 3 | 2313 |
| Juge | **juge** | « Monsieur André R. », « Mme 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** | *mot ne faisant partie d'aucune mention d'entité* | - | - |

[a] nombre de mentions d'entités dans le corpus annoté pour les expérimentations
[b] nombre médian de mentions par document dans le corpus annoté
[c] nombre total d'occurrences dans le corpus annoté
* Les statistiques sur les sommes d'argent ne concernent que 100 documents annotés (max=106, min=1, moyenne=17.77), contre 500 documents pour les autres entités.

Tableau 2.1 – Exemples d'entités et statistiques sur la base d'exemples annotés manuellement

On pourrait s'attendre à ce qu'une institution comme la justice respecte un modèle strict et commun à tous les tribunaux pour la rédaction des décisions pour permettre de facilement pouvoir les lire et les analyser. Malheureusement, même si les décisions décrivent des informations

de même nature, le modèle employé semble varier entre les juridictions. C'est ce que l'on remarque déjà au niveau de la transition entre sections. Au vu de leur rôle, il est évident que les sections devraient être séparées par des marqueurs bien précis. Une approche intuitive de sectionnement consisterait par conséquent à définir un algorithme capable de reconnaître automatiquement ces marqueurs de transition par l'utilisation d'expressions régulières. Cependant, les marqueurs retrouvés ne sont généralement pas standards. Les indicateurs de transitions sont en effet souvent différents d'une décision à l'autre ; ils peuvent correspondre à des titres ou des motifs à base de symboles (astérisques, tirets, etc.). Il arrive même parfois que la transition soit implicite et que l'on ne s'en rende compte que par la forme ou le contenu des lignes, au cours de la lecture. Même les marqueurs explicites sont hétérogènes. Lors de l'emploi de titres par exemple, la transition de l'entête à l'exposé du litige peut être indiquée par des titres comme « Exposé », « FAITS ET PROCÉDURES », « Exposé de l'affaire », « Exposé des faits », etc. Quant au dispositif, il est introduit généralement par l'expression « PAR CES MOTIFS » avec souvent quelques variantes qui peuvent être très simples (par exemple « Par Ces Motifs ») ou exceptionnelles (par exemple « P A R C E S M O T I F S : »). Dans certaines décisions, cette expression est remplacée par d'autres expressions comme « DECISION », « DISPOSITIF », « LA COUR », etc. Par ailleurs, lors de l'utilisation de symboles, il arrive qu'un même motif sépare différentes sections et même des paragraphes dans une même section. Des différences similaires apparaissent aussi pour les entités. Les noms de parties sont généralement placés après un mot particulier comme « APPELANTS » ou « DEMANDEUR » pour les demandeurs (appelants en juridiction de 2e degré), « INTIMES » ou « DEFENDEUR » pour les défendeurs (ou intimés), et « INTERVENANTS » pour les intervenants. Les noms des individus, sociétés et lieux commencent par une lettre majuscule, et sont entièrement en majuscules. Cependant, certains mots communs peuvent apparaître aussi en majuscules (par ex. APPELANTS, DÉBATS, ORDONNANCE DE CLÔTURE). Les entités peuvent contenir des chiffres (identifiants, dates, ...), des caractères spéciaux ( « / », « - »), des initiales (par ex. « A. ») ou abréviations. Dans l'entête, les entités apparaissent généralement dans le même ordre (par ex. les appelants avant les intimés, les intimés avant les intervenants). Cependant, on rencontre une multitude de types d'entités dans l'entête, contrairement aux autres sections où seules les normes nous intéressent. De plus, le texte est mieux structuré dans l'entête que dans les autres sections. Ces nombreuses différences entre décisions rendent inadéquates les expressions régulières.

Notre étude consiste à analyser l'application du Modèle Caché de Mar-

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

## 2.2 Extraction d'information par étiquetage de séquence

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

- Les **systèmes à recherche lexicale** sont conçus sur la base d'une liste d'entités préalablement connues, et leurs synonymes dans le domaine d'intérêt. Par exemple, dans le domaine juridique, un lexique pourrait contenir les identifiants de règles juridiques et les noms des juges. La liste des entités peut être fournie par des experts ou apprise à partir d'un ensemble de données annotées manuellement (phase d'apprentissage). Cependant, il s'avère très difficile de maintenir une telle liste car le domaine change régulièrement (nouvelles lois par ex.). De plus, les mentions d'entités peuvent avoir plusieurs variantes. Par exemple, la même règle juridique « Article 700 du code de procédure civile » peut être citée seule et en entier (« article 700 du code de procédure civile »), ou abrégée (« article 700 CPC »), ou encore avec d'autres règles (« articles 700 et 699 du code de procédure civile »). De plus, ces approches sont sujettes aux problèmes d'ambiguïté, par exemple lorsque différentes entités comprennent les mêmes mots. Ces problèmes ont largement limité ces premiers systèmes [Palmer & Day, 1997].
- Les **systèmes à base de règles** décrivent la variété des mentions d'entités en fonction de la régularité du contexte, de la structure et du lexique. Il existe plusieurs plate-formes et langages permettant de formaliser l'écriture des règles. Par exemple, dans le formalisme JAPE de Gate, Wyner [2010] détecte les énoncés de décisions à l'aide d'une règle qui sélectionne les phrases contenant un terme de jugement (*affirm, grant,* etc.) et suivies d'un nom de juge :

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

Ces systèmes présentent l'avantage de reposer sur des expressions déclaratives qui facilitent la maintenance (erreurs faciles à tracer et à expliquer) et l'expression directe des connaissances du domaine en règles [Waltl *et al.*, 2018]. Bien que parfois suffisant pour traiter des corpus modestes et spécialisés, ces systèmes sont très souvent limités en pratique. La définition manuelle de règles exige notamment des efforts considérables, en particulier pour le traitement de grands corpus. Par ailleurs, un ensemble donné de règles est difficilement réutilisable dans d'autres domaines ou sur des données n'intégrant pas exactement les subtilités linguistiques exprimées par les règles. Quelques approches adaptatives ont néanmoins été conçues pour surmonter ces limites tout en bénéficiant toujours de la facilité à expliquer le comportement des systèmes à base de règles [Siniakov, 2008; Chiticariu *et al.*, 2010].

- Les **systèmes basés sur l'apprentissage automatique** exécutent des classifieurs multi-classes sur des segments de texte. Par exemple, un algorithme traditionnel de classification comme le modèle bayésien naïf peut être entraîné pour détecter les noms de gènes en classifiant les mots d'un article scientifique [Persson, 2012]. Par ailleurs, les algorithmes d'étiquetage de séquences tels que le CRF classifient les mots tout en modélisant les transitions entre les labels [Finkel *et al.*, 2005]. Dans ce registre, les architectures d'apprentissage profond réalisent actuellement les meilleures performances sur de multiples tâches d'extraction d'information en général et de reconnaissance d'entités nommées en particulier [Lample *et al.*, 2016].

Certains travaux ont combiné différentes approches pour extraire les entités à partir de documents juridiques, par exemple, par la description de l'information contextuelle en utilisant des règles pour répondre au problème d'ambiguïté des méthodes à recherche lexicale [Mikheev *et al.*, 1999; Hanisch *et al.*, 2005]. Mais les systèmes basés sur l'apprentissage automatique sont les plus efficaces actuellement pour l'extraction d'information, en particulier les modèles graphiques probabilistes.

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

### 2.2.1 Les modèles graphiques probabilistes HMM et CRF

Nous avons choisi d'analyser l'application des modèles CRF et HMM car les comparaisons avec d'autres approches démontrent bien que les modèles probabilistes obtiennent les meilleurs résultats lors de l'extraction d'information dans les documents juridiques. Par exemple, dans Kríz *et al.* [2014], le modèle HMM a été comparé à l'Algorithme de Perceptron à Marges Inégales (PAUM) de Li *et al.* [2002] pour reconnaître les institutions et références d'autres décisions de justice, ainsi que les citations d'actes juridiques (loi, contrat, etc.) dans les décisions judiciaires de la République Tchèque. Les deux modèles ont donné de bonnes performances avec des scores $F_1$ de 89% et 97% pour le HMM utilisant les tri-grammes comme descripteurs de mots, et des scores $F_1$ de 87% et 97% pour le PAUM en utilisant des 5-grammes de lemmes et les rôles grammaticaux (*Part-Of-Speech tag*) comme descripteurs.

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

#### 2.2.1.1 Les modèles cachés de Markov (HMM)

Un modèle HMM[1] est une machine à états finis définie par un ensemble d'états $\{s_1, s_2, ..., s_m\}$. Un modèle HMM a pour fonction d'affecter une probabilité jointe $P(T, L) = \prod_{i=1}^{n} P(l_i|l_{i-1})P(t_i|l_i)$ à des paires de séquences d'observations $T = t_{1:n}$ et de séquences de labels $L = l_{1:n}$. Étant donné qu'un HMM est un modèle génératif, chaque label $l_i$ correspond à l'état $s_j$ dans lequel la machine a généré l'observation $t_i$. Il y a autant de labels candidats que d'états. Le processus d'étiquetage de $T$ consiste à déterminer la séquence de labels $L^*$ qui maximise la probabilité jointe

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

($L^* = \arg\max_L P(T, L)$). Une évaluation de toutes les séquences possibles de labels est nécessaire pour déterminer $L^*$. Pour éviter la complexité exponentielle $O(m^n)$ d'une telle approche, $n$ étant la longueur du texte et $m$ le nombre de labels candidats, l'algorithme de décodage Viterbi [Viterbi, 1967], basé sur de la programmation dynamique, permet d'obtenir une estimation de $L^*$. Cet algorithme utilise des paramètres estimés par apprentissage sur un corpus de textes annotés manuellement :

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

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

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

#### 2.2.1.2 Les champs conditionnels aléatoires à chaîne linéaire (CRF)

Même si l'algorithme Viterbi est aussi utilisé pour appliquer le modèle CRF à l'étiquetage de séquences, la structure du CRF diffère de celle du HMM. Au lieu de maximiser la probabilité jointe $P(L, T)$ comme le HMM, un modèle CRF [Lafferty *et al.*, 2001] cherche la séquence de labels $L^*$ qui maximise la probabilité conditionnelle suivante :

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

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

Les fonctions potentielles $f_j(\cdot)$ sont les caractéristiques utilisées par les modèles CRF. Deux types de fonctions caractéristiques sont définies : les caractéristiques de transition qui dépendent des labels aux positions courantes et précédentes ($l_{i-1}$ et $l_i$ resp.) et de $T$ ; et les caractéristiques d'état qui sont des fonctions de l'état courant $l_i$ et de la séquence $T$. Ces fonctions $f_j(\cdot)$ sont définies à l'aide de fonctions à valeurs binaires ou réelles $b(T, i)$ qui combinent les descripteurs d'une position $i$ dans $T$ [Wallach, 2004]. Pour labelliser les références aux règles de loi par exemple, un CRF pourrait inclure par exemple les fonctions potentielles pour étiqueter « 700 » dans ce contexte « ... l'article 700 du code de procédure civile ... » :

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

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

avec

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

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

$t_i$ étant l'observation (le mot) en position $i$ dans $T$, $POS_i$ étant le rôle grammatical de $t_i$ ($NUM$ = valeur numérique, $NOM$ = nom), et $NP1_i$ et $NS1_i$ sont les lemmes des mots avant et après $t_i$, respectivement. Les symboles *<unknown>* et *@card@* encodent les lemmes inconnus et ceux des nombres respectivement. Pouvant être activées au même moment, les fonctions $f_1$ et $f_2$ définissent des descripteurs se chevauchant. Avec plusieurs fonctions activées, la croyance dans le fait que $l_i = NORME$ 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 $\lambda = (\lambda_1, ..., \lambda_F)$ à partir de textes annotés manuellement $\{(T_1, L_1), ..., (T_M, L_M)\}$, $T_k$ étant un texte et $L_k$ la séquence de labels correspondants. La valeur optimale de $\lambda$ est celle qui maximise la fonction objectif $\sum_{k=1}^{M} \log P(L_k|T_k)$ sur les données d'entraînement. En général, outre le maximum de vraisemblance, cette optimisation

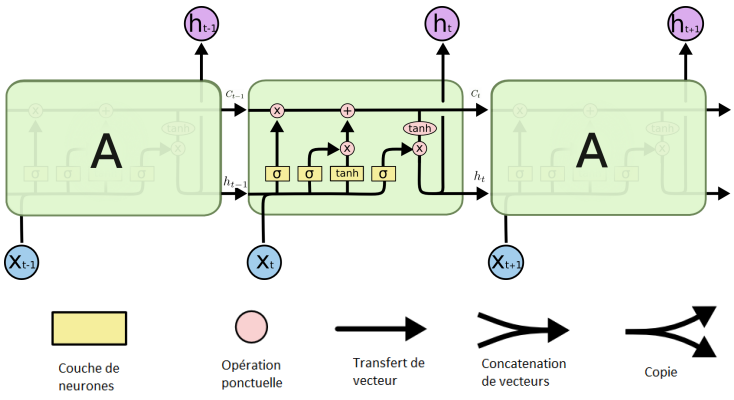
est résolue à l'aide de l'algorithme de descente de gradient dont l'exécution peut être accélérée à l'aide de l'algorithme L-BFGS de Liu & Nocedal [1989].

#### 2.2.1.3 CRF et réseaux de neurones artificiels

**Définitions** Un réseau de neurones artificiels est un algorithme d'apprentissage automatique dont le fonctionnement est inspiré de celui du cerveau [McCulloch & Pitts, 1943; Rosenblatt, 1958]. Un neurone reçoit des valeurs $x = [x_1, x_2, \ldots, x_l]$ en entrée, puis calcule un élément de sortie $y$ par une fonction de la combinaison pondérée des entrées (poids $W = [w_0, w_1, \ldots, w_l]$). Un réseau de neurones comprend des neurones structurés en couches. Les sorties d'une couche servent d'entrées à la couche suivante et les nœuds d'une couche ne sont pas connectés. Les poids sont déterminés lors d'une phase d'entraînement qui les ajuste afin de minimiser l'erreur entre les valeurs attendues $y$ en sorties et celles prédites $\hat{y}$ par le modèle. Traditionnellement, les réseaux de neurones sont à « propagation avant » (*feed-forward*) i.e. l'unique sens de propagation de l'information est de la couche d'entrée successivement vers a couche de sortie. Mais comme nous l'avons vue précédemment, la prise en compte du contexte (état précédent ou suivant) est très importante dans la modélisation des séquences. Les réseaux à propagation avant ne sont pas adaptés pour prédire la sortie à l'instant $t$ à partir de ses connaissances des instants précédents.

Les réseaux récurrents de neurones (*recurrent neural networks - RNN*) [Jordan, 1986; Elman, 1990] sont une architecture conçue pour modéliser les données séquentielles $X = X_{1:n}$. Le principe est de passer à l'instant $t$ en entrée du réseau, la sortie ou état du réseau de l'instant précédent $t-1$ en plus de l'observation courante $X_t$. Un LSTM est une variante particulière de RNN dont l'état est définie par la sortie $h_t$ du réseau et la sortie $C_t$ qui permet de gérer la mémoire à plus long terme (Figure 2.1 Page 34).

Les entrées et sorties d'un LSTM sont des vecteurs de nombres réels. L'entrée $X_i$ est la représentation vectorielle indépendante de la séquence entrée de l'observation en position $i$, par exemple celle du mot $t_i$ du texte $T = t_{1:n}$. La sortie $h_i$ représente le contexte de l'observation en position $i$ dans la séquence entrée. On peut ainsi chaîner des LSTM pour modéliser les textes entrées et y appliquer un CRF pour l'étiquetage. C'est le principe du BiLSTM-CRF de Lample *et al.* [2016] une des architectures les plus efficaces pour la l'étiquetage d'entités nommées. En effet, le BiLSTM comprend deux couches enchaînant des LSTM (Figure 2.2 Page 35). Une couche permet d'apprendre le contexte "gauche" des mots, les états étant propager dans le sens du début vers la fin du texte. L'autre couche apprend




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

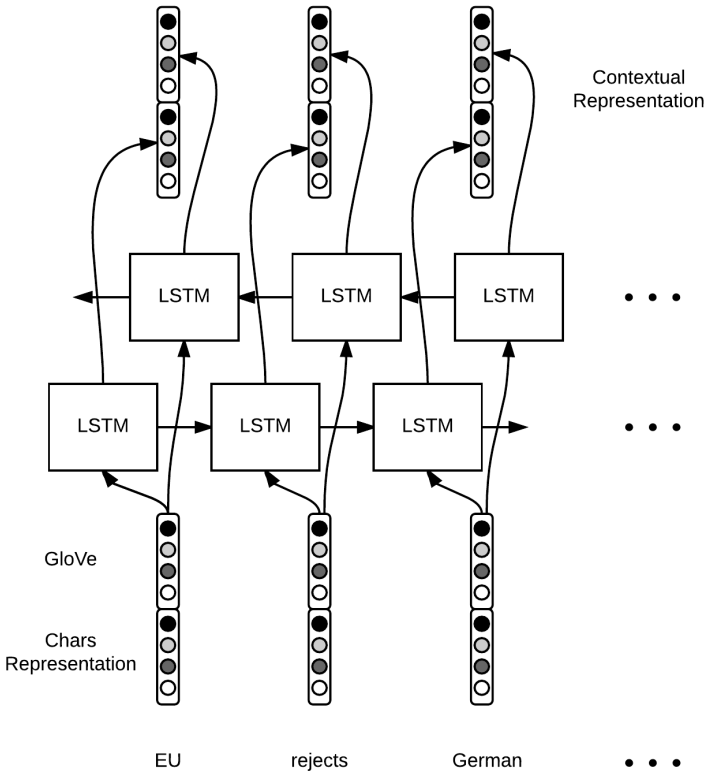
Figure 2.1 – Structure interne d'un LSTM dans une couche de réseau LSTM.

le contexte droit en propageant les informations dans le sens inverse. Les mots sont représentées en entrée sous forme vectorielle. Sur la Figure 2.2 Page 35 par exemple, il s'agit de la concaténation du plongement lexical des mot par la méthode Glove [Pennington *et al.*, 2014] et de la représentation contextuelle des caractères dans le mot (*Chars representation*).

### 2.2.2 Représentation des segments atomiques

La représentation des segments à labelliser occupe une place importante pour l'obtention de bons résultats avec les modèles décrits précédemment. Elle consiste généralement à décrire la forme et le contexte de chaque segment en lui assignant des attributs [Nadeau & Sekine, 2007; Sharnagat, 2014]. Ils peuvent être booléens (« le mot est il en majuscule ? »), numériques (nombre de caractères du mot), nominaux (par ex. le rôle grammatical d'un mot), ou définis par des expressions régulières (par ex. pour les numéros R.G. on peut avoir `dd/ddddd` 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 alors




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

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

s'avérer nécessaire. Cette sélection peut améliorer les performances d'étiquetage, et accélérer l'extraction des descripteurs, l'entraînement du modèle, ainsi que son application à de nouveaux textes [Kitoogo & Baryamureeba, 2007]. Elle peut aussi fournir une meilleure compréhension du comportement des modèles entraînés [Klinger & Friedrich, 2009]. Deux principales approches se distinguent. D'une part, les méthodes « filtrantes » (*filters*), comme l'information mutuelle, comparent individuellement les descripteurs à l'aide de scores qui ne sont pas nécessairement basés sur la performance. D'autre part, les méthodes « enveloppantes » (*wrappers*) comparent des sous-ensembles de descripteurs sur la base des performances d'évaluation qu'elles permettent d'obtenir (par exemple la $F_1$-mesure obtenue sur un ensemble d'exemples). Même si les méthodes filtrantes sont plus rapides, elles sont en général moins performantes car elles ne per-

mettent pas d'éviter les redondances, et ne prennent pas en compte l'effet de la combinaison de caractéristiques.

La définition manuelle des caractéristiques suivie de la sélection est souvent qualifiée de méthode forcée car elle dépend fortement de la capacité du concepteur du système à identifier les descripteurs appropriés.

### 2.2.3 Schéma d'étiquetage

Nous traitons d'entités dont les occurrences comprennent un ou plusieurs éléments atomiques. Pour améliorer les résultats d'un modèle d'étiquetage, certaines parties des entités peuvent être mises en évidence à travers une représentation appropriée de segments. La Figure 2.3 Page 36 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.

| | *composée* | *de* | *Madame* | *Martine* | *JEAN* | *,* | *Président* | *de* | *chambre* |
|---|---|---|---|---|---|---|---|---|---|
| IO | O | O | I-JUGE | I-JUGE | I-JUGE | O | I-FONCTION | I-FONCTION | I-FONCTION |
| BIO | O | O | B-JUGE | I-JUGE | I-JUGE | O | B-FONCTION | I-FONCTION | I-FONCTION |
| IEO | O | O | I-JUGE | I-JUGE | E-JUGE | O | I-FONCTION | I-FONCTION | E-FONCTION |
| BIEO | O | O | B-JUGE | I-JUGE | E-JUGE | O | B-FONCTION | I-FONCTION | E-FONCTION |

Figure 2.3 – Illustration des schémas d'étiquetage IO, BIO, IEO, BIEO

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

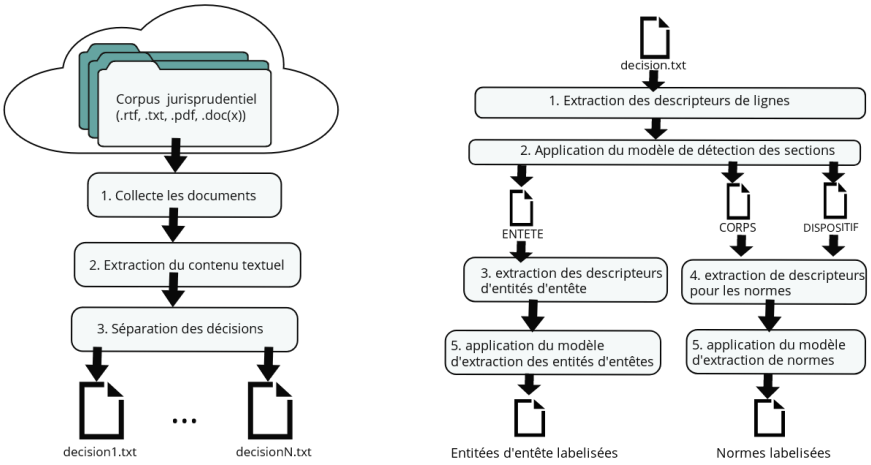
- B : début (*beginning*) ;
- I : intérieur (*inside*) ;
- E (ou L, ou M) : fin (*end* ou *last* ou *middle*) ;
- S (ou U, ou W) : singleton ou entité à segment unique (*single* ou *unit* ou *whole*) ;
- O : hors de toute entité (*outside*).

Le schéma IO utilisé par défaut ne met l'accent sur aucune partie et affecte le même label à tous les segments d'une même entité. D'autres schémas distinguent soit le premier élément (BIO), soit le dernier (IEO), soit les deux (BIEO). Les schémas IEO et BIO ont des variantes IEO1, BIO1, IOE2, et BIO2. Les modèles IOE2, et BIO2 utilisent resp. les préfixes E- et B- pour

étiqueter les entités à mot unique, contrairement à IEO1 et BIO1 qui utilisent plutôt le préfixe I- dans ce cas. Le modèle BIEO est souvent étendu sous la forme BIESO (ou BILOU) dans le cas où l'on souhaite distinguer les entités à un seul segment (par ex. ville ou numéro R.G.). Il est possible d'aller plus loin en mettant l'accent sur les mots avant (O-JUGE) et après (JUGE-O) l'entité (JUGE par exemple) et en indiquant le début (BOS-O, *begininning of sentence*) et la fin (O-EOS, *end of sentence*) du texte ou de la phrase. Le format ainsi obtenu est appelé BMEWO+ [Baldwin, 2009].

Un autre intérêt des schémas plus complexes que IO est de pouvoir distinguer des entités du même type qui se suivent sans être explicitement séparées (par exemple, des appelants mentionnés sur des lignes consécutives). Cet aspect est notamment important dans les décisions de justice par exemple lorsque des noms de parties sont listés dans la section ENTETE en n'étant séparés que d'un simple retour à la ligne.

## 2.3 Architecture proposée




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

Figure 2.4 – Application des modèles entraînés pour l'étiquetage de sections et entités.

Nous proposons de travailler uniquement avec le contenu textuel des documents. Ce contenu est extrait des documents téléchargés en éliminant les éléments inutiles, principalement des espaces vides. Ces éléments sont

typiques des documents formatés (`.rtf`, `.doc(x)`, `.pdf`). Ils ne fournissent pas une indication standard sur le début des sections. Le choix de ne pas exploiter le formatage des documents permet d'avoir à gérer un nombre plus faible de diversités entre les textes tout en appliquant le même processus de traitement à tout document indépendamment de son format d'origine. Une simple architecture d'étiquetage de sections et d'entités juridiques a été conçue avec cette uniformisation des documents comme point d'entrée (Figure 2.4 Page 37). Ainsi, les documents sont collectés puis prétraités suivant leur format d'origine (extraction du texte et séparation des décisions apparaissant dans le même document). Ensuite, après le sectionnement des décisions, les entités sont identifiées dans les différentes sections. Par ailleurs, comme segment atomique à étiqueter nous avons choisi les lignes pour la détection des sections, et les mots pour les entités.

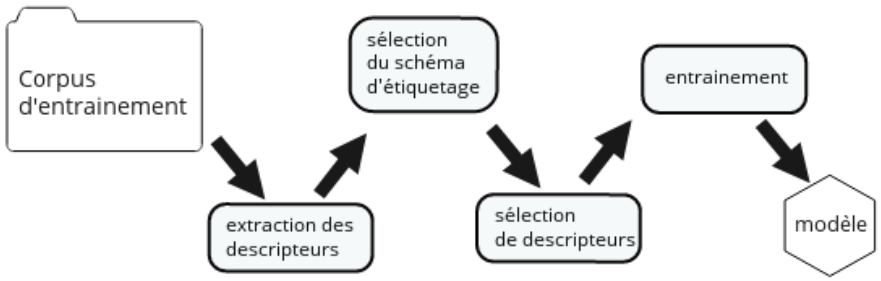



Figure 2.5 – Entrainement des modèles.

Les modèles HMM et CRF étant tous les deux supervisés, ils doivent être entraînés sur des exemples manuellement annotés pour estimer leurs paramètres. Nous proposons de sélectionner le schéma d'étiquetage et les sous-ensembles minimaux de caractéristiques manuellement définies, avant d'entraîner les modèles HMM et CRF (Figure 2.5 Page 38).

## 2.3.1 Définition manuelle de descripteurs candidats

### 2.3.1.1 Descripteurs pour la détection des sections

Nous considérons donc la ligne comme élément à étiqueter lors du sectionnement. Nous n'avons pas travaillé au niveau des mots afin d'éviter que des mots de la même ligne ne soient classés dans des sections différentes. L'étiquetage des phrases a aussi été évité car en découpant les

| Type | Descripteur |
|---|---|
| Forme | • la ligne entière (`token`)<br>• ses premiers mots (`t0, t1, t2`)<br>• sa longueur absolue (`absLength`)<br>• sa longueur relative (`relLength`) |
| Contexte | • le numéro de ligne (`absNum`)<br>• partie du document contenant la ligne (`relNum`)<br>• premiers mots de la ligne précédente (`p0, p1`)<br>• premiers mots de la ligne suivante (`n0, n1`)<br>• longueur absolue de ligne précédente (`nLength`)<br>• longueur relative de ligne précédente (`nRelLength`)<br>• longueur absolue de ligne suivante (`nLength`)<br>• longueur relative de ligne suivante (`nRelLength`) |

Tableau 2.2 – Descripteurs candidats de lignes pour les sections.

documents en phrases telles qu'elles sont entendues en français, on a parfois des segments qui s'étendent d'une section à une autre (absence de ponctuation). De plus, l'entête à davantage l'apparence d'un formulaire.

Plusieurs critères peuvent être utilisés pour différencier les sections, à savoir : la longueur des lignes (plus longues dans le corps, plus courtes dans l'entête), les premiers termes de certaines lignes (typiques de chaque section) et le nombre total de lignes. Un HMM ne supporte qu'un descripteur représentant 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 `LQ2` ($12 < longueur \leq 14$). Nous avons également catégorisé les parties du document afin de capturer une position relative de ligne. En effet, 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. Les descripteurs représentant les lignes sont résumés dans le Tableau 2.2 Page 39.

#### 2.3.1.2 Descripteurs pour la détection d'entités

La détection d'entités consiste, dans notre cas, à entraîner un modèle CRF ou HMM pour étiqueter les différents segments de texte (mot, ponctuation, numéro, identifiant) suivant qu'ils appartiennent ou non à la men-

tion d'une entité. Les deux modèles nécessitent des caractéristiques, dont certaines peuvent être définies sur la base de régularités directement observables dans les textes. Sur la base des observations de décision, nous avons défini la morphologie des mots en décrivant la forme des caractères du mots. Le lemme est aussi utilisé car il homogénéise les variantes du mot correspondant. Pour identifier les normes, nous définissons une liste de mots utilisés pour citer les règles juridiques (*article, code, loi, contrat, décret, convention, civil, pénal*, etc.). Cette liste permet d'indiquer si le mot est décrit est un mot-clé de norme. Le contexte des mots est défini en plus de la forme en décrivant les mots voisins. Par ailleurs, la position des mots par rapport à certains termes-clés semble indiquer la nature potentielle de ces mots. Par exemple, les appelants sont généralement listés avant le mot *appelants* et après le mot *intimés*. Il est également possible d'obtenir des descripteurs à partir du résultat d'autres tâches d'analyse de texte dont les rôles grammaticaux (*Part-of-Speech*) comme Chang & Sung [2005] et les modèles thématiques (*topic model*) comme Polifroni & Mairesse [2011] et Nallapati *et al.* [2010]. Le Tableau 2.3 Page 41 résume l'ensemble des descripteurs caractéristiques définis pour la détection des entités.

## 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 :** Meilleur sous-ensemble de descripteurs

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

| Type | Descripteur |
|---|---|
| Forme | • le mot (`token`)<br>• son lemme (`lemma_W0`)<br>• « commence-t-il par une lettre majuscule ? » (`startsWithCAP`)<br>• « est-il entièrement en majuscule ? » (`isAllCAP`)<br>• « est-ce une initiale solitaire tel que "B." ? » (`isLONELYINITIAL`)<br>• « contient-il un caractère de ponctuation ? » (`PUN-IN`)<br>• « n'est-ce qu'une ponctuation ? » (`isALLPUN`)<br>• « contient-il un caractère numérique ? » (`DIGIT-IN`)<br>• « ne contient-il que des chiffres ? » (`isALLDIGIT`)<br>• « S'agit-il d'un mot-clé de règles juridiques ? » (`isKEYWORD`) |
| Syntaxe | • rôle grammatical du mot (`POS`) |
| Sémantique | • thème du mot (`topic0`) |
| Contexte | • les mots précédents (`w-1, w-2`)<br>• les lemmes des mots précédents (`lemmaW-1, lemmaW-2`)<br>• les mots suivants (`w1, w2`)<br>• les lemmes des mots suivants (`lemmaW1, lemmaW2`)<br>• numéro de ligne (`lineNum`)<br>• position de l'élément dans la ligne (`numInLine`)<br>• « le document contient-il le mot *intervenant* ? » (`intervenantInText`)<br>• « le mot apparaît-il après *APPELANT*, *ENTRE*, et *DEMANDEUR* ? » (`isAfterAPPELANT`)<br>• « le mot apparaît-il après *INTIME*, *ET*, et *DEFENDEUR* » (`isAfterINTIME`)<br>• « le mot apparaît-il après *INTERVENANT* ? » (`isAfterINTERVENANT`)<br>• numéro de la ligne précédente contenant le mot (`lastSeenAt`)<br>• nombre d'occurrences du mot (`nbTimesPrevSeen`)<br>• rôles grammaticaux des mots voisins (`POSW-2, POSW-1, POSW1, POSW2`)<br>• thèmes des mots voisins (`w-2topic0`, `w-1topic0`, `w1topic0`, `w2topic0`) |

Tableau 2.3 – Descripteurs candidats de mots pour les mentions d'entités.

La première méthode, qui est la recherche bidirectionnelle (BDS) de Liu & Motoda [2012], combine la sélection séquentielle en avant (SFS) et la sélection séquentielle en arrière (SBS) en parallèle (Algorithme 1). La SFS recherche un sous-ensemble optimal, en commençant par un ensemble vide et en ajoutant le descripteur qui améliore le mieux l'efficacité du sous-ensemble sélectionné. Le critère d'efficacité dans notre cas est défini par

la $F_1$-mesure (Eq. 2.1). Contrairement à la SFS, la SBS commence par l'ensemble des candidats et supprime successivement les plus mauvais descripteurs. Une caractéristique ne peut être ajoutée dans $Y_{F_{k+1}}$ que si elle est présente dans $Y_{B_k}$.

**Algorithme 2 :** Sélection séquentielle avant à flottement

**Données :** Données annotées, $X$ liste de tous les descripteurs candidats
**Résultat :** Meilleur sous-ensemble de descripteurs

1 $Y_0 = \emptyset$;
2 $k = 0$;
3 **répéter**
4 $\quad x^+ = \underset{x \notin Y_k}{\operatorname{argmax}} F_1(Y_k + x)$ ; $Y_k = Y_k + x^+$;
5 $\quad x^- = \underset{x \in Y_k}{\operatorname{argmax}} F_1(Y_k - x)$;
6 $\quad$ **si** $F_1(Y_k - x^-) > F_1(Y_k)$ **alors**
7 $\quad\quad Y_{k+1} = Y_k - x^-$;
8 $\quad\quad X = X - x^-$;
9 $\quad\quad k = k + 1$;
10 $\quad\quad$ Rentrer à 5;
11 $\quad$ **sinon**
12 $\quad\quad$ 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 (Algorithme 2). En effet, le SFFS effectue des tests en arrière à chaque itération.

#### 2.3.2.2 Sélection pour le modèle HMM

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

## 2.4 Expérimentations et discussions

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

### 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 Developer [2]. Cet outil permet de marquer les passages à annoter en les surlignant à l'aide du pointeur de la souris ; ce qui allège l'annotation manuelle. Des balises XML sont rajoutées autour des passages sélectionnés, en arrière plan dans le document.

Chaque document annoté comprend en moyenne 260 lignes et 3900 mots environ. Les deux dernières colonnes du Tableau 2.1 Page 26 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 [1960]. Ces mesures d'accord inter-annotateur ont été calculées au niveau des caractères parce que certains mots peuvent être coupés par des annotations incorrectes (par ex. *<juridiction> cour d'appe </juridiction> l* contre *<juridiction> cour d'appel </juridiction>*), ou bien les annotateurs pourraient ne pas être d'accord si une apostrophe devrait être incluse ou pas dans l'annotation (par ex. *l'<norme>article 700* contre *<norme >l'article 700*). Les taux de Kappa de 0,705 et 0,974 ont été obtenus pour l'annotation des entités et des sections respectivement. D'après la catégorisation de Viera & Garrett [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. https ://gate.ac.uk/family/developer.html

#### 2.4.1.2 Mesures d'évaluation

Nous avons utilisé la précision, le rappel et la $F_1$-mesure comme mesures d'évaluation car elles sont généralement utilisées comme références en extraction d'information. La $F_1$-mesure se calcule à l'aide de la formule suivante :

$$F_1 = 2 \times \frac{Precision \times Rappel}{Precision + Rappel}. \tag{2.1}$$

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

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

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

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

**Evaluation au niveau entité (*entity-level*)** : cette évaluation mesure le taux d'entités parfaitement identifiées c'est-à-dire seulement celles dont les segments atomiques ont été tous correctement labellisés. Les valeurs de précision et rappel sont calculées sur les données de test pour chaque classe d'entité $e$ comme suit :

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

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

**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 TreeTagger [3] [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 2.4 Page 46 présente des mots représentatifs trouvés dans les premiers thèmes inférés. L'extraction des autres descripteurs a été implémentée pour cette expérimentation.

Les valeurs de précision, rappel, et $F_1$-mesure ont été calculées à l'aide du script d'évaluation de la campagne CoNLL-2002 [4]. Elles sont indiquées en pourcentage dans les tableaux de résultats d'évaluation des sections suivantes.

### 2.4.2 Sélection du schéma d'étiquetage

Dans le but d'évaluer comment la représentation de segments affecte les performances, nous avons implémenté quatre représentations (IO, 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 2.5 Page 47 sont les performances globales sur la base de test. Seul l'élément (mot/ligne) est utilisé comme descripteur. La durée d'entraînement est très longue, particulièrement pour la détection d'entités dans l'entête avec le CRF. Il semble évident que cette durée croisse avec le nombre de labels candidats de la section et la complexité du schéma d'étiquetage. En effet, 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

3. `http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger`
4. `http://www.cnts.ua.ac.be/conll2002/ner/bin/conlleval.txt`

| 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 2.4 – Mots représentatifs des 10 premiers thèmes sur les 100 inférés

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 $F_1$ pour éviter d'exécuter plusieurs fois l'entraînement pour les mêmes sous-ensembles de descripteurs, le processus de sélection est toujours resté très long pour les deux algorithmes. Nous avons testé individuellement chacun des descripteurs candidats pour les modèles HMM. Les résultats sont reportés dans le Tableau 2.6 Page 48.

Le résultat le plus remarquable est la forte réduction du nombre de

| Tâche | Modèle | Niveau atomique[a] | | | Niveau entité[a] | | | Durée[b] | Schéma |
|---|---|---|---|---|---|---|---|---|---|
| | | Précision | Rappel | $F_1$ | Précision | Recall | $F_1$ | | |
| Sections | CRF | 91.75 | 91.75 | 91.75 | 64.49 | 56.55 | 60.26 | 4.685 | IO |
| | | 88.95 | 88.95 | 88.95 | 48.12 | 38.26 | 42.63 | 11.877 | IEO2 |
| | | 87.09 | 87.09 | 87.09 | 46.79 | 37.20 | 41.45 | 12.256 | BIO2 |
| | | 86.00 | 86.00 | 86.00 | 58.98 | 41.86 | 48.97 | 35.981 | BIEO |
| | HMM | 32.64 | 32.64 | 32.64 | 22.16 | 18.91 | 20.41 | 6.564 | IO |
| | | 32.92 | 32.92 | 32.92 | 17.73 | 16.09 | 16.87 | 7.827 | IEO2 |
| | | 32.39 | 32.39 | 32.39 | 31.93 | 26.65 | 29.05 | 8.391 | BIO2 |
| | | 33.06 | 33.06 | 33.06 | 32.47 | 27.53 | 29.80 | 8.7 | BIEO |
| Entités d'entête | CRF | 86.86 | 78.96 | 82.73 | 80.84 | 65.17 | 72.17 | 70.525 | IO |
| | | 87.77 | 79.65 | 83.51 | 82.46 | 65.19 | 72.82 | 228.751 | IEO2 |
| | | 87.41 | 78.14 | 82.51 | 81.66 | 66.80 | 73.49 | 230.865 | BIO2 |
| | | 87.72 | 79.55 | 83.44 | 84.38 | 68.35 | 75.53 | 475.249 | BIEO |
| | HMM | 79.12 | 67.75 | 73.00 | 61.48 | 35.05 | 44.64 | 6.345 | IO |
| | | 78.82 | 68.69 | 73.40 | 66.63 | 40.16 | 50.11 | 8.298 | IEO2 |
| | | 80.68 | 67.48 | 73.49 | 70.37 | 45.32 | 55.14 | 7.908 | BIO2 |
| | | 80.05 | 69.01 | 74.12 | 74.73 | 50.77 | 60.46 | 9.973 | BIEO |
| Normes | CRF | 95.60 | 92.96 | 94.26 | 88.06 | 83.50 | 85.72 | 28 | IO |
| | | 95.40 | 93.18 | 94.27 | 88.75 | 85.65 | 87.17 | 32.136 | IEO2 |
| | | 95.20 | 93.30 | 94.24 | 85.65 | 83.13 | 84.37 | 50.769 | BIO2 |
| | | 95.46 | 91.57 | 93.47 | 88.83 | 84.71 | 86.72 | 50.566 | BIEO |
| | HMM | 89.83 | 88.78 | 89.30 | 73.74 | 75.02 | 74.37 | 41.389 | IO |
| | | 88.20 | 89.23 | 88.71 | 78.01 | 81.27 | 79.61 | 44.086 | IEO2 |
| | | 89.25 | 87.83 | 88.53 | 73.89 | 76.63 | 75.24 | 46.634 | BIO2 |
| | | 87.39 | 88.10 | 87.74 | 77.76 | 82.35 | 79.99 | 45.52 | BIEO |

[a] Résultats sur une simple division du jeu de données en 25% pour l'entraînement et 75% pour les tests (entraînement limité à 100 itérations maximum)
[b] Durée d'entraînement en secondes avant l'arrêt de l'entraînement

Tableau 2.5 – Comparaison des schémas d'étiquetage.

descripteurs par les algorithmes. En général, la moitié est éliminée par la sélection BDS, tandis que la méthode SFFS élimine beaucoup plus de candidats (par exemple en ne sélectionnant que 4 descripteurs parmi les 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 sections, la ligne suivante semble être beaucoup plus indicatrice que la première. Il est aussi intéressant de noter que les descripteurs basés sur notre observation apparaissent dans les sous-ensembles sélectionnés (par ex. `isAfterIntervenant`, `isKEYWORD`). Remarquons aussi que la longueur absolue des lignes (`absLength`) joue un rôle important dans l'identification des sections vu qu'il a été sélectionné à la fois pour le CRF et le HMM (sélection BDS). Avec ces sous-ensembles sélectionnés, les modèles sont plus performants que lorsqu'ils exploitent seulement le segment ou l'ensemble tout entier des candidats. Cette amélioration des résultats n'est pas très importante au regard de la longue durée d'exécution des algorithmes. Ainsi, un algorithme plus rapide et plus efficace devrait être utilisé.

| Tâche | Modèle | niveau atomique[a] | | | niveau entité[a] | | | Sous-ensemble |
|---|---|---|---|---|---|---|---|---|
| | | Précision | Rappel | $F_1$ | Précision | Rappel | $F_1$ | sélectionné |
| 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 2.6 – Performances des sous-ensembles sélectionnés de descripteurs.

### 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. Les Tableaux 2.7 et 2.8 présentent les résultats d'une validation croisée à 5 itérations, respectivement aux niveaux atomique et entité.

| | **HMM** | | | **CRF** | | |
|---|---|---|---|---|---|---|
| | *Precision* | *Rappel* | $F_1$ | *Precision* | *Rappel* | $F_1$ |
| **I-corps** | 92.46 | 95.25 | 93.83 | 99.57 | 99.69 | 99.63 |
| **I-dispositif** | 53.44 | 48.46 | 50.83 | 98.63 | 97.59 | 98.11 |
| **I-entete** | 97.91 | 91.93 | 94.83 | 99.51 | 99.55 | 99.53 |
| **Evaluation globale** | 90.63 | 90.63 | 90.63 | 99.48 | 99.48 | 99.48 |
| **I-appelant** | 34.46 | 16.87 | 22.65 | 84.34 | 76.27 | 80.1 |
| **I-avocat** | 85.17 | 98.75 | 91.46 | 98.02 | 98.15 | 98.09 |
| **I-date** | 75.67 | 72.45 | 74.02 | 98 | 96.6 | 97.3 |
| **I-fonction** | 88.81 | 64.46 | 74.7 | 95.23 | 95.13 | 95.18 |
| **I-formation** | 79.38 | 94.38 | 86.23 | 98.8 | 99.45 | 99.12 |
| **I-intervenant** | 82.07 | 38.04 | 51.98 | 83.38 | 68.26 | 75.07 |
| **I-intime** | 50.4 | 68.09 | 57.93 | 82.54 | 83.33 | 82.93 |
| **I-juge** | 73.4 | 88.73 | 80.34 | 97.55 | 97.23 | 97.39 |
| **I-juridiction** | 85.15 | 98.37 | 91.28 | 98.91 | 99.69 | 99.3 |
| **I-rg** | 68.53 | 22.14 | 33.47 | 97.81 | 97.44 | 97.62 |
| **I-ville** | 91.5 | 82.41 | 86.72 | 98.94 | 99.15 | 99.04 |
| **Evaluation globale** | 76.21 | 82.26 | 79.12 | 95.13 | 94.51 | 94.82 |
| **I-norme** | 88.23 | 93.7 | 90.89 | 97.14 | 96.09 | 96.62 |

Tableau 2.7 – Précision, Rappel, $F_1$-mesures pour chaque type d'entité et section au niveau atomique.

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

| | HMM | | | CRF | | |
|---|---|---|---|---|---|---|
| | *Precision* | *Rappel* | $F_1$ | *Precision* | *Rappel* | $F_1$ |
| **corps** | 0.99 | 0.99 | 0.99 | 89.57 | 90.1 | 89.83 |
| **dispositif** | 12.05 | 7.33 | 9.11 | 98.02 | 97.82 | 97.92 |
| **entete** | 10.47 | 10.5 | 10.48 | 92.11 | 92.48 | 92.29 |
| **Evaluation globale** | 7.22 | 6.27 | 6.71 | 93.22 | 93.47 | 93.34 |
| **appelant** | 17.84 | 5.6 | 8.52 | 84.05 | 77.29 | 80.53 |
| **avocat** | 44.29 | 39.15 | 41.56 | 90.97 | 90.3 | 90.63 |
| **date** | 66.87 | 62.15 | 64.43 | 97.96 | 96.6 | 97.27 |
| **fonction** | 89.84 | 64.13 | 74.84 | 96.89 | 96.94 | 96.92 |
| **formation** | 61.5 | 65.86 | 63.61 | 98.4 | 98.95 | 98.68 |
| **intervenant** | 14.29 | 4 | 6.25 | 62.5 | 40 | 48.78 |
| **intime** | 30.28 | 27.47 | 28.8 | 79.31 | 78.93 | 79.12 |
| **juge** | 73.54 | 83.21 | 78.07 | 96.58 | 96.35 | 96.47 |
| **juridiction** | 81.31 | 87.66 | 84.37 | 98.86 | 99.54 | 99.2 |
| **rg** | 68.53 | 22.41 | 33.77 | 97.57 | 98.02 | 97.79 |
| **ville** | 89.52 | 84.7 | 87.05 | 98.85 | 99.15 | 99 |
| **Evaluation globale** | 64.59 | 54.56 | 59.15 | 93.77 | 92.93 | 93.35 |
| **norme** | 71.94 | 78.45 | 75.05 | 92.66 | 91.38 | 92.01 |

Tableau 2.8 – Précision, Rappel, $F_1$-mesures pour chaque type d'entité et section au niveau entité.

## 2.4.5 Discussions

### 2.4.5.1 Confusion de classes

Certaines erreurs sont probablement dues à la proximité des entités de types différents. D'après la matrice de confusion des méta-données d'entête (Figure 2.6 Page 51), les *intervenants* sont parfois mal classifiés comme *intimé* en majorité (17 %), mais aussi comme *appelant* (4 %) ou *avocat* (2 %) probablement parce qu'il s'agit d'entités mentionnées les unes à la suite des autres dans l'entête (les *intervenants* sont généralement mentionnés juste après les *avocats* des *intimés*).

De plus, les intervenants apparaissent dans une très faible proportion de documents annotés. Par ailleurs, une quantité considérable d'*appelants* sont aussi classifiés comme *intimés* (16 %). Ce qui signifie que la transition entre la liste des appelants et celle des intimés est difficilement identifiable avec les descripteurs de mots que nous avons définis. La proximité crée aussi des confusions entre les sections Corps et Dispositif qui se suivent (Figure 2.7 Page 51).

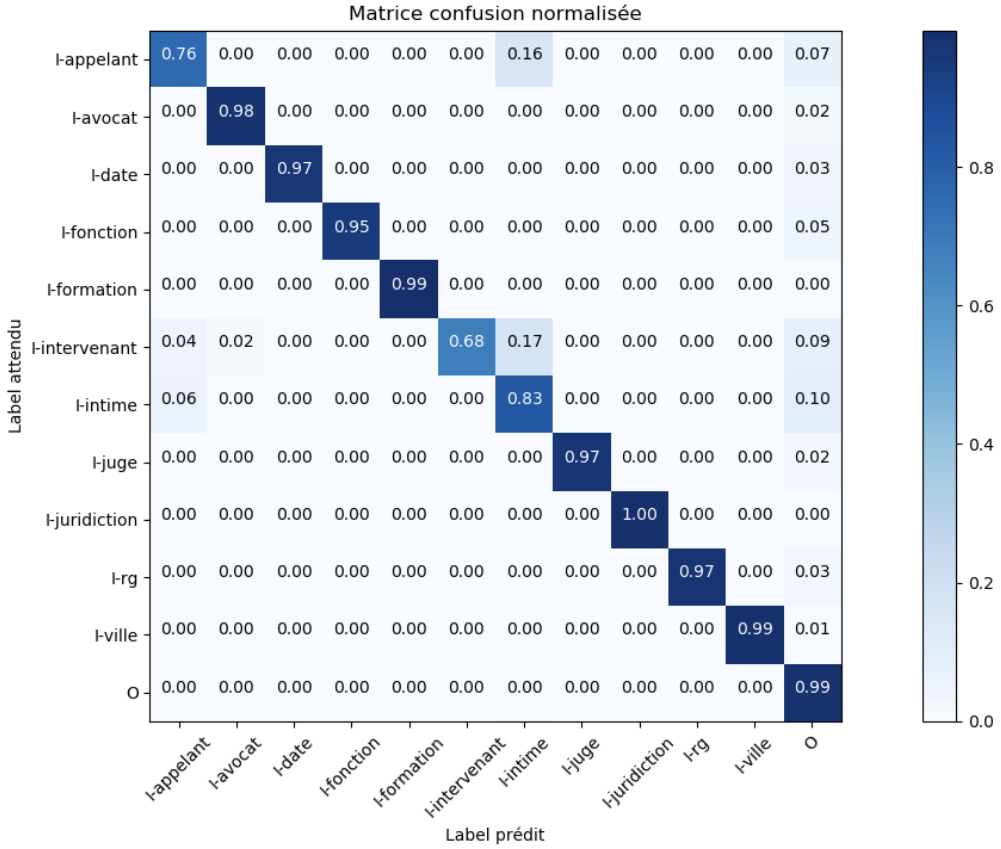



Figure 2.6 – Matrice de confusion entre méta-données d'entête avec le modèle CRF

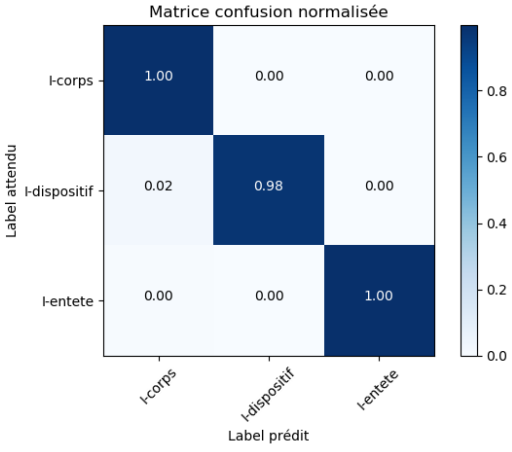



Figure 2.7 – Matrice de confusion entre lignes des sections avec le modèle CRF

#### 2.4.5.2 Redondance des mentions d'entités

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

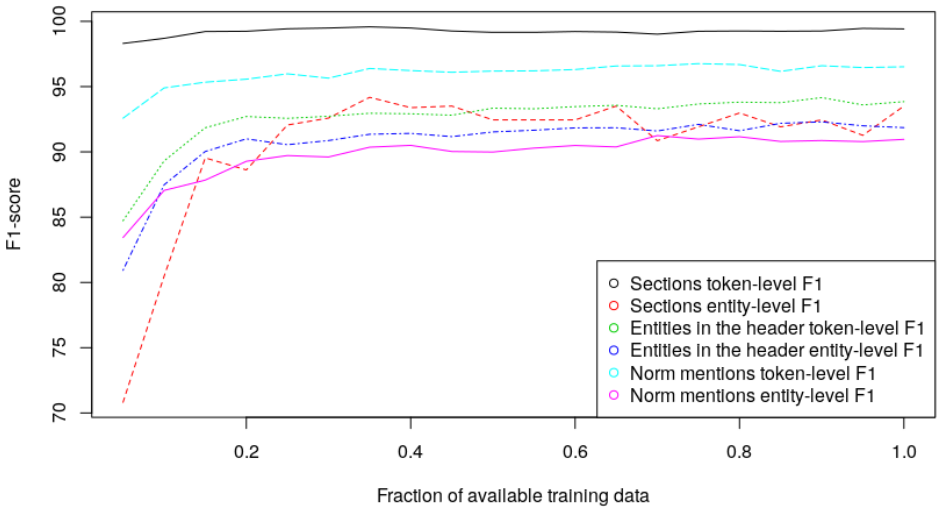
#### 2.4.5.3 Impact de la quantité d'exemples annotés

Des expérimentations ont été menées pour évaluer les variations des modèles lorsque l'on augmente le nombre de données d'entraînement. Pour cela, nous avons évalué différentes tailles de la base d'entraînement. Les données ont été divisées en $75\% - 25\%$ pour resp. l'entraînement et le test. 20 fractions de l'ensemble d'entraînement ont été utilisées (de 5% à 100%). A chaque session entraînement-test, le même jeu de test a été employé pour les différentes fractions de l'ensemble d'entraînement. Les courbes d'apprentissage des modèles CRF et HMM sont représentées resp. sur les Figures 2.8a (Page 53) et 2.8b (Page 53) .

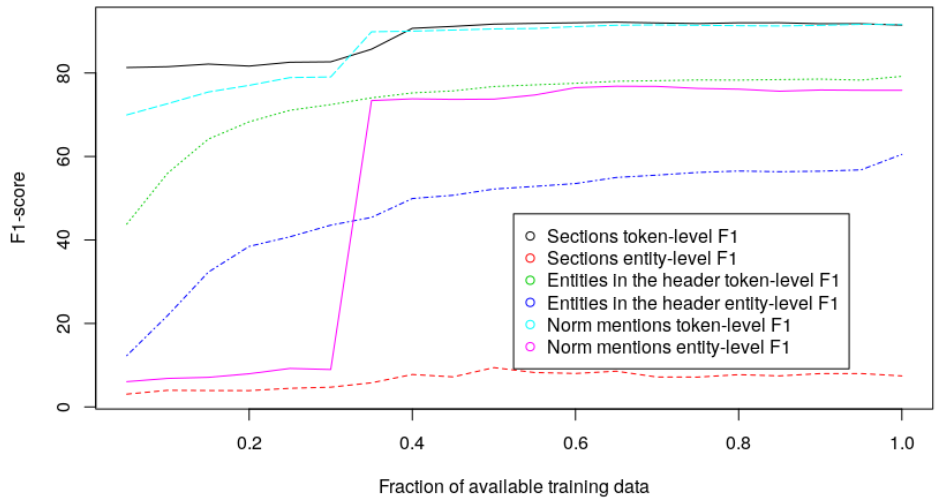
Il apparaît que les scores $F_1$ croissent avec le nombre de données d'entraînement pour les CRF et HMM, mais cette amélioration devient très faible au-delà de 60% de données d'entraînement quelle que soit la tâche. Il est possible que les exemples ajoutés par la suite partagent la même structure que celle de ceux qui ont été ajoutés auparavant. Ainsi, cette étude doit être étendue à la sélection des exemples les plus utiles. Raman & Ioerger [2003] ont démontré les avantages des algorithmes de sélection d'exemples combinés à celle des caractéristiques pour la classification. Les mêmes méthodes sont probablement applicables à l'étiquetage de séquences.

#### 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




(a) CRF




(b) HMM

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

validation croisée à 9 itérations. Le BiLSTM-CRF prend en entrée les plongements sémantiques Word2Vec [Mikolov *et al.*, 2013] des mots. Pour cela, nous avons entraîné des vecteurs de mots à partir d'un corpus jurisprudentiel de plus de 800K documents provenant de `www.legifrance.gouv.fr` avec l'implémentation [5] de Mikolov *et al.* [2013]. 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

5. `https://code.google.com/archive/p/word2vec/`

| | CRF + descripteurs manuels | | | BiLSTM-CRF | | |
|---|---|---|---|---|---|---|
| | *Precision* | *Rappel* | $F_1$ | *Precision* | *Rappel* | $F_1$ |
| **appelant** | 82.49 | 69.42 | 74.72 | 80.26 | 71.53 | 75.04 |
| **avocat** | 90.15 | 89.02 | 89.56 | 84.93 | 87.88 | 86.36 |
| **date** | 95.34 | 91.46 | 93.12 | 95.04 | 90.79 | 92.63 |
| **fonction** | 95.87 | 95.08 | 95.44 | 92.69 | 93.48 | 93.03 |
| **formation** | 96.91 | 91.31 | 93.7 | 91.05 | 89.47 | 89.84 |
| **intervenant** | 51.42 | 32.71 | 36.8 | 31.48 | 20 | 23.11 |
| **intime** | 76.01 | 79.15 | 77.22 | 67.7 | 75.43 | 70.83 |
| **juge** | 95.67 | 94.07 | 94.84 | 95.44 | 95.56 | 95.46 |
| **juridiction** | 98.55 | 98.25 | 98.33 | 97.95 | 99.22 | 98.57 |
| **rg** | 95.46 | 95.29 | 95.27 | 91.13 | 97.26 | 93.92 |
| **ville** | 98.33 | 93.01 | 94.71 | 91.43 | 95.34 | 93.3 |
| **norme** | 91.08 | 90.27 | 90.67 | 91.43 | 92.65 | 92.03 |
| **Evaluation globale** | 92.2 | 90.09 | 91.12 | 89.21 | 90.43 | 89.81 |

Tableau 2.9 – Comparaison entre le CRF avec des descripteurs définis manuellement et le BiLSTM-CRF au niveau entité.

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 2.9 Page 54). Etant donné que ces derniers permettent de mieux détecter certaines entités comme les *intervenants*, les *avocats* ou les numéro *R.G.*, et vice-versa pour les *normes* ou les *appelants* chez le BiLSTM-CRF, une combinaison des deux types de descripteurs pourrait améliorer les résultats actuels.

#### 2.4.5.5 Sectionnement en 4 sections pour l'extraction des demandes

| | CRF (%) | | |
|---|---|---|---|
| | *Precision* | *Rappel* | $F_1$ |
| **entete** | 99.80 | 99.54 | 99.67 |
| **litige** | 96.10 | 97.66 | 96.87 |
| **motifs** | 97.31 | 95.96 | 96.62 |
| **dispositif** | 99.00 | 98.49 | 98.72 |
| **Evaluation globale** | 97.55 | 97.55 | 97.55 |

Tableau 2.10 – Evaluation au niveau atomique de la détection de 4 sections à l'aide du CRF.

Le sectionnement peut permettre de mieux localiser les informations selon leur nature. Pour l'extraction des demandes nous avons dû passer d'un sectionnement à 3 sections comme étudié depuis le début de ce chapitre, vers un sectionnement à 4 sections (cf. Annexe § A.i Page 169). En

effet, la section Corps est remplacé par les deux sections qu'elle englobait : l'exposé du litige (Litige) et les motivations des juges (Motifs). Comme nous le faisons remarquer dans le chapitre 3, les demandes sont généralement énoncées dans la section Litige et les résultats dans le Dispositif. En expérimentant cet affinement du sectionnement par 5-*fold* validation croisée avec les mêmes documents annotés manuellement, les mêmes descripteurs caractéristiques de lignes, le schéma d'étiquetage IO, et le modèle CRF, nous observons les résultats du Tableau 2.10 Page 54 au niveau atomique. L'Entête reste aisément détectable ($F_1$-mesure=99.67%) contrairement aux autres sections qui ont quasiment 3% en moins en moyenne de $F_1$-mesures par rapport à l'entête.

## 2.5 Conclusion

L'application des modèles HMM et CRF dans le but de détecter des sections et des entités dans les décisions de justice est une tâche difficile. Ce chapitre a examiné les effets de divers aspects de la conception sur la qualité des résultats. En résumé, malgré une importante réduction du nombre de descripteurs, l'amélioration des résultats semble être insignifiante lorsque l'on sélectionne séparément la représentation du segment et le sous-ensemble de caractéristiques. Cependant, opter pour la bonne configuration en évaluant les approches de sélection combinées avec diverses représentations de segment pourrait peut-être offrir de meilleurs résultats. En raison de la longue durée de recherche du sous-ensemble optimal de descripteurs, il serait préférable d'utiliser un algorithme de sélection beaucoup plus rapide que les méthodes BDS et SFFS que nous avons expérimentées. De plus, même si les résultats s'améliorent avec l'augmentation de la taille de l'échantillon d'apprentissage, la mesure globale $F_1$ semble atteindre une limite très rapidement. Étant donné que certaines entités ne sont pas très bien détectées, il peut être avantageux d'ajouter des exemples appropriés afin de traiter ces problèmes spécifiques.

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

pas parfaitement à un nouvel ensemble de données (différents pays, différentes langues, différentes juridictions). Pour éviter les énormes efforts requis pour définir les fonctionnalités manuellement, il serait préférable d'utiliser des descripteurs appris automatiquement à partir de corpus étiquetés ou non, comme des mots incorporés.

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

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

# Chapitre 3

# Identification des demandes

***Résumé.*** *Ce chapitre aborde le problème d'identification automatique, dans une décision, des éléments structurants les demandes formulées. L'identification manuelle réalisée à travers la lecture exige beaucoup d'effort à cause de la complexité du contenu dans lequel les demandes sont mélangées à d'autres informations (des demandes de nature différente, des arguments, des faits, etc.). L'automatisation de cette tâche métier vise à aider les experts à rapidement comprendre les réclamations des parties et les réponses correspondantes des juges. Une demande est abstraite par cinq attributs : la norme qui la fonde, son objet, l'interprétation du résultat ($s_r$), le quantum demandé ($q_d$), et celui obtenu ($q_r$). La norme et l'objet forment ensemble la catégorie de la demande. L'annotation manuelle des données d'évaluation suit un protocole que nous avons précisément défini avec l'expert du projet. Ce protocole recommande des cycles d'annotation consistant chacun à constituer un ensemble de décisions et à y identifier toutes les demandes d'une seule catégorie donnée car il serait difficile d'annoter simultanément des données pour toutes les catégories qui sont très nombreuses. L'approche proposée extrait à chaque application les demandes d'une seule catégorie et est formulée en trois tâches. La présence de la catégorie est déterminée par classification de la décision. Ensuite, les quanta et le sens du résultat sont identifiés à proximité de termes appris de la catégorie dans les sections adéquates identifiées à l'aide d'un modèle à base de CRF comme décrit au chapitre 2. Enfin, les demandes sont formées en mettant en correspondance les éléments précédemment déterminés. Réalisées par validation croisées, nos expérimentations comparent une douzaine de méthodes statistiques d'extraction de termes-clés et quatre algorithmes de classification pour l'extraction de 6 catégories prédéfinies de demandes. Les résultats montrent que la détection de catégorie est facile quelque soit l'algorithme utilisé ($F_1$-mesure comprise entre 98.8 % et 100 %). Il résulte aussi que l'extraction des demandes nécessite de sélectionner la méthode d'extraction de terminologie la mieux adaptée à la catégorie. Cette sélection préalable permet d'observer, sur les données de test, des $F_1$-mesures comprises entre 33.09 % et 71.43 % pour les champs $q_d$, $q_r$, et $s_r$, et entre 28.65 % et 58.99 % pour les triplets $(q_d, s_r, q_r)$.*

## 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'issu 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 celles 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 ceux-ci 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 Page 59).

### 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 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.

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

***Source :*** *extraits de la décision 14/01082 de la cour d'appel de Saint-Denis (Réunion).*
***Légende :*** *énoncés simples en gris, références en bleu, et agrégations en marron.*

Figure 3.1 – Illustrations de la complexité des énoncés de demandes et de résultats.

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

#### 3.1.1.4 Quantum obtenu ou résultat

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

| Label | Nom | 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 dans le cadre du projet, et par conséquent, ils n'existent pas dans le langage juridique.*

Tableau 3.1 – Exemples de catégories de demandes

### 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 cf. § 2.4.5.5 et § A.i). 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 dans différents paragraphes qui correspondent soit à une partie, soit à un groupe de parties partageant les mêmes demandes (par ex. des époux). Les paragraphe sont parfois organisés 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 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 résul-

tat 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. ...*") méritent d'être résolues. Par ailleurs, les catégories de demandes sont nombreuses [1] mais ne sont pas toutes présentes à la fois dans les décisions. Tous ces aspects rendent difficile l'annotation manuelle des données de référence et la modélisation d'une approche d'extraction adéquate. Nous avons cependant identifié des 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 ;

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

2. Il existe un grand nombre de catégories (500+) ; ce qui rend difficile l'annotation d'exemples de référence pour couvrir toutes ces catégories.

L'idée est de pouvoir ajouter progressivement de nouvelles catégories. Nous avons par conséquent opté pour une extraction par catégorie. Cette stratégie permet par ailleurs d'ajouter facilement de nouvelles classes sans avoir à redéfinir les classes déjà entraînées. Une exécution du système d'extraction permet ainsi d'extraire les demandes d'une seule catégorie. Le problème est décomposé en deux 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.

## 3.2 Travaux connexes

Chacune des tâches précédentes se rapproche d'une tâche couramment traitée en fouille de texte. En effet, la détection de catégories dans les décisions peut être modélisée comme un problème de classification de documents. La tâche d'extraction se rapproche plus quant à elle 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 Extraction d'éléments structurés

Les demandes ressemblent aux structures telles que les relations ou les évènements. En effet, les champs définis par la compétition d'Extraction Automatique de Contenus ACE (*Automatic Content Extraction*), dans LDC [2008], pour les relations et LDC [2005] pour les évènements, se rapprochent de ceux visés lors de l'extraction des demandes comme l'illustre le Tableau Tableau 3.2 Page 63. 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 [LDC, 2008]** | **Événement [LDC, 2005]** | **Analogie chez les demandes** |
|---|---|---|---|
| **Type** | Org-Aff.Student-Alum | Die | Catégorie="Dommages-intérêts pour procédure abusive" |
| **Passage** (*extend*) | *Card graduated from the University of South Carolina* | "Il est mort hier d'une insuffisance rénale." | (*Figure 3.1*) |
| **Déclencheur (*trigger*)** | - | "mort" | "procédure abusive" |
| **Participants ou Arguments (*arguments*)** | Arg1="Card" Arg2="the University of South Carolina" | Victim-Arg="il" Time-Arg="hier" | Quantum-demandé="3000€" Quantum-obtenu="0 €" |
| **Classes (*attributes, classes*)** | Asserted | Polarity=POSITIVE, Tense=PAST | Sens-résultat="Rejeté" |

Tableau 3.2 – Exemples d'analogie entre relations, évènements et demandes

### 3.2.2 Approches d'extraction d'éléments structurés

L'extraction d'éléments structurés repose généralement sur une approche modulaire du problème qui le décompose en tâches plus simples. D'une part, on dispose de l'identification des déclencheurs [2] et des arguments. D'autre part, une mise en correspondance relie les arguments et déclencheurs qui participent à la même relation ou au même évènement. 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 traitements enchaînant des modules adaptés 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 & Hart, 1967] vs. classificateur d'entropie maximum [Nigam *et al.*, 1999]), pour extraire des champs d'évènements dans le corpus d'ACELDC [2005]. Bien que les différents modules soient plus faciles à développer, ce type d'architecture souffre de 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'évènements, Yang & Mitchell [2016] estiment la probabilité conditionnelle jointe du type d'entité $t_i$, les rôles des arguments $r_{i.}$ et les

2. Terme-clés indiquant la présence d'un évènement [LDC, 2005].

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 de potentiels arguments pour $i$, et $x$ est le document. Cette approche obtient 50.6% de $F_1$-mesure moyenne pour la détection des valeurs d'arguments et 48.4% pour leur classification dans leur rôle respectif. 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. Cette approche obtient 62.8% de $F_1$-mesure moyenne pour la détection des valeurs d'arguments et 55.4% pour leur classification dans leur rôle respectif.

L'annotation du corpus de l'ACE 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 mots »). 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 en temps avec la longueur des documents manipulés. 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 le problème d'identification des demandes judiciaires.

### 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 évènements. 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 calculées à partir des occurrences dans un corpus, à la différence des poids locaux (Tableau 4.1) 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) |
| $\overline{c}$ | la classe complémentaire ou négative |
| $D$ | ensemble global des documents de taille $N = \|D\|$ |
| $D_c$ | ensemble des documents de $c$ de taille $\|D_c\|$ |
| $D_{\overline{c}}$ | ensemble des documents de $\overline{c}$ de taille $\|D_{\overline{c}}\|$ |
| $N$ | nombre total de documents |
| $N_t$ | nombre de documents contenant $t$ |
| $N_{\overline{t}}$ | nombre de documents ne contenant pas $t$ |
| $N_{t,c}$ | nombre de documents de $c$ contenant $t$ |
| $N_{\overline{t},c}$ | nombre de documents de $c$ ne contenant pas $t$ |
| $N_{t,\overline{c}}$ | nombre de documents de $\overline{c}$ contenant $t$ |
| $N_{\overline{t},\overline{c}}$ | nombre de documents de $\overline{c}$ 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 & 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 textes où le but est plutôt de séparer des catégories [Wu *et al.*, 2017]. Elles se formulent comme suit :

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

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 :

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

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

#### 3.2.3.2 Métriques supervisées

Les métriques supervisées mesurent l'information contenue 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 $D_{\overline{c}}$. Elles sont ainsi mieux adaptées à la distinction entre catégories. Parmi les nombreuses métriques existantes, nous avons expérimenté les suivantes :

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

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

**Le gain d'information** $ig$ [Yang & 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 $c$ :

$$ig(t,c) = \begin{array}{l} \frac{N_{t,c}}{N}\log_2\left(\frac{N_{t,c}N}{N_t}\right) + \frac{N_{\bar{t},c}}{N}\log_2\left(\frac{N_{\bar{t},c}N}{N_{\bar{t}}|D_c|}\right) \\ \quad + \frac{N_{t,\bar{c}}}{N}\log_2\left(\frac{N_{t,\bar{c}}N}{N_t|D_{\bar{c}}|}\right) + \frac{N_{\bar{t},\bar{c}}}{N}\log_2\left(\frac{N_{\bar{t},\bar{c}}N}{N_{\bar{t}}|D_c|}\right) \end{array}$$

**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** $\chi^2$ [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 du $\chi^2$ [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 $c$. Une valeur positive de $ngl$ signifie que $t$ est corrélé avec $c$, lorsqu'une valeur négative signifie que $t$ est corrélé à $\bar{c}$.

**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 du label des éléments).

**Le coefficient de Maracuilo** (*mar*) [Marascuilo, 1966] qui se calcule par la formule :

$$mar(t,c) = \frac{\left( \begin{array}{l} (N_{t,c} - N_t N_{t,c}/N)^2 \\ \quad + (N_{t,\bar{c}} - N_t |D_{\bar{c}}|/N)^2 \\ \quad\quad + (N_{\bar{t},c} - |D_c| N_{\bar{t}}/N)^2 \\ \quad\quad\quad + (N_{\bar{t},} - N_{\bar{t}} |D_{\bar{c}}|/N)^2 \end{array} \right)}{N}.$$

C'est un test de proportion multivariée. Nous proposons de l'utiliser pour comparer les proportions d'occurrences d'un terme $t$ dans différents corpus.

**Le « delta lissé d'***idf* **»** , *dsidf* [Paltoglou & Thelwall, 2010], est une version lissée du delta *idf* (*didf*) de Martineau & Finin [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 & 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 favoriser 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 taille variée (nombre de mots). Pour compter rapidement les occurrences des n-grammes des corpus, nous avons utilisé la librairie SML [3] [Harispe *et al.*, 2013] lors des expérimentations.

## 3.3 Méthode

### 3.3.1 Détection des catégories par classification

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 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 classifieur bayésien naïf [Duda *et al.*, 1973], de l'arbre de décision C4.5 [Quinlan, 1993], des *k*-plus-proches-voisins (*k*NN) [Cover & Hart, 1967], de la machine à vecteurs de support (SVM) [Vapnik, 1995]. Ces algorithmes sont décrits en détail dans le chapitre 4 qui est axé sur la classification des documents. 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. 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 termes-clés

Diverses approches d'extraction d'information existent (§ 3.2.2). Il est 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 cou-

3. `http://www.semantic-measures-library.org`

ramment 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'extraction des demandes comme l'illustre la Figure 3.2 Page 70. 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 categorie="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 en 4 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 dans les sections Litige, et ceux des résultats respectivement dans la section Dispositif à l'aide des mots prédéfinis du Tableau 3.4 Page 71.

La recherche de passages à l'aide de listes de termes-clés prédéfinies a déjà été employé par Wyner [2010] pour annoter les énoncés de résultats en considérant toute phrase contenant un terme de jugement : *affirm*, *grant*,

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

Tableau 3.4 – Mots introduisant les énoncés de demandes et de résultats

*deny, reverse, overturn, remand, ...* Les mots du Tableau 3.4 Page 71 sont, en général, des verbes identiques pour les passages exprimant des demandes, des résultats de la décisions ou des résultats de jugements antérieurs. Ils sont à l'infinitif pour les demandes, au présent pour les résultats de la décisions, et au passé pour les résultats antérieurs. Pour localiser un énoncé, nous identifions ces verbes qui marquent le début des énoncés. La fin des énoncés est identifié par le prochain verbe introductif ou le prochain point (« . ») ou point-virgule (« ; »)

#### 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 $D_{\overline{c}}$ 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 chaque document de $D_{\overline{c}}$ à la concaténation des énoncés de demande et résultats contenant des sommes d'argent.
3. A l'aide d'une métrique global $g$, calculer le score des termes du corpus $D_c \cup D_{\overline{c}}$. Ce score est multiplié par le logarithme de la longueur du terme pour favoriser 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 $g'_{norm}(t,c) = \frac{g'(t,c) - \min\limits_{t_k}(g'(t_k,c))}{\max\limits_{t_k}(g'(t_k,c)) - \min\limits_{t_k}(g'(t_k,c))}$.
5. Trier les termes par ordre décroissant de score.

6. Sélectionner les premiers termes qui fournissent les meilleures performances 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 3.4 ;
4. relier les attributs ($q_d, s_r, q_r$) de chaque paire demande-résultat :
   (a) former les paires (énoncé de demande, énoncé de résultat) similaire (nous utilisons la métrique de « la plus longue sous-séquence commune » [Hirschberg, 1977; Bakkelund, 2009])
   (b) pour chaque paire d'énoncés formée, relier les quanta demandés et quanta résultats en considérant que les quanta correspondants apparaissent dans le même ordre dans les deux énoncés.

## 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. Sont discutées 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 3 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 $D_{\overline{c}}$ 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 3.5 ;

| IDENTIFICATION DE LA DECISION | | | DESCRIPTION DE LA PRETENTION | | | | DESCRIPTION DU RESULTAT | |
|---|---|---|---|---|---|---|---|---|
| Type | Ressort | RG | OBJET | NORME | QUANTUM DEMANDE | POSITION | RESULTAT | QUANTUM RESULTAT (obtenu) |
| CA | Lyon | 14/06911 | dommages-intérêts | 700 Code de Procédure Civile | 3 500,00 € | Demandeur initial | rejette | 0,00 € |
| CA | Lyon | 14/06911 | dommages-intérêts | 700 Code de Procédure Civile | 2 000,00 € | Défendeur initial | 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 3.5 – Extrait du tableau d'annotations manuelles des demandes.

La répartition des données d'évaluation est donnée par la Figure 3.3.

Il faut aussi noter que bien que l'annotation manuelle des demandes et des 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), $F_1$-mesure ($F_1$).

**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 $\{q_d, s_r, q_r\}$ sur les demandes extraites d'un corpus annotées $D$ de test. Nous utilisons les métriques traditionnellement employées en extraction d'information : la précision ($Precision_{c,x,D}$), le rappel ($Rappel_{c,x,D}$), et la F1-mesure ($F1_{c,x,D}$). Ces mesures sont définies à partir des nombres de vrais positifs ($TP$), faux positifs ($FP$) et faux négatifs ($FN$) calculés au niveau d'un document $d$ :

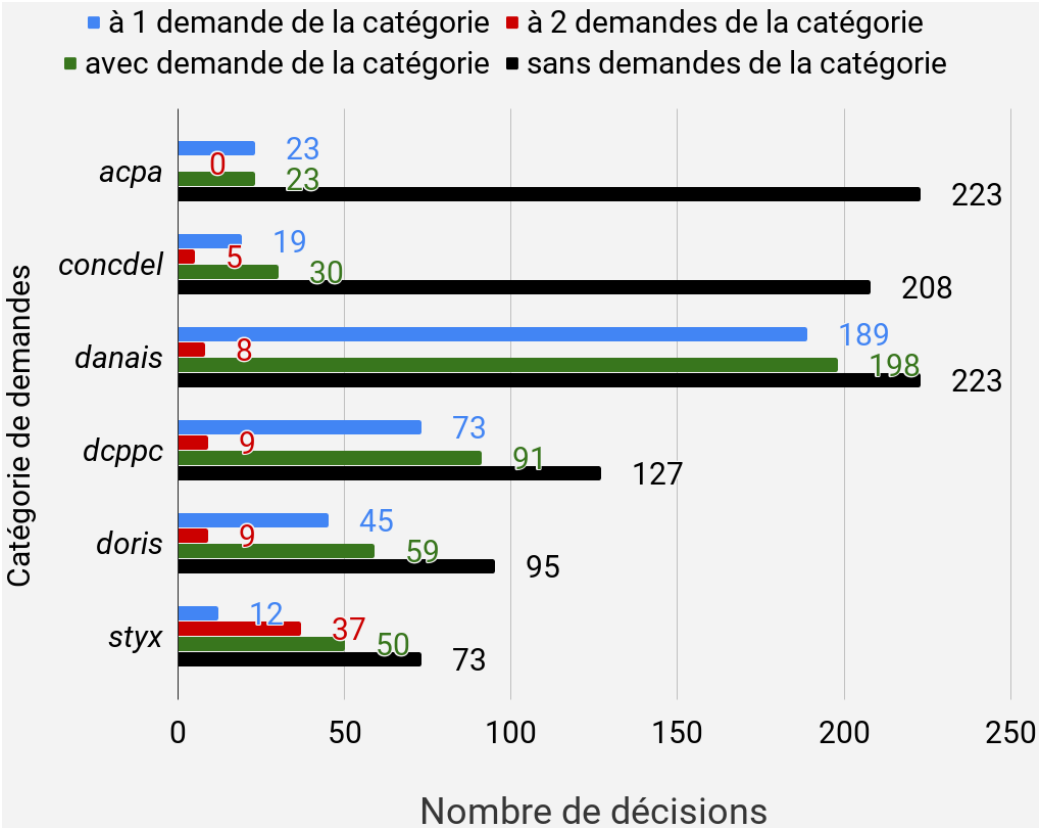



Figure 3.3 – Répartitions des demandes dans les documents annotées.

- $TP_{c,x,d}$ est le nombre de demandes extraites de $d$ par le système, qui sont effectivement de la catégorie $c$ (demandes correctes) ;
- $FP_{c,x,d}$ est le nombre de demandes extraites de $d$ par le système, mais qui ne sont pas des demandes de $c$ (demandes en trop) ;
- $FN_{c,x,d}$ 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 identiques.

### 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 C4.5 (implémenté sous l'appelation J48), les k-plus-proches-voisins (KNN), et le SVM. A chaque entraînement, s'exécute une sélection de modèles 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 3.6 Page 75.

| | NB | | | C4.5 | | | KNN | | | SVM | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $P$ | $R$ | $F_1$ | $P$ | $R$ | $F_1$ | $P$ | $R$ | $F_1$ | $P$ | $R$ | $F_1$ |
| *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, $F_1$ = $F_1$-mesure

Tableau 3.6 – Evaluation de la détection de catégories.

D'après les résultats, la détection de catégorie par classification binaire 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 $\overline{c}$ 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 3.7 Page 76 avec 0.5 comme seuil fixé).

Par conséquent, la métrique et le seuil doivent être bien sélectionnés en fonction de la catégorie de demandes traitée. En choisissant, pour ces hyper-paramètres, les valeurs les plus efficaces pour l'extraction sur la base d'apprentissage, les résultats du Tableau 3.8 Page 77 sont observés. Les améliorations sont à noter notamment pour trois catégories. Le score $F_1$ sur l'extraction des triplets $(q_d, s_r, q_r)$ passe de 54.55 au maximum à

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

Tableau 3.7 – Comparaison des pondérations globales suivant la $F_1$-mesure.

58.99 (plus de 4%) pour *acpa*, de 29.06 à 29.41 pour *dcppc*, de 19.64 à 29.08 (près de 10%) pour *doris*. Les baisses de performances observées pour les autres catégories est comparativement très faibles (moins de 2%).

Ces résultats détaillés font remarquer que les attributs, pris individuellement, présentent d'assez bonnes performances. Cependant, la mise en correspondance des attributs peine toujours à montrer des performances du même rang. On remarque néanmoins que les scores $F_1$ des triplets $(q_d, s_r, q_r)$ sont proches de celles des attributs qui présentent le plus de difficulté. En effet, la sélection préalable permet d'observer sur les données de test, des $F_1$-mesures comprises entre 33.09 % et 71.43 % pour les champs $q_d$, $q_r$, et $s_r$, et entre 28.65 % et 58.99 % pour les triplets $(q_d, s_r, q_r)$. L'échec de l'extraction des attributs est une des principales causes des faibles 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($F_1$-mesures entre 42.12 et 71.43 sauf pour *concdel*) que le quantum demandé $q_d$ ($F_1$-mesures entre 41.75 et 63.61 sauf pour *concdel*). Remarquons aussi que la précision est en général supérieure au rappel ; ce qui signifie que la méthode a plus tendance à éviter les valeurs erronées au risque de manquer un nombre important de valeurs correctes. Enfin, la proportion de documents parfaitement traités (dans lesquels toutes les demandes de la catégorie ont été extraites) reste inférieur à la moyenne même pour *acpa* donc le corpus ne comprend que des décisions à une seules demande de cette catégorie. L'unique terme-clef appris ne semble pas suffisant pour identifier toutes les demandes de *acpa* (l'« article 32-1 » est un bon indicateur par exemple). La catégorie *doris* en-

| | | | Données d'entraînement | | | | Données de test | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $c$ | Données | $\lvert V_c \rvert$ | $P$ | $R$ | $F_1$ | %Docs | $P$ | $R$ | $F_1$ | %Docs |
| *acpa* | $q_d$ | 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 |
| *concdel* | $q_d$ | 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 |
| *danais* | $q_d$ | 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 |
| *dcppc* | $q_d$ | 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 |
| *doris* | $q_d$ | 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 |
| *styx* | $q_d$ | 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, $F_1$ = $F_1$-mesure
%Docs : proportion de documents dont l'ensemble des données extraites est égale à l'attendu (documents parfaitement traités)
$\lvert V_c \rvert$ : nombre moyen de termes-clés identifiés pour la catégorie $c$

Tableau 3.8 – Résultats détaillés pour l'extraction des données avec sélection automatique de la méthode d'extraction des termes-clés

registre la plus faible valeur pour cette proportion, probablement à cause de la présence dans une même décision de plusieurs demandes pour des raisons très variés du trouble du voisinage (préjudice moral, nuisance sonore, préjudice matériel, préjudice de jouissance, etc.) donc malheureusement les termes-clés ne sont pas tous captés par les méthodes statistiques

employées (cf. Tableau 3.12 Page 79).

### 3.4.5 Analyse des erreurs

En extraction d'éléments structurés, on retrouve trois types d'erreurs [Yang & 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 d'après nos résultats, il est certain que les erreurs sont majoritairement dues aux données manquées comme le confirme le Tableau 3.9 Page 78.

| | 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 3.9 – 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 (pourcentages inférieurs à 100 dans les Tableaux 3.10 et 3.11). Par exemple, les quanta résultat de *doris* sont plus présents dans les 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 3.10 – 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 3.11 – Taux de quanta accordés ($q_r$) mentionnés dans les documents annotés

En second, la sélection des termes-clés n'est pas parfaite (Tableau 3.12). 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 utilisé 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 |
|---|---|
| *acpa* | amende civile |
| *concdel* | 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 |
| *danais* | 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 |
| *dcppc* | 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, |
| *doris* | 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 |
| *styx* | 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 3.12 – Premiers termes sélectionnés lors de la première itération de la validation croisée

Enfin, 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. Ces références aux décisions antérieures représentent une part importante des demandes des 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 la terminologie de la catégorie des demandes à extraire et autres connaissances du domaine judiciaire telles que les motifs d'énoncés de demandes et de résultats, 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. Même si nos résultats ont été obtenus à partir de terminologies apprises, une liste de termes fournis par les experts pourrait être plus précise et mettrait à l'abris des biais liés aux échantillons d'apprentissage. A cause de la forte dépendance aux subtilités de rédaction des décisions judiciaires, la méthode proposée rencontre des limites qui ne peuvent être surmontées qu'en la rendant 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 longs documents annotés.

# Chapitre 4

# Identification du sens du résultat

***Résumé.*** *L'extraction des demandes a été présentée dans le chapitre précédent comme l'identification du quantum demandé, du quantum résultat et du sens du résultat pour chaque demande d'une catégorie donnée. Le sens du résultat est l'information la plus importante pour le métier car elle donne une idée du taux d'acceptation des demandes dans les tribunaux pour une analyse descriptive du sens du résultat. Il a été proposé d'identifier le sens du résultat en interprétant simplement le verbe de décision de l'énoncé du résultat. Cette technique a l'inconvénient de dépendre de l'identification explicite du passage exprimant le résultat qui est donc une source supplémentaire d'erreurs. Le présent chapitre adresse cet inconvénient en reformulant l'identification du sens du résultat par la classification binaire (« accepte » / « rejette ») de la décision représentée en entrée sous forme vectorielle. Une décision pouvant comprendre plusieurs demandes de la catégorie traitée, nous ne traitons que le cas des décisions à une seule demande de la catégorie. Cependant, le défi de ce problème est le déséquilibre observé entre les 2 classes sur les données d'apprentissage. Nous observons en effet que la tendance est de rejeter une forte majorité des demandes, ce qui serait aussi le cas en général en justice pour la majorité des catégories. Sur la base de l'expertise d'une partie de l'encadrement de cette thèse, nous proposons deux adaptations de la méthode Gini-PLS de Mussard & Souissi-Benrejab [2018] pour la classification de textes. Nos expérimentations la compare à d'autres algorithmes de classification et sous différentes conditions de représentation dont divers modèles vectoriels et restriction du document à des sous-parties. Parmi les algorithmes expérimentés, Les meilleures résultats sont obtenus avec les arbres de classification. Nos propositions obtiennent néanmoins des scores d'évaluation proches de ceux des arbres.*

## 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 résul-

tat, 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 entrainer un algorithme à reconnaitre 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 basés sur nos observations des données annotées manuellement.

**Postulat 4.1.1** *Pour toute catégorie de demande, les décisions ne contenant qu'une demande de cette catégorie sont très largement majoritaires.*

Ce postulat est légitime car les statistiques sur les données labellisées de la Figure 3.3 Page 74 montrent bien que dans chaque catégorie, les décisions contiennent en majorité une seule demande d'une même catégorie. En effet, dans les données annotées, les décisions à une demande sont au nombre de 23 sur 23 (100%) pour *acpa*, 19 sur 30 (63,33%) pour *concdel*, 189 sur 198 (95,45%) pour *dnais*, 73 sur 91 (80.22%) pour *dcppc*, et 73 sur 91 (76,21%) pour *doris*. On remarque néanmoins l'exception de la catégorie STYX (dommage-intérêt sur l'article 700 CPC), où dans la majorité des 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 d'une catégorie peuvent se retrouver dans des décisions comprenant plus d'une demande de cette catégorie. Pour la catégorie *concdel* par exemple, 58 demandes ont été annotées manuellement (Figure 4.1 Page 83) mais 19 décisions sur 30 (63,33%) ont une seule demande de cette catégorie (Figure 3.3 Page 74). Ces 19 décisions comprennent donc seulement 32,75% ($100 \times 19/58$) des demandes annotées pour *concdel*. Par conséquent, 67.24% de ces demandes sont dans les décisions annotées ayant plus d'une demande de cette catégorie. 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 les sens de résultat ont majoritairement l'une de ces deux valeurs (Figure 4.1 Page 83). Les autres valeurs sont très rares.

Cette étude porte sur l'analyse de l'impact de différents aspects techniques en général impliqués dans la classification de textes qui consistent en général en une combinaison de représentations des documents et d'algorithmes de classification. Cette analyse permettra de savoir s'il existe une certaine configuration permettant de déterminer le sens du résultat à une demande sans identifier précisément cette dernière dans le document.

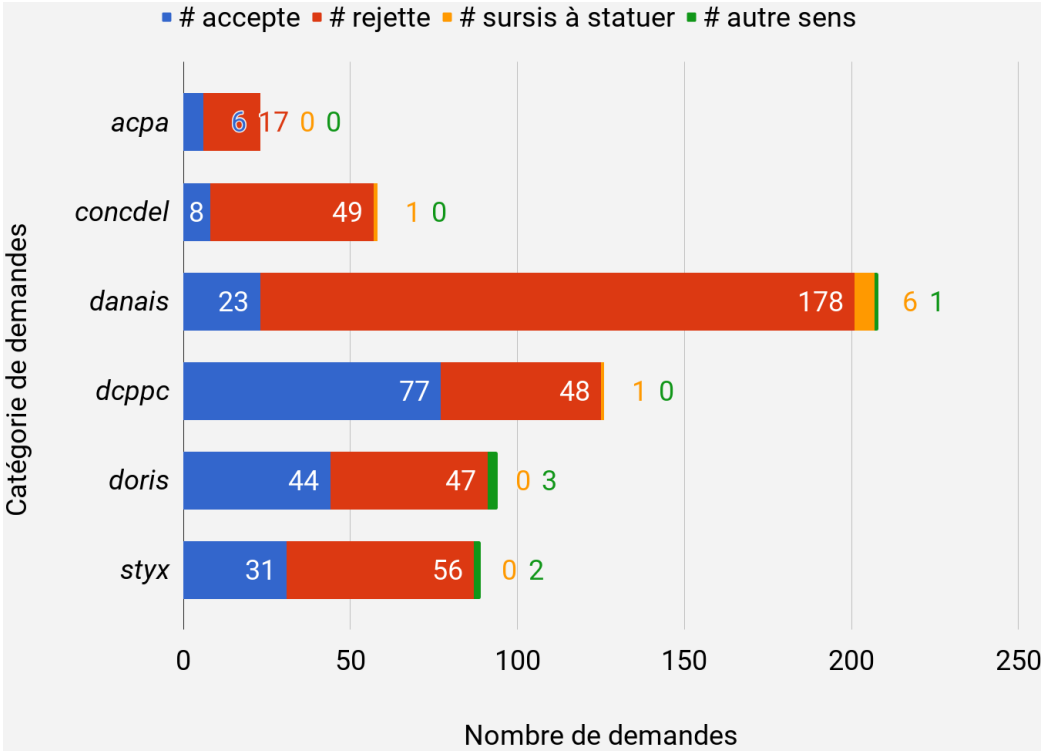



Figure 4.1 – Répartition des sens de résultat dans les données annotées.

Nous proposons l'algorithme Gini-PLS généralisé qui est une extension du modèle Gini-PLS simple. Il s'agit d'un nouveau modèle dans lequel un paramètre de régularisation va permettre de mieux adapter la régression aux informations se situant en queues de distribution tout en atténuant, comme dans le Gini-PLS simple, l'influence exercée par les valeurs aberrantes. Nous proposons également une nouvelle régression (LOGIT-Gini-PLS) qui est mieux adaptée à l'explication d'une variable cible lorsque cette dernière est une variable binaire. Ces deux modèles n'ont par ailleurs jamais été appliqués à de la classification de textes.

## 4.2 Classification de documents

La classification de textes permet d'organiser des documents dans des groupes prédéfinis. Elle reçoit depuis longtemps beaucoup d'attention. Deux choix techniques influencent principalement les performances : la représentation des textes et l'algorithme de classification. Dans la suite, la variable à prédire est notée $y$, et la base d'apprentissage comprend les

observations de l'échantillon $D = \{(x_i, y_i)_{i=1..n}\}$. $C$ représente l'ensemble des classes. Les notations du Tableau 3.3 sont utilisées dans cette section.

### 4.2.1 Représentation de textes

Chaque document $d$ est représenté sous une forme vectorielle du type TF-IDF (*term frequency - inverse document frequency*) proposé par Salton & Buckley [1988] dont chaque dimension $k$ est identifiée par un terme $t_k$. Tout document $d \in D$ est une séquence de mots $d = (d[1], \ldots, d[|d|])$, où $d_i$ est le mot à la position $i$ dans $d$. Sa représentation vectorielle est notée $\vec{d} = (\vec{d}[1], \vec{d}[2], \ldots, \vec{d}[m])$. Pour un modèle vectoriel de type TF-IDF de vocabulaire $V = \{t_1, t_2, \ldots, t_m\}$, $\vec{d}[k] = w(t_k, d)$ le poids du terme $t_k \in T$ dans le texte $d$ (cf. § 3.3.1). Le poids $w(t_k, d)$ affecté à ce dernier est le produit normalisé d'un poids global $g(t_k)$ de $t_k$ dans le corpus d'entraînement 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 euclidienne $\sqrt{\sum_k (w(t_k, d))^2}$. Le poids global est calculé à partir d'une des méthodes de la section § 3.2.3. Le poids local est calculé à partir de la fréquence d'occurrence du terme dans le document à l'aide d'une des méthodes du Tableau 4.1 Page 84.

| Description | Formule |
|---|---|
| Décompte brute du terme [Salton & Buckley, 1988] | $tf(t,d) =$ nombre d'occurrences de $t$ dans $d$ |
| Présence du terme [Salton & 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 & 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.*, 2009b] (avg représente la moyenne) | $logave(t,d) = \frac{1+\log tf(t,d)}{1+\log \operatorname{avg}_{t \in T} tf(t,d)}$ |

Tableau 4.1 – Métriques locales de pondération de termes.

### 4.2.2 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 dans des contextes détachés des considérations applicatives. Ces méthodes sont généralement basée sur une représentation des textes dans un espace vectoriel $\mathcal{X}$ d'entrée et délimitent une frontière entre les classes dans un espace multidimensionnel.

#### 4.2.2.1 Le classifieur bayésien naïf (NB)

Le classifieur naïf bayésien [Duda *et al.*, 1973] est un modèle probabiliste qui estime la probabilité qu'un texte appartienne à une classe à l'aide du théorème de Bayes [Raschka, 2014] :

$$\text{probabilité a posteriori} = \frac{\text{probabilité conditionnelle} \cdot \text{probabilité a priori}}{\text{évidence}}$$

La probabilité a posteriori peut être interprétée pour la classification de documents par la question "Quelle est la probabilité que le document $d$ soit de la classe $y = c \in C$?". La réponse à cette question se formalise comme suit :

$$P(y = c|d) = \frac{P(d|c)P(c)}{P(d)}, \forall c \in C$$

ou plus simplement $P(y = c|d) \propto P(c)P(d|c)$ car $P(d)$ ne change pas en fonction de la classe et peut donc être ignorée [Rish, 2001]. $d$ est catégorisé dans la classe $c$ pour laquelle $P(c|d)$ est maximale :

$$y = \underset{c \in C}{\text{argmax}}\, P(c|d).$$

La phase d'entrainement, appliquée à des exemples déjà labellisés, permet d'estimer les paramètres $P(c)$ et $P(d|c)$ qui servent à calculer $P(c|d)$.

$P(c)$ est estimée par la proportion de documents classés dans $c$ parmi les exemples d'apprentissage : $P(c) = \frac{|D_c|}{N}, \forall c \in C$.

$P(c|d)$ est estimé grâce l'hypothèse 4.2.1 d'indépendance conditionnelle des descripteurs (termes). Une hypothèse naïve dont la violation, par les données réelles, n'empêche pas le NB de bien fonctionner [Rish, 2001].

**Hypothèse 4.2.1 (indépendance conditionnelle des descripteurs)** *[Un modèle naïf bayésien étant de type génératif], étant donnée la catégorie du texte, la position de chaque mot dans le texte est générée indépendamment de tout autre mot.*

Si l'ensemble des termes de $d$ est $\{t_1, \ldots, t_m\} \subset V$ (vocabulaire), alors grâce à l'hypothèse 4.2.1, $P(d|c) = P(t_1, \ldots, t_m|c) = Prod_{k=1}^{m} P(t_k|c)$, et pour un terme $t_k$, la probabilité conditionnelle $P(t_k|c)$ est la proportion d'exemples de $c$ qui contiennent $t_k$ : $P(t_k|c) = \frac{N_{t_k,c}}{|D_c|}, \forall k \in \{1, \ldots, m\}$.

#### 4.2.2.2 Machine à vecteurs de support (SVM)

La classification binaire par une machine à vecteurs de support (SVM) [Vapnik, 1995] affecte à tout objet en entrée $x$ la classe $y$ qui correspond au coté d'un hyperplan, séparant les exemples d'entraînement des classes candidates, où $x$ se trouve. La phase d'apprentissage consiste à déterminer l'hyperplan optimal $w^T x + b = 0$ i.e. dont la marge [1] est maximale (Figure 4.2 [2]).

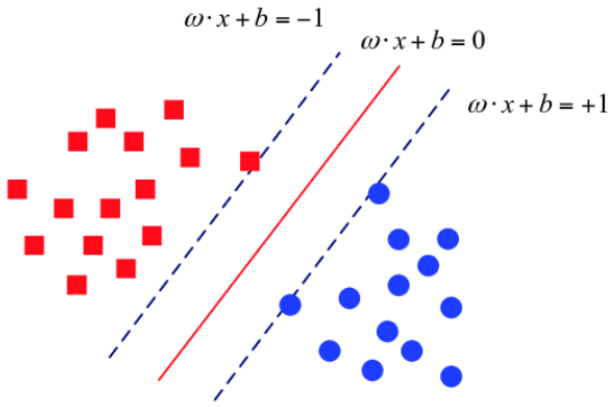



Figure 4.2 – Hyperplan optimal et marge maximale d'un SVM.

Le vecteur $w$ des poids des caractéristiques et le biais $b$ sont déterminés par le problème d'optimisation du « SVM à marges molles » de Cortes & Vapnik [1995] :

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2}\|w\|^2 + \mathbf{C}\sum_{i=1}^{N}\xi_i \\ \text{s.c.} \quad & y_i(w^T + b) \geq 1 - \xi_i, \xi_i \geq 0. \end{aligned}$$

où $N$ est le nombre de données d'entraînement, $\mathbf{C}$ est la constante prédéfinie de régularisation pour éviter un sur-apprentissage ou un sous-

1. La plus petite distance entre les exemples d'apprentissage et l'hyperplan séparateur.
2. `http://web.mit.edu/6.034/wwwbob/svm-notes-long-08.pdf`

apprentissage, les $\xi_i$ sont des variables ressort (*slack variables*) qui permettent à des points de se retrouver dans la marge.

Lorsque les exemples d'apprentissage des classes sont linéairement séparables, la classe d'un objet x correspond au signe de la fonction de décision $f(x) = w^T x + b$ :

$$y = signe(f(x)) = \begin{cases} -1 & \text{si } w^T x + b < 0 \\ +1 & \text{si } w^T x + b > 0. \end{cases}$$

Cependant, ils ne le sont pas toujours dans l'espace $\mathcal{X}$. Ainsi, une fonction « noyau » (*kernel*) $K : \mathcal{X} \longrightarrow \mathcal{F}$ doit être choisie pour transformer chaque donnée entrée $x$ de l'espace original $\mathcal{X}$ vers un nouvel espace $\mathcal{F}$ dit de caractéristiques dans lequel les classes sont linéairement séparables. Par conséquent, la fonction de classification s'écrit

$$f(x) = \sum_{i=1}^{N} \alpha_i K(x, x_i) + b$$

où les $\alpha_i$ sont les coefficients de la combinaison linéaire des exemples d'apprentissage égale à $w$ ($w = \sum_{i=1}^{N} \alpha_i x_i$) [Ben-Hur & Weston, 2010]. Parmi les multiples formes qu'il peut prendre, le noyau peut être, par exemple, soit linéaire($K(x, x_i) = x^T x_i + c$), soit polynomial ($K(x, x_i) = (\gamma x^T x_i + c)^d$), soit Gaussien ou RBF[3] ($K(x, x_i) = \exp -\gamma \|x - x_i\|$), soit une sigmoïde ($K(x, x_i) = \tanh(\gamma x^T x_i + c)$) [Amami *et al.*, 2013].

#### 4.2.2.3 k-plus-proches-voisins (kNN)

L'algorithme des $k$-plus-proches-voisins [Cover & Hart, 1967] est un algorithme simple qui consiste à affecter à un nouvel objet $x$ la classe majoritaire $y'$ parmi ceux des $k$ points d'exemples d'entrainement $\{(x_i, y_i)\}_{1 \leq i \leq k}$, les plus proches du point $x$ selon la fonction distance choisie. Ainsi, trois éléments clés influencent l'efficacité de la classification :

1. Si les données d'entraînement sont trop nombreuses, le processus de classification peut devenir coûteux en temps de calcul. En effet, la distance du nouvel objet à chaque point annoté est calculée.
2. Le nombre de voisins (c'est-à-dire la valeur de $k$) ne doit être trop petit pour limiter la sensibilité aux bruits / *outliers*. Il ne doit non plus être trop grand au risque d'avoir dans le voisinage trop de points d'une autre classe que celle attendue.

3. *radial base function*

3. La métrique de calcul de distance doit être adéquate pour le type de donnée et la tâche. Par exemple, la distance cosinus est souvent préférable à la distance euclidienne pour la classification de documents. En effet, la distance euclidienne se dégrade lorsque le nombre de dimensions augmente [Sohangir & Wang, 2017; Aggarwal *et al.*, 2001].

#### 4.2.2.4 Arbre de décision

Un arbre de décision est une structure arborescente qui associe un label prédéfini à des objets (classification), ou prédire la valeur d'une variable continue (régression). Il comprend des nœuds 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 nœuds terminaux qui correspondent chacun à une prédiction. La classification d'un objet $x$ consiste à faire passer successivement les tests en fonction des valeurs des attributs de $x$, de la racine jusqu'à une feuille dont le label est retourné comme classe de $x$ (Algorithme 3).

**Algorithme 3 :** Classification par arbre de décision

**Données :** Objet $x$, Arbre $A$
**Résultat :** label
$n := racine(A)$ ;
**tant que** *$n$ n'est pas une feuille* **faire**
    Effectuer sur $x$ le test associé à $n$;
    $n :=$ noeud fils de $n$ correspondant au résultat du test ;
**retourner** le label associé à la feuille $n$;

La construction de l'arbre (phase d'apprentissage) consiste à générer une hiérarchie de tests, aussi courte que possible, qui divise successivement l'ensemble $D$ d'exemples d'apprentissage en sous-ensembles disjoints de plus en plus purs [4]. L'arbre est construit de la racine aux feuilles en divisant les données d'entraînement $S_t$ à chaque nœud $(t)$ de sorte à minimiser le degré d'impureté des sous-ensembles d'exemples $S_{t_i}$ dans les nœuds fils $(t_i)$. Le critère de coupe est généralement défini à partir d'une métrique d'impureté comme par exemple :

- l'entropie de la distribution des classes dans $S_t$ :
  $h_C(S_t) = -\sum_{c \in C} [P(c|S_t) \log_2 P(c|S_t)]$ ;

4. homogénéité des labels

- 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 $c$. Parmi les critères de séparation les plus populaires associés à ces métriques d'impureté, 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 la moyenne des entropies des fils de $t$ :
  $ig(S_t, a) = h_C(S_t) - i(S_t, t, a) = h_C(S_t) - \sum_i \frac{|S_{t_i}|}{|S_t|} \cdot h_C(S_{t_i})$;
- 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 $S_{t_i}$ :
  $gr(S_t, t, a) = \frac{ig(S_t,t,a)}{h_t(S_t)}$ 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 que pour 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, soit uniquement en des partitions binaires suivant des tests booléens nécessitant de rechercher la division optimale. Les variables numériques sont divisées quant à elles soit par discrétisation de leur domaine en les transformant en variables catégoriques ordinales, soit en recherchant la meilleure division binaire parmi toutes les séparations possibles.

La construction de l'arbre est une division récursive qui peut continuer tant qu'il est possible d'améliorer la pureté des nœuds, ce qui peut engendrer un arbre très grand résultant en un sur-apprentissage [5], et une forte complexité temporelle et spatiale lors de la prédiction. Pour s'arrêter

5. 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 sur les données de test (erreur de test).

plus tôt ("pré-élagage"), plusieurs conditions sont possibles comme par exemple, l'atteinte d'un seuil minimum par la taille des données ($|S_t|$), ou l'atteinte par l'arbre d'une profondeur maximale, ou l'amélioration du critère de division est très faible, etc. Le post-élagage [6] est appliqué après construction de l'arbre toujours dans le but de minimiser le sur-apprentissage et la complexité. Le post-élagage peut être basé, par exemple, soit sur la réduction du taux d'erreur (éliminer successivement les feuilles si cela ne fait pas croître le taux d'erreur sur les données d'entrainement), soit sur la stratégie coût-complexité de Breiman *et al.* [1984].

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. Singh & Gupta [2014] comparent les deux algorithmes CART [Breiman *et al.*, 1984] (critère de « doublage », élagage coût-complexité) et C4.5 [Quinlan, 1993] (rapport des gains, élagage à réduction d'erreur).

#### 4.2.2.5 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 $|C|$ catégories ou variables qualitatives. Les analyses linéaires et quadratiques sont des méthodes probabilistes basées sur la probabilité conditionnelle d'appartenance d'un objet $x \in \mathbb{R}^m$ à une classe $c_k \in C$ :

$$P(y = c_k|x) = \frac{P(y = c_k)P(x|y = c_k)}{P(x)} = \frac{P(y = c_k)P(x|y = c_k)}{\sum_{j=1}^{|C|} P(y = c_j)P(x|y = c_j)}.$$

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

1. L'hypothèse de normalité statue que la probabilité conditionnelle $P(x|y)$ suit une loi normale multidimensionnelle :

$$P(x|y = c_k) = \mathcal{N}(\mu_k, V_k) = \frac{1}{\sqrt{(2\pi)^m det(V_k)}} e^{-\frac{1}{2}(x-\mu_k)^\top V_k^{-1}(x-\mu_k)},$$

6. Suppression de sous-arbres superflus après génération de l'arbre.

$\mu_k$ étant le centre de gravité conditionnel (centre de gravité des données d'entraînement de la classe $c_k \in C$), et $V_k$ la matrice de variance-covariance de la classe $c_k$. Le logarithme de $P(x|c_k)$ donne : $\ln(P(x|c_k)) \propto -\frac{m}{2}\ln(2\pi) - \frac{1}{2}\ln(det(V_k)) - \frac{1}{2}(x-\mu_k)^\top V_k^{-1}(x-\mu_k)$ [Ghojogh & Crowley, 2019]. Grâce à la proportionnalité de la probabilité conditionnelle à son logarithme, on déduit une fonction discriminante proportionnelle à $P(c_k|x)$ :

$$\delta_1(x, c_k) = -\frac{1}{2}\ln(det(V_k)) - \frac{1}{2}(x-\mu_k)^\top V_k^{-1}(x-\mu_k) + \ln(P(c_k))$$

en éliminant le terme $-\frac{m}{2}\ln(2\pi)$ car il est le même pour toutes les classes.

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

$$\forall j, k \in \{1, ..., |C|\}, V_j = V_k = V.$$

Cette hypothèse permet de simplifier $\delta_1(x, c_k)$ en :

$$\begin{aligned}\delta_2(x,k) &= -\tfrac{1}{2}\ln(det(V_k)) - \tfrac{1}{2}x^\top V^{-1}x - \mu_k^\top V^{-1}\mu_k + \mu_k^\top V^{-1}x + \ln(P(c_k)) \\ &= \mu_k^\top V^{-1}x - \tfrac{1}{2}\mu_k^\top V^{-1}\mu_k + \ln(P(c_k))\end{aligned}$$

car les termes $-\frac{1}{2}\ln(det(V_k))$ et $-\frac{1}{2}x^\top V^{-1}x$ sont indépendants des classes.

L'analyse discriminante linéaire (LDA) [Fisher, 1936] est définie à partir de la simplification de $P(x|c_k)$ sous ces deux hypothèses. Ainsi la classe de $x$ est $y = \underset{k\in\{1,...,|C|\}}{\text{argmax}}\ \delta_2(x, c_k)$. L'analyse discriminante quadratique (QDA) [McLachlan, 1992] quand à elle ne considère pas l'hétéroscédasticité (i.e. $\exists k \neq j, V_k \neq V_j$), et ne s'appuie que sur l'hypothèse de normalité. La classe de $x$ est par conséquent $y = \underset{k\in\{1,...,|C|\}}{\text{argmax}}\ \delta_1(x, c_k)$.

### 4.2.3 Algorithmes dédiés aux textes

Les algorithmes dédiés aux textes intègrent leur propre représentation de document, contrairement aux algorithmes opérant sur des espaces vectoriels aux axes et poids paramétrables à volonté comme le SVM.

#### 4.2.3.1 NBSVM

Le NBSVM [Wang & Manning, 2012] est un classifieur binaire qui consiste à appliquer une interpolation du classifieur naïf bayésien multinomial (MNB) et du SVM à représentation spécifique des textes entrés. Tout document entré est représenté par un vecteur $x = (x_1, x_2, \ldots, x_m)$ tel que $\forall i \in [1 \,..\, m], x_i$ est le nombre d'occurrences du terme $t_i \in V$. $x$ est d'abord transformé en un vecteur $\hat{x} = (\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_m)$ tel que :

$$\forall i \in [1 \,..\, m], \hat{x}_i = \mathbb{1}(x_i) = \begin{cases} 1 & \text{si } x_i > 0 \\ -1 & \text{sinon.} \end{cases}$$

$\hat{x}$ est ensuite transformé en $\tilde{x} = (\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_m)$ tel que :

$$\forall i \in [1 \,..\, m], \tilde{x}_i = \hat{r} \circ \hat{x}_i \text{ (produit par composant)}$$

où le vecteur $\hat{r}$ est défini par :

$$\hat{r} = \log\left(\frac{\hat{p}/||\hat{p}||_1}{\hat{q}/||\hat{q}||_1}\right)$$

avec $\hat{p} = \alpha + \sum_{i:\hat{x}_i=1} \hat{x}_i$ et $\hat{q} = \alpha + \sum_{i:\hat{x}_i=-1} \hat{x}_i$. La valeur de $\alpha$ est prédéfinie.

La classe de $x$ est prédite par : $y = signe(\mathbf{w}^\top \tilde{x} + b)$, avec $\mathbf{w} = (1 - \beta)\overline{w} + \beta w$ où $\overline{w} = ||w||_1/|V|$ et $\beta$ est un hyper-paramètre de valeur prédéfinie dans $[0; 1]$. $\mathbf{w}$ définie l'interpolation entre le MNB et le SVM pour que la classification par le MBN soit préférée à moins que le SVM soit très confiant. $w$ et $b$ sont appris lors de l'entraînement du SVM.

#### 4.2.3.2 fastText

Le classifieur fastText [Grave *et al.*, 2017] est un réseau de neurones dont l'architecture est donnée à la Figure 4.3 Page 93 [Zolotov & Kung, 2017]. Tout document $d$ à classifier est représenté un vecteur $x$ qui est la moyenne des vecteurs des mots de $d$ [Zolotov & Kung, 2017] :

$$x = \frac{1}{|d|}\sum_{i=1}^{|d|} \overrightarrow{d[i]}.$$

La couche cachée construit un vecteur $z = (z_1, z_2, \ldots, z_{|C|})$ présenté en entrée de la fonction de classification *softmax*. Cette dernière calcule la probabilité d'appartenance du document à chaque classe candidate :

$$\forall j \in [1 \,..\, |C|], P(c_j) = \frac{e^{z_j}}{\sum_{k=1}^{|C|} e^{z_k}}.$$

Ainsi, la classe de $d$ est $y = \underset{j \in \{1,\ldots,|C|\}}{\text{argmax}}\ P(c_j)$. Par ailleurs, $z = B \cdot x$ où $B$ est la matrice $|C| \times n$ des poids de la couche cachée ; $n$ étant le nombre d'unités de cette dernière. $B$ est apprise par l'entraînement du réseau sur un jeu de données annotées.

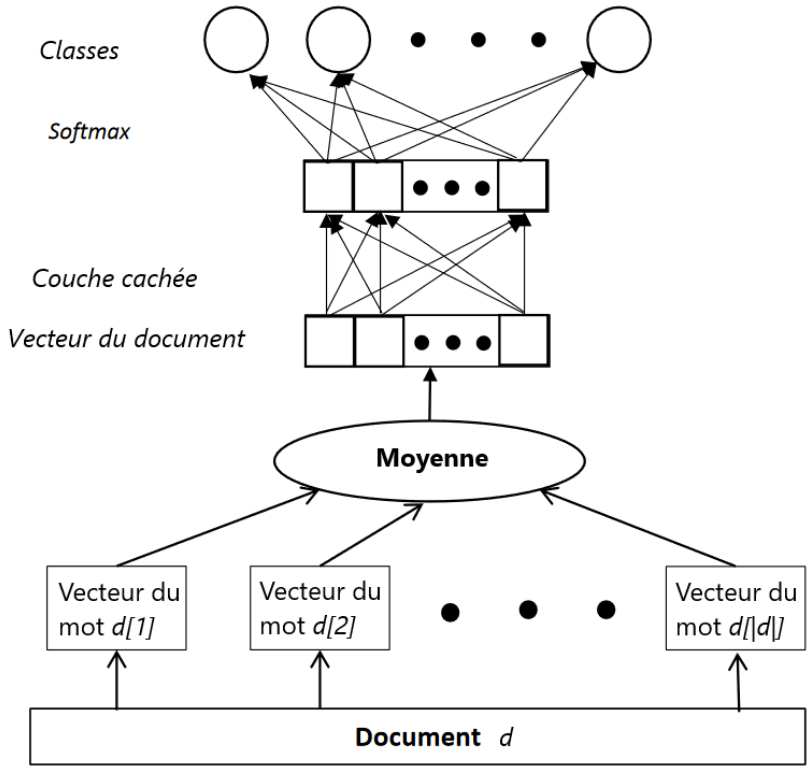



Figure 4.3 – Architecture du classifieur fastText.

### 4.2.4 Techniques d'amélioration de l'efficacité

La faible quantité [Ruparel *et al.*, 2013] et le déséquilibre des données sont susceptibles d'être des obstacles à l'entraînement des modèles de classification. De nombreuses techniques permettent néanmoins d'optimiser l'apprentissage en fonction des données. La sélection de modèle consiste à choisir les meilleures valeurs des hyper-paramètres (par exemple $C$ et $\gamma$ chez le SVM) en testant différentes combinaisons de valeurs candidates sur une fraction de la base d'entraînement (base de développement). La combinaison de classifieurs est aussi une méthode très étudiée [Kittler *et al.*, 1996; Kuncheva, 2004; Tulyakov *et al.*, 2008] notamment, par exemple, forêts aléatoires [Breiman, 2001], ou SVM ensembliste (*Ensemble SVM*) [Dong

& Han, 2005]. Par ailleurs, la représentation vectorielle des textes résulte généralement en des vecteurs de haute dimension dont les coordonnées sont en majorité nulles. Par conséquent, les techniques de réduction de dimensions, comme les analyses discriminantes, permettent de d'obtenir des vecteurs plus pertinents pour la classification.

## 4.3 Adaptations de la régression Gini-PLS pour la classification des textes

Cette section présente nos deux adaptations de la régression Gini-PLS [Mussard & Souissi-Benrejab, 2018] : une généralisation du Gini-PLS et une combinaison de cette dernière à la régression logistique. L'intérêt du Gini-PLS est de réduire la sensibilité aux valeurs aberrantes. C'est une extension de l'analyse des moindres carrés partiels PLS (*partial least square*) [Wold, 1966]. L'analyse PLS explique la dépendance entre une ou plusieurs variables $y$ (dite dépendantes) et des variables $x_1, x_2, ..., x_m$ (dites explicatives). Elle consiste principalement à transformer les variables explicatives en un nombre réduit de $h$ composantes principales orthogonales $t_1, ..., t_h$. Il s'agit donc d'une méthode de réduction de dimension au même titre que l'analyse en composantes principales, l'analyse discriminante linéaire (LDA), et l'analyse discriminante quadratique (QDA). Les composantes $t_h$ sont construites par étapes en appliquant l'algorithme du PLS de façon récurrente sur les données mal prédites (résidus). Plus précisément, à chaque itération $h$, la composante $t_h$ est calculée par la formule $t_h = w_{h1}x_1 + \cdots + w_{hj}x_j + \cdots + w_{hp}x_K$ dont les coefficients $w_{hj}$ sont à estimer. L'analyse PLS présente plusieurs avantages [Lacroux, 2011] dont la robustesse au problème de haute-dimension[7] et l'élimination du problème de multicolinéarité[8][Kroll & Song, 2013]. Ces problèmes sont susceptibles de survenir sur les petits corpus de textes comme dans notre cas. La méthode PLS est étendue et appliquée avec succès pour divers problèmes de régression [Lacroux, 2011] ou de classification de données en général [Liu & Rayens, 2007; Durif *et al.*, 2017; Bazzoli & Lambert-Lacroix, 2018], et de textes en particulier [Zeng *et al.*, 2007].

7. Lorsque le nombre de variables explicatives est très grand devant le nombre d'exemples d'entraînement ($N << m$).

8. La multicolinéarité est un problème qui survient lorsque certaines variables de prévision du modèle mesurent le même phénomène.

### 4.3.1 L'opérateur Gini covariance

Soit $\bar{x}_k$ la moyenne arithmétique de la variable explicative $x_k, \forall k \in [1 \,..\, m]$ sur les $N$ observations d'apprentissage. L'opérateur de Gini covariance proposé par Schechtman & Yitzhaki [2003], encore appelé opérateur co-Gini est donné par :

$$\text{cog}(x_\ell, x_k) := \text{cov}(x_\ell, F(x_k)) = \frac{1}{N}\sum_{i=1}^{N}(x_{i\ell} - \bar{x_\ell})(F(x_{ik}) - \bar{F}_{x_k}), \qquad (4.1)$$

où $F(x_k)$ est la fonction de répartition de $x_k$, $\bar{F}_{x_k}$ sa moyenne, avec $\ell \neq k = 1, \ldots, m$. Lorsque $k = \ell$ le co-Gini mesure la variabilité entre une variable et elle-même (l'équivalent de la variance mesurée sur la norme $\ell_2$). Le co-Gini est une mesure basée sur la distance de Manhattan (distance de métrique $\ell_1$), en effet :

$$\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}|x_{ik} - x_{jk}| = 4\text{cog}(x_k, x_k).$$

D'autre part, lorsque $k \neq \ell$, le co-Gini produit une mesure de la variabilité jointe entre deux variables. Puisque le co-Gini n'est pas symétrique :

$$\text{cog}(x_k, x_\ell) := \text{cov}(x_k, F(x_\ell)) = \frac{1}{N}\sum_{i=1}^{N}(x_{ik} - \bar{x_k})(F(x_{i\ell}) - \bar{F}_{x_\ell}).$$

Définissons les rangs croissants d'une variable aléatoire afin de fournir un estimateur de $F$,

$$R_\uparrow(x_{i\ell}) := NF(x_{i\ell}) = \begin{cases} \#\{x \leq x_{i\ell}\} & \text{si aucune observation similaire} \\ \frac{\sum_{i=1}^{p}\#\{x \leq x_{i\ell}\}}{p} & \text{s'il existe } p \text{ valeurs similaires } x_{i\ell}. \end{cases}$$

Alors, un estimateur du co-Gini est donné par,

$$\widehat{\text{cog}}(x_\ell, x_k) := \frac{1}{N}\sum_{i=1}^{N}(x_{i\ell} - \bar{x_\ell})(R_\uparrow(x_{ik}) - \bar{R}_{\uparrow_{x_k}}),\ \forall k, \ell = 1, \ldots, m, \qquad (4.2)$$

avec $\bar{R}_{\uparrow_{x_k}}$ la moyenne arithmétique du vecteur rang de la variable $x_k$.

### 4.3.2 Gini-PLS

Le premier algorithme Gini-PLS a été proposé par Mussard & Souissi-Benrejab [2018]. Nous le décrivons dans les lignes qui suivent. Il s'agit

d'une méthode de compression avec débruitage qui consiste à réduire les dimensions de l'espace généré par $X$ afin de trouver des composantes principales débruitées, dans le même esprit qu'une ACP débruitée, néanmoins l'approche est supervisée dans la mesure où une variable cible $y$ est prise en compte dans le changement d'espace. Le sous-espace formé par les composantes principales $\{t_1, t_2, \cdots\}$ est construit de telle sorte que le lien entre les variables explicatives $X = [x_1, x_2, \cdots, x_m]$ et la cible $y$ est maximisé.

• **Étape 1 :** La régression Gini permet de concevoir un nouveau type de liens entre la variable expliquée et les variables explicatives tout en évitant l'influence des valeurs aberrantes. Ceci est possible grâce notamment à l'opérateur co-Gini dans lequel le rôle de la variable explicative est remplacé par celui de son vecteur rang dans un espace muni d'une métrique $\ell_1$. Ainsi, il est possible de créer un nouveau vecteur de poids $w_1$ qui renforce le lien (co-Gini) entre la variable expliquée $y$ et les régresseurs $X$ dans le cadre d'une régression (linéaire ou non linéaire).
La solution du programme,

$$\max \operatorname{cog}(y, Xw_1) \text{ , s.c. } \|w_1\| = 1 \text{ , est}$$

$$w_{1j} = \frac{\operatorname{cog}(y, x_j)}{\sqrt{\sum_{k=1}^{m} cog^2(y, x_k)}} \text{ , } \forall j = 1 \ldots, p \text{ .}$$

La pondération est équivalente à :

$$w_{1k} = \frac{\operatorname{cov}(y, R(x_k))}{\sqrt{\sum_{k=1}^{m} \operatorname{cov}^2(y, R(x_k))}} \text{ , } \forall k = 1 \ldots, m \text{ .}$$

Comme dans la régression PLS, on régresse $y$ sur la composante $t_1$ qui est construite de la manière suivante :

$$t_1 = \sum_{k=1}^{m} w_{1k} x_k \implies y = \hat{c}_1 t_1 + \hat{\varepsilon}_1 \text{ .}$$

• **Étape 2 :** On régresse le vecteur rang de chaque régreseur $R(x_k)$ sur la composante $t_1$ par moindres carrés ordinaires afin de récupérer les résidus $\hat{U}_{(1)k}$ :

$$R(x_k) = \hat{\beta} t_1 + \hat{U}_{(1)k} \text{ , } \forall k = 1, \cdots, m \text{ .}$$

On construit le nouveau vecteur de pondération en utilisant les rangs des résidus des régressions partielles :

$$\max \operatorname{cog}(\hat{\varepsilon}_1, \hat{U}_{(1)} w_2) \text{ , s.c. } \|w_2\| = 1 \implies w_{2k} = \frac{\operatorname{cog}(\hat{\varepsilon}_1, \hat{U}_{(1)k})}{\sqrt{\sum_{k=1}^{m} \operatorname{cog}^2(\hat{\varepsilon}_1, \hat{U}_{(1)k})}}.$$

On utilise à présent les composantes $t_1$ et $t_2$ pour établir un lien entre $y$ et les régresseurs $x_k$ :

$$t_2 = \sum_{k=1}^{m} w_{2k} \hat{U}_{(1)k} \implies y = \hat{c}_1 t_1 + \hat{c}_2 t_2 + \hat{\varepsilon}_2 \, .$$

La validation croisée (décrite après l'étape $h$) permet de savoir si $t_2$ est significative.

• **Étape $h$ :** Les régressions partielles sont réitérées en ajoutant l'influence de $t_{h-1}$ :

$$R(x_k) = \beta t_1 + \cdots + \gamma t_{h-1} + \hat{U}_{(h-1)k} \text{ , } \forall k = 1, \cdots, m.$$

D'où, après maximisation :

$$w_{hk} = \frac{\operatorname{cog}(\hat{\varepsilon}_{h-1}, \hat{U}_{(h-1)k})}{\sqrt{\sum_{k=1}^{m} \operatorname{cog}^2(\hat{\varepsilon}_{h-1}, \hat{U}_{(h-1)k})}} \, ,$$

$$t_h = \sum_{k=1}^{m} w_{hk} \cdot \hat{U}_{(h-1)k} \implies y = \alpha_2 + c_1 t_1 + \cdots + c_h t_h + \varepsilon_h \, .$$

La procédure s'arrête lorsque la validation croisée indique que la composante $t_h$ n'est pas significative. L'algorithme Gini-PLS est valable si toutes les composantes $t_h$ et $t_l$ sont orthogonales, $\forall h \neq l$.

La validation croisée permet de trouver le nombre optimal $h > 1$ de composantes à retenir. Pour tester une composante $t_h$, on calcule la prédiction du modèle avec $h$ composantes comprenant l'observation $i$, $\hat{y}_{h_i}$, puis sans l'observation $i$, $\hat{y}_{h_{(-i)}}$. L'opération est répétée pour tout $i$ variant de 1 à $N$ : on enlève à chaque fois l'observation $i$ et on ré-estime le modèle [9].

9. Les observations peuvent être éliminées par blocs *Cf.* Tenenhaus [1998], p. 77.

Pour mesurer la robustesse du modèle, on mesure l'écart entre la variable prédite et la variable observée :

$$PRESS_h = \sum_{i=1}^{N} \left( y_i - \hat{y}_{h_{(-i)}} \right)^2 .$$

La somme des carrés résiduels obtenue avec le modèle à $(h-1)$ composantes est :

$$RSS_{h-1} = \sum_{i=1}^{N} \left( y_i - \hat{y}_{(h-1)_i} \right)^2 .$$

Le critère $RSS_h$ (*Residual Sum of Squares*) du modèle à $h$ composantes et $PRESS_h$ (*PRedicted Error Sum of Squares*) sont comparés. Leur rapport permet de savoir si le modèle avec la composante $t_h$ améliore la prédictibilité du modèle :

$$Q_h^2 = 1 - \frac{PRESS_h}{RSS_{h-1}} .$$

La composante $t_h$ est retenue si : $\sqrt{PRESS_h} \leq 0,95\sqrt{RSS_h}$. Autrement dit, lorsque $Q_h^2 \geq 0,0975 = (1 - 0,95^2)$, la nouvelle composante $t_h$ est significative, elle améliore la prévision de la variable $y$. Pour la significativité de la composante $t_1$, on utilise :

$$RSS_0 = \sum_{i=1}^{N} (y_i - \bar{y})^2 .$$

### 4.3.3 Régression Gini-PLS généralisée

Schechtman & Yitzhaki [2003] ont récemment généralisé l'opérateur co-Gini afin d'imposer plus ou moins de poids en queue de distribution. Notons $r_k = (R_\downarrow(x_{1k}), \ldots, R_\downarrow(x_{Nk}))$ le vecteur rang décroissant de la variable $x_k$, autrement dit, le vecteur qui assigne le rang le plus petit (1) à l'observation dont la valeur est la plus importante $x_{ik}$ :

$$R_\downarrow(x_{ik}) := \begin{cases} N + 1 - \#\{x \leq x_{ik}\} & \text{pas d'observation similaire} \\ N + 1 - \frac{\sum_{i=1}^{p} \#\{x \leq x_{ik}\}}{p} & \text{si } p \text{ observations similaires } x_{ik}. \end{cases}$$

L'opérateur co-Gini est généralisé grâce au paramètre $\nu$ :

$$\text{cog}_\nu(x_\ell, x_k) := -\nu \text{cov}(x_\ell, r_k^{\nu-1});\ \nu > 1. \tag{4.3}$$

Afin de bien comprendre le rôle de l'opérateur co-Gini, revenons sur la mesure du coefficient de corrélation linéaire généralisé au sens de Gini :

$$GC_\nu(x_\ell, x_k) := \frac{-\nu \text{cov}(x_\ell, r_k^{\nu-1})}{-\nu \text{cov}(x_\ell, r_\ell^{\nu-1})} ;\ GC_\nu(x_k, x_\ell) := \frac{-\nu \text{cov}(x_k, r_\ell^{\nu-1})}{-\nu \text{cov}(x_k, r_k^{\nu-1})}.$$

**Property 1 – Schechtman & Yitzhaki [2003] :**
(i) $GC_\nu(x_\ell, x_k) \leq 1$.
(ii) *Si les variables $x_\ell$ et $x_k$ sont indépendantes, pour tout $k \neq \ell$, alors* $GC_\nu(x_\ell, x_k) = GC_\nu(x_k, x_\ell) = 0$.
(iii) *Une transformation monotone des données $\varphi$ n'affecte pas le coefficient de corrélation,* $GC_\nu(x_\ell, \varphi(x_k)) = GC_\nu(x_\ell, x_k)$.
(iv) *Pour une transformation linéaire $\varphi$,* $GC_\nu(\varphi(x_\ell), x_k) = GC_\nu(x_\ell, x_k)$ [*comme le coefficient de corrélation de Pearson*].
(v) *Si $x_k$ et $x_\ell$ sont deux variables échangeables à une transformation linéaire près, alors* $GC_\nu(x_\ell, x_k) = GC_\nu(x_k, x_\ell)$.

Le rôle de l'opérateur co-Gini peut être expliqué de la manière suivante. Lorsque $\nu \to 1$, la variabilité des variables est atténuée de telle sorte que $\text{cog}_\nu(x_k, x_\ell)$ tend vers zéro (même si les variables $x_k$ et $x_\ell$ sont fortement corrélées). Au contraire, si $\nu \to \infty$ alors $\text{cog}_\nu(x_k, x_\ell)$ permet de se focaliser sur les queues de distribution $x_\ell$. Comme le montrent Olkin & Yitzhaki [1992], l'emploi de l'opérateur co-Gini atténue la présence d'outliers, du fait que le vecteur rang agit comme un instrument dans la régression de $y$ sur $X$ (régression par variables instrumentales).

Ainsi, en proposant une régression Gini-PLS basée sur le paramètre $\nu$, nous pouvons calibrer la puissance du débruitage grâce à l'opérateur co-Gini qui va localiser le bruit dans la distribution. Cette régression Gini-PLS généralisée devient une régression Gini-PLS régularisée où le paramètre $\nu$ joue le rôle de paramètre de régularisation.

#### 4.3.3.1 L'algorithme Gini-PLS généralisé

Dans ce qui suit nous généralisons la régression Gini-PLS de Mussard & Souissi-Benrejab [2018] avec renforcement du pouvoir de débruitage par l'intermédiaire du paramètre $\nu$.

La première étape consiste à trouver des poids de débruitage associés à chaque variable $x_k$ afin d'en déduire la première composante $t_1$ (ou première variable latente). Cette opération est bouclée jusqu'à la composante $t_{h^*}$, où $h^*$ est le nombre optimal de variable latentes. Ainsi, le modèle est estimé :

$$y = \sum_{h=1}^{h^*} c_h t_h + \varepsilon_h. \tag{4.4}$$

La statistique $VIP_{hj}$ est mesurée afin de sélectionner la variable $x_j$ qui a l'impact significatif le plus important sur le $y$ estimé. Les variables expli-

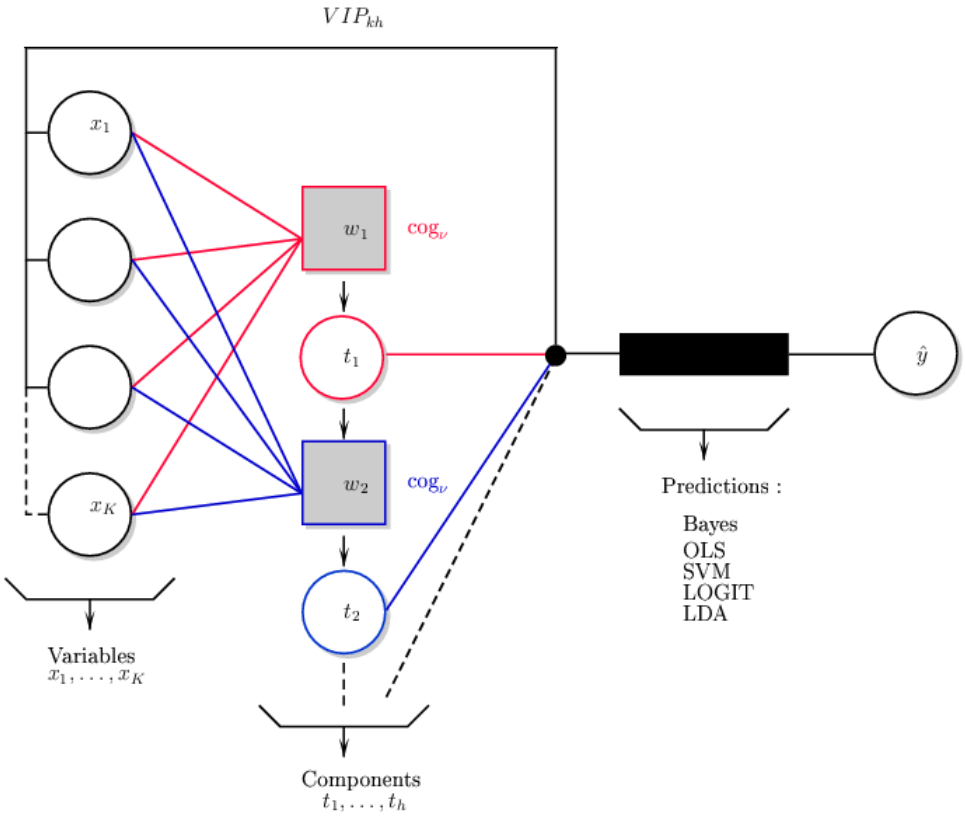



Figure 4.4 – Principe de l'algorithme Gini-PLS généralisé.

catives les plus significatives sont celles dont $VIP_{hj} > 1$ avec :

$$VIP_{hj} := \sqrt{\frac{m \sum_{\ell=1}^{h} Rd(y; t_\ell) w_{\ell j}^2}{Rd(y; t_1, \ldots, t_h)}}$$

et

$$Rd(y; t_1, \ldots, t_h) := \frac{1}{m} \sum_{\ell=1}^{h} \text{cor}^2(y, t_\ell) =: \sum_{\ell=1}^{h} Rd(y; t_\ell).$$

où $\text{cor}^2(y, t_\ell)$ est le coefficient de corrélation de Pearson entre $y$ et la composante $t_\ell$. Cette information est rétro-propagée dans le modèle (une seule fois) afin d'obtenir les variables latentes $t_{h^*}$ et leurs coefficients estimés $\hat{c}_{h^*}$ sur les données d'entraînement. La variable cible $y$ est ensuite prédite grâce à la formule (4.4) de la page 99. Cette formule a une valeur continue qui est transformée en valeur booléenne par,

$$classe(x) = \begin{cases} 0 & \text{si } y < 0.5 \\ 1 & \text{sinon.} \end{cases}$$

**Algorithme 4 :** Gini-PLS Généralisé (entraînement)

**Données :** $X$ (observations), $h_{max}$ (nombre maximum de composantes), $\nu_{max}$ (valeur maximale du paramètre $\nu$)
**Résultat :** Composantes principales $t_1, \ldots, t_{h^*}$

**répéter**
  **répéter**
    $\max \text{cog}_\nu(y, w_h X)$ s.t. $\|w_h\| = 1 \Longrightarrow$ poids $w_h$ de $X$ ;
    MCO équation : $y = \sum_h c_h t_h + \varepsilon_h$ ;
    MCO équation : $R(x_j) = \sum_h \beta_h t_h + \epsilon_k \; \forall k = 1, \ldots, m$ ;
    $X := (\hat{\epsilon}_1, \ldots, \hat{\epsilon}_m)$ ;
    $y := \hat{\varepsilon}_h$ ;
  **jusqu'à** $h = h_{max}$ $[h = h + 1]$;
  Mesurer $VIP_{kh}$, $Q_h^2$ ;
  Sélectionner le nombre optimal de composantes $h^*$ ;
**jusqu'à** $\nu = \nu_{max}$ $[\nu = \nu + 1]$;
Déduire le paramètre optimal $\nu^*$ qui minimise l'erreur ;
**retourner** $t_1, \ldots, t_{h^*}$;

#### 4.3.3.2 L'algorithme LOGIT-Gini-PLS généralisé

Comme nous le constatons dans l'algorithme Gini-PLS généralisé que nous avons proposé dans la section précédente, les poids $w_j$ proviennent de l'opérateur co-Gini appliqué à une variable booléenne $y \in \{0; 1\}$. Afin de trouver les poids $w_j$ qui maximisent le lien entre les variables $x_j$ et la variable cible $y$, nous proposons d'utiliser la régression LOGIT, autrement dit, une sigmoïde qui est mieux adaptée aux variables booléennes. Ainsi, dans chaque étape de la régression Gini-PLS nous remplaçons la maximisation du co-Gini par la mesure de la probabilité conditionnelle suivante :

$$P(y_i = 1/X = X_i) = \frac{\exp\{X_i\beta\}}{1 + \exp\{X_i\beta\}} \qquad \text{(LOGIT)}$$

où $X_i$ est la $i$-ème ligne de la matrice $X$ (observation des caractéristiques/-dimensions de la décision juridique $i$). L'estimation du vecteur $\beta$ se fait par maximum de vraisemblance. On en déduit alors les pondérations $w_j$ :

$$w_j = \frac{\beta_j}{\|\beta\|}$$

L'algorithme LOGIT-Gini-PLS généralisé est donc le suivant :

**Algorithme 5 :** LOGIT-Gini-PLS Généralisé (entraînement)

**Données :** $X$ (observations), $h_{max}$ (nombre maximum de composantes), $\nu_{max}$ (valeur maximale du paramètre $\nu$)
**Résultat :** Composantes principales $t_1, \ldots, t_{h^*}$

**répéter**
  **répéter**
    LOGIT équation : $\Longrightarrow$ poids $w_j$ de $X$ ;
    MCO équation : $y = \sum_h c_h t_h + \varepsilon_h$ ;
    $X := (\hat{\epsilon}_1, \ldots, \hat{\epsilon}_K)$ ;
    $y := \hat{\varepsilon}_h$ ;
  **jusqu'à** $h = h_{max}$ $[h = h + 1]$;
  Mesurer $VIP_{kh}$, $Q^2_h$ ;
  Sélectionner le nombre optimal de composantes $h^*$ ;
**jusqu'à** $\nu = \nu_{max}$ $[\nu = \nu + 1]$;
Déduire le paramètre optimal $\nu^*$ qui minimise l'erreur ;
**retourner** $t_1, \ldots, t_{h^*}, \nu^*$;

## 4.4 Expérimentations et résultats

Nous discutons ici les performances de divers algorithmes populaires et l'impact de la quantité et du déséquilibre des données, de l'heuristique, et de la restriction explicite des documents aux passages relatifs à la catégorie de demandes, ainsi que leur capacité à faire abstraction des autres demandes du document. Ces expériences visent aussi à comparer l'efficacité du Gini-Logit-PLS par rapport à d'autres analyses discriminantes. Comme Im *et al.* [2017], nous comparons différentes combinaisons d'algorithmes de classification et méthodes de pondération de termes (utilisées pour la représentation des textes). Ces combinaisons représentent un plus de 600 configurations expérimentées dont :

- 12 algorithmes de classification : NB, SVM, KNN, LDA, QDA, Arbre, fastText, NBSVM, Gini-PLS (Gini-PLS généralisé), Logit-PLS [Tenenhaus, 2005], GiniLogitPLS (LOGIT-Gini-PLS généralisé), Standard-PLS (le PLS standard) ;
- 11 pondérations globales de termes : (cf. §3.2.3) : $\chi^2$, $dbidf$, $\Delta_{DF}$, $dsidf$, $gss$, $idf$, $ig$, $mar$, $ngl$, $rf$, $avg_{global}$ (moyenne des métriques globales) ;

- 6 pondérations locales de termes (cf. Tableau 4.1 Page 84) : $tf$, $tp$, $logtf$, $atf$, $logave$, et $avg_{local}$ (moyenne des métriques locales).

### 4.4.1 Protocole d'évaluation

Deux métriques d'évaluation sont utilisées : la précision et la $F_1$-mesure. Pour tenir compte du déséquilibre entre les classes, la macro-moyenne est préférée. Il s'agit de l'agrégation de la contribution individuelle de chaque classe. Elle est calculée à partir des macro-moyennes de la précision ($P_{macro}$) et du rappel ($R_{macro}$) sont calculées en fonction des nombres moyens de vrais positifs ($\overline{TP}$), faux positifs ($\overline{FP}$), et faux négatifs ($\overline{FN}$) comme suit [Van Asch, 2013] : $P_{macro} = \frac{\overline{TP}}{\overline{TP}+\overline{FP}}$, $R_{macro} = \frac{\overline{TP}}{\overline{TP}+\overline{FN}}$.

Les données utilisées sont une restriction, des données du chapitre précédent, aux documents n'ayant qu'une seule demande annotée pour chacune des catégories de demande. Le déséquilibre entre les classes est illustré par la Figure 4.5. En effet, les demandes sont en majorité rejetées pour les catégories *acpa*, *concdel*, *danais* et *styx*. Le contraire est observé pour *dcppc*, et le rapport est légèrement équilibré pour *doris*.

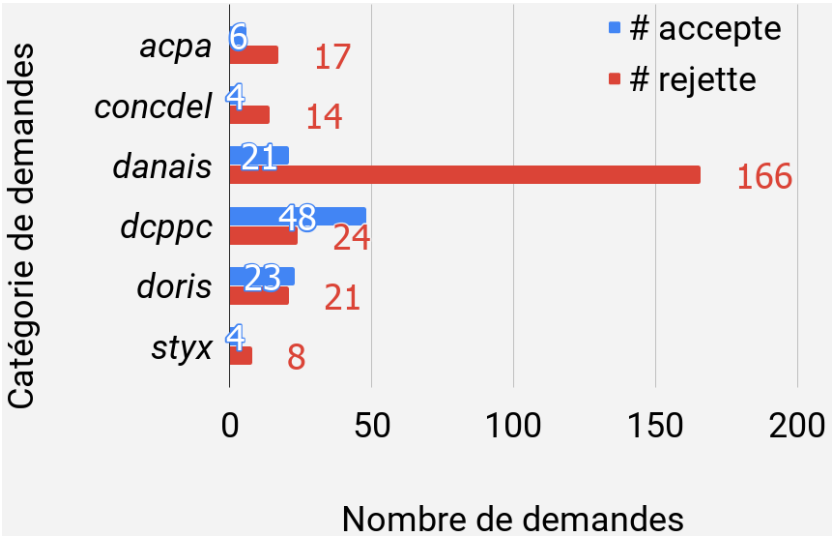



Figure 4.5 – Répartition des documents entre *accepte* et *rejette*.

L'efficacité des algorithmes dépend souvent des méta-paramètres dont il faut déterminer des valeurs optimales. La librairie *scikit-learn* [Pedregosa *et al.*, 2011] implémente deux stratégies de recherche des ces valeurs : RandomSearch et GridSearch. Malgré la rapidité de la méthode RandomSearch, elle est non déterministe et les valeurs qu'elle trouve donnent une

prédiction moins précise que les valeurs par défaut. Idem pour la méthode GridSearch, qui est très lente, et donc peu pratique face au grand nombre de configurations à évaluer. Par conséquent, les valeurs utilisées pour les expérimentations sont les valeurs définies par défaut (Tableau 4.2).

| Algorithmes | Hyper-paramètres |
|---|---|
| SVM | $C = 1.0; \gamma = \frac{1}{\lvert V \rvert \times var(X)}; noyau = RBF$ (fonction de base radiale) |
| KNN | $k = 5$ |
| LDA | $solver = svd, n\_components = 10$ |
| QDA | |
| Arbre | critère de séparation=gini |
| NBSVM | $n$-grammes de 1 à 3 mots |
| Gini-PLS | $h_{max} = 10$ |
| Logit-PLS | $h_{max} = 10$ |
| Gini-Logit-PLS | $h_{max} = 10; \nu = 14$ |

Tableau 4.2 – Valeurs utilisées pour les hyper-paramètres des algorithme.

### 4.4.2 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 4.3 que les arbres sont en moyenne meilleurs sur l'ensemble des catégories même si en moyenne la $F_1$-mesure moyenne est limité à 0.668. Les résultats des extensions du PLS ne sont pas très éloignées de ceux des arbres avec des différences de $F_1$ à moins de 0.1 (si on choisit le bon schéma de représentation).

| **Représentation** | **Algorithme** | $F_1$ | **min** | **Cat. min** | **max** | **Cat. max** | $meilleur(F_1)$ **-** $F_1$ | **max - min** | **rang** |
|---|---|---|---|---|---|---|---|---|---|
| $tf - gss$ | Arbre | 0.668 | 0.5 | *doris* | 0.92 | *dcppc* | 0 | 0.42 | 1 |
| $tf - avg_{global}$ | LogitPLS | 0.648 | 0.518 | *danais* | 0.781 | *dcppc* | 0.02 | 0.263 | 13 |
| $tf - avg_{global}$ | StandardPLS | 0.636 | 0.49 | *danais* | 0.836 | *dcppc* | 0.032 | 0.346 | 24 |
| $tf - \Delta_{DF}$ | GiniPLS | 0.586 | 0.411 | *danais* | 0.837 | *dcppc* | 0.082 | 0.426 | 169 |
| $tf - \Delta_{DF}$ | 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 4.3 – Comparaison des combinaisons représentation+algorithme proposées avec les arbres, fastText et NBSVM pour la détection du sens du résultat.

Les scores $F_1$ moyens des algorithmes NBSVM et fastText n'excédent en général pas 0.5 malgré qu'ils soient spécialement conçus pour les textes. On peut estimer qu'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 toutes les données test avec le label (sens) majoritaire i.e. le rejet, et par conséquent, ils ne détectent quasiment pas d'acceptation de demande. Le cas des catégories *doris* et *dcppc* pour le NBSVM ($F_{1macro} = 0.834$) tend à démontrer la forte sensibilité aux cas négatifs de ces algorithmes puisque même avec presque autant de labels "accepte" que "rejette", la $F_1$-mesure de "rejette" est toujours supérieure à celle de "accepte" (Tableau 4.4).

| Cat. Dmd. | Algo. | Préc. | Préc. équi. | err-0 | err-1 | $F_1(0)$ | $F_1(1)$ | $F_{1macro}$ |
|---|---|---|---|---|---|---|---|---|
| *dcppc* | NBSVM | 0.875 | 0.812 | 0 | 0.375 | 0.916 | 0.752 | **0.834** |
| *danais* | fastText | 0.888 | 0.5 | 0 | 1 | 0.941 | 0 | 0.47 |
| *danais* | NBSVM | 0.888 | 0.5 | 0 | 1 | 0.941 | 0 | 0.47 |
| *concdel* | fastText | 0.775 | 0.5 | 0 | 1 | 0.853 | 0 | 0.437 |
| *concdel* | NBSVM | 0.775 | 0.5 | 0 | 1 | 0.873 | 0 | 0.437 |
| *acpa* | fastText | 0.745 | 0.5 | 0 | 1 | 0.853 | 0 | 0.426 |
| *acpa* | NBSVM | 0.745 | 0.5 | 0 | 1 | 0.853 | 0 | 0.426 |
| *doris* | NBSVM | 0.5 | 0.492 | 0.167 | 0.85 | 0.63 | 0.174 | 0.402 |
| *dcppc* | fastText | 0.667 | 0.5 | 0 | 1 | 0.8 | 0 | 0.4 |
| *styx* | fastText | 0.667 | 0.5 | 0 | 1 | 0.8 | 0 | 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 = "rejette" et 1 == "accepte"
Cat. Dmd. : Catégorie de demandes
Algo. : algorithme
err-0 : taux d'erreur dans la classe "rejette"
err-1 : taux d'erreur dans la classe "accepte"
Préc. : précision globale ($accuracy = \frac{TP}{N}$)
Préc. équi. : $\frac{1}{2}(accuracy(0) + accuracy(1))$

Tableau 4.4 – Évaluation de fastText et NBSVM pour l'identification du sens du résultat par catégorie de demandes.

Les algorithmes PLS dépassent systématiquement les performances ($F_1$-mesure) de fastText et NBSVM de 10 à 20 à points, ce qui tend à démontrer l'efficacité des techniques PLS dans leur rôle de réduction des dimensions. Les algorithmes Gini-PLS ne semblent pas mieux fonctionner que les algorithmes classiques PLS. On peut supposer que la réduction de dimensions se réalise en conservant encore trop de bruit dans les données. Ceci est confirmé par les résultats des arbres qui restent très mitigés pour lesquels la $F_1$-mesure (0,668) dépasse à peine celle du Logit-PLS (0,648). Il semble donc nécessaire de procéder à un zonage dans le document qui permettrait de mieux cerner les informations pertinentes et de ce fait réduire le bruit.

### 4.4.3 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), énoncés aux termes de la catégorie dans les Motifs (contexte). Les combinaisons passages-représentation vectorielle-algorithme sont comparées dans le Tableau 4.5. 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).

| Catégorie | Zone | Représentation | Algorithme | $F_1$ |
|---|---|---|---|---|
| *acpa* | **demande_resultat_a_resultat_context** | $tf - dbidf$ | **Arbre** | **0.846** |
| | litige_motifs_dispositif | $tf - dbidf$ | StandardPLS | 0.697 |
| | litige_motifs_dispositif | $tf - avg_{global}$ | LogitPLS | 0.683 |
| *concdel* | **litige_motifs_dispositif** | $tf - gss$ | **Arbre** | **0.798** |
| | motifs | $tf - idf$ | GiniLogitPLS | 0.703 |
| | context | $logave - dbidf$ | StandardPLS | 0.657 |
| *danais* | **demande_resultat_a_resultat_context** | $avg_{local} - \chi^2$ | **Arbre** | **0.813** |
| | demande_resultat_a_resultat_context | $atf - avg_{global}$ | LogitPLS | 0.721 |
| | demande_resultat_a_resultat_context | $atf - avg_{global}$ | StandardPLS | 0.695 |
| *dcppc* | **demande_resultat_a_resultat_context** | $tf - \chi^2$ | **Arbre** | **0.985** |
| | demande_resultat_a_resultat_context | $tf - \chi^2$ | LogitPLS | 0.94 |
| | litige_motifs_dispositif | $tp - mar$ | StandardPLS | 0.934 |
| *doris* | **litige_motifs_dispositif** | $tp - dsidf$ | **GiniPLS** | **0.806** |
| | litige_motifs_dispositif | $tp - dsidf$ | GiniLogitPLS | 0.806 |
| | litige_motifs_dispositif | $atf - ig$ | StandardPLS | 0.772 |
| *styx* | **motifs** | $tf - dsidf$ | **Arbre** | **1** |
| | demande_resultat_a_resultat_context | $logave - dsidf$ | GiniLogitPLS | 0.917 |
| | litige_motifs_dispositif | $tf - rf$ | GiniPLS | 0.833 |

Tableau 4.5 – Impact de la restriction des documents à certains passages sur l'identification du sens du résultat.

Après réduction de la taille du document, force est de constater que les arbres fournissent d'excellents résultats, suivis de très près par nos algorithmes GiniPLS et LogitGiniPLS. Par exemple, dans la catégorie dcppc (voir Tableau 3.5), les performances des arbres ($F_1$ = 0,985) dépassent légèrement les algorithmes LogitPLS (0,94) et PLS standard (0,934). Dans la catégorie concdel, les performances des arbres ($F_1$ = 0,798) sont encore suivis de près par les algorithmes GiniLogitPLS (0,703) et PLS standard (0657). Le cas le plus intéressant concerne les troubles du voisinage (catégorie doris). Ces décisions comportent bien souvent de multiples informations qu'il est parfois difficile de synthétiser, même pour un humain. L'argumentation exposée dans *doris* peut porter sur de multiples informations

(problèmes de vues, d'ensoleillement, etc.) si bien que les éléments factuels qui conditionnent l'identification du résultat des juges sont parfois complexes. Ces multiples informations sont de plus parfois très singulières, on va par exemple mentionner une seule fois le rôle d'un expert dans une décision, alors que les éléments apportés par ce dernier auront une incidence décisive sur la décision. Ce genre d'information isolée peut être soit sous-représentée soit sur-representée suivant la représentation vectorielle considérée, ce que nous appelons les données aberrantes. Notre algorithme GiniPLS (de même que notre GiniLogitPLS) semble être particulièrement adapté à cette catégorie de demande. Les ($F_1$-mesures trouvées dans cette catégorie s'élèvent à 0,806 (pour GiniPLS et GiniLogitPLS) et à 0,772 pour le StandardPLS alors que les arbres de décisions ne font pas parties des algorithmes pertinents pour cette catégorie de demande (non classés permis les trois meilleurs algorithmes). Ce résultat nous conforte dans l'idée que nos algorithmes GiniPLS peuvent parfois concurrencer les arbres de décisions qui font office de référence dans la littérature. Ce résultat permettrait d'envisager à l'avenir d'inclure nos algorithmes GiniPLS dans les méthodes ensemblistes afin d'élargir le spectre des algorithmes robustes aux valeurs aberrantes et qui jouent en même temps un rôle de compression des données.

## 4.5 Conclusion

L'étude de ce chapitre tente de simplifier l'extraction du sens du résultat rendu par les juges sur une demande de catégorie donnée. Elle a consisté à formuler le problème comme une tâche de classification de documents. On évite ainsi de passer par la détection ad-hoc [10] des passages et données à l'aide de termes-clés qui est un inconvénient de la méthode à règles du chapitre précédent car elle n'est peut-être pas généralisable à tous types de décisions (i.e. il pourrait être nécessaire d'établir de nouvelles listes de mots-clés pour d'autres domaines). Au total dix algorithmes de classification on été expérimentés sur 55 méthodes de représentations vectorielles de textes. Nous avons remarqué que les résultats de classification sont principalement influencés par 3 caractéristiques de nos données. Tout d'abord, le très faible nombre d'exemples d'entraînement défavorise certains algorithmes (sensibilité aux valeurs aberrantes ou *outliers*), comme par exemple fastText qui nécessite plusieurs milliers d'exemples pour mettre à jour le pas du gradient (*learning rate*). Ensuite, le fort dés-

10. i.e. spécialement conçue pour nos données.

équilibre entre les classes ("accepte" vs. "rejette") rend difficile la reconnaissance de la classe minoritaire qui est généralement la classe "accepte". Le fort gap entre les erreurs sur "rejette" et celles sur "accepte", ainsi que les bons résultats obtenus sur *dcppc* en sont la preuve. Enfin, la présence d'autres catégories de demande dans le document dégrade l'efficacité de la classification parce que les algorithmes ne parviennent pas seuls à retrouver les éléments en rapport direct avec la catégorie choisie. Ceci est démontré par l'impact positif de la restriction du contenu à classer à certains passages particuliers, même si la restriction adéquate est fonction de la catégorie.

Au final, les arbres de décision sont adaptés pour la tâche, mais l'usage du Gini-PLS et du Gini-Logit-PLS permet d'obtenir des performances assez proches de celles des arbres. Il serait intéressant de combiner ces variantes de l'analyse PLS, à d'autres comme le Sparse-PLS qui pourrait peut-être aider à résoudre le problème de vecteurs/matrices creuses dont sont victimes les représentations vectorielles de texte. Il existe aussi un grand nombre d'architectures neuronales pour la classification de document et de très grands nombres de métriques de pondération de termes pour la représentation des textes, mais aucune ne semble s'adapter à toutes les catégories. Par conséquent, une étude sur l'usage des représentations par plongement sémantique comme Word2Vec [Mikolov *et al.*, 2013], Sent2Vec [Pagliardini *et al.*, 2018] ou Doc2Vec [Le & Mikolov, 2014] serait intéressante.

# Chapitre 5

# Découverte des circonstances factuelles

***Résumé.*** *Le présent chapitre s'intéresse à découvrir les situations qui permettent généralement de formuler une catégorie donnée de demande. Nous avons montré dans le Chapitre 3 qu'il est facile, par classification, de déterminer si une décision traite d'une catégorie donnée de demande. Ce problème considéré comme résolu, il est donc facile de rassembler les décisions relatives à une catégorie. Nous proposons donc de découvrir les circonstances factuelles en formant des groupes de documents similaires définissants les situations recherchées. L'importance pour le métier est de connaître les différentes situations dans lesquelles les demandes d'une catégorie sont généralement formulée. L'expert peut ainsi distinguer les décisions qui lui sont utiles de celles qui le sont moins suivant que leur circonstances factuelles sont similaires ou pas à celle de son cas d'étude. L'expert du projet a annoté les circonstances factuelles d'une catégorie concernant l'action en responsabilité des avocats. La compréhension des notions de circonstances factuelles et de similarité entre décision peut être subjective d'un juriste à un autre. Par conséquent, nous proposons de déterminer, sur le corpus annoté, la représentation qui correspond le mieux à la compréhension de l'annotateur (validation non supervisée sur le regroupement manuel). Cette représentation est ensuite appliquée aux autres catégories de demandes avec l'hypothèse qu'elle permettra d'identifier des circonstances factuelles de même nature. Par ailleurs, nous proposons une approche d'apprentissage de distance sémantique entre deux documents de la même catégorie à l'aide d'un algorithme de régression. La métrique apprise lors des expérimentations donne de meilleurs résultats avec l'algorithme des K-moyennes en comparaison avec d'autres distances expérimentées.*

## 5.1 Introduction

Les circonstances factuelles définissent les contextes possibles dans lesquels une catégorie de demande peut être formulée. Les analyses descriptives ou prédictives ne prennent sens que lorsqu'elles sont appliquées à un ensemble de décisions aux circonstances similaires. Par exemple, il serait imprudent de considérer toutes les décisions pour analyser les chances d'acceptation d'une demande de dommages et intérêts fondée sur l'« ar-

ticle 700 du code de procédure civile ». Les taux d'acceptation ou de rejet peuvent être différents entre des affaires de licenciement et celles portant sur les troubles anormaux du voisinage, et même plus spécifiquement entre des troubles de voisinage entre particuliers et entreprises. Il est indispensable de travailler uniquement avec des décisions similaires à la situation d'intérêt. L'identification des circonstances factuelles devient donc une étape préalable indispensable à l'analyse du résultat. Malheureusement, les circonstances sont très diverses et quasi infinies pour être identifiées par classification supervisée à l'aide d'annotation manuelle d'exemples comme dans les chapitres précédents. Il est donc plus adéquat d'adopter une approche non-supervisée capable de découvrir les circonstances factuelles à partir d'un corpus de documents d'une même catégorie de demandes. Plus précisément, la méthode doit construire des sous-ensembles de décisions partageant des situations similaires. L'objectif de ce chapitre est d'expérimenter des algorithmes de regroupement (*clustering*) et des métriques de similarité généralement utilisées sur les textes. Ce chapitre propose aussi une méthode d'apprentissage d'une distance qui est basée sur la transformation de documents, et montre qu'une telle métrique permet de bien mesurer la (dis-) similarité sémantique définie par les circonstances factuelles.

## 5.2 Catégorisation non-supervisée de documents

Cette section fait une synthèse bibliographique de différents aspects qui rentrent dans la conception d'un système de regroupement de documents. Elle aborde principalement le choix de l'algorithme, la définition d'une mesure de similarité, la représentation des documents, la détermination du nombre de groupes (appelés *clusters*), et l'évaluation de la catégorisation générée. Le corpus à catégoriser est noté $\mathcal{D}$ et comprend $N$ documents. La catégorisation obtenue est un ensemble de clusters $C = \{C_1, C_2, \cdots, C_K\}$, $K$ étant le nombre de clusters formés.

### 5.2.1 Algorithmes de catégorisation non-supervisé

La catégorisation de documents a pour objectif d'identifier, sans supervision [1], une organisation pertinente (pour le domaine expert) de l'ensemble $\mathcal{D}$ en construisant des groupes représentants des catégories inconnues au départ. Ces groupes, appelés *clusters*, peuvent être disjoints

1. Sans utiliser des exemples annotés.

ou se chevaucher, organisés de manière plate ou hiérarchique suivant les contraintes du domaine expert. L'algorithme à utiliser dépend généralement de la forme qu'on souhaite donner à l'organisation.

#### 5.2.1.1 Partitionnement disjoint

Pour réaliser des partitions distinctes [2] (*hard clustering*), des algorithmes tels que celui des K-moyennes (*K-means*) [Forgey, 1965] et celui des K-medoïdes (*K-medoids*) [Kaufman & Rousseeuw, 1987] sont les plus simples [Balabantaray *et al.*, 2015]. Ces deux algorithmes fonctionnent de manière similaire, et nécessitent que le nombre $K$ de clusters soit prédéfini. Ils commencent par une définition aléatoire de $K$ centres initiaux de clusters (*centroïdes*) et l'affectation des différents documents au cluster dont le centre est le plus proche. S'en suit une boucle dans laquelle le centroïde est recalculé (le point dont la somme des distances aux membres du cluster est minimale) et les documents sont réaffectés chacun au cluster dont le centroïde est le plus proche. L'algorithme s'arrête si aucune amélioration n'est plus observée, ce qui se traduit soit par l'atteinte d'une valeur minimale prédéfinie de l'erreur de catégorisation [3] ou d'une mesure d'évaluation non supervisée (§ 5.2.5.2). La différence entre l'algorithmes des K-moyennes et celui des K-medoïdes tient principalement au fait que les centroïdes du premier ne sont pas nécessairement des points (documents) de l'ensemble d'origine, mais des points moyennes des représentations vectorielles des membres du cluster, contrairement à l'algorithme des K-medoïdes qui ne considère comme centres que des documents de $\mathcal{D}$. Cette différence donne l'avantage au K-medoïdes de ne pas dépendre d'une représentation vectorielle nécessaire au calcul de la moyenne, mais elle a aussi l'inconvénient d'augmenter sa complexité en temps et en espace car il faut calculer et stocker la distance entre toutes les paires de documents. Il existe plusieurs autres algorithmes de partitionnement dont le principe est différent de celui des K-moyennes. Par exemple, l'algorithme DBSCAN (*Density-based spatial clustering of applications with noise*) [Ester *et al.*, 1996] ne prend pas en paramètre le nombre de clusters à construire. Il est défini sur le concept de régions de densité caractérisées par la distance minimale $\epsilon$ autorisée entre deux points d'une même région, et le nombre maximal de points qui doivent être dans le voisinage de rayon $\epsilon$ d'un point pour que ce voisinage soit une région de densité (le point central est appelé "point noyau" (*core point*). Le principe du DBSCAN est de construire les clusters successivement en reliant les régions (voisinages) dont les noyaux sont à

2. Chaque document n'appartient qu'à un seul cluster.
3. Somme des distances au carré entre les points et leur centre respectif.

distance plus ou moins inférieure à $\epsilon$. Les points qui sont seuls dans leur cluster sont qualifiés de points aberrants (*outliers*).

La catégorisation spectrale est une autre méthode efficace de partitionnement qui effectue préalablement une réduction de dimensions à l'aide du spectre [4] de la matrice de similarité $M \in \mathbb{R}^{N \times N}$ [5] des données avant d'appliquer un algorithme traditionnel comme celui des K-moyennes. Les dimensions du nouvel espace sont définies par les vecteurs propres de la matrice Laplacienne $L$ de $M$ [Shi & Malik, 2000; Von Luxburg, 2007] qui peut être normalisée ($L = T^{-1/2}(T - S)T{-}1/2$) ou pas ($L = T - M$), $T$ étant la matrice diagonale déduite de $M$ i.e. $T_{ii} = \sum_j M_{ij}$.

Il est aussi possible d'utiliser les arbres de décision pour améliorer les résultats des K-moyennes. En effet, les forêts aléatoires [Breiman, 2001] permettent d'estimer la similarité entre deux points. Le principe consiste à générer un ensemble de $n$ points synthétiques, et d'entraîner une forêt aléatoire à une classification binaire supervisée avec les points originaux considérés dans la classe des "originaux" et les données synthétiques dans la seconde classe des "synthétiques" [Afanador *et al.*, 2016]. Une forêt aléatoire construit des arbres de décision sur des parties de l'ensemble d'apprentissage, auxquelles on a retiré une ou plusieurs variables prédictives. La similarité entre 2 points est la proportion d'arbres dans lesquels ces points se trouvent dans le même nœud feuille. Cette métrique "apprise" peut-être par la suite utilisée dans un algorithme comme les K-moyennes.

#### 5.2.1.2 Catégorisation avec chevauchements

Les regroupements avec chevauchement sont intéressants parce qu'il est possible qu'une décision traite de plusieurs circonstances factuelles. Lorsque des chevauchements sont observables entre clusters [6], un degré d'appartenance (*membership degree*) d'un document à chaque groupe est estimé par une fonction $u_{ij}, \forall d_i \in \mathcal{D}, \forall j \in [1..K]$ [Baraldi & Blonda, 1999]. Ce degré d'appartenance est employé dans des algorithmes de partitionnement "flou" comme l'algorithme des c-moyennes flou (FCM) [Bezdek *et al.*, 1984; Hathaway *et al.*, 1989], ou le fuzzy c-Medoids (FDMdd) [Krishnapuram *et al.*, 2001], ou la version améliorée IFKM (*improved fuzzy K-medoids*)[Sabzi *et al.*, 2011]. Nefti & Oussalah [2004] proposent par ailleurs les C-moyennes floues probabilistes qui sont une variante du FCM pour laquelle la somme des degrés d'appartenance d'un document aux clusters

4. Le spectre d'une matrice est l'ensemble de ses valeurs propres
5. $M_{ij}$ est la mesure de la similarité entre les élément $d_i$ et $d_j$ de $\mathcal{D}$.
6. Un chevauchement est l'appartenance d'un document à plusieurs groupes.

est de 1.

Ces algorithmes consistent en deux étapes principales [Sabzi *et al.*, 2011] :

1. l'estimation des degrés d'appartenance de chaque instance $d_i \in \mathcal{D}$ à chaque cluster $j \in [1..K]$ de centroïde $z_j$ est réalisée par la minimisation de la fonction objectif $P(\mathcal{D}, Z) = \sum_{i=1}^{N} \sum_{j=1}^{K} \left[u_{ij} r(d_i, z_j)\right]$ [Krishnapuram *et al.*, 2001] améliorée par Sabzi *et al.* [2011] en :

$$P(\mathcal{D}, Z) = \sum_{i=1}^{N} \sum_{j=1}^{K} \left[u_{ij} r(d_i, z_j)\right] + \lambda \sum_{i=1}^{N} \sum_{j=1}^{K} \left[u_{ij} \log_2(u_{ij})\right]$$

$$s.c. \sum_{j=1}^{K} u_{ij} = 1, 0 \leq u_{ij} < 1$$

dont la valeur approximative de la solution est

$$u_{ij} = \frac{\exp\left(\frac{-r(d_i, z_j)}{\lambda}\right)}{\sum_{l=1}^{k} \exp\left(\frac{-r(d_l, z_j)}{\lambda}\right)},$$

$r(d_i, z_j)$ étant la distance entre $d_i$ et $z_j$

2. Le nouveau centre de chaque cluster $C_j$ est redéfini comme étant la moyenne des membres de chaque cluster chez le FCM. Mais pour les K-medoïdes flous, il s'agit du membre $z_j$ dont la somme des distances pondérées[7] aux autres membres est minimale :

$$\forall j \in [1 \,..\, K], z_j = \underset{d_q \in C_j}{\operatorname{argmin}} \sum_{d_i \in C_j} \left[u_{ij} r(d_i, d_q)\right].$$

Ainsi l'objectif de l'entraînement des algorithmes de la catégorisation floue est double : déterminer les degrés optimaux d'appartenance $u_{ij}$ et l'ensemble $Z$ des centroïdes.

#### 5.2.1.3 Catégorisation hiérarchique

La catégorisation hiérarchique consiste à construire une hiérarchie de clusters. Le regroupement hiérarchique ascendant ou regroupement *agglomératif* (*Agglomerative Clustering*) est une technique de catégorisation hiérarchique qui commence par autant de clusters que de documents (chacun des groupes comprenant un document). Ensuite, l'algorithme détecte

7. La distance est pondérée par le degré d'appartenance de l'autre membre.

et fusionne successivement les paires de groupes dont la fusion vérifie un critère (par exemple la paire est celle dont la distance est la plus petite et/ou proche d'une valeur seuil donnée, ou bien la fusion forme un groupe à inertie [8]minimale), jusqu'à ce que tous les documents soient dans un unique groupe (racine). Pour déterminer le partitionnement optimal, le nombre de clusters doit être déterminé par l'une des diverses techniques existantes [Thorndike, 1953; Salvador & Chan, 2004].

### 5.2.2 Métriques de dis-similarité

Les algorithmes de catégorisation dépendent de la distance utilisée qui doit être bien choisie pour que le résultat révèle au mieux la sémantique visée. Une distance $Dis$ est une fonction réelle d'une paire de documents $(d, d')$ qui mesure le degré de différence ou dis-similarité entre $d$ et $d'$ en satisfaisant aux propriétés suivantes $\forall d, d', d'' \in \mathcal{D}$ [Harispe *et al.*, 2015; Wang & Sun, 2015] :

1. $Dis(d, d') \geq 0$ ("non-négativité")
2. $Dis(d, d') = 0 \Leftrightarrow d = d'$ (identité des indiscernables)
3. $Dis(d, d') = Dis(d', d)$ (symétrie)
4. $Dis(d, d'') \leq Dis(d, d') + Dis(d', d'')$ (inégalité triangulaire)

La métrique peut être normalisée ($\forall (d, d') \in \mathcal{D} \times \mathcal{D}; 0 \leq Dis(d, d') \leq 1$). Dans ce cas, la relation entre la similarité $Sim$ et la dis-similarité $Dis$ est définie par $Sim(d, d') = 1 - Dis(d, d')$.

Parmi les nombreuses métriques généralement utilisées sur les textes [Huang, 2008; Vijaymeena & Kavitha, 2016; Afzali & Kumar, 2018], on retrouve par exemple :

- Les distances de Minkowski $Dis(d, d') = \left\| \vec{d} - \vec{d'} \right\|_{Lp} = \sqrt[p]{\sum_{i=1}^{m} |\vec{d}[i] - \vec{d}[i]|^p}$, dont font partie la distance euclidienne $Dis_{euclidienne}$ ($p = 2$) et la distance de Manhattan $Dis_{manhattan}$ ($p = 1$).
- La distance de Bray & Curtis [1957] : $Dis_{braycurtis}(d, d') = \frac{\sum_{i=1}^{m} |\vec{d}[i] - \vec{d'}[i]|}{\sum_{i=1}^{m} |\vec{d}[i] + \vec{d'}[i]|}$ [Huang, 2008].
- La similarité cosinus basée sur le cosinus de l'angle entre $\vec{d}$ et $\vec{d'}$ par la formule : $Sim_{cos}(d, d') = \frac{\vec{d}^t \vec{d'}}{\|\vec{d}\| \|\vec{d'}\|}$. Pour un modèle vectoriel du

8. L'inertie d'un cluster est la variance de ses points c'est-à-dire la somme des erreurs (distance d'un membre au centre) au carré.

type TF-IDF, cette formulation considère que tous les termes du vocabulaire $T$ sont différents et ne partagent aucune relation. Sidorov *et al.* [2014] la corrigent en proposant la fonction *soft-cosine* utilisant la matrice de similarité entre termes $S = \{s_{ij}\}_{1 \leq i,j \leq m}$ :

$$Sim_{soft-cos}(d, d') = \frac{\vec{d}^T \cdot S \cdot \vec{d'}}{\sqrt{\vec{d}^T \cdot S \cdot \vec{d}} \cdot \sqrt{\vec{d'}^T \cdot S \cdot \vec{d'}}} = \frac{\sum\limits_{1 \leq i,j \leq m} s_{ij} \vec{d}[i] \vec{d'}[j]}{\sqrt{\sum\limits_{1 \leq i,j \leq m} s_{ij} \vec{d}[i] \vec{d}[j]} \sqrt{\sum\limits_{1 \leq i,j \leq m} s_{ij} \vec{d'}[i] \vec{d'}[j]}}.$$

$S$ peut être calculée à partir de n'importe quelle métrique comme la distance d'édition de Levenshtein [Sidorov *et al.*, 2014], la similarité cosinus entre plongements lexicaux [Charlet & Damnati, 2017, 2018], ou la similarité WordNet. La fonction cosinus étant comprise entre -1 et +1, la distance déduite $Dis_{cos}(d, d') = 1 - Sim_{cos}(d, d')$ est comprise entre 0 et 2.

- Le coefficient similarité de Jaccard [1901] : $Sim_{Jaccard}(d, d') = \frac{\vec{d}^T \vec{d'}}{\|\vec{d}\|^2 + \|\vec{d'}\|^2 - \vec{d}^T \vec{d'}}$ [Huang, 2008] qui donne la distance $Dis_{jaccard}(d, d') = 1 - Sim(d, d')$.
- La similarité basée sur le coefficient de corrélation de Pearson est calculée comme suit [Huang, 2008] :

  $Sim_{pearson}(d, d') = \frac{\sum\limits_{i=1}^{m} \vec{d}[i] \cdot \vec{d'}[i] - TF_d \cdot TF_{d'}}{\sqrt{[m \sum\limits_{i=1}^{m} \vec{d}[i]^2 - TF_d^2][m \sum\limits_{i=1}^{m} \vec{d'}[i]^2 - TF_{d'}^2]}}$, avec $TF_d = \sum\limits_{i=1}^{m} \vec{d}[i]$. Sa distance est déduite par la formule :

  $$Dis_{pearson}(d, d') = \begin{cases} 1 - Sim_{pearson}(d, d') & \text{si } Sim_{pearson}(d, d') \geq 0 \\ |Sim_{pearson}(d, d')| & \text{si } Sim_{pearson}(d, d') < 0. \end{cases}$$

- « La distance du déménageur de mot » (*word mover's distance - WMD*) [Kusner *et al.*, 2015] tiens compte de la dis-similarité sémantique entre les mots dans l'estimation de la distance entre documents. En effet, elle est la solution optimale du problème de transport suivant [9] :

$$\begin{aligned} Dis_{wmd}(d, d') = \min_{T>0} \quad & \sum_{i,j=1}^{m} T_{ij} c(i, j) \\ \text{s.c.} \quad & \sum_{j=1}^{m} T_{ij} = \vec{d}[i], \forall i \in [1 \,..\, m]; \sum_{i=1}^{m} T_{ij} = \vec{d'}[j], \forall j \in [1 \,..\, m] \end{aligned}$$

$m$ est le nombre de mots considérés ; $T$ est une matrice dont $T_{ij}$ est interprété comme étant la quantité du mot $i$ de $d$ qui va au mot $j$ dans $d'$ ("voyage"). $c(i, j)$ est la distance euclidienne entre les vecteurs des

9. Valeur minimale du coût cumulatif pondéré nécessaire pour déplacer tous les mots de $d$ à $d'$ i.e. transformer $d$ en $d'$.

mots $i$ et $j$ ; $\vec{d}[i] = \frac{compte(i,d)}{\sum_{k=1}^{m} compte(k,d)}$, $compte(i,d)$ étant le nombre d'occurrences du mot $i$ dans $d$.

### 5.2.3 Représentation des textes

#### 5.2.3.1 Modèle vectoriel

La formulation des distances (cf. § 5.2.2) s'applique généralement à une représentation vectorielle des textes. Le chapitre 3 décrit différentes métriques de pondération des termes qui permettent de définir des variantes du modèle TF-IDF. Cependant, il s'agit de modèles basés uniquement sur le lexique des textes, et par conséquent, les distances appliquées sur ces représentations correspondent à une similarité plus lexicale que sémantique. Il existe quelques techniques permettant de définir des dimensions par des thématiques qui transparaissent dans le corpus, et apportant par conséquent plus de sémantique à la représentation vectorielle.

#### 5.2.3.2 Réduction de dimension

Les méthodes expérimentées ici sont non supervisées [10] et génèrent de nouvelles dimensions [11].

**L'analyse en composantes principales (ACP)** [Burrows, 1992] est une technique de transformation de l'espace originel en un espace de dimension réduite préservant au mieux l'inertie du nuage originel [12]. L'intérêt de l'ACP se traduit par une meilleur interprétation des données dans le nouvel espace. Son principe consiste à construire successivement les composantes principales qui sont des combinaisons linéaires des observations ; chaque composante étant orthogonale à la suivante. Plus précisément, les colonnes de la matrice document-terme $A$ de dimensions $N \times m$ sont préalablement transformée en des variables centrées réduites [13] résultant en une matrice $\hat{A}$. Ensuite, l'ACP décompose $\hat{A}$ en un produit de trois matrices : $U$ la matrice $N \times N$ des vecteurs propres de $\hat{A}\hat{A}^T$, $S$ la matrice

10. Elles ne prennent en compte aucune classification prédéfinie des documents.

11. Par opposition aux algorithmes de sélection de caractéristiques comme le BDS et le SFFS expérimentées au Chapitre 2.

12. Distance entre les individus pris 2 à 2, ou dispersion autour du barycentre.

13. Centrer-réduire une variable $X$ d'espérance $\mu$ et d'écart-type $\sigma$ revient à obtenir une variable $\hat{X}$ d'espérance nulle et variance à 1 en calculant pour chaque valeur $X_i$ de $X$, $\hat{X}_i = \frac{X_i - \mu}{\sigma}$.

diagonale $m \times m$ des valeurs propres de la plus grande à la plus petite, et $V$ la matrice $m \times N$ des vecteurs propres de $\hat{A}^T\hat{A}$. Un nombre $q$ de composantes (vecteurs propres $\hat{U}_q = U[1, N; 1, q]$) peut être sélectionné pour la réduction de dimension suivant la variance totale qu'on souhaite conserver. Ainsi, tout vecteur $\vec{d}$ est réduit à $q$ dimensions en le multipliant par $\hat{U}_q : \hat{\vec{d}}_q = \hat{U}_q\vec{d}$. La méthode la plus simple et populaire pour déterminer $q$ est la règle dite de "Kaiser" ou de "Kaiser-Guttman" [Guttman, 1954; Kaiser, 1960] même si elle tend à extraire beaucoup plus de composantes que nécessaire [Bandalos & Boehm-Kaufman, 2010]. Le critère de Kaiser juge que seules les valeurs propres supérieures ou égales à la moyenne des valeurs propres sont plus informatives que les variables initiales.

**L'allocation latente de Dirichlet** (ALD) [Blei *et al.*, 2003] est une technique qui extrait un nombre $q$ de thématiques à partir du corpus $\mathcal{D}$. Elle introduit les "variables latentes" comme pont pour modéliser les relations entre les textes et les termes. Chaque document est caractérisé comme une distribution de Dirichlet sur les variables latentes (thématiques), et chaque thématique est caractérisée par une autre distribution de Dirichlet sur tous les termes. La réduction de dimension est déduite de la première distribution en obtenant pour chaque document, un vecteur de taille $q$ représentant la distribution de probabilité des thématiques dans le document. Un des défis de la modélisation thématique est la détermination d'une valeur optimale du nombre de thèmes $q$. Parmi plusieurs valeurs candidates, celle qui maximise la cohérence du modèle peut être choisie. La cohérence du modèle est la moyenne des cohérences des thèmes. La cohérence de chaque thème peut être considérée comme étant la moyenne de la similarité entre les paires de termes de la description du thème [14]. Fang *et al.* [2016] démontrent que la cohérence est bien estimée avec la similarité cosinus des plongements lexicaux des termes.

**L'analyse latente sémantique** (ALS) [Dumais *et al.*, 1988; Deerwester *et al.*, 1990] réalise une décomposition en valeurs singulières (SVD) de la matrice $A$ des scores de co-occurrence document-terme. Plus précisément, l'ALS applique la décomposition SVD directement sur $A$, contrairement à l'ACP qui centre-réduit cette dernière au préalable. L'ALS permet ainsi de réduire la grande dimension lexicale des documents à un espace de thématiques de dimensions définies par le nombre de valeurs propres choisies. Comme pour l'ALD, le nombre $q$ de thématiques peut être sélectionné

14. Les $n$ premiers termes les plus fréquents du thème.

comme étant celui qui maximise la cohérence.

**La factorisation de matrice non-négative** (FMN) [Paatero & Tapper, 1994] factorise la matrice $A$ des scores non-négatifs de co-occurrence document-terme, en deux matrices $W$ et $H$ sans élément négatif, dont le produit est une approximation de $A$. Il s'agit en effet d'une méthode de modélisation de thématiques dans laquelle $W$ est décrite comme la matrice $N \times q$ de relation entre les termes et les thèmes découverts dans les documents, et $H$ est la matrice $q \times m$ des poids d'appartenance des documents aux thèmes. Le calcul de $W$ et $H$ consiste à minimiser la fonction objectif : $\frac{1}{2}\|A - WH\|_F^2 = \sum_{i=1}^{N}\sum_{i=1}^{m}(A_{ij} - (WH)_{ij})^2$, où $\|\cdot\|_F$ désigne la norme matricielle de Frobenius [15], et $A_{ij} = w(t_j, d_i)$, $d_i \in \mathcal{D}$. Comme pour l'ALD, le nombre $q$ de composantes (ou thématiques) peut être sélectionné comme étant celui qui maximise la cohérence.

**Barycentre des termes** A partir de vecteurs de mots appris à l'aide de méthodes comme Word2Vec [Mikolov *et al.*, 2013] ou GloVe [Pennington *et al.*, 2014], il est souvent proposé de considérer le document comme le barycentre des mots qu'il contient, et de le représenter comme la moyenne des vecteurs de termes pondérés par leur poids dans le document (par exemple TF-IDF) [Le & Mikolov, 2014; Charlet & Damnati, 2018; Arora *et al.*, 2017]. Tout vecteur $\vec{d}$ est réduit en un vecteur $\hat{\vec{d}}_q$ de $q$ dimensions correspondant à la taille des vecteurs de termes :

$$\hat{\vec{d}}_q[k] = \frac{1}{\sum_{i=1}^{m}\vec{d}[i]}\sum_{i=1}^{m}\vec{d}[i]\cdot\vec{t}_i[k]$$

$\vec{t}_i[k]$ étant le plongement sémantique du mot $t_i$.

### 5.2.4 Sélection du nombre optimal de groupes

Au delà de l'algorithme à utiliser, le nombre $K$ approprié de clusters ne doit pas être prédéfini mais déterminé automatiquement, puisqu'il est difficile de savoir à l'avance le nombre de groupes. Une méthode très connue est celle du « coude » (ou « genou ») [Halkidi *et al.*, 2001], qui est basée sur

15. $\|X\|_F = \sqrt{\sum_{i=1}^{N}\sum_{i=1}^{m} X_{ij}^2}$.

le principe de base des algorithmes de partitionnement (e.g. K-moyennes) i.e. minimiser le critère d'inertie : $J(K) = \sum_{j=1}^{K} \sum_{d_i \in C_j} \left\| \vec{d_i} - \overrightarrow{\overline{d_j}} \right\|^2$, $C_j$ étant ensemble des objets du cluster $j$, et $\overrightarrow{\overline{d_j}}$ le centre du cluster $j$. La méthode du coude consiste à essayer différentes valeurs consécutives de $K$, puis à choisir celle qui correspond au coude de la courbe du critère d'inertie $J(K)$ c'est-à-dire le point à partir duquel la décroissance de la courbe commence à être très faible. Le choix de ce coude est visuel et peut être ambigu (plusieurs valeurs de $K$ sur le coude par exemple).

La méthode de la silhouette moyenne [Rousseeuw, 1987] est une alternative moins ambigüe qui consiste à choisir comme valeur optimale de $K$, celle qui maximise le critère de la largeur moyenne de la silhouette : $\overline{s_K}(C) = \frac{1}{K} \sum_{i=1}^{N} s(d)$. La largeur $s(d)$ de la silhouette est un indice qui compare la ressemblance d'un document $d$ aux autres membres de son cluster $C_t$ par rapport à sa ressemblance aux autres clusters $C_l, l \neq t : s(d) = \frac{b(d)-a(d)}{\max\{a(d),b(d)\}}$ où $a(d) = \frac{1}{|C_t|} \sum_{d' \in C_t} Dis(d,d')$, et $b(d) = \min_{l \neq t} \frac{1}{|C_l|} \sum_{d' \in C_t} Dis(d,d')$, pour $d \in C_l$. $K$ est optimal lorsque $\overline{s_K}(C)$ est maximale. Les valeurs de ce dernier varient entre -1 (pire valeur) et +1 (meilleure valeur). Les valeurs proches de zéro indiquent que les clusters se chevauchent en $x$, et il est difficile de savoir à quel cluster $x$ doit être affecté. Une valeur négative indique que $x$ a été affecté à cluster inapproprié.

## 5.2.5 Validation de la catégorisation

La validation peut être supervisée ou non suivant l'emploie ou pas des affectations manuelles de documents à des groupes attendus.

### 5.2.5.1 Métriques supervisées ou indices externes

Les métriques couramment utilisées mesurent la ressemblance entre deux catégorisations $X = \{X_1, X_2, ..., X_r\}$ et $Y = \{Y_1, Y_2, ..., Y_s\}$ :

- l'indice ajusté par chance de Rand (*ajusted Rand index* - ARI) [Hubert & Arabie, 1985] corrigent l'indice de Rand (RI) [Rand, 1971] pour obtenir une valeur très proche de 0 pour les catégorisations aléatoires et exactement 1 lorsque les clusters sont identiques aux classes attendues. En effet, l'indice de Rand prend ses valeurs en pratique dans l'intervalle $[0.5; 1]$, et par conséquent, considère une valeur de base très élevée. ARI est calculé à l'aide du tableau de contingence résumant les chevauchements que partagent $X$ et $Y$ (Tableau 5.1) par la

formule :

$$ARI(X,Y) = \frac{\sum_{i,j}\binom{n_{ij}}{2} - \frac{\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}}{\binom{N}{2}}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \frac{\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}}{\binom{N}{2}}}$$

avec $\binom{n}{2} = \frac{n(n-1)}{2}$. $n_{i,j} = |X_i \cap Y_j|$, $a_i$ et $b_j$ proviennent du tableau de contingence (Tableau 5.1).

| | $Y_1$ | $Y_2$ | $\cdots$ | $Y_s$ | $\sum$ |
|---|---|---|---|---|---|
| $X_1$ | $n_{11}$ | $n_{12}$ | $\cdots$ | $n_{1s}$ | $a_1$ |
| $X_2$ | $n_{21}$ | $n_{22}$ | $\cdots$ | $n_{2s}$ | $a_2$ |
| $\cdots$ | $\cdots$ | $\cdots$ | $\ddots$ | $\cdots$ | $\cdots$ |
| $X_r$ | $n_{r1}$ | $n_{r2}$ | $\cdots$ | $n_{rs}$ | $a_r$ |
| $\sum$ | $b_1$ | $b_2$ | $\cdots$ | $b_s$ | |

Tableau 5.1 – Tableau de contingence des chevauchement entre les catégorisations $X = \{X_1, X_2, ..., X_r\}$ et $Y = \{Y_1, Y_2, ..., Y_s\}$

ARI a des valeurs dans $[-1; 1]$. Une valeur négative indique que la catégorisation obtenue s'accorde moins bien avec l'attendu qu'une catégorisation aléatoire.

- L'information mutuelle normalisée (NMI) [Kvalseth, 1987; Strehl *et al.*, 2000; Vinh *et al.*, 2010] normalise l'information mutuelle entre $X$ et $Y$ par une agrégation de leur entropie respective. Par exemple, l'incertitude symétrique [Kvalseth, 1987] est une variante qui utilise la moyenne comme fonction d'agrégation : $NMI(X,Y) = \frac{2 \cdot I(X,Y)}{H(Y)+H(Y)}$, avec $I(X,Y) = H(X) - H(X|Y) = \sum_{i=1}^{r}\sum_{j=1}^{s} \frac{n_{ij}}{N} \log_2 \frac{n_{ij}/N}{a_i b_j/N}$ et $H(X) = \sum_{X_i \in X} (-p(X_i)\log_2 p(X_i))$, $p(X_i) = \frac{a_i}{N}$ et $p(Y_j) = \frac{b_j}{N}$. La meilleure catégorisation est celle qui a la plus grande valeur.
- Au regard de leur formulation, les métriques ARI et NMI sont plus focalisées sur la différence de volume entre les clusters de deux catégorisations. D'autres méthodes appelées « mesures de comptage de paires » (*pair counting measures*) mesurent la capacité du modèle à mettre deux documents similaires (de labels identiques dans les données annotées) dans le même groupe, et des documents dis-similaires (de labels différents dans les données annotées) dans des clusters différents. Parmi ces mesures, on retrouve par exemple, la précision, le rappel, et la $F_1$-mesure qui sont définies par les métriques de base suivantes [Manning *et al.*, 2009a] :

- un vrai positif (TP) survient si le modèle place deux documents similaires dans le même cluster (groupe généré par le modèle) ;
- un faux négatif (FN) survient si deux documents similaires sont dans des clusters différents ;
- un vrai négatif (TN) survient si deux documents dissemblables se retrouve dans deux clusters différents ;
- un faux positif (FP) survient si deux documents dissemblables sont dans le même cluster.

#### 5.2.5.2 Métriques non-supervisées ou indices internes

La cohésion et la séparation des clusters sont les principaux indices internes. La cohésion mesure le degré de proximité entre objets d'un cluster à partir du carré de la somme des erreurs (distance entre un point et le centre du cluster dont il est membre) dans les clusters. La formule est $WCSS(C) = \sum_{j=1}^{K} \sum_{d \in C_j} (Dis(d, z_j))^2$, où $C = \{C_1, C_1, \cdots, C_K\}$ est l'ensemble des clusters de la catégorisation, $z_j$ le centre de $C_j$, et $Dis(d, z_j)$ la distance (généralement euclidienne) entre un point $d$ et $z_j$. La séparation quant à elle mesure l'éloignement de chaque cluster des autres à partir du carré de la somme des distances entre clusters : $BCSS(C) = \sum_{j=1}^{K} |C_j|(\overline{z} - z_j)$, $\overline{z}$ étant le centre de tous les documents. Le coefficient de silhouette de Rousseeuw [1987] (cf. § 5.2.4) combine les idées de cohésion et séparation mais individuellement pour chaque document.

## 5.3 Apprentissage d'une distance basée sur la transformation de document

Nous définissons une métrique $Dis_{\mathcal{M}}$ qui est fonction des transformations permettant de passer d'un document $d$ à un autre $d'$ :

$$\begin{array}{rccl} Dis_{\mathcal{M}} & : & \mathcal{D} \times \mathcal{D} & \rightarrow & \mathbb{R} \\ & & d, d' & \mapsto & Dis_{\mathcal{M}}(d, d') = f(\mathcal{M}_{d,d'}). \end{array} \tag{5.1}$$

$\mathcal{D}$ est le corpus. $\mathcal{M}_{d,d'}$ est l'ensemble des modifications de $d$ permettant d'obtenir $d'$ i.e. les paires de mots différents $(d[k], d'[k])$ telles que le mot $d[k]$ a été remplacé par $d'[k]$. $f$ est une fonction qui croît avec le nombre de modifications. Après une légère modification, le sens d'un texte reste

assez similaire à celui de l'original. Tandis qu'après un grand nombre de modifications, le sens du texte est très différent de l'original.

Pour des documents de même taille, cette distance peut, par exemple, se formuler comme étant la proportion de mots modifiés :

$$Dis_{\mathcal{M}}(d, d') = f(\mathcal{M}_{(d,d')}) = \frac{|\mathcal{M}_{(d,d')}|}{|d|} \tag{5.2}$$

Par contre, pour des textes de tailles différentes, il est impossible de savoir les positions où des mots ont été supprimés ou ajoutés, et par conséquent, il devient impossible de calculer leur distance. La distance étant une valeur continue, en entraînant un modèle de régression sur un ensemble de paires de documents pour lesquelles la distance est connue, il est possible de la prédire pour des paires de documents de taille quelconque. Nous proposons de générer une base synthétique de paires de documents dont l'un est un document du corpus original mais l'autre est le résultat de substitutions et suppressions de mots du premier. En contrôlant ces modifications, il est facile de calculer une valeur de $Dis_{\mathcal{M}}$ pour chaque paire générée de documents, même s'ils sont de tailles différentes (en considérant la suppression d'un mot comme son remplacement par le « mot vide »).

### 5.3.1 Génération d'une base d'apprentissage

La génération de la base synthétique nécessite de définir une formulation de la fonction $f(\mathcal{M}_{d,d'})$ pour les documents de taille égale, comme par exemple celle de l'Equation 5.2. Cette formulation considère que toutes les modifications ont la même importance c'est-à-dire qu'une substitution de mots contraires est équivalente à une substitution de mots similaires. Pour corriger cette limite, chaque modification peut être pondérée par la distance entre les mots substitués (le vecteur du « mot vide » étant nul) :

$$Dis_{\mathcal{M}}(d, d') = f(\mathcal{M}_{(d,d')}) = \frac{\sum_{(d[k],d'[k])\in\mathcal{M}_{(d,d')}} Dis_{cos}(\overrightarrow{d[k]}, \overrightarrow{d'[k]})}{|d|} \tag{5.3}$$

$d$ est un document du corpus original $\mathcal{D}$, et $d'$ est le résultat d'une transformation contrôlée de $d$. $\overrightarrow{d[k]}$ désigne le plongement lexical du mot $d[k]$. Pour garantir la symétrie et la réflexivité de la métrique, nous imposons respectivement $Dis_{\mathcal{M}}(d, d') = Dis_{\mathcal{M}}(d', d)$ et $Dis_{\mathcal{M}}(d, d) = Dis_{\mathcal{M}}(d', d') = 0, \forall d \in \mathcal{D}$ sur le jeu d'entraînement généré. L'algorithme 6 de génération de documents synthétiques contrôle le taux de modifications à effectuer

sur le document original grâce à un seuil donnée $0 \leq p \leq 1$. En variant $p$, plusieurs documents $d'$ sont générés pour chaque $d \in \mathcal{D}$ pour former un jeu d'apprentissage $B_{\mathcal{M}} = \{((d_1, d_2), Dis(d_1, d_2))_i\}_{1 \leq i \leq |B_{\mathcal{M}}|}$.

**Algorithme 6 :** Transformation de document

**Données :** document $d \in \mathcal{D}$, valeur seuil $p$, ensemble $W$ des mots
**Résultat :** $d'$, $\mathcal{M}_{(d,d')}$

$d' = []$;
$\mathcal{M}_{(d,d')} = \emptyset$;
**pour** $k \in [1 \,..\, |d|]$ **faire**
  $v = valeur\_elle\_alatoire\_entre(0, 1)$;
  **si** $v < p$ **alors**
    $d'[k] = modifie\_mot(d[k], W)$ ; // mot aléatoire de $W$ différent de $d[k]$;
    $\mathcal{M}_{d[k],d'[k]} = \mathcal{M}_{(d,d')} \cup \{(d[k], d'[k])\}$;
  **sinon**
    $d'[k] = d[k]$;
**retourner** $d', \mathcal{M}_{(d,d')}$;

### 5.3.2 Entraînement de la métrique

Sur $B_{\mathcal{M}}$, un modèle de régression peut être entrainé pour prédire la distance entre deux documents quelconques $d_i$ et $d_j$ en fonction de leur représentation vectorielle. Ce modèle de régression $Reg_{\mathcal{M}}$ peut être utilisé comme distance dans un algorithme de catégorisation comme celui des K-moyennes. Cependant, les modèles de régression ne supportent généralement qu'un seul vecteur en entrée, et pas deux comme en dispose la base $B_{\mathcal{M}}$. Les vecteurs $\vec{d_i}$ et $\vec{d_j}$ doivent donc être agrégés en un seul. Pour des vecteurs de documents à composantes non-négatives, une bonne agrégation est la soustraction car la soustraction des vecteurs de documents similaires résulte en un vecteur proche du nul. La fonction d'estimation automatique de la distance entre $x$ et $y$ s'écrit : $Dis_{\mathcal{M}}(d_i, d_j) = Reg_{\mathcal{M}}(\vec{d_i} - \vec{d_j})$.

### 5.3.3 Utilisation pour le regroupement des documents

Nous proposons dans un premier temps de sélectionner la représentation vectorielle pour laquelle la distance sépare au mieux les groupes manuellement annotées et rapproche au mieux les éléments à l'intérieur de

ces groupes. La représentation [16] optimale maximise la largeur moyenne de la silhouette de la catégorisation manuelle. L'idée étant d'avoir une représentation qui optimise à la fois les métriques supervisées et non supervisées de validation. La représentation ainsi déterminée est utilisée ensuite pour la catégorisation sur les corpus non annotés.

## 5.4 Expérimentations et résultats

Cette section discute de la validité, l'adéquation, et l'efficacité de la métrique apprise en comparaison avec d'autres distances. La validité de la métrique est établie si cette dernière respecte les propriétés d'une distance. L'adéquation de la métrique avec le problème à résoudre mesure la capacité de la métrique à estimer une distance très faible entre documents de mêmes circonstances factuelles, et une similarité très faible entre documents de circonstances différentes, indépendamment de tout algorithme de catégorisation. Enfin, l'efficacité de la métrique est liée à la qualité de la catégorisation résultant de l'application d'un algorithme de regroupement utilisant cette distance.

### 5.4.1 Données

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" (*arcpa*) qui concerne les contentieux impliquant des avocats. L'annotateur (juriste) a organisé 81 documents [17] en 4 circonstances factuelles, avec 6 documents appartenant simultanément chacun à 2 groupes (chevauchements) :

- cas *a* (46 documents) : il s'agit d'un avocat qui est négligent et envoie son assignation de manière tardive ;
- cas *b* (20 documents) : il s'agit d'un avocat qui n'a pas donné un conseil opportun, qui n'a pas soulevé le bon argument ;
- cas *c* (18 documents) : un avocat qui n'a pas rédigé un acte valide ou réussi à obtenir un avantage fiscal ;
- cas *d* (3 documents) : il s'agit d'un avocat attaqué par son adversaire et non par son propre client.

16. Combinaison d'un modèle vectoriel et d'une méthode de réduction.
17. Sur 85 documents disponibles de catégorie *arcpa*.

Nous avons expérimenté le regroupement uniquement sur les 74 documents n'appartenant qu'à un seul des cas $L = \{a, b, c\}$ (les chevauchements ne sont pas traités).

Les terminologies des cas se distinguent bien (Tableau 5.2). Par conséquent, une représentation des documents qui mettrait en évidence de tels termes permettrait de retrouver les circonstances factuelles.

| Corpus | Terminologie |
|---|---|
| *arcpa* | chance, perte chance, avocat, perte, diligence, chance obtenir, perdre, client, devoir conseil, manquement |
| *cas a* | chance, perte chance, chance succès, perte, client, préjudice indemnisable, article code commerce, indemnisable, condamnation emporter, emporter nécessairement rejet |
| *cas b* | défense intérêt, intérêt client, avocat, contractuel égard, responsabilité contractuel droit, responsabilité professionnel avocat, contractuel droit commun, assurer défense intérêt, civil avocat, grief articuler |
| *cas c* | rédacteur acte, rédacteur, avocat rédacteur acte, avocat rédacteur, qualité rédacteur acte, rédaction acte, qualité rédacteur, projet acte, prendre initiative conseiller, initiative conseiller |
| *cas d* | revêtir aucun, revêtir aucun caractère, article code, article code procédure, faire référence aucun, fautif madame, civil profit autre, civil depuis, mention expresse, moyen dont |

*10 premiers termes lémmatisés de 1 à 3 mots sélectionnés à l'aide du coefficient de corrélation ngl (cf. § 3.2.3.2)*

Tableau 5.2 – Terminologies de la catégorie *arcpa* et de ses circonstances factuelles manuellement annotées.

Pour l'évaluation non supervisée, les corpus des chapitres 3 et 4 sont aussi employés. Ils sont notés $\mathcal{D}_{acpa}$, $\mathcal{D}_{concdel}$, $\mathcal{D}_{danais}$,$\mathcal{D}_{dcppc}$, $\mathcal{D}_{doris}$, $\mathcal{D}_{styx}$.

### 5.4.2 Protocole et outils logiciels

La métrique apprise est entraînée sur une base générée, puis nous l'évaluons sur le corpus annoté $\mathcal{D}_{arcpa}$ restreint aux 74 documents. Les documents sont pré-traités avant leur représentation vectorielle. Ce pré-traitement consiste à les sectionner (chapitre 2), à les restreindre à la section Motifs, à mettre en minuscule et lemmatiser leur contenu, puis à y éliminer la ponctuation et des mots inutiles (*stop words*) car ils sont généralement indépendants de toute catégorie. Après pré-traitement, les documents ont une taille allant de 208 à 3812 mots dont une moyenne de 1381 mots. Pour générer les données d'entrainement de $Dis_M$, le vocabulaire $W$ utilisé est restreint aux mots du corpus original $D$ sur lequel il faut appliquer les catégorisations. La représentation vectorielle emploie des modèles de type

TF-IDF (poids local × poids global × facteur de normalisation, cf. § 3.3.1) avec des n-grammes de 1 à 3 mots. Les poids globaux sont appris sur la discrimination entre deux corpus $\mathcal{D}_c$ et $\mathcal{D}_{\bar{c}}$ ($|\mathcal{D}_{\overline{arcpa}}| = 427$ documents) c'est-à-dire la terminologie d'une catégorie $c$ de demandes.

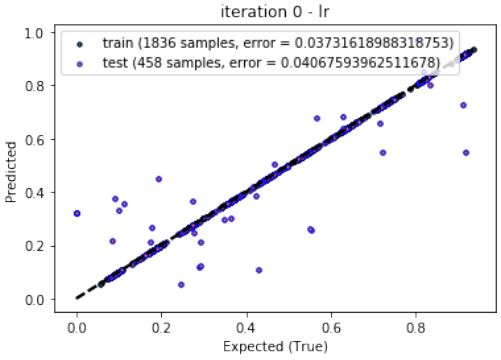
La librairie Python Scikit-Learn [Pedregosa *et al.*, 2011] a été utilisée pour les implémentations des algorithmes de réduction de dimension, et ceux du DBSCAN, de la catégorisation spectrale (*Spectral Clustering*), et du regroupement hiérarchique (*Agglomerative Clustering*). Les implémentations des C-moyennes floues probabilistes (*Probabilistic FCM*) et des arbres aléatoires (*Random Forest*) proviennent respectivement des librairies *scikit-cmeans 0.1* [18] et RandomForestClustering [19]. La distance WMD est celle implémentée dans la librairie Gensim de Řehůřek & Sojka [2010]. Les expérimentations utilisent un modèle de plongement lexical de type GLoVE [Pennington *et al.*, 2014] entraîné sur un corpus de 800k décisions (de Légifrance) lémmatisées avec des fenêtres de contexte de 15 mots. Ce modèle dispose de 32445 mots représentés par des vecteurs de 300 dimensions. Le vecteur $\vec{t}$ d'un terme $t$ de $n$ mots $(w_1, w_2, \cdots, w_n)$ est obtenu en concaténant leur vecteur respectif de la droite vers la gauche : $\vec{t} = [\vec{w}_1 \vec{w}_2 \cdots \vec{w}_n]$.

### 5.4.3 Validité de la distance apprise

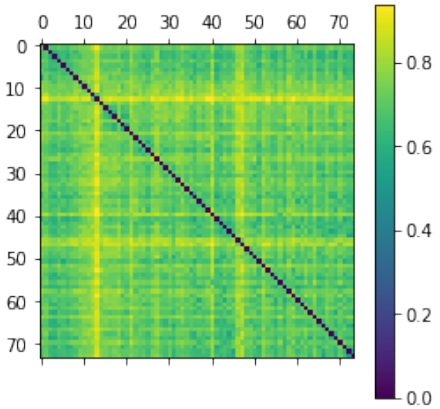
Pour rappel, l'objectif ici est de vérifier que la métrique $Dis_{\mathcal{M}}$ conserve les propriétés de distance après à la suite de son entraînement sur le jeu de données synthétique. Pour la catégorie *arcpa*, la base d'entraînement $B_{\mathcal{M}}$ comprend 935 documents dont 10 documents synthétiques générés pour chacun des 85 documents. Sur un modèle TF-IDF, la régression linéaire approxime bien la distance proposée $Dis_{\mathcal{M}}$ car elle a un faible taux d'erreur (Figure 5.1a).

D'après la matrice des distances entre les 74 documents de $\mathcal{D}_{arcpa}$ (Figure 5.1b), les propriétés de "non-négativité", d'identité des indiscernable, et de symétrie sont respectée car toutes les valeurs sont non-négatives, seule la diagonale est nulle, et la matrice est symétrique. De plus, toutes les distances sont comprises entre 0 et 1, et par conséquent la métrique est normalisée. Nous vérifions pareillement la propriété d'inégalité triangulaire en vérifiant que $Dis_{\mathcal{M}}(d, d'') - (Dis_{\mathcal{M}}(d, d') + Dis_{\mathcal{M}}(d', d'')) \leq 0, \forall (d, d', d'') \in \mathcal{D}_{arcpa} \times \mathcal{D}_{arcpa} \times \mathcal{D}_{arcpa}$.

18. `https://bm424.github.io/scikit-cmeans/index.html`
19. `https://github.com/joshloyal/RandomForestClustering/`




(a) Différence entre valeurs prédites et valeurs attendues



(b) Matrice des distances entre documents

Figure 5.1 – Validité de la distance apprise.

### 5.4.4 Sélection de la représentation optimale des textes

En considérant la catégorisation manuelle de $\mathcal{D}_{arcpa}$, différentes représentation vectorielles peuvent être comparées, à l'aide de la silhouette, sur leur habilité à séparer les clusters manuels de documents dans l'espace. Nous comparons ici les combinaisons de différents poids locaux (Tableau 4.1 Page 84), poids globaux (cf. §3.2.3) et méthodes de réduction de dimensions (cf. §5.2.3.2). Le Tableau 5.3 présente la meilleure représentation de largeur moyenne de silhouette sur les données annotées pour chaque distance. Pour les réductions par ALD, ALS, et FNM, le nombre de thèmes est déterminé entre 4 et 10.

| **Distance** | **Base**[a] | **Silhouette optimale (pondération, réduction, dim.)** |
|---|---|---|
| $Dis_{jaccard}$ | 0.001 | 0.212 (TP-NGL, FNM, 4) |
| $Dis_{cos}$ | 0.002 | 0.202 (TP-NGL, FNM, 4) |
| $Dis_{M}$ | -0.049 | 0.195 (TP-NGL, FNM, 4) |
| $Dis_{braycurtis}$ | 0.002 | 0.182 (TP-NGL, FNM, 4) |
| $Dis_{euclidienne}$ | 0.001 | 0.168 (TP-NGL, FNM, 4) |
| $Dis_{manhattan}$ | -0.019 | 0.17 (TP-NGL, FNM, 4) |
| $Dis_{pearson}$ | 0.014 | 0.057 (TP-CHI2, aucune, 19763) |
| $Dis_{wmd}$ | -0.096 | - |

[a] occurrence de mots pour $Dis_{wmd}$, et TF-IDF pour les autres distances.

Tableau 5.3 – Meilleures représentations sur la catégorisation manuelle.

Le modèle vectoriel TP-NGL réduit à 4 dimensions par la factorisation de matrice non négative (FNM) est préféré par la majorité des dis-

tances. Nous remarquons que la pondération globale supervisée (NGL et CHI2) met en évidence non seulement la terminologie de la catégorie de demande mais aussi celle des circonstances factuelles associées. La FNM marche mieux en moyenne pour toutes les classes avec des silhouettes maximales comprises entre 0.052 et 0.212, suivie de l'ALS (0.048 – 0.119) et de l'ACP (0.029 – 0.079). Les scores maximaux de silhouette observés par l'ALD (-0.032 – -0.001) sont moins bons que la représentation sans réduction (0.001 – 0.008). La réduction par la méthode du barycentre des termes quant à elle donne un score de silhouette compris entre -0.048 et 0.013 au maximum sur l'ensemble des distances, avec une moyenne de 0.002. Les représentations sélectionnées sont utilisées dans la suite.

Le Tableau 5.3 classe aussi les distances en fonction de leur adaptabilité à la tâche suivant les scores de silhouette obtenus sur un regroupement considéré comme parfait (car manuel). La $Dis_{\mathcal{M}}$ se replace bien grâce à la représentation optimale. $Dis_{wmd}$ n'a pas été adaptée avec un modèle vectoriel plus adéquat que le sac-de-mots. Par conséquent, elle présente un score de -0.096 légèrement moins bon que celui d'un regroupement aléatoire (score nul).

### 5.4.5 Catégorisation dans le corpus annoté manuellement

#### 5.4.5.1 Nombre prédéfini de *clusters*

Le Tableau 5.4 présente les mesures ARI, NMI et $F_1$-mesure de la catégorisation par K-moyennes et K-medoïdes sur $\mathcal{D}_{arcpa}$, avec $K = 3$.

Les valeurs de silhouette se reflètent plus sur les indices ARI et NMI, mais moins sur la $F_1$-mesure. Suivant le score $F_1$, $Dis_M$, semble le mieux adaptée pour les K-moyennes, et $Dis_{cos}$ pour les K-medoïdes. Mais les distances, $Dis_{jaccard}$, $Dis_{cos}$, $Dis_{euclidienne}$ semblent proposer un meilleur compromis entre les différents critères de validation.

#### 5.4.5.2 Nombre de *clusters* déterminé automatiquement

Nous analysons ici la détermination du nombre de clusters par la méthode de la silhouette pour chaque distance (Tableau 5.5 Page 130). La sélection de $K$ est effectué pour les valeurs entre 2 et 30. $Dis_{\mathcal{M}}$ retrouve le nombre attendu avec les K-moyennes. 4 est la plus choisie par les distances, et 4 donne une $F_1$-mesure au maximum à 0.551 et un ARI de 0.398 avec $Dis_{cos}$. Les valeurs déterminées (entre 2 et 6) sont très proches de la valeur attendue 3 pour toutes les distances. Notons aussi que l'efficacité de $Dis_{\mathcal{M}}$ et $Dis_{cos}$ reste presque aussi élevée qu'avec un $K$ prédéfini

| Distance | Algorithme | Silhouette | ARI | NMI | R | P | $\mathbf{F_1}$ |
|---|---|---|---|---|---|---|---|
| $Dis_{\mathcal{M}}$ | K-moyennes | 0.403 | 0.411 | 0.427 | 0.574 | 0.648 | **0.607** |
| $Dis_{\mathcal{M}}$ | K-medoïdes | 0.398 | 0.321 | 0.340 | 0.483 | 0.591 | 0.532 |
| $Dis_{braycurtis}$ | K-moyennes | 0.370 | 0.364 | 0.382 | 0.545 | 0.603 | 0.570 |
| $Dis_{braycurtis}$ | K-medoïdes | 0.358 | 0.272 | 0.292 | 0.444 | 0.540 | 0.487 |
| $Dis_{cosine}$ | K-moyennes | 0.422 | 0.389 | 0.406 | 0.556 | 0.616 | 0.583 |
| $Dis_{cosine}$ | K-medoïdes | **0.448** | 0.437 | **0.455** | **0.656** | 0.598 | **0.626** |
| $Dis_{euclidean}$ | K-moyennes | 0.372 | **0.417** | **0.434** | 0.591 | 0.603 | 0.592 |
| $Dis_{euclidean}$ | K-medoïdes | 0.369 | 0.392 | 0.409 | 0.566 | **0.672** | 0.615 |
| $Dis_{jaccard}$ | K-moyennes | **0.442** | 0.371 | 0.389 | 0.554 | 0.600 | 0.574 |
| $Dis_{jaccard}$ | K-medoïdes | 0.431 | **0.440** | **0.455** | 0.529 | 0.645 | 0.581 |
| $Dis_{manhattan}$ | K-moyennes | 0.390 | 0.376 | 0.394 | 0.567 | 0.582 | 0.571 |
| $Dis_{manhattan}$ | K-medoïdes | -0.059 | 0.097 | 0.127 | 0.479 | 0.422 | 0.448 |
| $Dis_{pearson}$ | K-moyennes | 0.434 | 0.088 | 0.117 | 0.585 | 0.487 | 0.530 |
| $Dis_{pearson}$ | K-medoïdes | -0.019 | 0.111 | 0.136 | 0.421 | 0.476 | 0.447 |
| $Dis_{wmd}$ | K-medoïdes | 0.105 | -0.004 | 0.024 | 0.333 | 0.401 | 0.364 |

Tableau 5.4 – Evaluation de la catégorisation par K-moyennes et K-medoïdes sur $\mathcal{D}_{arcpa}$ avec le nombre de groupes prédéfini à $K = 3$.

(Tableau 5.4 Page 129). $Dis_{\mathcal{M}}$ et $Dis_{cos}$ semblent ainsi former de bonnes combinaison avec respectivement les K-moyennes et les K-medoïdes. Par ailleurs, seules $Dis_{manhattan}$ et $Dis_{wmd}$ obtiennent de très faibles valeurs pour les indices ARI et NMI. Même si ces valeurs sont inférieures pour les autres distances, ces dernières, en particulier $Dis_{\mathcal{M}}$, $Dis_{jaccard}$ et $Dis_{cos}$, parviennent quand même à un bon compromis entre les 3 critères ARI, NMI et $F_1$. $Dis_{jaccard}$ et $Dis_{cos}$ sont efficaces à la fois avec les K-moyennes et les K-medoïdes. Elles parviennent à associer une bonne validation non-supervisée (silhouette) à une bonne validation non-supervisée.

La Figure 5.2 Page 130 montre l'évolution de la valeur de la silhouette en fonction du nombre de clusters pour $Dis_{\mathcal{M}}$ utilisé dans les K-moyennes. Malgré les dents de scie très prononcées, le score de la silhouette atteint son pic le plus élevé (maximum global) au niveau de la valeur optimale de $K$.

#### 5.4.5.3 Autres algorithmes de catégorisation

Avec la représentation sélectionnée TP-NGL, nous appliquons d'autres algorithmes de catégorisation sur $\mathcal{D}_{arcpa}$ (Tableau 5.6 Page 131). L'algorithme de regroupement à chevauchement des C-moyennes floues (*Probabilistic FCM*) est utilisé pour partitionner le corpus en choisissant un seul cluster pour chaque document (celui pour qui le degré d'appartenance est maximal). Seul le regroupement hiérarchique et les C-moyennes floues

| Distance | Algorithme | K | Silhouette | ARI | NMI | R | P | $F_1$ |
|---|---|---|---|---|---|---|---|---|
| $Dis_{\mathcal{M}}$ | K-moyennes | **3** | 0.438 | **0.407** | **0.423** | 0.552 | 0.654 | **0.599** |
| $Dis_{\mathcal{M}}$ | K-medoïdes | 6 | 0.453 | 0.359 | 0.395 | 0.298 | 0.669 | 0.413 |
| $Dis_{braycurtis}$ | K-moyennes | 4 | 0.473 | 0.383 | 0.407 | 0.446 | 0.658 | 0.532 |
| $Dis_{braycurtis}$ | K-medoïdes | 5 | 0.448 | 0.344 | 0.375 | 0.331 | 0.645 | 0.437 |
| $Dis_{cosine}$ | K-moyennes | 4 | 0.528 | 0.383 | 0.407 | 0.446 | 0.658 | 0.532 |
| $Dis_{cosine}$ | K-medoïdes | 4 | 0.526 | **0.398** | **0.421** | 0.464 | 0.680 | **0.551** |
| $Dis_{euclidean}$ | K-moyennes | 5 | 0.478 | 0.365 | 0.395 | 0.341 | 0.670 | 0.452 |
| $Dis_{euclidean}$ | K-medoïdes | 5 | 0.456 | 0.313 | 0.346 | 0.335 | 0.619 | 0.434 |
| $Dis_{jaccard}$ | K-moyennes | 4 | 0.570 | 0.367 | 0.391 | 0.439 | 0.643 | 0.522 |
| $Dis_{jaccard}$ | K-medoïdes | 4 | **0.560** | 0.389 | 0.412 | 0.451 | 0.666 | 0.538 |
| $Dis_{manhattan}$ | K-moyennes | 4 | 0.482 | 0.376 | 0.400 | 0.452 | 0.657 | 0.535 |
| $Dis_{manhattan}$ | K-medoïdes | 5 | 0.452 | 0.368 | 0.397 | 0.345 | 0.675 | 0.456 |
| $Dis_{pearson}$ | K-moyennes | 2 | **0.611** | 0.054 | 0.072 | 0.746 | 0.453 | 0.564 |
| $Dis_{pearson}$ | K-medoïdes | 2 | 0.171 | 0.152 | 0.166 | 0.598 | 0.482 | 0.534 |
| $Dis_{wmd}$ | K-medoïdes | 2 | 0.332 | -0.016 | 0.002 | 0.545 | 0.397 | 0.459 |

Tableau 5.5 – Evaluation de la catégorisation par K-moyennes et K-medoïdes sur $\mathcal{D}_{arcpa}$ avec détermination du nombre de clusters basée sur la silhouette.

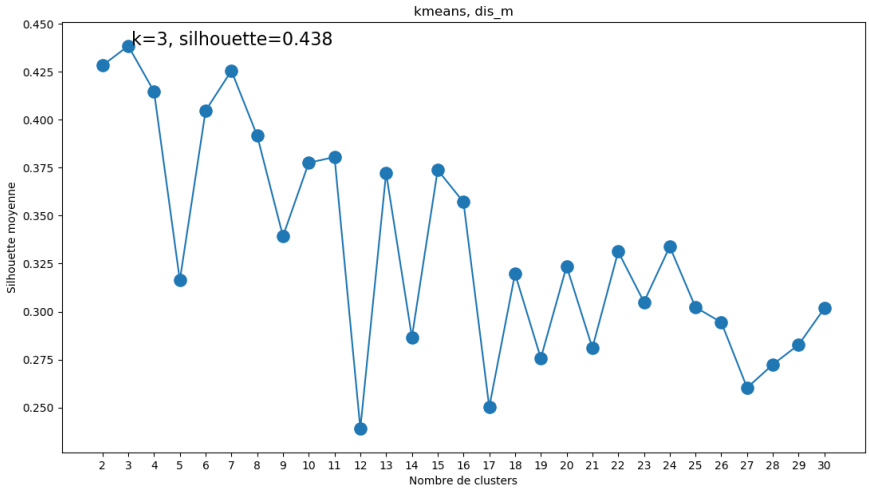



Figure 5.2 – Evolution de la silhouette pour les K-moyennes et la distance apprise.

parviennent à trouver un nombre de clusters (4) assez proche de celui attendu (3). Les forêts aléatoires ont une très basse $F_1$-mesure du fait du grand nombre de clusters déterminé. Par ailleurs, la tâche ne semble pas correspondre ni à la catégorisation par densité qui tend toujours à mettre

tous les documents dans un même cluster, ni à la catégorisation spectrale qui sélectionne un très grand nombre de clusters.

| Algorithme | K | Silhouette | ARI | NMI | R | P | $F_1$ |
|---|---|---|---|---|---|---|---|
| Spectral Clustering | 19 | 0.352 | 0.193 | 0.317 | 0.069 | 0.632 | 0.124 |
| DBSCAN | 2 | -1.000 | 0.000 | 0.000 | 1.000 | 0.398 | 0.570 |
| Agglomerative Clustering | 4 | 0.475 | 0.355 | 0.381 | 0.428 | 0.567 | 0.487 |
| Probabilistic FCM | 4 | 0.521 | 0.394 | 0.417 | 0.444 | 0.657 | 0.530 |
| Random Forest | 11 | 0.272 | 0.228 | 0.303 | 0.127 | 0.598 | 0.210 |

Tableau 5.6 – Evaluation de la catégorisation proposée par plusieurs algorithmes sur $\mathcal{D}_{arcpa}$ avec détermination du nombre de clusters basée sur la silhouette.

Le score de silhouette des C-moyennes floues probabilistes (Figure 5.3 Page 131) présente moins de dents de scie que les K-moyennes (Figure 5.2 Page 130) en fonction du nombre de clusters. Même si le $K = 4$ du pic le plus élevé n'est pas la valeur optimale attendue 3, le score de silhouette décroit au-delà du pic plus rapidement avec les C-moyennes floues qu'avec les K-moyennes.

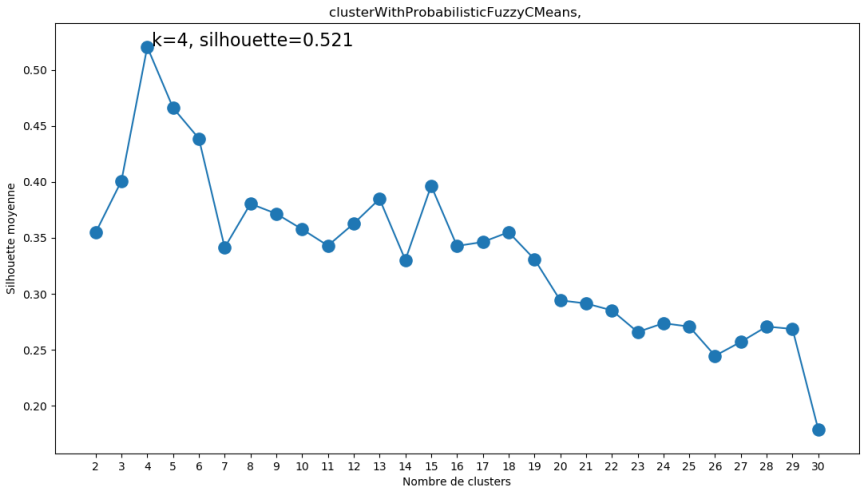



Figure 5.3 – Évolution de la silhouette pour les C-moyennes floues probabilistes

### 5.4.6 Catégorisation des corpus non annotés manuellement

Les matrices documents-termes des autres catégories sont construites à base de la représentation sélectionnée sur la catégorisation manuelle

(TP-NGL FMN, 4 dimensions). Le Tableau 5.7 présente les valeurs de silhouette ($\bar{s}_K$) obtenues par les K-moyennes et le K-medoïdes sur ces matrices. Les nombres déterminés de clusters restent bas (entre 2 et 6) bien qu'ils soient sélectionnés entre 2 et 30.

| $\mathcal{D}$ | Distance | Algorithme | K | $\bar{s}_K$ |
|---|---|---|---|---|
| $\mathcal{D}_{acpa}$ (21*) | $Dis_{\mathcal{M}}$ | K-medoïdes | 2 | 0.769 |
| | $Dis_{\mathcal{M}}$ | K-moyennes | 2 | 0.769 |
| | $Dis_{cosine}$ | K-medoïdes | 5 | 0.694 |
| | $Dis_{cosine}$ | K-moyennes | 3 | 0.762 |
| | $Dis_{jaccard}$ | K-medoïdes | 2 | 0.499 |
| | $Dis_{jaccard}$ | K-moyennes | 3 | 0.769 |
| $\mathcal{D}_{concdel}$ (30) | $Dis_{\mathcal{M}}$ | K-medoïdes | 4 | 0.639 |
| | $Dis_{\mathcal{M}}$ | K-moyennes | 4 | 0.624 |
| | $Dis_{cosine}$ | K-medoïdes | 4 | 0.675 |
| | $Dis_{cosine}$ | K-moyennes | 4 | 0.632 |
| | $Dis_{jaccard}$ | K-medoïdes | 5 | 0.719 |
| | $Dis_{jaccard}$ | K-moyennes | 5 | 0.702 |
| $\mathcal{D}_{danais}$ (198) | $Dis_{\mathcal{M}}$ | K-medoïdes | 4 | 0.403 |
| | $Dis_{\mathcal{M}}$ | K-moyennes | 2 | 0.442 |
| | $Dis_{cosine}$ | K-medoïdes | 4 | 0.475 |
| | $Dis_{cosine}$ | K-moyennes | 4 | 0.471 |
| | $Dis_{jaccard}$ | K-medoïdes | 3 | 0.483 |
| | $Dis_{jaccard}$ | K-moyennes | 3 | 0.482 |

| $\mathcal{D}$ | Distance | Algorithme | K | $\bar{s}_K$ |
|---|---|---|---|---|
| $\mathcal{D}_{dcppc}$ (91) | $Dis_{\mathcal{M}}$ | K-medoïdes | 2 | 0.451 |
| | $Dis_{\mathcal{M}}$ | K-moyennes | 2 | 0.781 |
| | $Dis_{cosine}$ | K-medoïdes | 2 | 0.549 |
| | $Dis_{cosine}$ | K-moyennes | 2 | 0.925 |
| | $Dis_{jaccard}$ | K-medoïdes | 2 | -0.016 |
| | $Dis_{jaccard}$ | K-moyennes | 3 | 0.820 |
| $\mathcal{D}_{doris}$ (59) | $Dis_{\mathcal{M}}$ | K-medoïdes | 2 | 0.509 |
| | $Dis_{\mathcal{M}}$ | K-moyennes | 3 | 0.527 |
| | $Dis_{cosine}$ | K-medoïdes | 5 | 0.549 |
| | $Dis_{cosine}$ | K-moyennes | 4 | 0.586 |
| | $Dis_{jaccard}$ | K-medoïdes | 3 | 0.600 |
| | $Dis_{jaccard}$ | K-moyennes | 4 | 0.645 |
| $\mathcal{D}_{styx}$ (50) | $Dis_{\mathcal{M}}$ | K-medoïdes | 2 | 0.669 |
| | $Dis_{\mathcal{M}}$ | K-moyennes | 2 | 0.669 |
| | $Dis_{cosine}$ | K-medoïdes | 5 | 0.695 |
| | $Dis_{cosine}$ | K-moyennes | 4 | 0.705 |
| | $Dis_{jaccard}$ | K-medoïdes | 6 | 0.635 |
| | $Dis_{jaccard}$ | K-moyennes | 4 | 0.690 |

* Nombre de documents
$\bar{s}_K$ : largeur moyenne de la silhouette

Tableau 5.7 – Evaluation non-supervisée des K-moyennes et K-medoïdes sur $\mathcal{D}_{acpa}, \mathcal{D}_{concdel}, \mathcal{D}_{danais}, \mathcal{D}_{dcppc}, \mathcal{D}_{doris}, \mathcal{D}_{styx}$.

La silhouette étant bonne en général, on s'attend à ce que les groupes soient en majorité formés effectivement de décisions partageant les mêmes circonstances factuelles. Il est possible de se faire une idée des circonstances factuelles découvertes en observant leur terminologie qu'on peut extraire à l'aide d'une pondération globale supervisée comme le coefficient *ngl*. Considérons par exemple le regroupement avec les K-medoïdes combinés à la distance cosinus pour laquelle des résultats intéressants ont été obtenus sur $\mathcal{D}_{arcpa}$. Pour la catégorie *concdel* [20], on obtient 4 circonstances factuelles dont les champs lexicaux (Tableau 5.8 Page 133) semblent définir les cas de contrefaçon pour le cluster 0, de publicité déloyale pour le cluster 1, de recrutement d'un salarié par un concurrent pour le cluster 2, et de contrats avec des partenaires pour le dernier cluster.

Pour la catégorie *doris* [21], on obtient 5 circonstances factuelles dont les champs lexicaux (Tableau 5.9 Page 134) semblent définir les cas d'excès de trouble pour le cluster 0, de différents entre copropriétaires d'un im-

20. *concdel* : dommages-intérêts pour concurrence déloyale.
21. *doris* : dommages-intérêts pour trouble anormal de voisinage

| Cluster | Terminologie |
|---|---|
| *0* | distinctif, reproduire, code propriété intellectuel, contrefaçon, attaque, dépôt marque, circonstance intervenir, intervenir jugement, intervenir jugement déférer, caractère distinctif |
| *1* | publicitaire, action concurrence, défaut qualité agir, action concurrence déloyal, qualité agir, fichier client, force chose juger, fonder demande titre, date transfert, celui -ci fonder |
| *2* | acte concurrence, acte concurrence déloyal, clause non concurrence, non concurrence, clause non, entreprise concurrent, démarchage, démarcher, salarié, massif |
| *3* | non concurrencer, clause non concurrencer, tout droit, résilier contrat, marcher, préjudice invoquer, détournement, compte entre partie, contrat @card@ juillet, compte entre |

*10 premiers termes de 1 à 3 mots sélectionnés à l'aide du coefficient de corrélation ngl (cf. § 3.2.3.2)*

Tableau 5.8 – Terminologies des circonstances factuelles découvertes en combinant les K-medoïdes et la distance cosinus sur $\mathcal{D}_{concdel}$.

meuble pour le cluster 1, de toit dépassant la limite entre deux habitations pour le cluster 2, et d'ouvrage dépassant la hauteur limite autorisée pour le dernier cluster. Ces analyses sont partiellement sujettes à interprétation.

## 5.5 Conclusion

Les circonstances factuelles organisent les décisions d'une même catégorie de demande mais sont illimitées car elles correspondent aux faits courants de la vie. Leur découverte est indispensable afin de rapprocher les litiges non décidés des cas similaires de la jurisprudence. Ce chapitre aborde ce problème comme une tâche de catégorisation non supervisée de documents. La proposition faite ici est double : (i) l'apprentissage d'une métrique de dis-similarité en considérant qu'un document est obtenu par transformation de tout autre document, (ii) l'exploitation de la faible quantité de catégorisations manuelles pour sélectionner la représentation de texte qui correspond au mieux à la sémantique des circonstances factuelles. Le schéma sélectionné permet de transformer de nouveaux corpus non annotés afin d'y découvrir les circonstances factuelles par catégorisation non supervisée. Les expérimentations montrent une amélioration considérable par rapport au modèle de base TF-IDF. La silhouette reste néanmoins faible, ce qui signifie que la réduction de dimension par FNM est efficace mais il faudrait la combiner avec de meilleurs modèles vectoriels ou mieux l'intégrer au processus de catégorisation. Une approche sem-

| Cluster | Terminologie |
|---|---|
| *0* | excéder inconvénient, inconvénient normal, excéder inconvénient normal, normal voisinage, inconvénient normal voisinage, inconvénient, trouble excéder inconvénient, trouble excéder, excéder, normal |
| *1* | copropriétaire, syndicat copropriétaire, syndicat, condamner in, anormal voisinage, trouble anormal voisinage, in, trouble anormal, syndic, jouissance subir |
| *2* | deux fond \| fonds, séparatif deux fond \| fonds, limite séparatif deux, ordonner démolition, séparatif deux, implanter, condamner démolir, devoir établir toit, devoir établir, toit manière |
| *3* | manière plus, chose manière plus, chose manière, usage prohiber loi, prohiber loi règlement, prohiber loi, absolu, usage prohiber, manière plus absolu, plus absolu |
| *4* | situer zone, hauteur @card@ mètre, hauteur dépasser, appel contester, vitrer, dont hauteur dépasser, urbaniser, recevabilité <unknown> appel, cahier charge lotissement, charge lotissement |

*10 premiers termes de 1 à 3 mots sélectionnés à l'aide du coefficient de corrélation ngl (cf. § 3.2.3.2)*

Tableau 5.9 – Terminologies des circonstances factuelles découvertes en combinant les K-medoïdes et la distance cosinus sur $\mathcal{D}_{doris}$.

blable à celle proposée par Xie & Xing [2013], basée sur l'allocation latente de Dirichlet, mériterait d'être étudiée. Néanmoins, cette sélection de représentation permet d'obtenir une assez bonne efficacité de catégorisation sur le corpus annoté. En effet, nous obtenons, avec respectivement les K-moyennes et les K-medoïdes, 0.599 et 0.551 de $F_1$-mesure, correspondant à 0.407 et 0.398 de ARI, et 0.423 et 0.421 de NMI pour un nombre de clusters déterminé automatiquement. Ces résultats se traduisent aussi sur la largeur moyenne de la silhouette autant pour le corpus annoté (0.438 et 0.526 respectivement) que pour les six corpus non annotés utilisés (entre 0.403 et 0.770 au maximum pour toutes les catégories). La métrique apprise s'accorde mieux avec les K-moyennes que les autres distances selon différents indices de validation, et même pour la détermination du nombre de clusters. Par ailleurs, la représentation vectorielle n'est sélectionnée que sur un seul corpus dans ce chapitre. Il serait intéressant à l'avenir d'annoter plusieurs corpus pour sélectionner la meilleure représentation en moyenne sur l'ensemble de ces catégorisations manuelles. De plus, les expérimentations doivent être étendues au regroupement à chevauchement pour les affaires concernant plus d'une circonstance factuelle.

# Chapitre 6

# Application à l'analyse descriptive d'un grand corpus de décisions jurisprudentielles

Ce chapitre décrit des résultats statistiques observés sur un corpus de décisions d'appel formé de la base CAPP de la DILA [2019] (65k décisions en XML) et 10k décisions de cour d'appel de formats divers collectés à partir d'autres sources. La base CAPP fournit un ensemble de méta-données de référence pour chaque décision notamment la juridiction, la date, la ville. De nouvelle données y sont régulièrement ajoutées dans le dépôt en ligne [1]. Il est donc facile d'observer la répartition des décisions entre les villes (Figure 6.1 Page 135) et entre les années (Figure 6.2 Page 136).

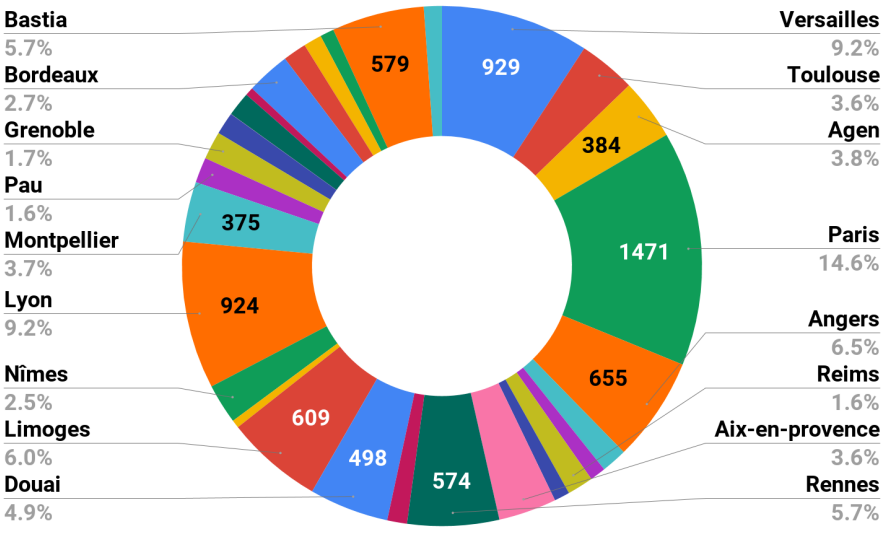



Figure 6.1 – Répartition des décisions de la base CAPP entre villes.

1. Le dépot de CAPP est accessible à partir de `https://www.data.gouv.fr/fr/datasets/capp/`

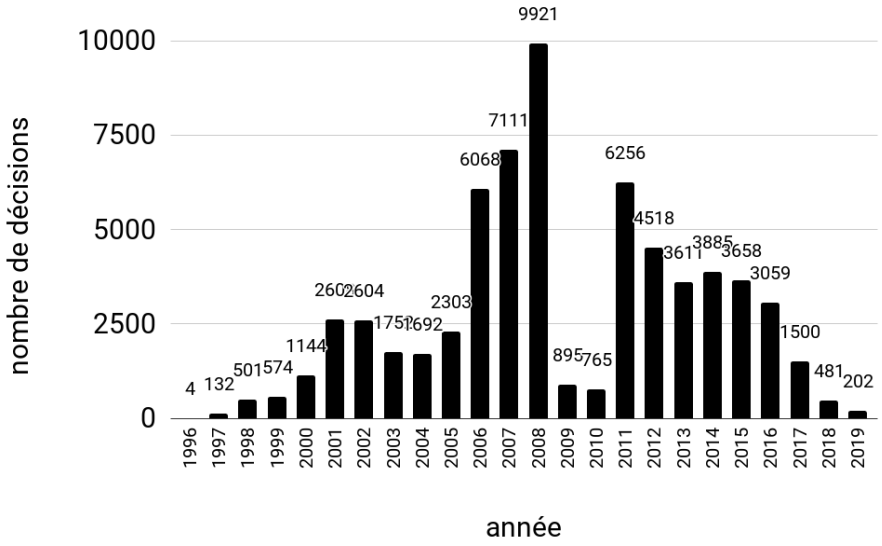



Figure 6.2 – Nombre de décisions de la base CAPP par an.

30 villes sont couvertes pour des décisions qui s'étalent entre 1996 et 2019. La répartition n'est pas égales entre les villes. 6 villes ont moins de 100 décisions : Carcassonne (1), Chambéry (50), Nancy (52), Besançon (87), Amiens (97), et Bourges (99). Les cinq (5) villes qui fournissent le plus de décisions fournissent chacune plus de 600 décisions : Paris (14.6%), Versailles (9.2%), Lyon (9.2%), Angers (6.5%), et Limoges (6.0%).

La base CAPP couvre par ailleurs des juridictions de nature autre que les cours d'appels qui représentent néanmoins plus de 97% de CAPP. On y retrouve par exemple des décisions du conseil de prud'hommes, du tribunal de grande instance, du tribunal d'instance, de juridiction de proximité, du Tribunal Supérieur d'Appel, du tribunal de commerce, du tribunal de première instance etc.

Les connaissances jurisprudentielles ont été extraites à partir de ce corpus non structuré à l'aide des approches dans cette thèse. Après cette extraction, les décisions de la base de données sont réparties entre les villes identifiées automatiquement comme sur la Figure 6.1 Page 135 et dans le temps comme sur la Figure 6.2 Page 136. Les demandes extraites se répartissent comme suit : 476 *acpa*, 409 *concdel*, 160 *danais*, 0 *dcppc*, 34 *doris*, et 45928 *styx*.

La structuration des données dans la base de données permet de mieux comprendre la jurisprudence à l'aide de graphiques appropriés. Une application de visualisation dédiée a notamment été développée par PRYSIAZHNIUK [2017]. Les analyses des sections suivantes sont restreintes

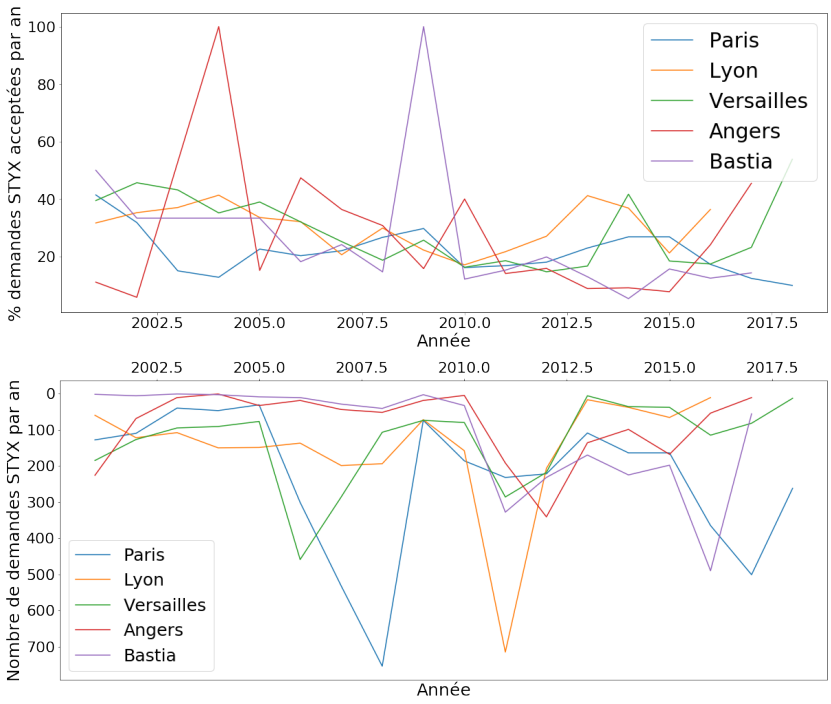



Figure 6.3 – Evolution du sens du résultat des demandes *styx* dans le temps (années) à Paris, Lyon, Versailles, Angers, Bastia.

aux 5 villes ayant les plus grands nombres de décisions : Paris, Lyon, Versailles, Angers, Bastia ; sur la période 2000-2019.

## 6.1 Analyse du sens du résultat

A partir des données extraites, l'évolution du pourcentage de demandes acceptées peut être observée sur une courbe. En traçant une telle courbe pour chaque ville, il est possible de comparer les villes. Par exemple, pour les dommages intérêts sur l'article 700 du Code de Procédure Civile (*styx*), la Figure 6.3 Page 137 compare l'évolution du sens du résultat entre les villes citées précédemment. On remarque que les demandes sont beaucoup plus rejetées qu'acceptées. Pour chaque année, le nombre total de demandes doit être associée pour savoir si le pourcentage de succès est réellement interprétable et comparable à celui des autres années.

La visualisation par l'application de PRYSIAZHNIUK [2017] permet de comparer les villes en observant sur un arbre l'épaisseur des branches associées aux catégories de demande (Figure Figure 6.4 Page 138). On peut ainsi facilement observer quelles villes acceptent les demandes d'une cer-

taine catégorie plus que d'autres par exemple.

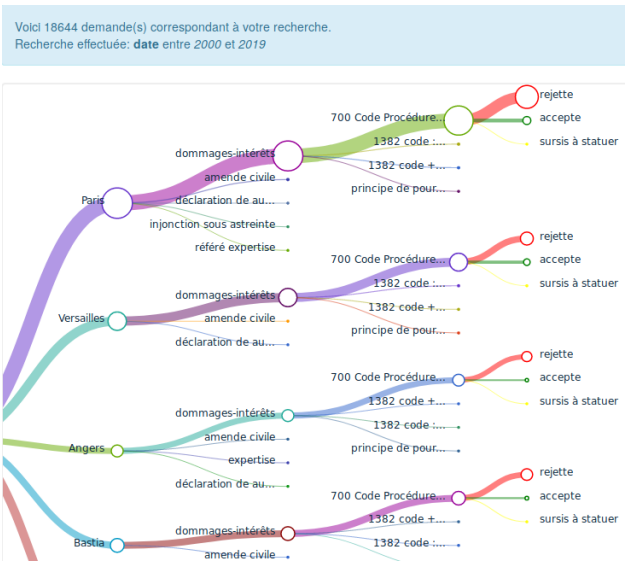



Figure 6.4 – Comparaison des Paris, Lyon, Versailles, Angers, Bastia sur l'acceptation des demandes *styx* à partir d'une visualisation arborée.

## 6.2 Analyse des quanta

### 6.2.1 Evolution dans le temps

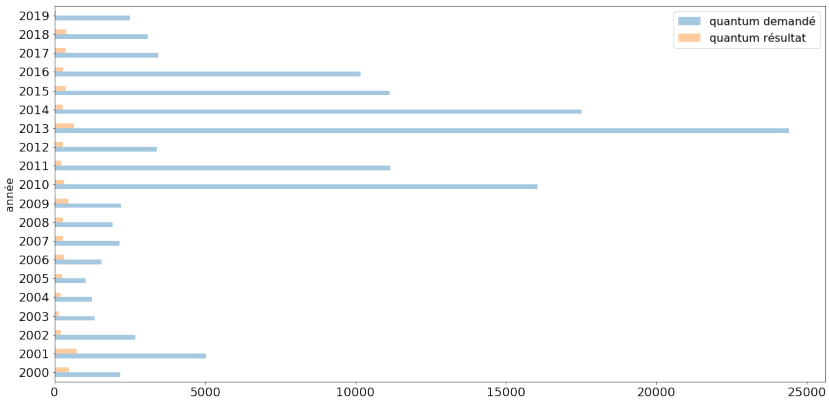



Figure 6.5 – Evolution des quanta moyens par année des demandes *styx* entre 2000 et 2019.

De même l'évolution des quanta demandés et accordés peut être facilement visualisée par un diagramme en barre comme celui de la Figure 6.5 Page 138 qui correspond aux demandes *styx* entre 2000 et 2019. Même si le nombre total de demandes est à prendre en compte, un tel diagramme donne un aperçu des sommes d'argent moyennes demandées et accordées chaque année. Malheureusement, une seule valeur aberrante très élevée a un impact négatif sur l'interprétation de la moyenne. On observe par exemple une moyenne particulièrement haute en 2013 (Figure 6.5 Page 138). On préfèrera des diagrammes boîtes (*box plot*) comme celui de l'évolution des quanta accordés de moins de 10k € à Bastia (Figure 6.6 Page 140). Par exemple, même si les médianes en 2001 et 2002 sont presque égales, le quanta accordés non nuls ont été très proches en 2001 qu'en 2002 où la distribution est plus large.

### 6.2.2 Variabilité dans les territoires

Pour avoir une idée du montant que l'on peut recevoir pour une catégorie de demande, l'évolution des valeurs généralement accordées peut être comparée entre deux villes en visualisant les diagrammes boîtes des quanta accordés dans ces villes. La Figure 6.6 Page 140 permet d'effectuer des comparaisons entre Bastia et Lyon. En 2008 par exemple, les quanta accordés sont plus proches entre eux à Lyon qu'à Bastia.

### 6.2.3 Quantum demandé vs. quantum accordé

La prédiction du quantum résultat doit définir un modèle dont la forme s'accorde avec celle du nuage de points ($x$ = quantum demandé, $y$ = quantum accordé) correspondant. D'après les nuages de points observés pour Paris, Bastia, Angers et Lyon (Figure 6.7 Page 141), le quantum demandé ne semble pas suffisant seul pour déterminer le quantum accordé [2]. Il sera ainsi nécessaire de tenir compte des circonstances factuelles et autres spécificités du cas traité qui permettront de filtrer les décisions sur lesquelles se basera l'apprentissage. On remarque néanmoins une ressemblance de forme entre les nuages de points des différentes villes. On observe empiriquement une caractéristique du droit qui est « l'impossibilité d'accorder plus qu'une somme demandée ».

2. Différentes valeurs de quantum résultat sont observées pour la même valeur de quantum demandé.

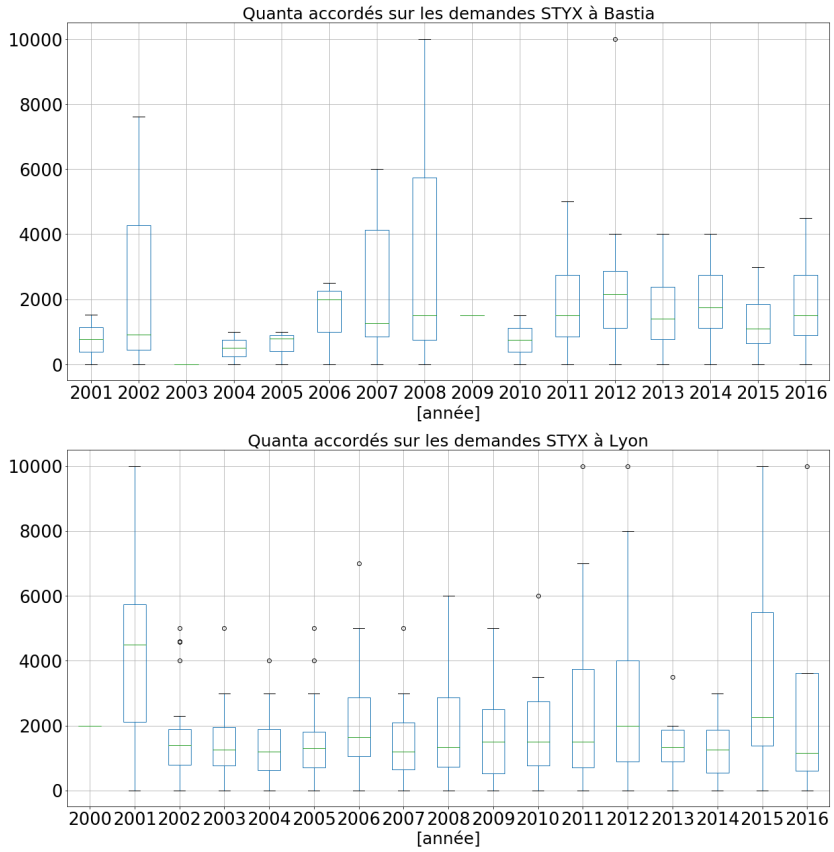



Figure 6.6 – Evolution des quanta accordés (< 10k €) par année sur les demandes *styx* entre 2000 et 2016 à Bastia et à Lyon.

## 6.3 Conclusion

Les démonstrations de ce chapitre donnent quelques exemples de statistiques qui informent de l'état de la jurisprudence à partir d'informations extraites à l'aide des approches proposées dans cette thèse. Les analyses du sens du résultat et des quanta sont les principales applications directes de la chaîne de traitement développée. Ce chapitre se limite aux filtres sur l'année, la ville, et la catégorie de demande, mais les analyses peuvent déjà être affinées en associant d'autres filtres comme des mot-clés, les normes appliquées, ou le type de juridiction. Les analyses pourront être enrichies grâce l'extraction future de nouvelles informations comme les motivations des juges et de meilleurs modèles d'identification de circonstances factuelles.

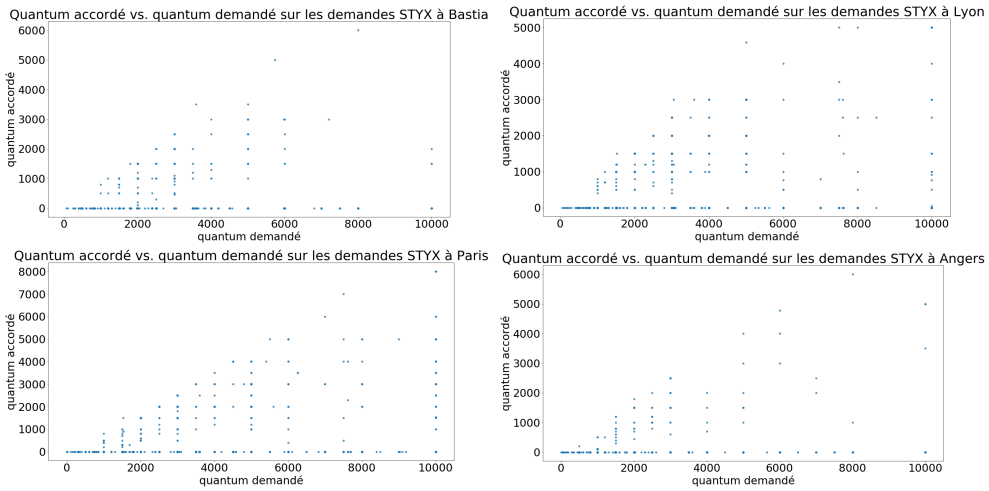



Figure 6.7 – Nuages des points (quantum accordé, quantum demandé) pour les demandes *styx* entre 2000 et 2019 à Paris, Bastia, Angers et Lyon (quantum demandé < 10000) .

# Conclusion générale

## i Évaluation des contributions

Cette thèse porte essentiellement sur la proposition et l'exploration d'approches adressant des problèmes d'analyses de données textuelles rencontrés lors de l'étude de corpus jurisprudentiels par des experts juristes. Trois problèmes principaux y sont abordés. Premièrement, l'annotation, dans les documents, des sections de textes et des entités juridiques, est traitée afin d'aider les experts à se repérer dans le document et à améliorer leur recherche de décisions judiciaires. Le chapitre 2 démontre empiriquement, sur des documents annotés manuellement, l'efficacité de l'application de modèles probabilistes d'étiquetage de séquences, HMM et CRF, sur les deux tâches. Par la suite, l'extraction de données relatives aux demandes, suivant leur catégorie juridique, est discutée dans les chapitres 3 et 4. Le problème impose d'effectuer les extractions pour une catégorie de demandes à la fois car il est impossible d'annoter suffisamment de données pour toutes les catégories prédéfinies. Pour cela nous proposons de filtrer à l'entrée les documents de la catégorie à traiter par une classification binaire. Ensuite, nous proposons une approche identifiant les attributs des demandes à l'aide de termes-clés prédéfinis et appris. Cette méthode, bien que dépendante d'heuristiques, parvient à reconnaître un grand nombre de demandes avec plus ou moins de difficultés selon les catégories traitées. Ensuite, la classification de documents est expérimentée comme approche plus généraliste. Sur l'ensemble des algorithmes explorés, les extensions de l'analyse PLS, appliquées ici pour la première fois sur du texte, démontrent une efficacité proche de celle du meilleur algorithme testé, l'arbre de décision. L'utilité de la restriction des documents à des passages relatifs à la catégorie est aussi observée empiriquement. Enfin, le chapitre 5 aborde la problématique de similarité entre deux textes dans un contexte de catégorisation non supervisée des documents. Le but est ici de révéler les circonstances factuelles faisant appel à une catégorie de demande particulière. Une approche d'apprentissage de distance est proposée : elle repose sur le coût d'une transformation d'un des deux textes en l'autre. Cette distance est comparée à d'autres métriques avec

l'algorithme des K-moyennes dans des expérimentations qui explorent différents aspects des problèmes de regroupement comme la détermination du nombre de clusters ou la représentation de documents. En somme, les problèmes abordés sont variés et très importants dans le métier des experts juristes. Le chapitre 6 illustre en l'occurrence la riche visibilité sur la jurisprudence qui devient facilement accessible à ces derniers grâce à l'extraction automatique des connaissances jurisprudentielles.

## ii Critique du travail

Cette thèse est limitée par la faible quantité des données employées pour les expérimentations. Cette dernière est révélatrice de la lenteur et de la pénibilité liée à l'annotation manuelle des jeux de données d'évaluation. Par conséquent, il est difficile d'avoir une estimation du taux de données manquées par l'expert ou du degré différence entre son annotation manuelle et celles qu'auraient pu réaliser d'autres juristes. De plus, ne disposant la plupart du temps que d'un expert, l'annotation manuelle n'a été évaluée que pour le problème de détection des sections et entités juridiques. Par ailleurs, un grand nombre de méthodes de la littérature n'ont pas été expérimentées pour deux raisons principales. D'une part, la littérature regorge de très nombreuses méthodes répondant aux divers problèmes traités ici. D'autre part, certaines méthodes intéressantes ne sont pas adaptés à nos conditions d'expérimentation. Par exemple, les réseaux neurones profonds sont réputés gourmandes en données d'entraînement que nous ne disposons pas. Nous avons aussi expliqué par exemple que les méthodes proposées pour l'extraction des évènements exploite une annotation manuelle qui renseigne sur la position exacte où se trouve les données ciblées dans le texte. Les données d'apprentissage pour l'identification des demandes sont répertoriées, au contraire, dans un tableau à l'extérieur des documents d'origine.

## iii Travaux futurs de recherche

Les propositions données dans la conclusion des chapitres 2 à 6 pour poursuivre les travaux peuvent être résumées en 4 catégories principale. En premier, l'exploration de méthodes plus récentes que celles étudiées dans ce manuscrit permettra d'étendre les résultats expérimentaux. Ensuite, la formalisation des problèmes abordés permettra de définir des approches plus théoriques. Par exemple, la formalisation des demandes

comme des relations, entre quantum demandé et quantum accordé, permettra d'explorer le cadre probabiliste et neuronal de la littérature en matière d'extraction des relations. Puis, l'exploration d'autres formulations des problèmes permettra probablement de découvrir des méthodes plus efficaces. Par exemple, on peut percevoir la détermination des circonstances factuelles comme une tâche de modélisation de thématiques (*topic modeling*). Enfin, les études menées méritent d'être étendues sur d'autres aspects. Par exemple, la détection d'entités juridiques doit être étendue la résolution qui unifie les mentions variantes d'une entité sous un identifiant prédéfinir ou à définir automatiquement. Cette résolution est importante pour l'automatisation d'autres tâches du métier comme l'anonymisation des décisions judiciaires.

Il faut aussi remarquer qu'il reste encore des types d'information dont le problème d'extraction n'est pas abordé par cette thèse. Par exemple, les raisons, qui font pencher les juges en faveur d'une décision sur une demande, sont indispensables pour être capable d'anticiper la prise de décision des juges. L'extraction des raisons concernera l'identification et l'analyse des arguments des parties et les motivations des juges.

Par ailleurs, il faudra aussi mieux évaluer la qualité des annotations manuelles expertes ce qui révélera le niveau d'accord non seulement sur les données annotées mais aussi sur leur perception des informations ciblées comme les circonstances factuelles qui semblent restées subjectives.

Cette thèse est l'un des premiers travaux de recherche d'une telle diversité de problèmes sur les décisions de justice françaises. Ainsi, elle ouvre la voie à bien des problématiques comme l'analyse des réseaux de normes, l'anonymisation des décisions, ou l'analyse des arguments, déjà largement étudiés dans d'autres pays, notamment aux États-Unis. En cela, cette thèse encourage la recherche en analyse de données textuelles à s'intéresser à l'analyse automatique de la jurisprudence française dont les défis, la disponibilité d'un grand volume de données et la lucrativité du domaine judiciaire ne rendent ce champ d'application que plus attractif. Les cas d'utilisation des données extraites sont très nombreuses pour la recherche en droit, l'aide à la décision des juristes, pour l'enseignement du droit, mais aussi et surtout pour l'accessibilité des profanes au droit par une estimation automatique de leurs risques judiciaires.

# Bibliographie


Afanador, Nelson Lee, Smolinska, Agnieszka, Tran, Thanh N., & Blanchet, Lionel. 2016. Unsupervised random forest : a tutorial with case studies. *Journal of Chemometrics*, **30**(5), 232–241.

Afzali, Maedeh, & Kumar, Suresh. 2018. An Extensive Study of Similarity and Dissimilarity Measures Used for Text Document Clustering using K-means Algorithm. *International Journal of Information Technology and Computer Science (IJITCS)*, **9**, 64–73.

Aggarwal, Charu C., Hinneburg, Alexander, & Keim, Daniel A. 2001. On the surprising behavior of distance metrics in high dimensional space. *Pages 420–434 of : International Conference on Database Theory*. Springer.

Agrawal, Rakesh, Srikant, Ramakrishnan, *et al.* 1994. Fast algorithms for mining association rules. *Pages 487–499 of : Proc. 20th int. conf. very large data bases, VLDB*, vol. 1215.

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

Aletras, Nikolaos, Tsarapatsanis, Dimitrios, Preoţiuc-Pietro, Daniel, & Lampos, Vasileios. 2016. Predicting judicial decisions of the European Court of Human Rights : A Natural Language Processing perspective. *PeerJ Computer Science*, **2**, e93.

Aleven, Vincent. 2003. Using Background Knowledge In Case-based Legal Reasoning : A Computational Model And An Intelligent Learning Environment. *Artificial Intelligence*, **150**(1-2), 183–237.

Aleven, Vincent, & Ashley, Kevin D. 1997. Evaluating A Learning Environment For Case-based Argumentation Skills. *Pages 170–179 of : Proceedings of the 6th international conference on artificial intelligence and law (ICAIL)*. ACM.

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

Amami, Rimah, Ayed, Dorra Ben, & Ellouze, Noureddine. 2013. Practical Selection of SVM Supervised Parameters with Different Feature Representations for Vowel Recognition. *International Journal of Digital Content Technology and its Applications (JDCTA)*, **7**(9).

Amarappa, S., & Sathyanarayana, S. V. 2015. Kannada named entity recognition and classification (NERC) based on multinomial naïve bayes (MNB) classifier. *International Journal on Natural Language Computing (IJNLC)*, **4**(4).

Ancel, Pascal. 2003. *Les décisions d'expulsion d'occupants sans droit ni titre - Connaissance empirique d'un contentieux hétérogène*. Tech. rept. Ministère de la Justice. `https://halshs.archives-ouvertes.fr/halshs-00798914/document`.

Andrew, Judith Jeyafreeda, & Tannier, Xavier. 2018. Automatic Extraction of Entities and Relation from Legal Documents. *Pages 1–8 of : Proceedings of the Seventh Named Entities Workshop*.

Arora, Sanjeev, Liang, Yingyu, & Ma, Tengyu. 2017. a Simple But Tough-to-beat Baseline For Sentence Embeddings. *In : Proceedings of 5th International Conference on Learning Representations (ICLR)*.

Ashley, Kevin D. 1990. *Modeling Legal Arguments : Reasoning With Cases And Hypotheticals*. MIT press.

Ashley, Kevin D., & Brüninghaus, Stefanie. 2009. Automatically classifying case texts and predicting outcomes. *Artificial Intelligence and Law*, **17**(2), 125–165.

Bakkelund, Daniel. 2009. An LCS-based string metric. *Olso, Norway : University of Oslo*.

Balabantaray, Rakesh Chandra, Sarma, Chandrali, & Jha, Monica. 2015. *Document Clustering Using K-means And K-medoids*. arXiv preprint arXiv :1502.07938 [cs.IR].

Baldwin, Breck. 2009. *Coding chunkers as taggers : IO, BIO, BMEWO, and BMEWO+*. `https://lingpipe-blog.com/2009/10/14/coding-chunkers-as-taggers-io-bio-bmewo-and-bmewo/`.

Bandalos, Deborah L., & Boehm-Kaufman, Meggen R. 2010. Four common misconceptions in exploratory factor analysis. *Pages 81–108 of : Statistical and methodological myths and urban legends*. Routledge.

Baraldi, Andrea, & Blonda, Palma. 1999. A survey of fuzzy clustering algorithms for pattern recognition. I. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, **29**(6), 778–785.

Bazzoli, Caroline, & Lambert-Lacroix, Sophie. 2018. Classification based on extensions of LS-PLS using logistic regression : application to clinical and multiple genomic data. *BMC bioinformatics*, **19**(1), 314.

Ben-Hur, Asa, & Weston, Jason. 2010. A User's Guide to Support Vector Machines. *Chap. 13, pages 223–239 of : Data Mining Techniques for the Life Sciences*. Totowa, NJ : Humana Press.

Bench-Capon, Trevor J.M. 1997. Arguing With Cases. *Pages 85–100 of : Proceedings of The Tenth Conference of The Foundation for Legal Knowledge Systems (JURIX'97)*.

Berka, Petr. 2011. NEST : A Compositional Approach to Rule-Based and Case-Based Reasoning. *Advances in Artificial Intelligence*, **2011**, 15.

Bezdek, James C, Ehrlich, Robert, & Full, William. 1984. FCM : The fuzzy c-means clustering algorithm. *Computers & Geosciences*, **10**(2-3), 191–203.

Blei, David M., Ng, Andrew Y., & Jordan, Michael I. 2003. Latent Dirichlet Allocation. *the Journal of Machine Learning Research*, **3**, 993–1022.

Bommarito, Michael James, Katz, Daniel Martin, & Detterman, Eric. 2018 (June). *LexNLP : Natural Language Processing and Information Extraction For Legal and Regulatory Texts*. Available at SSRN : https ://ssrn.com/abstract=3192101 or http ://dx.doi.org/10.2139/ssrn.3192101.

Bray, J Roger, & Curtis, John T. 1957. An ordination of the upland forest communities of southern Wisconsin. *Ecological monographs*, **27**(4), 325–349.

Breiman, Leo. 2001. Random Forests. *Machine Learning*, **45**(1), 5–32.

Breiman, Leo, Friedman, J. H., Olshen, R. A., & Stone, C. J. 1984. *Classification and Regression Trees*. Statistics/Probability Series. Belmont, California, U.S.A. : Wadsworth Publishing Company.

Brown, Ralf D. 2013. Selecting and weighting n-grams to identify 1100 languages. *Pages 475–483 of : International Conference on Text, Speech and Dialogue*. Springer.

Brüninghaus, Stefanie, & Ashley, Kevin D. 2001. Improving the representation of legal case texts with information extraction methods. *Pages 42–51 of : Proceedings of the 8th international conference on Artificial intelligence and law*. ACM.

Bruninghaus, Stefanie, & Ashley, Kevin D. 2003. Predicting outcomes of case based legal arguments. *Pages 233–242 of : Proceedings of the 9th international conference on Artificial intelligence and law*. ACM.

Burrows, John F. 1992. Not unles you ask nicely : The interpretative nexus between analysis and information. *Literary and Linguistic Computing*, **7**(2), 91–109.

Cardellino, Cristian, Teruel, Milagro, *et al.* 2017. A Low-cost, High-coverage Legal Named Entity Recognizer, Classifier And Linker. *Pages 9–18 of : Proceedings of the 16th edition of the International Conference on Articial Intelligence and Law*. ACM.

Chang, Yu-shan, & Sung, Yun-Hsuan. 2005. *Applying name entity recognition to informal text*. Tech. rept. Stanford University. CS224N/Ling237 Final Project Report.

Charlet, Delphine, & Damnati, Geraldine. 2017. Simbow at semeval-2017 task 3 : Soft-cosine semantic similarity between questions for community question answering. *Pages 315–319 of : Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*.

Charlet, Delphine, & Damnati, Géraldine. 2018. Similarité textuelle pour l'association de documents journalistiques. *In : 15e Conférence en Recherche d'Information et Applications (CORIA)*.

Chau, Michael, Xu, Jennifer J, & Chen, Hsinchun. 2002. Extracting Meaningful Entities From Police Narrative Reports. *Pages 1–5 of : Proceedings of the 2002 annual national conference on Digital government research*. Digital Government Society of North America.

Chiticariu, Laura, Krishnamurthy, Rajasekar, Li, Yunyao, Reiss, Frederick, & Vaithyanathan, Shivakumar. 2010. Domain adaptation of rule-based

annotators for named-entity recognition tasks. *Pages 1002–1012 of : Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Cohen, Jacob. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, **20**(1), 37–46.

Cortes, Corinna, & Vapnik, Vladimir. 1995. Support-vector networks. *Machine Learning*, **20**(3), 273–297.

Cover, Thomas, & Hart, Peter. 1967. Nearest Neighbor Pattern Classification. *IEEE Transactions on Information Theory*, **13**(1), 21–27.

Cretin, Laurette. 2014. L'opinion des Français sur la justice. *INFOSTAT JUSTICE*, **125**(Janvier).

Deerwester, Scott, Dumais, Susan T., Furnas, George W., Landauer, Thomas K., & Harshman, Richard. 1990. Indexing By Latent Semantic Analysis. *Journal Of The American Society For Information Science*, **41**(6), 391–407.

DILA. 2019. *Base de données CAPP*. Page d'accueil à `https://www.data.gouv.fr/fr/datasets/capp/`, fichiers CAPP_20180315-195806.tar.gz à CAPP_20190805-214041.tar.gz et Freemium_capp_global_20180315-170000.tar.gz téléchargés à partir de `ftp://echanges.dila.gouv.fr/CAPP/`.

Dong, Yan-Shi, & Han, Ke-Song. 2005. Boosting SVM classifiers by ensemble. *Pages 1072–1073 of : Special Interest Tracks And Posters Of The 14th International Conference On World Wide Web*. ACM.

Dozier, Christopher, Kondadadi, Ravikumar, Light, Marc, Vachher, Arun, Veeramachaneni, Sriharsha, & Wudali, Ramdev. 2010. Named entity recognition and resolution in legal text. *Pages 27–43 of : Semantic Processing of Legal Texts*. Springer.

Duda, Richard O., Hart, Peter E., *et al.* 1973. *Pattern Classification And Scene Analysis*. Vol. 3. New York : John Wiley & Sons.

Dumais, Susan T., Furnas, George W., Landauer, Thomas K., Deerwester, Scott, & Harshman, Richard. 1988. Using Latent Semantic Analysis To Improve Access To Textual Information. *Pages 281–285 of : Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM.

Durif, Ghislain, Modolo, Laurent, Michaelsson, Jakob, Mold, Jeff E., Lambert-Lacroix, Sophie, & Picard, Franck. 2017. High dimensional classification with combined adaptive sparse PLS and logistic regression. *Bioinformatics*, **34**(3), 485–493.

Elman, Jeffrey L. 1990. Finding Structure In Time. *Cognitive science*, **14**(2), 179–211.

Emmanuel, Barthe. 20 janvier 2010. *Arrêts des cours d'appel : la base JURICA enfin en service chez Lexbase*. `https://www.precisement.org/blog/Arrets-des-cours-d-appel-la-base.html`.

Ester, Martin, Kriegel, Hans-Peter, Sander, Jörg, Xu, Xiaowei, *et al.* 1996. A Density-based Algorithm For Discovering Clusters In Large Spatial Databases With Noise. *Pages 226–231 of : KDD*, vol. 96.

Fang, Anjie, Macdonald, Craig, Ounis, Iadh, & Habel, Philip. 2016. Using word embedding to evaluate the coherence of topics from twitter data. *Pages 1057–1060 of : Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM.

Finkel, Jenny Rose, Grenager, Trond, & Manning, Christopher. 2005. Incorporating Non-local Information Into Information Extraction Systems By Gibbs Sampling. *Pages 363–370 of : Proceedings of the 43rd annual meeting on association for computational linguistics*. Association for Computational Linguistics.

Fisher, Ronald A. 1936. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, **7**(2), 179–188.

Forgey, Edward. 1965. Cluster analysis of multivariate data : Efficiency vs. interpretability of classification. *Biometrics*, **21**(3), 768–769.

Frank, Eibe, Hall, Mark A., & Witten, Ian H. 2016. *The WEKA workbench*. Fourth edn. Morgan Kaufmann. `https://www.cs.waikato.ac.nz/ml/weka/Witten_et_al_2016_appendix.pdf`. Page 128 p.

Frantzi, Katerina, Ananiadou, Sophia, & Mima, Hideki. 2000. Automatic recognition of multi-word terms :. the c-value/nc-value method. *International journal on digital libraries*, **3**(2), 115–130.

Galavotti, Luigi, Sebastiani, Fabrizio, & Simi, Maria. 2000. Experiments on the use of feature selection and negative evidence in automated text categorization. *Pages 59–68 of : International Conference on Theory and Practice of Digital Libraries*. Springer.

Genesereth, Michael. 2015. Computational Law : The Cop in the Backseat. *The standford Center for Legal Informatics hosted the third annual FutureLaw 2015 conference*.

Ghojogh, Benyamin, & Crowley, Mark. 2019. *Linear and quadratic discriminant analysis : Tutorial*. arXiv preprint arXiv :1906.02590 [stat.ML].

Grave, E., Mikolov, T., Joulin, A., & Bojanowski, P. 2017 (April 3-7). Bag of tricks for efficient text classification. *Pages 427–431 of : Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, vol. 2.

Grishman, Ralph, & Sundheim, Beth. 1996. Message understanding conference-6 : A brief history. *In : COLING 1996 Volume 1 : The 16th International Conference on Computational Linguistics*, vol. 1.

Guttman, Louis. 1954. Some necessary conditions for common-factor analysis. *Psychometrika*, **19**(2), 149–161.

Halkidi, Maria, Batistakis, Yannis, & Vazirgiannis, Michalis. 2001. On clustering validation techniques. *Journal of intelligent information systems*, **17**(2-3), 107–145.

Hanisch, Daniel, Fundel, Katrin, *et al.* 2005. ProMiner : rule-based protein and gene entity recognition. *BMC bioinformatics*, **6**(1), 14.

Harispe, Sébastien, Ranwez, Sylvie, Janaqi, Stefan, & Montmain, Jacky. 2013. The semantic measures library and toolkit : fast computation of semantic similarity and relatedness using biomedical ontologies. *Bioinformatics*, **30**(5), 740–742.

Harispe, Sébastien, Ranwez, Sylvie, Janaqi, Stefan, & Montmain, Jacky. 2015. Semantic similarity from natural language and ontology analysis. *Synthesis Lectures on Human Language Technologies*, **8**(1), 1–254.

Hathaway, Richard J., Davenport, John W., & Bezdek, James C. 1989. Relational duals of the c-means clustering algorithms. *Pattern recognition*, **22**(2), 205–212.

Hirschberg, Daniel S. 1977. Algorithms For The Longest Common Subsequence Problem. *Journal of the ACM (JACM)*, **24**(4), 664–675.

Huang, Anna. 2008. Similarity measures for text document clustering. *Pages 9–56 of : Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand*, vol. 4.

Hubert, Lawrence, & Arabie, Phipps. 1985. Comparing partitions. *Journal of classification*, **2**(1), 193–218.

Im, Chan Jong, Mandl, Thomas, *et al.* 2017. Text Classification for Patents : Experiments with Unigrams, Bigrams and Different Weighting Methods. *International Journal of Contents*, **13**(2).

Jaccard, Paul. 1901. Etude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bulletin de la Société Vaudoise Sciences Naturelles*, **37**, 547–579.

Jeandidier, Bruno, & Ray, Jean-Claude. 2006. 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.

Jones, K. Sparck, Walker, Steve, & Robertson, Stephen E. 2000. A Probabilistic Model Of Information Retrieval : Development And Comparative Experiments. *Information Processing & Management*, **36**(6), 809–840.

Jordan, Michael I. 1986. *Serial Order : A Parallel Distributed Processing Approach. Technical Report, June 1985 - March 1986*. Tech. rept. California Univ., San Diego, La Jolla (USA). Inst. for Cognitive Science.

Kaiser, Henry F. 1960. The application of electronic computers to factor analysis. *Educational and psychological measurement*, **20**(1), 141–151.

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

Katz, Daniel Martin, Bommarito II, Michael J, & Blackman, Josh. 2017. A general approach for predicting the behavior of the Supreme Court of the United States. *PloS one*, **12**(4), e0174698.

Kaufman, Leonard, & Rousseeuw, Peter J. 1987. Clustering By Means Of Medoids. *Page 405–416 of :* Yadolah Dodge (ed), *Statistical Data Analysis Based on the L1-Norm*. North Holland/Elsevier. Amsterdam.

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

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

Kittler, Josef, Hater, Mohamad, & Duin, Robert P.W. 1996. Combining classifiers. *Pages 897–901 of : Proceedings of 13th international conference on pattern recognition*, vol. 2. IEEE.

Kittler, Josef, Hatef, Mohamad, Duin, Robert PW, & Matas, Jiri. 1998. On combining classifiers. *IEEE transactions on pattern analysis and machine intelligence*, **20**(3), 226–239.

Klinger, Roman, & Friedrich, Christoph M. 2009. Feature subset selection in conditional random fields for named entity recognition. *Pages 185–191 of : Proceedings of the International Conference RANLP-2009*.

Konkol, Michal, & Konopík, Miloslav. 2015. Segment representations in named entity recognition. *Pages 61–70 of : International Conference on Text, Speech, and Dialogue*. Springer.

Krishnapuram, Raghu, Joshi, Anupam, Nasraoui, Olfa, & Yi, Liyu. 2001. Low-Complexity Fuzzy Relational Clustering Algorithms For Web Mining. *IEEE transactions on Fuzzy Systems*, **9**(4), 595–607.

Kríz, Vincent, Hladká, Barbora, Dedek, Jan, & Necaský, Martin. 2014. Statistical Recognition of References in Czech Court Decisions. *Pages 51–61 of :* Gelbukh, Alexander and Espinoza, Félix Castro and Galicia-Haro, Sofía N. (ed), *Human-Inspired Computing and Its Applications : 13th Mexican International Conference on Artificial Intelligence, MICAI 2014, Tuxtla Gutiérrez, Mexico, November 16-22, 2014. Proceedings, Part I*. Cham : Springer International Publishing.

Kroll, Charles N, & Song, Peter. 2013. Impact of multicollinearity on small sample hydrologic regression models. *Water resources research*, **49**(6), 3756–3769.

Kumar, Sushanta, Reddy, P Krishna, Reddy, V Balakista, & Singh, Aditya. 2011. Similarity analysis of legal judgments. *Page 17 of : Proceedings of Compute 2011 - Fourth Annual ACM Bangalore Conference*. ACM.

Kuncheva, Ludmila I. 2004. *Combining pattern classifiers : methods and algorithms*. John Wiley & Sons.

Kusner, Matt, Sun, Yu, Kolkin, Nicholas, & Weinberger, Kilian. 2015. From word embeddings to document distances. *Pages 957–966 of : International Conference on Machine Learning*.

Kvalseth, Tarald O. 1987. Entropy and correlation : Some comments. *IEEE Transactions on Systems, Man, and Cybernetics*, **17**(3), 517–519.

Lacroux, Alain. 2011. Les avantages et les limites de la méthode «Partial Least Square »(PLS) : une illustration empirique dans le domaine de la GRH. *Revue de gestion des ressources humaines*, **80**(2), 45–64.

Lafferty, John, McCallum, Andrew, & Pereira, Fernando C. N. 2001. Conditional random fields : probabilistic models for segmenting and labeling sequence data. *International Conference on Machine Learning*.

Lamanda, Vincent. 2010. *Discours du Premier Président de la Cour de Cassation Vincent Lamanda lors de l'audience solennelle de début d'année 2010*. `https://www.courdecassation.fr/institution_1/occasion_audiences_59/debut_annee_60/discours_m._lamanda_14858.html`.

Lample, Guillaume, Ballesteros, Miguel, Subramanian, Sandeep, Kawakami, Kazuya, & Dyer, Chris. 2016. *Neural architectures for named entity recognition*. arXiv preprint arXiv :1603.01360 [cs.CL].

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

Langlais, Eric, & Chappe, Nathalie. 2009. *Analyses économiques du droit : principes, méthodes, résultats*. Editions de Boeck Université. Chap. 4. Analyse économique de la résolution des litiges.

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

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

Le, Quoc, & Mikolov, Tomas. 2014. Distributed representations of sentences and documents. *Pages 1188–1196 of : International conference on machine learning*.

Leith, Philip. 2010. The rise and fall of the legal expert system. *European Journal of Law and Technology*, **1**(1), 179–201.

Li, Jianqiang, Zhao, Shenhe, Yang, Jijiang, Huang, Zhisheng, Liu, Bo, Chen, Shi, Pan, Hui, & Wang, Qing. 2018. WCP-RNN : a novel RNN-based approach for Bio-NER in Chinese EMRs. *The Journal of Supercomputing*, 1–18.

Li, Yaoyong, Zaragoza, Hugo, Herbrich, Ralf, Shawe-Taylor, John, & Kandola, Jaz. 2002. The perceptron algorithm with uneven margins. *Pages 379–386 of : ICML*, vol. 2.

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

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

Liu, Jingjing, Pasupat, Panupong, Cyphers, Scott, & Glass, Jim. 2013. ASGARD : A Portable Architecture For Multilingualdialogue Systems. *Pages 8386–8390 of : 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.

Liu, Yushu, & Rayens, William. 2007. PLS and dimension reduction for classification. *Computational Statistics*, **22**(2), 189–208.

Llewellyn, Karl Nickerson. 1962. *Jurisprudence : Realism in Theory and Practice*. The University of Chicago Press.

Love, Nathaniel, & Genesereth, Michael. 2005. Computational law. *Pages 205–209 of : Proceedings of the 10th international conference on Artificial intelligence and law*. ACM.

Ma, Yinglong, Zhang, Peng, & Ma, Jiangang. 2018. *An Efficient Approach to Learning Chinese Judgment Document Similarity Based on Knowledge Summarization*. arXiv preprint arXiv :1808.01843 [cs.AI].

Mandal, Arpan, Ghosh, Kripabandhu, Bhattacharya, Arnab, Pal, Arindam, & Ghosh, Saptarshi. 2017. Overview of the FIRE 2017 IRLeD

Track : Information Retrieval from Legal Documents. *Pages 63–68 of : FIRE (Working Notes)*.

Manning, Christopher D., Raghavan, Prabhakar, & Schütze, Hinrich. 2009a. Flat clustering. *Chap. 16, pages 349–375 of : Introduction to information retrieval*. Cambridge : Cambridge university press.

Manning, Christopher D, Raghavan, Prabhakar, & Schütze, Hinrich. 2009b. Scoring, term weighting and the vector space model. *Chap. 6, pages 109–133 of : Introduction to information retrieval*. Cambridge : Cambridge university press.

Marascuilo, Leonard A. 1966. Large-sample multiple comparisons. *Psychological bulletin*, **65**(5), 280.

Martineau, Justin, & Finin, Tim. 2009. Delta TFIDF : An Improved Feature Space for Sentiment Analysis. *In : Third International AAAI Conference on Weblogs and Social Media (ICWSM)*.

McCallum, Andrew Kachites. 2012. *MALLET : A Machine Learning for Language Toolkit*. `http://mallet.cs.umass.edu/`.

McCulloch, Warren S., & Pitts, Walter. 1943. A Logical Calculus Of The Ideas Immanent In Nervous Activity. *The bulletin of mathematical biophysics*, **5**(4), 115–133.

McLachlan, Geoffrey J. 1992. *Discriminant analysis and statistical pattern recognition*. John Wiley & Sons.

Medvedeva, Masha, Vols, Michel, & Wieling, Martijn. 2018. Judicial Decisions of the European Court of Human Rights : Looking into the Crystal Ball. *In : Proceedings of the Conference on Empirical Legal Studies*.

Mikheev, Andrei, Moens, Marc, & Grover, Claire. 1999. Named entity recognition without gazetteers. *Pages 1–8 of : Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.

Mikolov, Tomas, Chen, Kai, Corrado, Greg, & Dean, Jeffrey. 2013. Efficient estimation of word representations in vector space. *In : Proceedings of the International Conference on Learning Representations (ICLR)*.

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

Moens, Marie-Francine. 2002. What information retrieval can learn from case-based reasoning. *Pages 83–91 of : Legal Knowledge and Information Systems.* Amsterdam : T.J.M. Bench-Capon, A. Daskalopulu and R.G.F. Winkels (eds.), for Jurix 2002 : The Fifteenth Annual Conference.

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

Muhlenbach, Fabrice, & Sayn, Isabelle. 2019 (June). Artificial Intelligence and Law : What Do People Really Want ? : Example of a French Multidisciplinary Working Group. *Pages 224–228 of : Proceedings of the 17th International Conference on Artificial Intelligence and Law*. ACM.

Mussard, Stéphane, & Souissi-Benrejab, Fattouma. 2018. Gini-PLS Regressions. *Journal of Quantitative Economics*, April, 1–36.

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

Nair, Akhil M., & Wagh, Rupali Sunil. 2018. Similarity Analysis of Court Judgements Using Association Rule Mining on Case Citation Data - A Case Study. *International Journal of Engineering Research and Technology*, **11**(3), 373–381.

Nallapati, Ramesh, Surdeanu, Mihai, & Manning, Christopher. 2010. Blind domain transfer for named entity recognition using generative latent topic models. *Pages 281–289 of : Proceedings of the NIPS 2010 Workshop on Transfer Learning Via Rich Generative Models*.

Nazarenko, Adeline, & Wyner, Adam. 2017. Legal NLP Introduction. *Traitement automatique de la langue juridique / Legal Natural Language Processing - Revue TAL*, **58**(2), 7–19.

Nefti, Samia, & Oussalah, Mourad. 2004. Probabilistic-fuzzy clustering algorithm. *Pages 4786–4791 of : 2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583)*, vol. 5. IEEE.

Ng, Hwee Tou, Goh, Wei Boon, & Low, Kok Leong. 1997. Feature selection, perceptron learning, and a usability case study for text categorization. *Pages 67–73 of : ACM SIGIR Forum*, vol. 31. ACM.

Nguyen, Thien Huu, Cho, Kyunghyun, & Grishman, Ralph. 2016. Joint Event Extraction via Recurrent Neural Networks. *Pages 300–309 of : HLT-NAACL*.

Nigam, Kamal, Lafferty, John, & McCallum, Andrew. 1999. Using maximum entropy for text classification. *Pages 61–67 of : IJCAI-99 Workshop on Machine Learning for Information Filtering*, vol. 1.

Olkin, Ingram, & Yitzhaki, Shlomo. 1992. Gini regression analysis. *International Statistical Review/Revue Internationale de Statistique*, 185–196.

Paatero, Pentti, & Tapper, Unto. 1994. Positive matrix factorization : A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, **5**(2), 111–126.

Pagliardini, Matteo, Gupta, Prakhar, & Jaggi, Martin. 2018. Unsupervised Learning of Sentence Embeddings using Compositional n-Gram Features. *In : NAACL 2018 - Conference of the North American Chapter of the Association for Computational Linguistics*.

Palm, Rasmus Berg, Hovy, Dirk, Laws, Florian, & Winther, Ole. 2017. End-to-End Information Extraction without Token-Level Supervision. *In : Proceedings of the Workshop on Speech-Centric Natural Language Processing*.

Palmer, David D., & Day, David S. 1997. A statistical profile of the named entity task. *Pages 190–193 of : Proceedings of the fifth conference on Applied natural language processing*. Association for Computational Linguistics.

Paltoglou, Georgios, & Thelwall, Mike. 2010. A study of information retrieval weighting schemes for sentiment analysis. *Pages 1386–1395 of : Proceedings of the 48th annual meeting of the association for computational linguistics*. Association for Computational Linguistics.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., *et al.* 2011. Scikit-learn : Machine Learning in Python. *Journal of Machine Learning Research*, **12**, 2825–2830.

Pennington, Jeffrey, Socher, Richard, & Manning, Christopher. 2014. Glove : Global vectors for word representation. *Pages 1532–1543 of : Proceedings Of The 2014 Conference On Empirical Methods In Natural Language Processing (EMNLP)*.

Persson, Caroline. 2012. *Machine Learning for Tagging of Biomedical Literature*. Tech. rept. Technical University of Denmark, DTU Informatics.

Polifroni, Joe, & Mairesse, François. 2011. Using Latent Topic Features for Named Entity Extraction in Search Queries. *Pages 2129–2132 of : INTERSPEECH*.

Poole, David, & Mackworth, Alan. 2017. *Artificial Intelligence : Foundations of Computational Agents*. Cambridge University Press. Chap. 7 Supervised Machine Learning.

Price, Patti J. 1990 (Jun). Evaluation Of Spoken Language Systems : The ATIS Domain. *Pages 91–95 of : Proceedings of the Speech and Natural Language Workshop of the Human Language Technology Conference*.

PRYSIAZHNIUK, Anastasiia. 2017. *Application Web permettant la recherche d'information dans les décisions de justice - Stage Master1 au LGI2P/IMT Mines Alès*. Tech. rept. Université de Montpellier.

Pudil, Pavel, Novovičová, Jana, & Kittler, Josef. 1994. Floating search methods in feature selection. *Pattern recognition letters*, **15**(11), 1119–1125.

Quinlan, J. Ross. 1993. C4.5 : Programming for machine learning. *Morgan Kauffmann*, **38**, 48.

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

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

Rand, William M. 1971. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, **66**(336), 846–850.

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

Ravi Kumar, V., & Raghuveer, K. 2012. Legal documents clustering using latent dirichlet allocation. *International Journal of Applied Information Systems (IJAIS)*, **2**(6), 34–37.

Řehůřek, Radim, & Sojka, Petr. 2010. Software Framework for Topic Modelling with Large Corpora. *Pages 45–50 of : Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta : ELRA. `http://is.muni.cz/publication/884893/en`.

Rish, Irina. 2001. An Empirical Study Of The Naive Bayes Classifier. *Pages 41–46 of : IJCAI 2001 Workshop On Empirical Methods In Artificial Intelligence*, vol. 3. IBM New York.

Rosenblatt, Frank. 1958. The Perceptron : A Probabilistic Model For Information Storage And Organization In The Brain. *Psychological Review*, **65**(6), 386.

Rousseeuw, Peter J. 1987. Silhouettes : A Graphical Aid To The Interpretation And Validation Of Cluster Analysis. *Journal Of Computational And Applied Mathematics*, **20**, 53–65.

Ruparel, Nidhi H, Shahane, Nitin M, & Bhamare, Devyani P. 2013. Learning from small data set to build classification model : A survey. *Internationla Journal of Computer Applications*, **975**(8887), 23–26.

Sabzi, Akhtar, Farjami, Yaghoub, & ZiHayat, Morteza. 2011. An Improved Fuzzy K-medoids Clustering Algorithm With Optimized Number Of Clusters. *Pages 206–210 of : Proceedings of the 11th International Conference on Hybrid Intelligent Systems (HIS)*. IEEE.

Salton, Gerard, & Buckley, Christopher. 1988. Term-weighting Approaches In Automatic Text Retrieval. *Information Processing & Management*, **24**(5), 513–523.

Salton, Gerard, & McGill, Michael J. 1983. *Introduction To Modern Information Retrieval*. New York, NY, USA : McGraw-Hill, Inc.

Salton, Gerard, Wong, Anita, & Yang, Chung-Shu. 1975. A Vector Space Model For Automatic Indexing. *Communications of the ACM*, **18**(11), 613–620.

Salvador, Stan, & Chan, Philip. 2004. Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. *Pages 576–584 of : 16th IEEE International Conference on Tools with Artificial Intelligence*. IEEE.

Schechtman, Edna, & Yitzhaki, Shlomo. 2003. A family of correlation coefficients based on the extended Gini index. *The Journal of Economic Inequality*, **1**(2), 129–146.

Schmid, Helmut. 1994. TreeTagger - a part-of-speech tagger for many languages. *Page 154 of : Proceedings of International Conference on New Methods in Language Processing*.

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

Sharnagat, Rahul. 2014. *Named entity recognition : A literature survey*. Tech. rept. Center For Indian Language Technology.

Shi, Jianbo, & Malik, Jitendra. 2000. Normalized Cuts and Image Segmentation. *IEEE Transactions On Pattern Analysis And Machine Intelligence*, **22**(8), 888–905.

Shulayeva, Olga, Siddharthan, Advaith, & Wyner, Adam. 2017. Recognizing cited facts and principles in legal judgements. *Artificial Intelligence and Law*, **25**(1), 107–126.

Sidorov, Grigori, Gelbukh, Alexander, Gómez-Adorno, Helena, & Pinto, David. 2014. Soft similarity and soft cosine measure : Similarity of features in vector space model. *Computación y Sistemas*, **18**(3), 491–504.

Singh, Sonia, & Gupta, Priyanka. 2014. Comparative study ID3, CART and C4.5 decision tree algorithm : a survey. *International Journal of Advanced Information Science and Technology (IJAIST)*, **27**, 97–103.

Siniakov, Peter. 2008. *GROPUS an Adaptive Rule-based Algorithm for Information Extraction*. Ph.D. thesis, Freie Universität Berlin.

Sohangir, Sahar, & Wang, Dingding. 2017. Improved sqrt-cosine similarity measurement. *Journal of Big Data*, **4**(1), 25.

Sparck Jones, Karen. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, **28**(1), 11–21.

Strehl, Alexander, Ghosh, Joydeep, & Mooney, Raymond. 2000. Impact of similarity measures on web-page clustering. *Page 64 of : Workshop on artificial intelligence for web search (AAAI 2000)*, vol. 58.

Şulea, Octavia-Maria, Zampieri, Marcos, Malmasi, Shervin, Vela, Mihaela, P. Dinu, Liviu, & van Genabith, Josef. 2017a (June). Exploring the Use of Text Classification in the Legal Domain. *Page 5 of : Proceedings of 2nd Workshop on Automated Semantic Analysis of Information in Legal Texts*. ASAIL'2017, London, United Kingdom.

Şulea, Octavia-Maria, Zampieri, Marcos, Vela, Mihaela, & van Genabith, Josef. 2017b. Predicting the Law Area and Decisions of French Supreme Court Cases. *Pages 716–722 of : Proceedings of the International Conference on Recent Advances in Natural Language Processing, RANLP 2017*.

Tenenhaus, Michel. 1998. *La régression PLS : théorie et pratique*. Editions TECHNIP.

Tenenhaus, Michel. 2005. La regression logistique PLS. *Chap. 12, pages 263–276 of :* Droesbeke, Jean-Jacques and Lejeune, Michel and Saporta, Gilbert (ed), *Modèles statistiques pour données qualitatives*. Editions Technip.

Thakker, Dhaval, Osman, Taha, & Lakin, Phil. 2009. *GATE JAPE Grammar Tutorial*. `https://gate.ac.uk/sale/thakker-jape-tutorial/GATE%20JAPE%20manual.pdf`.

Thenmozhi, D., Kannan, Kawshik, & Aravindan, Chandrabose. 2017. A Text Similarity Approach for Precedence Retrieval from Legal Documents. *Pages 90–91 of : Proceedings of Forum for Information Retrieval Evaluation - FIRE (Working Notes)*.

Thorndike, Robert L. 1953. Who belongs in the family ? *Psychometrika*, **18**(4), 267–276.

Tjong Kim Sang, Erik F., & De Meulder, Fien. 2003. Introduction to the CoNLL-2003 Shared Task : Language-independent Named Entity Recognition. *Pages 142–147 of : Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*. CONLL '03. Stroudsburg, PA, USA : Association for Computational Linguistics.

Tulyakov, Sergey, Jaeger, Stefan, Govindaraju, Venu, & Doermann, David. 2008. Review of classifier combination methods. *Pages 361–386 of : Machine learning in document analysis and recognition*. Springer.

Tumonis, Vitalius. 2012. LEGAL REALISM & JUDICIAL DECISION-MAKING. *Jurisprudencija*, **19**(4).

Ulmer, S. Sidney. 1963. Quantitative analysis of judicial processes : Some practical and theoretical applications. *Law and Contemporary Problems*, **28**(1), 164–184.

Van Asch, Vincent. 2013. *Macro- and micro-averaged evaluation measures*. Tech. rept. Computational Linguistics & Psycholinguistics (CLiPS), Belgium. `https://pdfs.semanticscholar.org/1d10/6a2730801b6210a67f7622e4d192bb309303.pdf`.

Vapnik, Vladimir N. 1995. *The Nature of Statistical Learning Theory*. Springer.

Viera, Anthony J., & Garrett, Joanne M. 2005. Understanding interobserver agreement : the kappa statistic. *Family Medicine*, **37**(5), 360–363.

Vijaymeena, M.K., & Kavitha, K. 2016. A survey on similarity measures in text mining. *Machine Learning and Applications : An International Journal*, **3**(2), 19–28.

Vinh, Nguyen Xuan, Epps, Julien, & Bailey, James. 2010. Information theoretic measures for clusterings comparison : Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, **11**(Oct), 2837–2854.

Vinyals, Oriol, Fortunato, Meire, & Jaitly, Navdeep. 2015. Pointer networks. *Pages 2692–2700 of : Advances in Neural Information Processing Systems*.

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

Von Luxburg, Ulrike. 2007. A tutorial on spectral clustering. *Statistics and computing*, **17**(4), 395–416.

Wallach, Hanna M. 2004. *Conditional Random Fields : An Introduction*. Tech. rept. University of Pennsylvania Department of Computer and Information Science.

Waltl, Bernhard, Matthes, Florian, Waltl, Tobias, & Grass, Thomas. 2016. LEXIA - A Data Science Environment for Semantic Analysis of German Legal Texts. *In : IRIS : Internationales Rechtsinformatik Symposium*. Salzburg, Austria.

Waltl, Bernhard, Landthaler, Jörg, Scepankova, Elena, Matthes, Florian, Geiger, Thomas, Stocker, Christoph, & Schneider, Christian. 2017a. Automated extraction of semantic information from German legal documents. *In : IRIS : Internationales Rechtsinformatik Symposium. Association for Computational Linguistics*.

Waltl, Bernhard, Bonczek, Georg, Scepankova, Elena, Landthaler, Jörg, & Matthes, Florian. 2017b. Predicting the Outcome of Appeal Decisions in Germany's Tax Law. *Pages 89–99 of : International Conference on Electronic Participation*. Springer.

Waltl, Bernhard, Bonczek, Georg, & Matthes, Florian. 2018. Rule-based Information Extraction : Advantages, Limitations, And Perspectives. *Jusletter IT*, Feb.

Wang, Fei, & Sun, Jimeng. 2015. Survey on distance metric learning and dimensionality reduction in data mining. *Data Mining and Knowledge Discovery*, **29**(2), 534–564.

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

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

Wold, Herman. 1966. Estimation of principal components and related models by iterative least squares. *Multivariate Analysis*, 391–420.

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

Wu, Harry, & Salton, Gerard. 1981. A comparison of search term weighting : term relevance vs. inverse document frequency. *Pages 30–39 of : ACM SIGIR Forum*, vol. 16. ACM.

Wyner, Adam, & Peters, Wim. 2010. Lexical Semantics and Expert Legal Knowledge towards the Identification of Legal Case Factors. *Pages 127–136 of : JURIX*, vol. 10.

Wyner, Adam, Mochales-Palau, Raquel, Moens, Marie-Francine, & Milward, David. 2010. Approaches to text mining arguments from legal cases. *Pages 60–79 of : Semantic Processing of Legal Texts : where the Language of Law Meets the Law of Language*. Berlin, Heidelberg : Springer-Verlag.

Wyner, Adam Z. 2010. Towards annotating and extracting textual legal case elements. *Informatica e Diritto : special issue on legal ontologies and artificial intelligent techniques*, **19**(1-2), 9–18.

Xiao, Richard. 2010. Corpus Creation. *Chap. 7, page 146–165 of :* Nitin Indurkhya and Fred J . Damerau (ed), *Handbook of Natural Language Processing*, Second edn. Chapman and Hall.

Xie, Pengtao, & Xing, Eric P. 2013. Integrating document clustering and topic modeling. *In : Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence (UAI2013)*.

Yadav, Vikas, & Bethard, Steven. 2018. A Survey on Recent Advances in Named Entity Recognition from Deep Learning models. *Pages 2145–2158 of : Proceedings of the 27th International Conference on Computational Linguistics*.

Yang, Bishan, & Mitchell, Tom. 2016. Joint Extraction of Events and Entities within a Document Context. *Pages 289–299 of : Proceedings of NAACL-HLT*.

Yang, Yiming, & Pedersen, Jan O. 1997. A comparative study on feature selection in text categorization. *Pages 412–420 of : ICML*, vol. 97.

Zeng, Xue-Qiang, Wang, Ming-Wen, & Nie, Jian-Yun. 2007. Text classification based on partial least square analysis. *Pages 834–838 of : Proceedings of the 2007 ACM symposium on Applied computing*. ACM.

Zhu, Xiaojin. 2010. *Conditional Random Fields*. CS769 Spring 2010 Advanced Natural Language Processing.

Zolotov, Vladimir, & Kung, David. 2017. *Analysis and optimization of fasttext linear text classifier*. arXiv preprint arXiv :1702.05531 [cs.CL].

# Annexes

## A.i Exemple de décision judiciaire annotée

```
<?xml version="1.0" encoding="utf-8"?>
<decision>

<entete>
<juridiction> Cour d' appel </juridiction> , <ville> Lyon </
    ville> , <formation> 6e chambre </formation> , <date> 17 Mars
     2016 </date> - n° <rg> 14/06777 </rg>
<juridiction> Cour d' appel </juridiction>
<ville> Lyon </ville>
<formation> 6e chambre </formation>
<date> 17 Mars 2016 </date>
Répertoire Général : <rg> 14/06777 </rg>
X / Y
Contentieux Judiciaire
R.G : <rg> 14/06777 </rg>
Décision du
Juge de l' exécution de LYON
Au fond
du 29 juillet 2014
RG : 2014/04851
ch n°
<appelant> V. </appelant>
C/
<intime> C. </intime>
<intime> C. </intime>
<intime> C. </intime>
RÉPUBLIQUE FRANÇAISE
AU NOM DU PEUPLE FRANÇAIS
<juridiction> COUR D' APPEL </juridiction> DE <ville> LYON </
    ville>
<formation> 6ème Chambre </formation>
ARRÊT DU <date> 17 Mars 2016 </date>
APPELANTE :
<appelant> Mme Monique V. </appelant>
née le 25 Juillet 1944 à [ ... ]
[ ... ]
[ ... ]
```

Représentée par <avocat> Me Chrystelle P. , avocat au barreau de LYON </avocat>
( bénéficie d' une aide juridictionnelle Partielle numéro 2014/024291 du
11/09/2014 accordée par le bureau d' aide juridictionnelle de LYON )
INTIMES :
<intime> Mme Sylvianne C. </intime> prise en sa qualité d' héritière de Madame Jeannine C.
née le 24 Juillet 1957 à [ ... ]
[ ... ]
[ ... ]
Représentée par <avocat> la SCP ELISABETH L. DE M. & L. L. , avocat au barreau de LYON </avocat>
Assistée par <avocat> Me Isabelle L. , avocat au barreau de LYON </avocat>
<intime> M. Patrick C. </intime> pris en sa qualité d' héritier de Madame Jeannine C.
né le 23 Mai 1953 à [ ... ]
[ ... ]
[ ... ]
Représenté par <avocat> la SCP ELISABETH L. DE M. & L. L. , avocat au barreau de LYON </avocat>
Assisté par <avocat> Me Isabelle L. , avocat au barreau de LYON </avocat>
<intime> M. Thierry C. </intime> pris en sa qualité d' héritier de Madame Jeannine C.
né le 13 Mai 1956 à [ ... ]
[ ... ]
[ ... ]
Représenté par <avocat> la SCP ELISABETH L. DE M. & L. L. , avocat au barreau de LYON </avocat>
Assisté par <avocat> Me Isabelle L. , avocat au barreau de LYON </avocat>
Date de clôture de l' instruction : 28 Avril 2015
Date des plaidoiries tenues en audience publique : 02 Février 2016
Date de mise à disposition : 17 Mars 2016
Composition de la Cour lors des débats et du délibéré :
– <juge> Claude VIEILLARD </juge> , <fonction> président </fonction>
– <juge> Olivier GOURSAUD </juge> , <fonction> conseiller </fonction>
– <juge> Catherine CLERC </juge> , <fonction> conseiller </fonction>
assistés pendant les débats de Charlotte LENOIR , greffier
A l' audience , Olivier GOURSAUD a fait le rapport , conformément à l' article 785
du code de procédure civile .

```
Arrêt Contradictoire rendu publiquement par mise à disposition
    au greffe de la
cour d' appel , les parties en ayant été préalablement avisées
    dans les conditions
prévues à l' article 450 alinéa 2 du code de procédure civile ,
Signé par <juge> Claude VIEILLARD </juge> , <fonction> président
     </fonction> , et par Martine SAUVAGE , greffier , auquel
la minute a été remise par le magistrat signataire .
* * * *
</entete>

<litige>
FAITS , PROCÉDURE , MOYENS ET PRÉTENTIONS DES PARTIES
Suite à un prêt de 10.000 € consenti le 29 janvier 2010 par Mme
    C. à Mme Monique
V. , celle -ci a remis à la première un chèque de 7.400 € devant
     solder sa dette .
Le dit chèque étant revenu impayé , un certificat de non
    paiement a été délivré
par la Société Générale et un titre exécutoire délivré par
    huissier de justice .
Par jugement en date du 4 avril 2013 , le tribunal d' instance
    de Lyon , statuant
en matière de saisie des rémunérations , a autorisé Mme V. à s'
    acquitter de sa
dette liquidée à 7.690 , 01 € par mensualités de 20 € , le
    premier devant
intervenir le 15 mai 2013 , et dit qu' à défaut de paiement
    selon les modalités
prévues , la saisie des rémunérations pourrait être dénoncée à
    son employeur à
l' initiative du créancier .
Par un arrêt en date du 28 novembre 2013 , la cour d' appel de
    Lyon statuant sur
appel d' un précédent jugement du 2 mai 2012 , a confirmé ce
    jugement en ce qu' il
avait rejeté une demande de mainlevée d' un commandement de
    payer délivré le 8
septembre 2011 mais , le réformant sur la demande de délais , a
    autorisé Mme V. à
payer sa dette en 23 mensualités de 150 € et le solde à la 24ème
     , ces délais
étant assortis d' une clause de déchéance du terme .
Par acte d' huissier en date du 5 février 2014 , Mme C. a fait d
    élivrer à Mme V.
un commandement de payer la somme de 7.400 € en principal aux
    fins de saisie
vente .
Par exploit d' huissier en date du 2 avril 2014 , Mme Monique V.
```

a fait assigner
Mme C. devant le juge de l'exécution du tribunal de grande instance de Lyon aux
fins de suspendre les effets de ce commandement et d'être autorisée à continuer
à s'acquitter de sa dette sur la base du jugement ayant statué en matière de
saisie des rémunérations, soit par mensualités de 20 euros par mois.
Mme Sylvianne C., M. Patrick C. et M. Thierry C., héritiers de Mme C., décédée
entre temps, sont intervenus volontairement à l'instance et ont sollicité la
nullité de l'assignation et subsidiairement le rejet des prétentions de Mme V.
et sa condamnation à leur payer des dommages et intérêts.
Par jugement en date du 29 juillet 2014 auquel il est expressément référé pour
un exposé plus complet des faits, des prétentions et des moyens des parties, le
juge de l'exécution du tribunal de grande instance de Lyon a :
– débouté Mme Monique V. de toutes ses demandes,
– condamné Mme Monique V. à payer à Mme Sylvianne C., M. Patrick C. et M.
Thierry C., chacun en qualité d'héritiers de Mme Jeanine C., une somme de 400 €
à titre de dommages et intérêts pour abus de procédure,
– condamné Mme Monique V. à payer à Mme Sylvianne C., M. Patrick C. et M.
Thierry C., chacun en qualité d'héritiers de Mme Jeanine C., une indemnité de
300 € en application de <norme> l'article 700 du code de procédure civile </norme>,
– condamné Mme Monique V. aux entiers dépens de l'instance.
Par déclaration en date du 13 août 2014, Mme Monique V. a interjeté appel de
cette décision.
Dans le dernier état de ses conclusions en date du 10 novembre 2014, Mme V.
demande à la cour de :
– la dire et juger recevable et bien fondée en son appel,
y faisant droit,
– débouter les consorts C. de l'ensemble de leurs demandes,
– réformer le jugement rendu le 29 juillet 2014 par le juge de l'exécution du
tribunal de grande instance de Lyon en toutes ses dispositions,
et statuant à nouveau,
– suspendre les effets du commandement aux fins de saisie vente du 5 février

2014 ,
– dire qu' elle continuera de s' acquitter de sa dette par mensualités de 20 euros
par mois , sur le fondement du jugement rendu par le tribunal d' instance statuant
en matière de saisie sur rémunérations ,
– dire et juger qu' elle n' a commis aucun abus de procédure en saisissant le juge
de l' exécution du tribunal de grande instance de Lyon ,
– condamner solidairement les consorts C. à lui payer une somme de 900 € au
titre de <norme> l' article 700 du code de procédure civile </norme> ,
– condamner solidairement les consorts C. aux entiers dépens de première
instance et d' appel , dont distraction au profit de Me P. , dans les conditions de
<norme> l' article 699 du code de procédure civile </norme> .
Mme V. fait valoir que :
– elle a parfaitement respecté la décision rendue par le tribunal d' instance
statuant en matière de saisie des rémunérations et s' est acquittée de la somme
mensuelle de 20 € ,
– alors qu' elle respectait cet échéancier , Mme C. a procédé de nouveau à une
voie d' exécution forcée par l' intermédiaire d' un autre huissier de justice en
lui faisant délivrer le commandement de payer litigieux et ce sur le fondement
de l' arrêt de la cour d' appel de Lyon du 28 novembre 201 , arrêt postérieur à
celui du tribunal d' instance ,
– les deux décisions accordant des délais différents pour une même dette étaient
manifestement contradictoires ce qui justifiait l' existence d' une difficulté
d' exécution et la saisine du juge de l' exécution .
Dans leurs conclusions en date du 5 janvier 2015 , Mme Sylvianne C. , M. Patrick
C. et M. Thierry C. , chacun en sa qualité d' héritier de Mme Jeanine C. , intimés ,
demandent à la cour de :
– confirmer le jugement en toutes ses dispositions ,
– condamner Mme V. à leur payer la somme de 1.500 € au titre de <norme> l' article 700 du
code de procédure civile </norme> ,
– condamner Mme V. aux entiers dépens de première instance et d' appel lesquels

```
seront distraits au profit de la scp L. de M. & L. , conformé
    ment à <norme> l' article 699
du code de procédure civile </norme> .
Les consorts C. font valoir que :
- le juge de l' exécution a constaté que Mme C. disposait bien d
    ' un titre
exécutoire lui permettant d' agir contre Mme V. laquelle n' a
    pas respecté les
délais de paiement de 150 € mensuels qui lui ont été accordés
    par la cour
d' appel le 28 novembre 2013 , ce qui rendait sa créance
    exigible ,
- la cour d' appel s' est prononcée sur la base d' un précédent
    commandement en
date du 6 septembre 2011 aux fins de saisie vente et il n' y
    avait pas de
contrariété de jugement puisque le jugement du tribunal d'
    instance avait pour
seul effet de statuer en matière d' exécution sur la demande de
    saisie des
rémunérations et ne privait pas le créancier de procéder à d'
    autres voies
d' exécution pour obtenir le paiement de sa créance .
L' ordonnance de clôture est intervenue le 28 avril 2015 et l'
    affaire a été
plaidée à l' audience du 2 février 2016 .
</litige>

<motifs>
MOTIFS DE LA DÉCISION
La cour constate au préalable que le jugement n' est pas remis
    en cause en ce
qu' il a rejeté l' exception de nullité de l' assignation ,
    motif tiré de ce que les
consorts C. ne rapportaient pas la preuve d' un grief résultant
    de l' irrégularité
commise .
Suivant exploit du 5 février 2014 , Mme Jeanine C. , aux droits
    de laquelle
viennent aujourd'hui les consorts C. , a fait délivrer à Mme
    Monique V. un
commandement aux fins de saisie vente .
Le premier juge a relevé à bon droit par application de <norme>
    l' article L 221-1 du
code des procédures civiles d' exécution </norme> que du fait d'
     un certificat de non
paiement et du titre exécutoire délivré par huissier de justice
    , Mme C.
disposait d' un titre exécutoire .
```

Il est constant et non contesté que Mme V. n' a pas respecté les délais octroyés
par la cour d' appel de Lyon dans son arrêt du 28 novembre 2013 qui l' avait
autorisée à payer sa dette en 23 mensualités de 150 € et le solde à la 24ème ,
ces délais étant assortis d' une clause de déchéance du terme .
Mme V. se prévaut des dispositions d' un précédent jugement statuant sur une
demande de saisie de ses rémunérations formée par Mme C. l' ayant autorisée à
s' acquitter de sa dette par mensualités de 20 € et fait valoir qu' elle a
respecté ces délais .
Toutefois , si en application des <norme> article 510 4ème alinéa du code de procédure
civile et L 221−8 du code de l' organisation judiciaire </norme> , le juge du tribunal
d' instance lorsqu' il connaît de la saisie des rémunérations , exerce les pouvoirs
du juge de l' exécution et a ainsi compétence , après signification d' un
commandement ou d' un acte de saisie , pour accorder un délai de grâce , cette
attribution ne fait que lui conférer les pouvoirs du juge de l' exécution dans le
seul domaine de sa compétence , celui de la saisie des rémunérations .
Ainsi , l' autorité de chose jugée attachée à cette décision se limite à suspendre
les effets de la saisie des rémunérations et à les conditionner au respect des
délais accordés ainsi que l' a d'ailleurs relevé le juge d' instance dans sa
décision .
Elle n' interdit pas au créancier , ainsi que l' a justement rappelé le premier
juge , de procéder à d' autres voies d' exécution .
Il n' y a donc pas contrariété entre les deux décisions qui ont accordé des
modalités de délais de paiement différentes .
Le jugement est confirmé en ce qu' il a débouté Mme V. de sa demande tendant à
voir suspendre les effets du commandement aux fins de saisie vente et à être
autorisée à s' acquitter de sa dette par mensualités de 20 € .
Il n' est pas justifié en l' espèce d' un abus de procédure , alors que Mme V. qui a
pu se méprendre sur les effets du premier jugement lui ayant

```
    accordé des délais ,
a , contrairement à ce qu' a retenu le premier juge , respecté
    les termes de ce
jugement en s' acquittant de sa dette par mensualités de 20 €
    ainsi qu' il ressort
des justificatifs qu' elle produit aux débats .
Il convient ainsi de débouter les consorts C. de leur demande en
     dommages et
intérêts , le jugement étant réformé de ce chef .
La Cour estime par contre que l' équité commande à nouveau de
    faire application
de <norme> l' article 700 du code de procédure civile en cause d
    ' appel </norme> au profit des
intimés et il convient de leur allouer à ce titre la somme de
    1.000 € .

</motifs>

<dispositif>

PAR CES MOTIFS
La Cour , statuant publiquement et contradictoirement ,
Confirme le jugement entrepris en toutes ses dispositions sauf
    en ce qu' il a
condamné Mme Monique V. à payer aux consorts C. une somme de 400
     € à chacun à
titre de dommages et intérêts pour abus de procédure .
Statuant à nouveau de ce chef ,
Déboute les consorts C. de leur demande en dommages et intérêts
    .
Condamne Mme Monique V. à payer en cause d' appel aux consorts C
    . la somme de
MILLE euroS ( 1.000 € ) au titre de <norme> l' article 700 du
    code de procédure civile </norme> .
Condamne Mme Monique V. aux dépens de l' instance d' appel , é
    tant précisé qu' elle
est bénéficiaire de l' aide juridictionnelle , et accorde à la
    scp L. de M. & L. ,
avocat , le bénéfice de <norme> l' article 699 du code de procé
    dure civile </norme> .
LE GREFFIER LE PRESIDENT
_ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __
     __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ _
Décision antérieure
LYON Juge de l' exécution 29 Juillet 2014 2014/04851
</dispositif>

</decision>
```