CLEMI: Extraction de connaissances pour les données juridiques

Quentin Gruchet, Zoubida Kedad, Stéphane Lopes, Mohamad Rihany, Anaïs Szkopinski

CLEMI: Extraction de connaissances pour les données juridiques

Q. Gruchet Z. Kedad S. Lopes M. Rihany A. Szkopinski

21 mars 2024

Résumé

The aim of the CLEMI project is to analyze the phenomenon of software counterfeiting by analysing court decisions. These are made available in Open Data thanks to the API Judilibre [api].

Our contribution to this project consisted in collecting this data, integrating it into a database, and then enabling it to be queried and analyzed by our colleagues in sociology and law. In addition, we carried out some preliminary analyses on these data to characterize decisions dealing with software counterfeiting.

This report presents the IT aspects of the work carried out as part of the CLEMI project. It has been performed during a 6-month first-year Master's internship and a 3-month post-doctoral fellowship. First, we describe the data collection process. A preprocessing step was then applied to prepare the data for integration into a database. Several analysis tasks derived from Natural Language Processing (NLP) were applied to the data. Finally, the results were formatted for use by other project members.

1 Introduction

L'objectif du projet CLEMI est d'analyser le phénomène de la contrefaçon de logiciels en étudiant des décisions de justice. Ces dernières sont mises à disposition en Open Data grâce à l'API Judilibre [api].

Notre contribution dans ce projet a consisté à collecter ces données, à les intégrer dans une base de données pour ensuite permettre leur interrogation et leur analyse par nos collègues sociologues et juristes. De plus, nous avons réalisé quelques analyses préliminaires sur ces données afin de caractériser les décisions traitant de contrefaçon de logiciels.

Ce rapport présente les aspects informatiques du travail effectué dans le cadre du projet CLEMI et a été réalisé lors d'un stage de Master première année de 6 mois et d'un post-doctorat de 3 mois. Dans un premier temps, nous détaillons la collecte des données. Une étape de prétraitement a ensuite été appliquée pour préparer les données à leur intégration dans une base de données. Plusieurs tâches d'analyse issues du Traitement Automatique du Langage naturel (*TAL* ou *NLP* en anglais) ont été appliquées à ces données. Enfin, les résultats ont été mis en forme pour être exploités par les autres membres du projet. Ce rapport est structuré selon ces différentes étapes.

Remerciements Ce travail a été soutenu par la Maison des Sciences de l'Homme Paris-Saclay dans le cadre de la subvention *22-MA-02*.

2 Collecte des données

L'API Judilibre

Les données auxquelles nous nous intéressons sont des décisions de justice disponibles en *Open Data*. Elles comportent le texte de la décision accompagné de quelques métadonnées. Elles sont rendues disponibles grâce à l'initiative Judilibre qui vise à mettre à disposition du public les décisions de justice. L'ensemble du jeu de données peut être récupéré par l'intermédiaire de l'API Judilibre [api].

Cette dernière assure la publication des décisions rendues publiquement par la Cour de cassation, enrichies et pseudonymisées [api]. L'API est mise à disposition via le portail PISTE et permet de rechercher en plein texte ou suivant des critères spécifiques parmi l'ensemble de ces décisions. Les données disponibles via l'API sont également celles de la version du site de la Cour de cassation. La documentation technique de l'API est disponible sur le dépôt github du projet.

Récupération des données

Les identifiants des décisions des cours d'appel peuvent être récupérés en interrogeant le site de la Cour de cassation. Trois requêtes sont lancées pour cela : une pour les décisions contenant les mots-clés « *contrefaçon* » et « *logiciel* », une pour « *contrefaçon* » et une pour « *logiciel* ». Les décisions sont ensuite récupérées à partir de leurs identifiants en utilisant API Judilibre.

Les identifiants et les décisions concernant la cour de cassation sont directement récupérés à partir de API Judilibre.

Enfin, le jeu de données est conservé dans une base de données avec le système de gestion de bases de données orienté documents MongoDB.

Caractéristiques des données

Au moment de la rédaction de ce rapport, le jeu de données comporte 11558 décisions, 6510 issues de la cour de cassation et 5048 provenant des cours d'appel. Parmi les décisions de la cour de cassation, 1403 disposent des attributs `timeline` et `contested`. Pour les cours d'appel, 1276 possèdent un attribute `timeline` (sans `contested`).

Les décisions sont décrites par un ensemble de métadonnées comportant jusqu'à 22 attributs[1] (cf. table 1).

Attribut	Nb. occu.	Nb. dist.	Valeurs (valeur : nb. occu.)
`source`	11558	3	'dila' : 4992, 'jurica' : 4358, 'jurinet' : 2208
`chamber`	10868	267	'Chambre sociale' : 1486, 'Chambre criminelle' : 1171, ...
`solution`	11558	12	'Rejet' : 4175, 'Cassation' : 2098, 'Irrecevabilité' : 130, ...
`type`	11348	4	'Arrêt' : 8971, 'Ordonnance' : 107, « Demande d'avis » : 1
`formation`	2205	4	'Formation restreinte hors RNSM/NA' : 1240, ...

TABLE 1 – Statistiques sur quelques attributs du jeu de données.

Les décisions peuvent aussi contenir un attribut multivalué `thème`. Dans le jeu de données, 27119 thèmes sont utilisés dont 8526 distincts (cf. table 2)

1. {'_id' : 11558, 'numbers' : 11558, 'publication' : 11558, 'timeline' : 11558, 'visa' : 11558, 'rapprochements' : 11558, 'source' : 11558, 'text' : 11558, 'decision_date' : 11558, 'jurisdiction' : 11558, 'number' : 11558, 'solution' : 11558, 'forward' : 11558, '___v' : 11558, 'update_date' : 11556, 'type' : 11348, 'chamber' : 10868, 'contested' : 6207, 'themes' : 5557, 'ecli' : 3582, 'formation' : 2205, 'summary' : 1490}

Valeur	Nb. occu.
« Demande d'indemnités liées à la rupture du contrat de travail CDI ou CDD, son exécution ou inexécution »	1201
« contrefaçon »	655
« marque de fabrique »	442
« conditions »	417
« propriete litteraire et artistique »	344
« protection »	332
« portée »	287
« définition »	248
« Demande d'indemnités ou de salaires »	246

TABLE 2 – Quelques valeurs de l'attribut `thème`.

3 Préparation des données

Le prétraitement est une étape importante du processus d'analyse de données. Il s'agit de sélectionner, nettoyer et adapter les données aux futures tâches d'analyse.

3.1 Nettoyage des données

Le nettoyage des données est le processus de détection et de correction des enregistrements corrompus ou inexacts d'un jeu de données. Dans le cas de données textuelles comme les décisions de justice, un ensemble de traitements est nécessaire pour ensuite utiliser les algorithmes de NLP.

Le texte des décisions est traité en supprimant la ponctuation et les *stop words* et en appliquant une lemmatisation au texte résultant. La lemmatisation est une technique ramenant tout mot à sa forme neutre canonique. Enfin, la racinisation (*stemming*) est chargée de regrouper les différentes formes infléchies des mots à leur racine, ayant la même signification. Un exemple de traitement des données est donné dans la table 3.

Opération	Données
Phrase originale	« La demanderesse invoque, à l'appui de son pourvoi, le moyen unique de cassation annexé au présent arrêt. »
Lemmatisation [2]	'demanderesse', 'invoque', 'appui', 'son', 'pourvoi', 'moyen', 'unique', 'cassation', 'annexé', 'présent', 'arrêt'
Élimination des stop words	'demanderesse', 'invoque', 'appui', 'pourvoi', 'moyen', 'unique', 'cassation', 'annexé', 'présent', 'arrêt'
Racinisation	'demanderess', 'invoqu', 'appui', 'pourvoi', 'moyen', 'uniqu', 'cassat', 'annex', 'présent', 'arrêt'

TABLE 3 – Nettoyage des données.

3.2 Préparation des données

Afin d'analyser le texte, à chaque mot est associé un identifiant unique. Pour ce faire, chaque décision est considérée comme un ensemble de *tokens* issus de la phase de nettoyage. L'ensemble de tous les tokens est ensuite utilisé pour créer un dictionnaire qui associe chaque mot à un identifiant unique pour le jeu de données.

Le corpus de décisions est finalement converti en *sac de mots* (*bag-of-words*) contenant l'identifiant du mot et sa fréquence dans chaque document. Par exemple, une décision sera représentée par une liste de couples *(identifiant, fréquence)* du type $[(8, 3), (9, 1), (11, 1), \ldots$

4 Tâches d'analyse

Comme expliqué en introduction, les tâches d'analyse réalisées dans cette section visent à caractériser les décisions portant sur la contrefaçon de logiciels. Pour cela, plusieurs approches issues du domaine du NLP ou de l'apprentissage machine sont mises en œuvre : la modélisation thématique (*Topic Modeling*), qui identifie un ensemble de sujets abordés par des documents, la Reconnaissance d'Entités Nommées (*NER*), qui met en évidence les termes et expressions d'une catégorie donnée (personnes, lieux, ...), le *clustering*, qui regroupe des entités en fonction de leur similarité en utilisant une mesure adaptée, et l'extraction de relations, qui identifie les liens existants entre les éléments d'une phrase.

4.1 Topic Modeling

La modélisation thématique est une technique d'apprentissage automatique non supervisée capable d'analyser un ensemble de documents afin de mettre en évidence les thèmes qu'ils abordent. L'Allocation de Dirichlet Latente (LDA), développée par David Blei, Andrew Ng et Michael Jordan en 2003 est un algorithme pour le topic modeling [BNJ03]. Elle nous indique quels sont les thèmes présents dans une collection de documents en observant les mots qu'elle contient et en produisant une distribution des thèmes.

Une implémentation en Python de l'algorithme LDA est fournie dans la bibliothèque Gensim. Cette dernière permet de représenter des documents textuels sous la forme de vecteurs pour ensuite en extraire la structure sémantique en examinant les co-occurrences statistiques des motifs dans le corpus.

L'algorithme LDA impose de préciser le nombre de sujets cherchés dans le corpus. Pour cette analyse, ce paramètre a été fixé à 20 sujets. Après visualisation et analyse, nous avons découvert que deux sujets étaient liés à l'informatique. La figure 1 représente les 30 premiers mots-clés de ces thèmes.

Même si certains des thèmes mis en évidence sont en rapport avec l'informatique, aucun ne semble directement lié à la contrefaçon de logiciels. Il est possible de ce thème soit peu représenté dans l'ensemble des décisions analysées ce qui ne permet pas de le faire émerger avec cette approche.

4.2 Named-Entity Recognition

La reconnaissance des entités nommées (NER) est la tâche d'identification et de catégorisation des informations clés (entités) dans un texte (personnes, lieux, ...). Une entité peut être un mot ou une série de mots qui se réfère à un même sujet. Chaque entité détectée est classée dans une catégorie prédéterminée. L'utilisation d'un algorithme de NER permet d'extraire des informations sur les entités (noms pertinents tels que lieux, personnes, emplacements, etc.) du langage naturel afin de mieux comprendre le texte. Par exemple, un modèle NER peut détecter la suite de mots « Albert Einstein » dans un texte et le classer dans la catégorie « Personne ».

Dans notre cas, l'utilisation de cette tâche d'analyse pourrait permettre de détecter la présence de termes en rapport avec notre sujet d'étude dans les décisions.

Parmi les nombreuses implémentations d'algorithme NER, nous avons utilisé l'approche proposée par spaCy. Il s'agit d'une bibliothèque libre et gratuite pour le traitement du langage naturel en Python. Elle intègre un grand nombre d'algorithmes pour différentes tâches de NLP et supporte plusieurs langues dont le français.

Le modèle NER pré-entraîné de spaCy est capable d'identifier un ensemble prédéfini de catégories (cf. table 4).

Il est également possible d'entraîner le modèle NER de spaCy afin d'identifier les éléments pertinents en fonction des besoins de l'utilisateur. Dans notre contexte, nous avons ajouté une

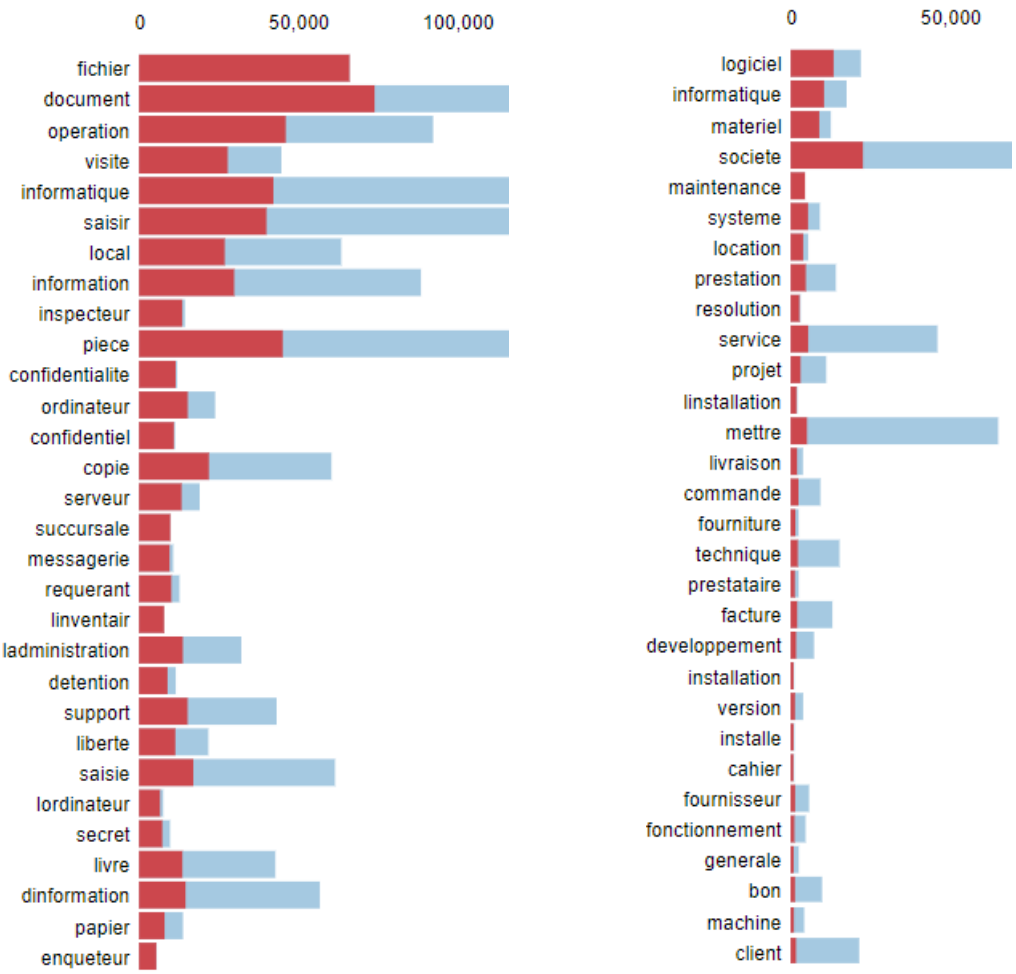

FIGURE 1 – Les 30 premiers mots-clés des thèmes 8 et 12.

Catégorie	**Description**
PERSON	People, including fictional.
NORP	Nationalities or religious or political groups
FAC	Buildings, airports, highways, bridges, etc.
ORG	Companies, agencies, institutions, etc.
GPE	Countries, cities, states.
LOC	Non-GPE locations, mountain ranges, bodies of water.
PRODUCT	Objects, vehicles, foods, etc. (Not services.)
EVENT	Named hurricanes, battles, wars, sports events, etc.
WORK_OF_ART	Titles of books, songs, etc.
LAW	Named documents made into laws.
LANGUAGE	Any named language.
DATE	Absolute or relative dates or periods.
TIME	Times smaller than a day.
PERCENT	Percentage, including « % ».
MONEY	Monetary values, including unit.
QUANTITY	Measurements, as of weight or distance.
ORDINAL	first, second, etc.
CARDINAL	Numerals that do not fall under another type
MISC	Miscellaneous entities

TABLE 4 – Catégories NER prédéfinies pour spaCy.

catégorie représentant les termes en rapport avec la contrefaçon de logiciels. Des mots-clés ont été fournis par deux sources différentes : la première provenant des experts du domaine, et la seconde des 30 premiers mots-clés extraits des deux thèmes pertinents issus du topic modeling présentés dans la section précédente.

Après avoir sélectionné les deux sujets pertinents, nous obtenons 763 documents appartenant à ces thèmes. Nous les avons séparés en un ensemble d'entraînement de 300 documents et un ensemble de test de 463 documents. Les documents de l'ensemble d'entraînement sont décomposés en phrases dont les éléments pertinents sont annotés avec la nouvelle catégorie « Contrefaçon de logiciels » (CFL).

Prenons comme exemple la phrase suivante :

« Si le bénéficiaire d'une licence de marque peut intervenir à une instance en contrefaçon afin de réclamer la réparation du préjudice qui lui est propre, l'action en contrefaçon ne peut être exercée que par le titulaire de la marque ; ».

Les entités sont identifiées selon leurs index dans la phrase (index du début et de fin du mot-clé). Ici, le mot-clé « licence » appartient à la catégorie (CFL). L'indice du premier caractère de ce mot-clé est 27, et l'indice du dernier caractère est 34. Le résultat final est une liste des positions de ces entités : [[27, 34, « CFL »], [79, 90, « CFL »], [167, 178, « CFL »]] que l'on peut visualiser sur la phrase :

« Si le bénéficiaire d'une licence de marque peut intervenir à une instance en contrefaçon afin de réclamer la réparation du préjudice qui lui est propre, l'action en contrefaçon ne peut être exercée que par le titulaire de la marque ; ».

L'efficacité de l'approche est limitée par la faible taille du jeu de données d'entraînement puisque ce dernier nécessite une annotation manuelle des décisions. Une possibilité serait d'annoter automatiquement des décisions en utilisant des expressions rationnelles (*regular expressions*) pour ensuite entraîner le modèle NER.

4.3 Clustering

Appliquer un algorithme de clustering sur les données permet de les partitionner selon un certain critère. Les groupes ainsi formés, dénommés *clusters*, sont ensuite analysés pour déterminer si l'un ou plusieurs d'entre eux sont en rapport avec la contrefaçon de logiciels.

De nombreux algorithmes de clustering sont décrits dans la littérature [Ezu+22 ; XW05]. Parmi ceux-ci, l'algorithme *k-means* est l'un des plus populaires [ASI20]. La version naïve de l'algorithme débute en choisissant aléatoirement k *centroïdes* représentant les barycentres des clusters. Les éléments sont ensuite affectés au cluster le plus « proche » et les centroïdes sont mis à jour. L'algorithme itère jusqu'à ce que les affectations se stabilisent.

Plusieurs points délicats sont à prendre en compte lors de l'application de k-means. Tout d'abord, le nombre de clusters est requis pour exécuter l'algorithme ce qui a un fort impact sur la forme des clusters. Ensuite, l'initialisation aléatoire des centroïdes peut conduire à une convergence vers un minimum local parfois éloigné de l'optimal. Enfin, le calcul de distance nécessaire pour comparer les éléments entre eux peut rendre difficile son application à des types de données non numériques.

Dans notre étude, nous avons utilisé l'implémentation de k-means fournie dans la bibliothèque Python scikit-learn. La mesure de similarité utilisée est la distance entre deux points dans l'espace des documents. Cela nécessite donc de convertir les décisions de justice en vecteurs numériques. Deux approches ont été évaluées : (i) un document est représenté par les mesures *Term Frequency-Inverse Document Frequency* (*TF-IDF*) de chacun des termes qu'il contient, (ii) un document est représenté par les mots les plus fréquents qu'il contient et deux documents sont comparés avec la similarité de Jaccard.

4.3.1 k-means et TF-IDF

La mesure TF-IDF est calculé à partir de la fréquence des termes dans un document (*Term Frequency*) corrigée par l'inverse de la fréquence des termes dans l'ensemble du corpus (*Inverse Document Frequency*). L'intuition derrière la mesure TF est qu'un document comportant un grand nombre d'occurrences d'un terme a de forte chance de traiter de ce terme. Par contre, si ce dernier est également présent dans de nombreux documents, il ne permettra pas de caractériser un document particulier (mesure IDF).

$$tf\_idf(t,d) = \frac{freq_{t,d}}{\sum_k freq_{k,d}} \log(\frac{|D|}{|\{d' \mid t \in d'\}|})$$

L'algorithme k-means appliqué aux décisions transformées par TF-IDF avec un paramètre $k = 22$ a formé les clusters de la table 5.

Id	#déc.	Mots
0	485	attendu, chambr, part, societ, cassat, infract, procédur, arrêt, prévenu, pénal
1	1061	attendu, demand, part, arrêt, civil, contrefaçon, droit, auteur, œuvr, societ
2	426	demand, licenci, temp, titr, pai, employeur, supplémentair, salari, travail, heur
3	560	licenci, employeur, titr, salair, prim, euros, contrat, societ, salari, travail
4	438	civil, part, euros, pai, matériel, demand, résili, locat, contrat, societ
5	1124	titr, concurrent, contrat, procédur, part, cet, civil, euros, demand, societ
6	504	commercialis, original, arrêt, auteur, concurrent, deloyal, dessin, contrefaçon, societ, model
7	156	libert, autoris, fichi, piec, ordon, administr, societ, sais, visit, fiscal
8	360	arret, du, un, est, qu, une, en, et, le, que
9	439	concurrent, cet, droit, consider, franc, demand, produit, contrefaçon, societ, marqu
10	534	médecin, demand, moral, post, licenci, mme, employeur, harcel, salari, travail
11	326	contrat, madam, euros, titr, demand, salari, licenci, travail, societ, monsieur
12	397	part, demand, attendu, moyen, arrêt, contrefaçon, invent, revend, societ, brevet
13	576	civil, conseil, chambr, avocat, moyen, pourvoi, attendu, cassat, arrêt, societ
14	460	contrefaçon, sign, dénomin, servic, usag, confus, désign, produit, societ, marqu
15	781	entrepris, titr, réel, sérieux, mme, societ, employeur, travail, salari, licenci
16	386	rapporteur, attendu, conseil, mme, mémoir, cassat, arrêt, criminel, pénal, chambr
17	177	monsieur, societ, demand, titr, contrat, employeur, salari, licenci, travail, madam
18	111	employeur, attendu, effect, forf, heur, rémuner, salari, salari, temp, travail
19	367	dénomin, commercial, moyen, pourvoi, contrefaçon, cassat, attendu, arrêt, societ, marqu
20	201	bancair, prêt, euros, mme, pai, compt, societ, cred, chequ, banqu
21	629	euros, employeur, activ, social, sécur, mme, caiss, salari, travail, societ

TABLE 5 – Clusters des décisions avec leur taille et les termes principaux.

4.3.2 k-means et Jaccard

La similarité de Jaccard, également appelée indice de Jaccard ou coefficient de Jaccard, est calculée comme le rapport entre la taille de l'intersection entre deux échantillons de données et la taille de l'union de ces mêmes échantillons. Elle est représentée par la formule : $J(A,B) = \frac{|A \cap B|}{|A \cup B|}$. Elle est utilisée pour trouver la similarité ou le chevauchement entre deux vecteurs binaires, numériques ou chaînes de caractères.

Dans notre jeu de données, la *méthode du coude* permet de fixer le paramètre k à 3. L'algorithme k-means a ensuite permis d'extraire les clusters (cf. figure 2).

Les 3 cartes de distance interthème (*topic modeling*) pour les clusters sont données dans les figures 3, 4 et 5.

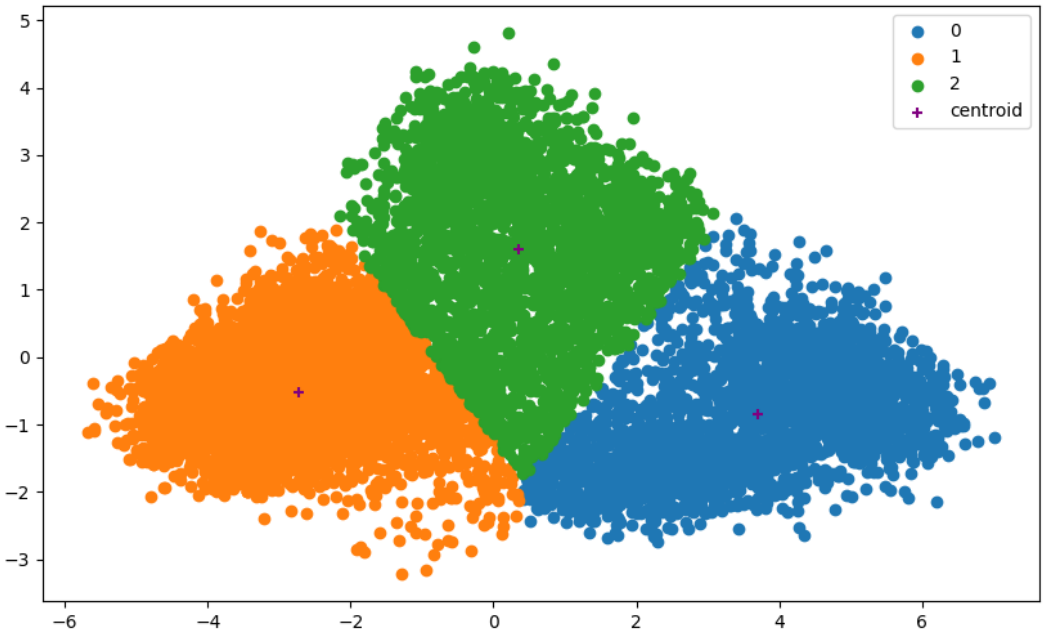

FIGURE 2 – Répartition en clusters pour les 100 mots les plus fréquents.

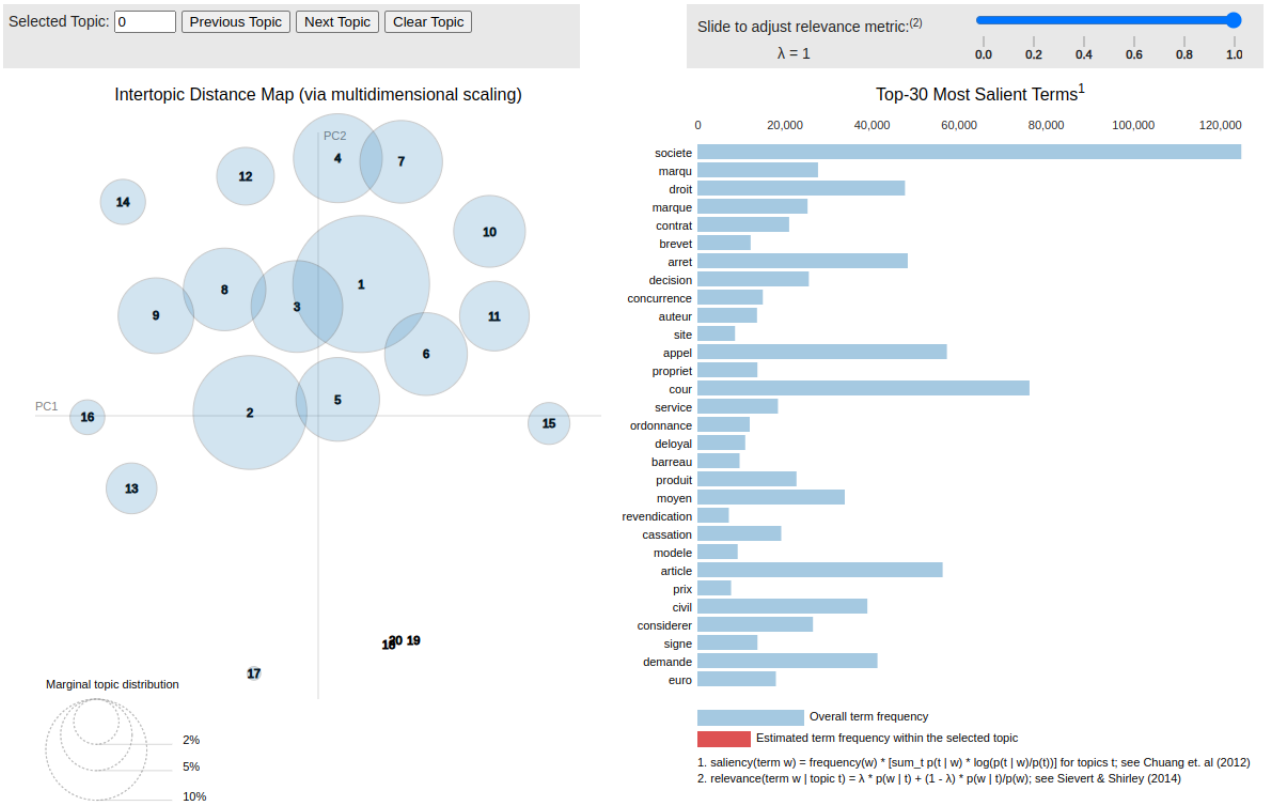

FIGURE 3 – Distances interthèmes et top-30 des termes les plus marquants (cluster 1).

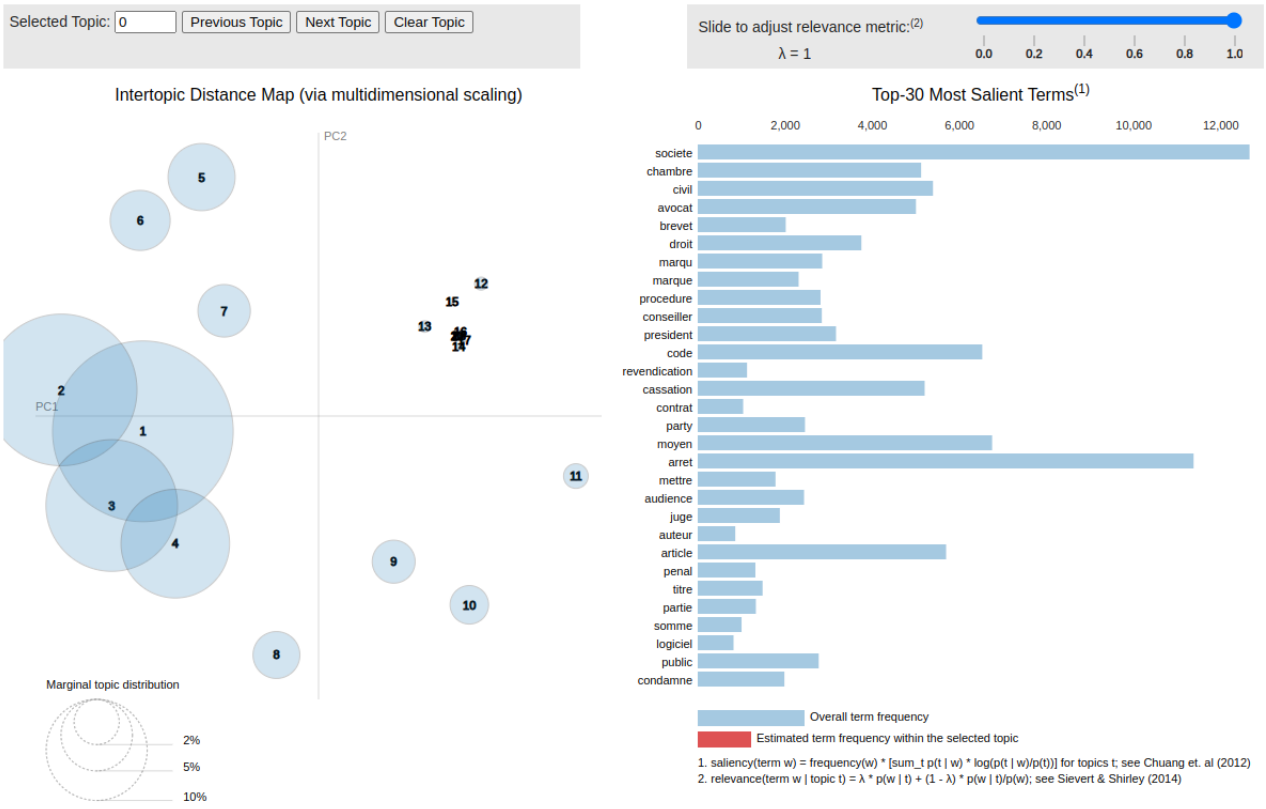

FIGURE 4 – Distances interthèmes et top-30 des termes les plus marquants (cluster 2).

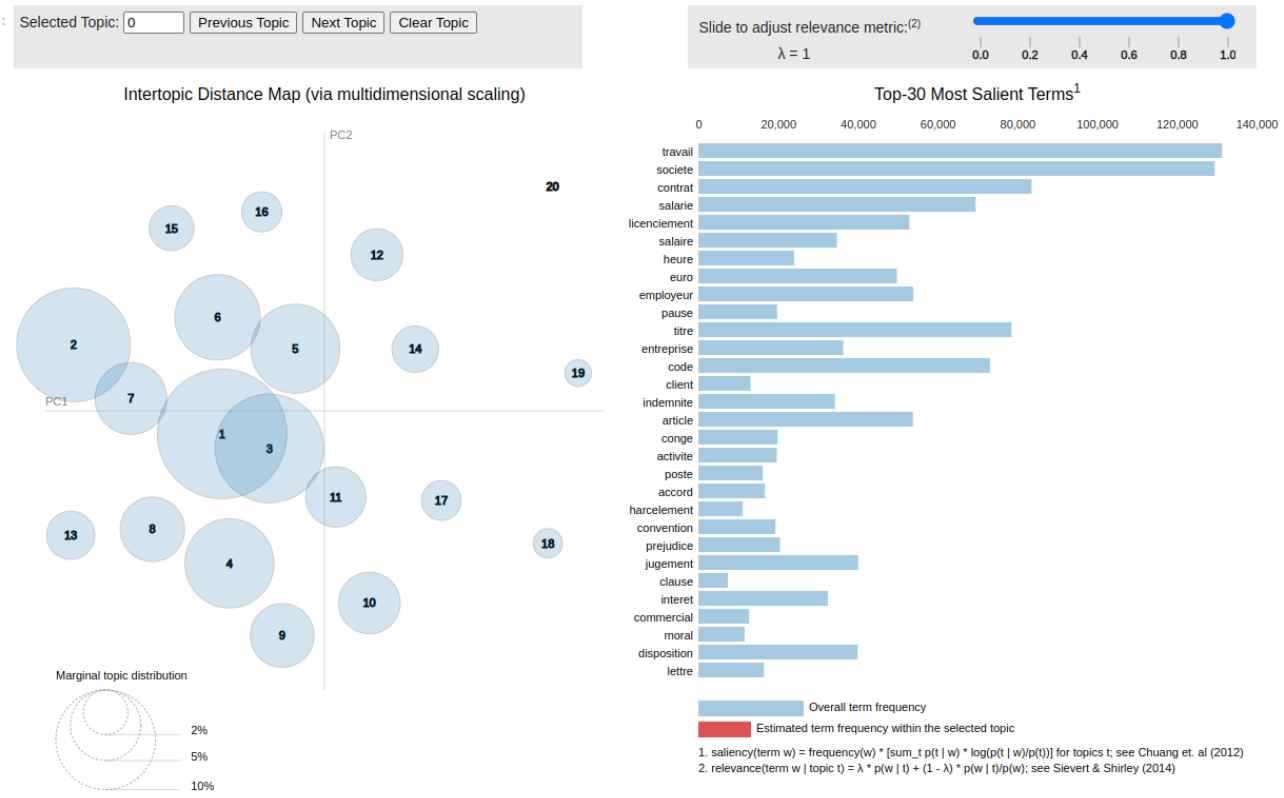

FIGURE 5 – Distances interthèmes et top-30 des termes les plus marquants (cluster 3).

4.4 Relation extraction

L'extraction de relations (*Relation extraction* ou *RE*) est la tâche qui consiste à extraire du texte les relations sémantiques qui existent entre deux ou plusieurs entités. Divers types d'approches sont présentés dans la littérature pour l'extraction de relations [DBB23 ; NJM21 ; PPB17 ; BB07]. Ces relations peuvent être de différents types. Par exemple, « Paris est en France » indique une relation « est en » entre Paris et la France.

Cette relation peut être décrite à l'aide de triplets comme *(Paris, est en, France)*. L'extraction de relations est la composante clé de la construction de graphes de connaissances relationnelles, et elle est d'une importance cruciale pour les applications de traitement du langage naturel telles que la recherche structurée, l'analyse des sentiments, les systèmes de questions/réponse et le résumé. Dans notre cas, l'idée est de permettre de représenter les décisions sous la forme d'un graphe de connaissances afin d'identifier une certaine structuration dans ces textes.

Dans ce projet, nous avons expérimenté l'algorithme non supervisée *ReVerb* qui identifie et extrait automatiquement les relations entre les entités d'une phrase [FSE11]. Il est conçu pour l'extraction d'informations à l'échelle du Web, où les relations cibles ne peuvent pas être spécifiées à l'avance. ReVerb prend du texte brut en entrée et produit des triples (argument1, phrase de relation, argument2). Par exemple, étant donné la phrase « Les bananes sont une excellente source de potassium », le triple (bananes, être source de, potassium) est extrait. La particularité de cet algorithme est d'utiliser une contrainte syntaxique et une contrainte lexicale afin d'améliorer la précision de l'extraction. La première s'appuie sur les balises POS (*Part-Of-Speech tag*) et des patterns pour identifier les relations formées d'un verbe (e.g. *inventait*), d'un verbe suivi d'une préposition (e.g. *habitait dans*) ou d'un verbe suivi de noms, adjectifs ou adverbes terminés par une préposition (e.g. *a un numéro atomique de*). La seconde vérifie que les relations possèdent un nombre d'occurences suffisant avec des arguments variés dans l'ensemble du corpus.

5 Visualisation des résultats

La plupart des résultats d'analyse ont été rendu disponibles par l'intermédiaire d'une application accessible en ligne. Par exemple, la liste des clusters est consultable et les décisions faisant partie de chaque cluster sont visualisables (cf. figure 6).

Pour chaque décision, un nuage de mots a été calculé et l'entête construit (cf. figure 7).

De plus, le texte des décisions est annoté avec les entités extraites lors de l'analyse (cf. figure 8).

Enfin, un ensemble de statistiques a été calculé sur le jeu de données (cf. figure 9).

6 Conclusion

Le projet CLEMI a pour but de caractériser les décisions de justice en rapport avec la contrefaçon de logiciels. En ce qui concerne les aspects informatiques, cela nous a permis de débuter l'étude de nouvelles données et de nouvelles problèmatiques liées ainsi que d'obtenir quelques résultats préliminaires. La première contribution a porté sur les moyens d'accès aux données et à leur collecte. Elle s'est poursuivie par leur étude qui a permis de définir les prétraitements adéquats pour poursuivre l'analyse. Un ensemble de tâches d'analyse exploratoires a ensuite été réalisé afin d'expérimenter plusieurs approches pour la caractérisation des décisions en rapport avec la contrefaçon de logiciels. Finalement, ces résultats ont été regroupés au sein d'une application en ligne permettant de les visualiser simplement.

Tableau regroupant les Clusters identifiés avec les mots qui le composent.

Show 10 entries Search:

Cluster	Mots	Nombre de décisions dans le cluster
0	attendu, chambr, part, societ, cassat, infract, procédur, arrêt, prévenu, pénal	485
1	attendu, demand, part, arrêt, civil, contrefaçon, droit, auteur, oeuvr, societ	1061
2	demand, licenci, temp, titr, pai, employeur, supplémentair, salari, travail, heur	426
3	licenci, employeur, titr, salair, prim, euros, contrat, societ, salari, travail	560
4	civil, part, euros, pai, matériel, demand, résili, locat, contrat, societ	438
5	titr, concurrent, contrat, procédur, part, cet, civil, euros, demand, societ	1124
6	commercialis, original, arrêt, auteur, concurrent, déloyal, dessin, contrefaçon, societ, model	504
7	libert, autoris, fichi, piec, ordon, administr, societ, sais, visit, fiscal	156
8	arret, du, un, est, qu, une, en, et, le, que	360
9	concurrent, cet, droit, consider, franc, demand, produit, contrefaçon, societ, marqu	439

Showing 1 to 10 of 22 entries Previous 1 2 3 Next

FIGURE 6 – Liste des clusters.

Les divers résultats ont été présenté à notre collègue juriste lors de points réguliers. Les regroupements effectués par les algorithmes de clustering ont été tout particulièrement discutés afin de déterminer s'ils étaient conformes à ce qui était attendu et s'il était nécessaire d'adapter les phases de prétraitements ou d'analyse. À ce stade, les résultats préliminaires disponibles actuellement sont une première étape vers la caractérisation des décisions mais laissent ouverts bon nombre de questions.

Concernant la collecte et la préparation des données, plusieurs pistes sont à explorer. Les métadonnées disponibles avec chaque décision peuvent être considérées et intégrées à certaines analyses. De plus, les décisions, même si ce sont des données textuelles, suivent des canevas assez précis. Il serait intéressant d'identifier les patterns de décisions et de les traiter de façon adéquat pour améliorer la précision des traitements ultérieurs. Enfin, la question de la mise à jour des données et de leur intégration dans les différents processus n'a pas été abordée dans ce projet.

Les approches d'analyse proposées dans ce projet sont très préliminaires et ont permis d'explorer « en largeur » et partiellement ce qu'il est possible de faire. D'une part, les approches de *topic modeling* et de *clustering* sont très utiles pour avoir une vision d'ensemble des données. Pour améliorer leur intérêt, il reste à affiner les processus qui les mettent en œuvre. D'autre part, la transformation des décisions en graphe de connaissance permettrait de visualiser une sorte de résumé de chaque décision tout en offrant la possibilité de les manipuler avec d'autres outils informatiques.

Finalement, la présentation des données et des résultats devra être affinée pour parfaire leur utilité pour les autres membres du projet.

Références

[api] API.GOUV.FR, éd. *API Judilibre*. URL : https://api.gouv.fr/les-api/api-judilibre.

[ASI20] Mohiuddin AHMED, Raihan SERAJ et Syed Mohammed Shamsul ISLAM. « The k-means Algorithm : A Comprehensive Survey and Performance Evaluation ». In :

Decision n°5fca28d1c8c4b77eec00dae5

FIGURE 7 – Nuage de mots et entête d'une décision.

En conclusion, il est demandé de :

- confirmer ne toutes ses dispositions l'ordonnance du JLD du TGI de PARIS du 11 juin 2019 ;

- rejeter toutes autres demandes, fins et conclusions ;

- condamner les appelantes au paiement de la somme de 2 000 € sur le fondement de l' article 700 du code de procédure civile .

SUR CE

I ' Sur la régularité de la procédure

A ' sur la régularité de la saisine du JLD

L'administration fiscale précise que lors d'une demande d'autorisation, les agents déposent les pièces constituant les éléments soumis à l'appréciation du JLD et présentent au juge les habilitations des personnes appelées à intervenir.

En l'espèce, l'ordonnance rendue par le JLD du TGI de Paris vise la requête de l'administration fiscale qui a été présentée le 4 juin 2019 par [D] [XO]. Le JLD précise dans sa décision que les copies des habilitations nominatives de l'agent qui a présenté la requête et de ceux désignés pour l'exécution des opérations « tous spécialement habilités par le Directeur général des Impôts ou le Directeur général des Finances Publiques en application des dispositions de l' article L. 16 B du LPF » lui ont été présentées et il précise en page 1, que la copie de l'habilitation nominative de l'agent qui a présenté la requête, à savoir M. [D] [XO] « spécialement habilité par le Directeur général des Finances Publiques en application des articles L. 16 B et R. 16 B-1 du livre des procédures fiscales » lui a été présentée.

Ainsi, en énonçant dans le dispositif de l'ordonnance que les copies des habilitations nominatives des agents autorisés lui ont été présentées, le juge satisfait aux exigences légales.

FIGURE 8 – Texte annoté d'une décision.

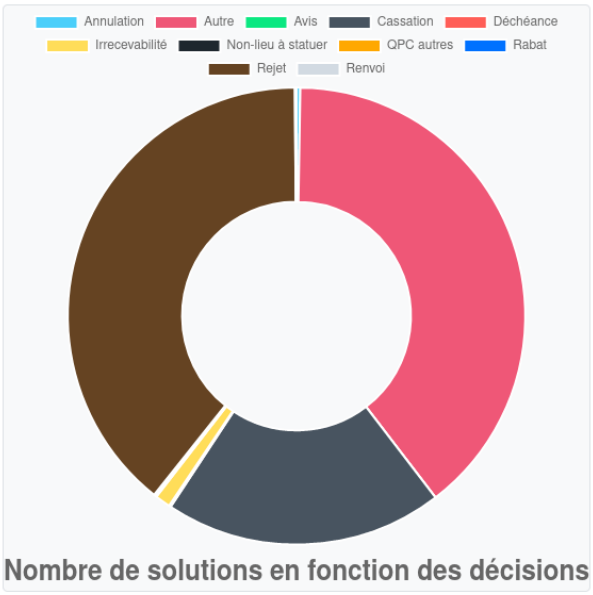

FIGURE 9 – Exemple de statistiques sur les données.

Electronics 9.8 (2020). ISSN : 2079-9292. DOI : 10.3390/electronics9081295. URL : https://www.mdpi.com/2079-9292/9/8/1295.

[BB07] Nguyen BACH et Sameer BADASKAR. « A Review of Relation Extraction ». 2007. URL : https://www.cs.cmu.edu/~nbach/papers/A-survey-on-Relation-Extraction.pdf.

[BNJ03] David M. BLEI, Andrew Y. NG et Michael I. JORDAN. « Latent Dirichlet Allocation ». In : *Journal of Machine Learning Research* 3 (2003), p. 993-1022. URL : http://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf.

[DBB23] Kartik DETROJA, C.K. BHENSDADIA et Brijesh S. BHATT. « A survey on Relation Extraction ». In : *Intelligent Systems with Applications* 19 (2023), p. 200244. ISSN : 2667-3053. DOI : https://doi.org/10.1016/j.iswa.2023.200244. URL : https://www.sciencedirect.com/science/article/pii/S2667305323000698.

[Ezu+22] Absalom E. EZUGWU et al. « A comprehensive survey of clustering algorithms : State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects ». In : *Engineering Applications of Artificial Intelligence* 110 (2022), p. 104743. ISSN : 0952-1976. DOI : https://doi.org/10.1016/j.engappai.2022.104743. URL : https://www.sciencedirect.com/science/article/pii/S095219762200046X.

[FSE11] Anthony FADER, Stephen SODERLAND et Oren ETZIONI. « Identifying relations for open information extraction ». In : *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. EMNLP '11. Edinburgh, United Kingdom : Association for Computational Linguistics, 2011, p. 1535-1545. ISBN : 9781937284114.

[NJM21] Zara NASAR, Syed Waqar JAFFRY et Muhammad Kamran MALIK. « Named Entity Recognition and Relation Extraction : State-of-the-Art ». In : *ACM Comput. Surv.* 54.1 (fév. 2021). ISSN : 0360-0300. DOI : 10.1145/3445965. URL : https://www.researchgate.net/publication/345315661_Named_Entity_Recognition_and_Relation_Extraction_State_of_the_Art.

[PPB17] Sachin PAWAR, Girish PALSHIKAR et Pushpak BHATTACHARYYA. « Relation Extraction : A Survey ». 2017. URL : https://www.researchgate.net/publication/321823883_Relation_Extraction_A_Survey.

[XW05] Rui XU et Donald WUNSCH. « Survey of Clustering Algorithms ». In : *Neural Networks, IEEE Transactions on* 16 (juin 2005), p. 645-678. DOI : 10.1109/TNN.2005.845141.