Extracting claims information from judgments using text classification and term weighting

No Author Given

No Institute Given

Abstract. The analysis of judgments is very important for legal practitioners to understand how judges rule. In our way to find a solution to assist them, we are dealing with the task of extracting the claimed quantum, the result polarity, and the granted quantum for each individual claim of a considered type in French court decisions. we propose also a baseline approach that uses term weighting and text classification to learn useful features related to a category in order to locate the information in the right documents.

Keywords: information extraction, document classification, term weighting, claims, court decisions

1 Introduction

A court decision is a document summarizing facts, parties claims, results or solutions and reasonings of a legal case. Judicial decisions are essential for lawyers because they are used to gather and analyze decisions to solve problems at hand or advise their clients. The analysis of judgments can indeed provide an invaluable insight about the application of the law that can serve numerous applications and studies, e.g. (i) for providing a competitive advantage for handling future cases - recall that justice is complex and its language is barely understandable to allow non-lawyer individuals to estimate the legal risk of their actions without the help of a legal expert [1], (ii) for detecting variations on legal decisions w.r.t. to specific variables such as the location of the court – there is indeed a need for automatic descriptive analyses of decisions for analyzing the application of the law. Then, how to leverage the existing corpus of decisions to analyze and even predict judges' decision making knowing that the subjective interpretation of legal rules makes the application of the law somehow non-deterministic? This question is of big interest to several companies such as LexisNexis with its Lex-Machina[1] system, and some new French startups such as Predictice[2] and Case Law Analytics[3] are working on it. To answer that question, a comprehensive analysis of cases is necessary. But it should implies a large collection of documents (more than 2.5 millions of decisions are pronounced in France every year),

[1] https://lexmachina.com
[2] http://predictice.com
[3] Http://caselawanalytics.com

and it would requires a lot of time and money if done manually. With the aim to assist legal experts with an automatic analysis tool, we are addressing the extraction of the amount of money claimed (the claimed quantum), the corresponding answer of judges i.e. whether the claim was accepted or rejected (the polarity or meaning of the result) and the amount of money the judges ordered the defendant to pay (result or granted quantum).

Our aim is to enrich a knowledge base of judgments structured as shown on Table 1. In this table, Each column describes an information on the claim and each line correspond to an individual claim. Let's look at the claim of the line 442. The first three columns identify the court decision with resp. the type of court (e.g. CA = appeal court), the city (e.g. Lyon), the registry identifier number in the court (e.g 14/06911). The next three columns describe the claim made by a party. We organize the claims into categories predefined by an expert annotator with two informations : the object or what is claimed (e.g. damages) and the norm or legal basis (e.g Article 700 of the Code of Civil Procedure). The last two columns describe how the judges answer the claim i.e. the result polarity (e.g. accept) and the quantum granted (e.g. 1500 euros). Claims are understandable according to their category i.e. object + norm (e.g. damages + Article 700). Since there are lots of categories (at least 500), we define the extraction task by category first to ease the annotation process of the evaluation data and secondly to enable some analyses on a category of claim such as comparing the frequency of accepted and rejected claims.

Table 1. Structure of the labeled data with two claims from the same document.

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

This task is difficult because court decisions are unstructured texts with multiple claims of different categories and written with some implicit and aggregated statements, and also references to previous judgments.

2 Evaluation protocol and metrics

For a category c_i of the set C of existing categories, the data are labeled in a table like described previously. The annotator should also supply the corpus D_{c_i} of the raw documents from which the claims were extracted, with also some documents that do not contain the category $D_{\overline{c_i}}$ in order to ensure the system does not extract claims from such decisions. We should first define a standard way to know when an information have been correctly extracted. The quanta

are amounts of money thus they are comparable if converted into numbers. As for the result polarity, its value is a category among *accepte, rejette* and *sursis à statuer* (i.e. "accept", "reject", "stay of proceedings") and thus it is very easy to evaluate its extraction. Let's say we want to evaluate a system on I a set of types of information of claims over a testing corpus $D = D_{c_i} \cup D_{\overline{c_i}}$. I might be a subset of $\{Q_{DMD}, S_{RST}, Q_{RST}\}$ where $Q_{DMD}, S_{RST}, Q_{RST}$ denotes resp. the requested quantum, the meaning of the result, and the granted quantum. During the evaluation, we are going to match successively the tuples extracted from D_j with those in the gold standard annotation of D_j. At the level of the document: TP_{c_i,I,D_j} is the number of claims (tuples of type I) correctly extracted from D_j, FP_{c_i,I,D_j} is the number of claims extracted from D_j but wrongly classify in c_i, and finally, FN_{c_i,I,D_j} is the number of claims in D_j that were missed. At the level of the corpus then, $TP_{c_i,I,D} = \sum_{j=1}^{|D|} TP_{c_i,I,D_j}$, $FP_{c_i,I,D} = \sum_{j=1}^{|D|} FP_{c_i,I,D_j}$, and $FN_{c_i,I,D_j} = \sum_{j=1}^{|D|} FN_{c_i,I,D_j}$. Finally, at the level of the corpus, we can define the precision, the recall, and the f1-score :

$$Precision_{c_i,I,D} = \frac{TP_{c_i,I,D}}{TP_{c_i,I,D}+FP_{c_i,I,D}} \; ; \; Recall_{c_i,I,D} = \frac{TP_{c_i,I,D}}{TP_{c_i,I,D}+FN_{c_i,I,D}}$$

$$F1_{c_i,I,D} = 2 \times \frac{Precision_{c_i,I,D} \times Recall_{c_i,I,D}}{Precision_{c_i,I,D} + Recall_{c_i,I,D}}$$

3 Term weighting and document categorization

3.1 Global term weighting schemes

Global term weighting metrics are commonly used as feature selection methods for text categorization and information retrieval. The unsupervised methods (e.g. *idf*) compute a score of a term all over the corpus independently of any category while the supervised metrics (e.g. χ^2, ngl, gss) compute a correlation score between a term t_k and a category c_i over a labeled corpus. Supervised weights are aggregated over the classes using aggregation functions such as *max*, or *min* to have a single weight. We study the metrics described in table 2 with the following notations :

- N = number of documents (docs.) in a training dataset $D = D_{c_i} \cup D_{\overline{c_i}}$
- $N_{c_i} = |D_{c_i}|$, $N_{\overline{c_i}} = |D_{\overline{c_i}}|$, $DF_{t_k,c_i} = N_{t_k,c_i}/N_{c_i}$, $DF_{t_k,\overline{c_i}} = N_{t_k,\overline{c_i}}/N_{\overline{c_i}}$
- $N_{t_k}(resp. N_{\overline{t_k}})$ = number of docs. in D with (resp. without) t_k
- N_{t_k,c_i} (resp. $N_{\overline{t_k},c_i}$) = number of docs. of D_{c_i} with (resp. without) t_k
- $N_{t_k,\overline{c_i}}$ (resp. $N_{\overline{t_k},\overline{c_i}}$) = number of docs. out of D_{c_i} with (resp. without) t_k

There is a lot of comparing studies on how these metrics behave for the problem of document categorization [2, 3]. We are also interested in global metrics to determine the terms that indicate the mention of information.

Table 2. Global term weighting methods studied in this work: supervised methods are noted $f(t_k, c_i)$ and unsupervised methods are noted $g(t_k)$.

Description	Metric formula
Inverse document frequency [4]: simply computes the importance score of t_k all over a corpus D	$idf(t_k) = \log_2(\frac{N}{N_{t_k}})$
Delta Document Frequency	$deltadf(w, c_i) = DF_{t_k,c_i} - DF_{t_k,\overline{c_i}}$
Test of Marascuilo	$mar(t_k, c_i) =$ $\frac{\left(\begin{array}{l}(N_{t_k,c_i} - N_{t_k}N_{t_k,c_i}/N)^2 \\ \quad + (N_{t_k,\overline{c_i}} - N_{t_k}N_{\overline{c_i}}/N)^2 \\ \quad\quad + (N_{\overline{t_k},c_i} - N_{c_i}N_{\overline{t_k}}/N)^2 \\ \quad\quad\quad + (N_{\overline{t_k},} - N_{\overline{t_k}}N_{\overline{c_i}}/N)^2\end{array}\right)}{N}$
Chi square	$chi2(t_k, c_i) = \frac{N((N_{t_k,c_i}N_{\overline{t_k},\overline{c_i}}) - (N_{t_k,\overline{c_i}}N_{\overline{t_k},c_i}))^2}{N_{t_k}N_{\overline{t_k}}N_{c_i}N_{\overline{c_i}}}$
Correlation coefficient of "Ng, Goh, Low" [5] :	$ngl(t_k, c_i) = \frac{\sqrt{N}((N_{t_k,c_i}N_{\overline{t_k},\overline{c_i}}) - (N_{t_k,\overline{c_i}}N_{\overline{t_k},c_i}))}{\sqrt{N_{t_k}N_{\overline{t_k}}N_{c_i}N_{\overline{c_i}}}}$
Coefficient of "Galavotti, Sebastiani, and Simi" [6]	$gss(t_k, c_i) = (N_{t_k,c_i}N_{\overline{t_k},\overline{c_i}}) - (N_{t_k,\overline{c_i}}N_{\overline{t_k},c_i})$
Relevance frequency	$rf(t_k, c_i) = \log\left(2 + \frac{N_{t_k,c_i}}{max(1,N_{t_k,\overline{c_i}})}\right)$
Information gain	$ig(t_k, c_i) =$ $\begin{array}{l}(N_{t_k,c_i} * \log(N_{t_k,c_i}/(N_{t_k}N_{c_i}))) \\ \quad + (N_{\overline{t_k},c_i} * \log(N_{\overline{t_k},c_i}/(N_{\overline{t_k}}N_{c_i}))) \\ \quad\quad + (N_{t_k,\overline{c_i}} * \log(N_{t_k,\overline{c_i}}/(N_{t_k}N_{\overline{c_i}}))) \\ \quad\quad\quad + (N_{\overline{t_k},\overline{c_i}} * \log(N_{\overline{t_k},\overline{c_i}}/(N_{\overline{t_k}}N_{\overline{c_i}})))\end{array}$
Kulback-Leibler divergence	$kld(t_k, c_i) = (N_{t_k,c_i}/N_{t_k}) * \log(\frac{N_{t_k,c_i}N}{N_{t_k}N_{c_i}})$
Delta Smoothed IDF	$dsidf(t_k, c_i) = \log(\frac{(N_{\overline{c_i}}N_{t_k,c_i})+0.5}{(N_{c_i}N_{t_k,\overline{c_i}})+0.5}$
Delta BM25 IDF	$dbidf(t_k, c_i) = \log(\frac{(N_{\overline{c_i}} - N_{t_k,\overline{c_i}}+0.5)*(N_{t_k,c_i}+0.5)}{(N_{c_i} - N_{t_k,c_i}+0.5)*N_{t_k,\overline{c_i}}+0.5)}$

3.2 Text classification using term weighting

Traditional classification algorithms such as support vector machines (SVM) or K-nearest neighbor are vector based models that are strong baselines for several classification tasks. Even in the legal domain, SVM [7] and Ensemble learning methods [8] behave very well for predicting the law area and the decision of judges of the French Supreme Court. To use these algorithms for text classifi-

cation, the documents are represented as vectors using a method such as the Bag-of-Words (BoW) that defines a vector space model where the dimensions are identified by a set of terms learned from a training corpus. The number of occurrences of the terms is consider while their ordering is ignored. Given a term t_k among the dimensions $T = t_1, t_2, ..., t_{|T|}$, and a document D_j, the weight score $w(t_k, D_j)$ of t_k for D_j is the product of a local weight $lw(t_k, D_j)$ based on the frequency of t_k in D_j (table 3), a global weight $gw(t_k)$ (table 2), and a normalization factor $nf(D_j)$ to avoid issues related to the difference of length between documents [9]: $w(t_k, D_j) = lw(t_k, D_j) \times gw(t_k) \times nf(D_j)$. The cosine factor is a very popular normalization factor: $nf_{cos}(D_j) = 1/\sqrt{\sum_{k=1}^{|T|} w^2(t_k, D_j)}$.

Table 3. Local weighting metrics used in this work

Description	Metric formula
Raw term frequency [9] : number of times t_k occurs in D_j	$tf(t_k, D_j) = occ(t_k, D_j)$
Term presence [9] : all the terms weight the same inside a document	$tp(t_k, D_j) = \begin{cases} 1 \text{ , if } tf(t_k, D_j) > 0 \\ 0 \text{ , otherwise} \end{cases}$
Logarithm of the term frequency :	$logtf(t_k, D_j) = 1 + \log\left(tf(t_k, D_j)\right)$
Augmented normalized term frequency [9]: k is set such that the normalized weight should lie in $[0.5, 1]$	$atf(t_k, D_j) = k + (1-k)\frac{tf(t_k, D_j)}{\max_{t_k \in T} tf(t_k, D_j)}$
Averaged term frequency based normalization [10]	$logave(t_k, D_j) = \frac{1+\log tf(t_k, D_j)}{1+\log \operatorname{avg}_{t_k \in T} tf(t_k, D_j)}$

avg denotes the average.
$occ(t_k, D_j)$ denotes the number of times t_k occurs in D_j

The dimensions of the vector space model are usually reduced to a minimum set of discriminative features by defining a threshold on $gw(\cdot)$ after its normalization over T using the formula $\left(gw(t_k) = \frac{gw(t_k) - \min_{t_i \in T}(gw(t_i))}{\max_{t_i \in T}(gw(t_i)) - \min_{t_i \in T}(gw(t_i))}\right)$.

4 A baseline approach based on term weighting

Our approach takes a court decision at input, and detects the present categories, and finally, for each detected category, the category-specific claim recognizer extracts the claimed quanta, the result polarity and the obtained quantum, and matches them(Figure 1).

Stage 1 - preprocessing court decisions : Documents are available in various formats (DOC, RTF, PDF, text, ...) and thus they are all converted into

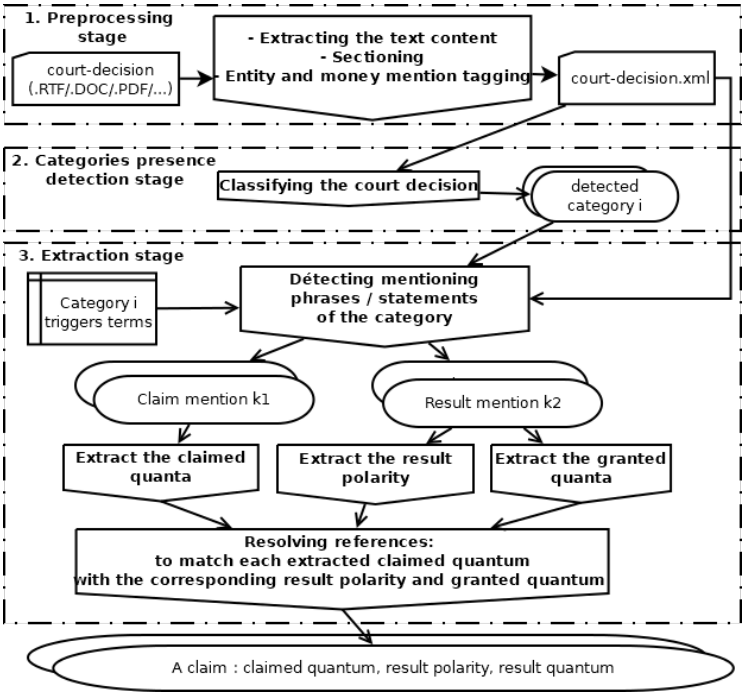

Fig. 1. Claim extraction proposed approach

text to enable the same processing on every document. Documents are then sectioned and money mentions are tagged by using an entity labeling approach similar to [11]. We split the documents into 5 sections : the Header that contains meta-data, the Litigation section containing facts, previous procedures, claims and arguments of the parties, the Reasonings section where judges detail their arguments and results, the Dispositions section that summarizes the result of the cases, the signature foot part.

Stage 2 - categories presence detection : The system learns the best meta-parameters (i.e the best classifier algorithm , the best global and local weights, and the best global weight thresholds) for each category to train a binary classifier that can detect its presence inside documents. With a cross-validation process on the training dataset, the system chooses the combination of meta-parameters that has the best result. Only the main part (*Litigation + Reasoning + Dispositions*) of the documents are used for the classification.

Stage 3 - claim-result information extraction : Here we assume that the claims are in the section *Litigation* and the results are in the section *Dispositions*. Our strategy is to zone around amounts of money to identify the statement of

Table 4. Some introducing words of statements of claims and results

Claims	**Results** (organised per polarity)		
	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, surseoit, sursoyons*	*déboute, déboutons, rejette, rejettons*

claims and results. A zone is just a substring going from the introducing word before the money mention to the introducing word or at the point after the money mention. An example is given by the Figure 2. We define manually some introducing words (Table 4). After the statements have been identified, those that are related to the category are selected through a zone weighting based strategy: since the quanta are money amounts, the idea here is to compute a weight for each candidate zone by summing the weights of the trigger terms present in the zone. The selected zones are those that have a weight greater than a learned threshold. The system learned a threshold for claims and another one for results. The quanta are then extracted by taking the money mentions that are the closest to a trigger. The result polarity is determined according to its introducing word. Finally, after matching the claim statement to the corresponding result statement with a text distance score, the claimed quanta is matched to the quanta result by assuming that the quanta in the result phrase appear in the same order as in the claim phrase since there might be multiple quanta in the same statement. However, a claimed quantum with no matched result is considered to have been rejected.

" ... débouter M. S. de l' ensemble de ses demandes
- le <**claim category="acpa"**>condamner à payer une <trigger category="acpa">amende civile</trigger> de <money> 1.500 euros </money> pour procédure abusive ...
- le</**claim**> condamner à payer la somme ..."

Fig. 2. Example of a claim statement zoned aroung the trigger *amende civile*

The parameters to learn are the optimal set of trigger terms, the optimal thresholds of claims and results. The terms might be n-grams of various length. For example, in our experiments, we learn tersm with 3,2, and 1 word. The training dataset is split into two parts D and $D^{validation}$. The zones around the quanta from the ground truth of D are used as a positive class associated to (c_i) and the zones around money that are not quanta of reference are used as the negative class associated to $\overline{c_i}$ to learn the therms that are pro c_i (i.e. $DF_{t_k,c_i} > DF_{t_k,\overline{c_i}}$). After ranking the terms, the learning approach then selects

the terms successively from the most important one (algorithm 1). A term is kept in the optimal list if it improves the current F1-score.

Algorithm 1: Parameters learning algorithm for the extraction stage

Data: D, $D^{validation}$, X = list of the ranked terms x_k
Result: optimal term subset Y_k, the minimal thresholds of claim zones $optW_{claim}$ and of result zones $optW_{result}$

Start with $Y_0 = \emptyset$; $k = 0$; maxF1Score = 0;
repeat
  remove x_k from X ;
  Mark the terms $Y_k + x_k$ in the quanta zones of D;
  wt_{claim} = min weight of the claimed quanta zones of D;
  wt_{result} = min weight of the result quanta zones of D;
  if *maxF1-score* $<$ *F1-score*$(Y_k + x, wt_{claim}, wt_{result}, D^{validation})$ **then**
    $Y_{k+1} = Y_k + x_k$;
    $optW_{claim} = wt_{claim}$;
    $optW_{result} = wt_{result}$;
    max_F1-score = F1-score$(Y_k + x, wt_{claim}, wt_{result}, D^{validation})$;
  else
    $Y_{k+1} = Y_k$;
until $X = \emptyset$ *or convergence of the F1-score*;
return Y_k, $optW_{claim}$, $optW_{result}$;

5 Experiments

The data we used for the experiments were labeled manually by a legal scholar. The dataset contains 610 claims extracted from 431 decisions (decisions of D_{c_i} only) category distributed as shown on figure 3 (the supplied $D_{\overline{c_i}}$ are represented by the grey bars). There are 6 categories that we call respectively : *acpa* - financial penalty for abusive procedure, concdel - damages for unfair competition, *danais* - damages for abusive procedure, *dcppc* - statement of claim on the liabilities side of the insolvency proceedings, *doris* - damages for neighborhood disorder, *styx* - damages on the basis of Article 700 of the Code of Civil Procedure. The category *styx* (article 700) is present in almost all the court decisions, so it was difficult to find decisions to make a $D_{\overline{styx}}$. To create that corpus, the annotator took some decisions from which he remove all the claims of the category *styx*.

We tested 4 classifiers implemented by the Java data mining library Weka [12]: the naive Bayes classifier NB, the decision tree J48, the K-nearest neighbor KNN, and the SVM. As shown in table 5, all the algorithms seem to give very good results for any category, it is obvious that the terms related to the categories are very discriminative so that a baseline algorithm is suitable for the detection of the presence of claim categories in court decisions. It is important to notice that

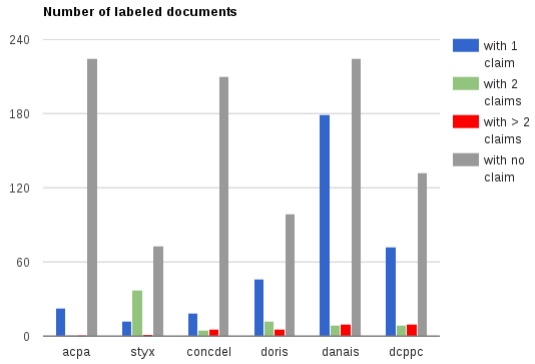

Fig. 3. Dataset: Number of documents per category and claims distribution

this results are observed because the best weighting metrics and the global weight threshold were learned. On the other side, the results of the extraction stage alone are detailed in table 6. The first remarkable observation if the difference of the degree of difficulty between the categories. While the some are barely extractable, *acpa, danais styx* seem easier to extract even though all the results are still low. We also notice that there is not a global weighting metric that is the best for every categories. That is the reason why we add the choice of the best metric in the training process.

Table 5. Results of a 5-fold cross-validation for the category detection (P= Precision, R=Recall, F1=F1score)

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

There might be several source of errors that make us get such low results. According to the expert annotator, *concdel, dcppc* and *doris* are such categories that have multiple sub-categories because there are various prejudices among the cases that are related to them and it is difficult to capture most of the relevant terms with the small size of the available labeled data. So the global weight metrics miss some trigger terms and since the triggers are the keys of the

Table 6. Result of a 5-fold cross-validation for only the extraction stage (R=Recall, F1=F1score)

	acpa		concdel		danais		dcppc		doris		styx	
Weight	R	F1	R	F1	R	F1	R	F1	R	F1	R	F1
*chi*2	0.52	0.54	0.074	0.075	0.425	0.432	0.17	0.199	0.091	0.11	0.373	0.442
dbidf	0.52	0.54	0.174	0.147	0.275	0.386	0.156	0.189	0.01	0.017	0.371	0.44
deltadf	0.52	0.54	0.264	0.149	0.463	0.461	0.17	0.199	0.207	0.17	0.37	0.441
dsidf	0.57	0.59	0.024	0.044	0.077	0.131	0	0	0	0	0.279	0.333
gss	0.52	0.54	0.264	0.149	0.463	0.461	0.17	0.199	0.207	0.17	0.37	0.441
idf	0.05	0.057	0	0	0	0	0	0	0	0	0	0
ig	0.26	0.098	0.228	0.038	0.015	0.025	0	0	0.01	0.017	0	0
kld	0.39	0.371	0.223	0.141	0.395	0.413	0.155	0.186	0.105	0.125	0.382	0.435
mar	0.52	0.54	0.293	0.155	0.463	0.461	0.17	0.199	0.238	0.172	0.37	0.441
ngl	0.52	0.54	0.074	0.073	0.425	0.433	0.17	0.199	0.071	0.08	0.362	0.429
rf	0.52	0.54	0.06	0.086	0.323	0.422	0.111	0.149	0.055	0.072	0.349	0.409

approach, the system consequently misses a lot of claims or results like shown in table 7. On the other hand, the matching of claimed quantum to result might also contribute to the errors. Although we do not have a token-level labeled sample to evaluate our matching approach, by comparing the difference between the results on single information with the performance on tuples of information, we can know if the matching contributes a lot to the errors. Table 7 detailed the results obtained with the best global metric for some information and tuples of information. The results on single informations are a little bit better than the results on the tuples so the matching is not perfect but the detection of the individual informations remains the greater contributor to the errors.

Table 7. Recognition stage: 5-fold cross-validation results on some tuples of information with the best global weight metric of each category (R=Recall, F1=F1score)

	Q_{DMD}		Q_{RST}		S_{RST}		$Q_{RST} + S_{RST}$		$Q_{DMD} + Q_{RST} + S_{RST}$	
	R	F1	R	F1	R	F1	R	F1	R	F1
acpa	0.61	0.634	0.74	0.79	0.74	0.79	0.74	0.79	0.57	0.59
concdel	0.414	0.21	0.393	0.203	0.364	0.195	0.364	0.195	0.293	0.155
danais	0.501	0.498	0.549	0.545	0.535	0.532	0.535	0.532	0.463	0.461
dcppc	0.224	0.26	0.479	0.561	0.55	0.643	0.471	0.553	0.17	0.199
doris	0.369	0.27	0.363	0.264	0.38	0.282	0.342	0.25	0.238	0.172
styx	0.455	0.539	0.467	0.554	0.456	0.543	0.45	0.535	0.373	0.442

To evaluate whether or not the prior detection of categories improves the results, we combined and experimented the two stages on the same cross-validation splits that we did to get tables 6 and 7. The results we obtained are shown in table 8. We observe that while the classification improves the performances on the most difficult categories, it decreases the performances on the easiest ones. So the classification might be useful only for some categories.

Table 8. Classification + recognition stages: 5-fold cross-validation results on some tuples of information (R=Recall, F1=F1score)

	Q_{DMD}		Q_{RST}		S_{RST}		$Q_{RST} + S_{RST}$		$Q_{DMD} + Q_{RST} + S_{RST}$	
	R	F1	R	F1	R	F1	R	F1	R	F1
acpa	0.560	0.595	0.700	0.757	0.700	0.757	0.700	0.757	0.520	0.545
concdel	0.317	0.319	0.374	0.366	0.345	0.351	0.345	0.351	0.263	0.268
danais	0.312	0.433	0.326	0.453	0.321	0.447	0.317	0.440	0.283	0.394
dcppc	0.224	0.261	0.463	0.546	0.534	0.628	0.455	0.537	0.178	0.211
doris	0.298	0.300	0.317	0.320	0.334	0.347	0.296	0.302	0.192	0.197
styx	0.383	0.448	0.388	0.455	0.386	0.458	0.380	0.446	0.319	0.376

6 Conclusion

We presented the problem of extracting claim-result tuples of information from court decisions, an evaluation method and a first baseline that detects the statements of claims and results, then extracts and matches the informations. This problem is very important because its solution might be the key to do efficient automatic analyses of judgments in order to assist legal experts in their comprehension of court rulings. Our current approach learns automatically the triggers that indicate mentions of claims and results and leverages them to locate the informations. We only processed explicitly expressed claims and results that have a simple and resembling pattern but there is still more information in other parts of the document. We have not yet exploit the references to previous judgments and the informations detailed in the Reasonings section. We intend to extend our current approach to those parts to improve our results and to get a full baseline. Beside that, our experiments show that some categories are very difficult than others, thus it is important to find better solutions to that problem. The resemblance between claim extraction and event extraction makes us believe that the same techniques might apply in both cases. For example, the term weighting metrics might improve the detection of triggers in an open event task. On the other hand, since it is quite difficult to get token-level annotations of claims in documents, real world data oriented methods such as the end-to-end deep learn-

ing approach described by [13] are interested to extend in order to deal with the multi- sentence and multi-instance challenge of claim extraction.

References

[1] Cretin, L.: L'opinion des français sur la justice. INFOSTAT JUSTICE **125** (Janvier 2014)

[2] Dave, K.: Study of feature selection algorithms for text-categorization. (2011)

[3] Wu, H., Gu, X., Gu, Y.: Balancing between over-weighting and under-weighting in supervised term weighting. Information Processing & Management **53**(2) (2017) 547–557

[4] Sparck Jones, K.: A statistical interpretation of term specificity and its application in retrieval. Journal of documentation **28**(1) (1972) 11–21

[5] Ng, H.T., Goh, W.B., Low, K.L.: Feature selection, perceptron learning, and a usability case study for text categorization. In: ACM SIGIR Forum. Volume 31., ACM (1997) 67–73

[6] Galavotti, L., Sebastiani, F., Simi, M.: Experiments on the use of feature selection and negative evidence in automated text categorization. In: International Conference on Theory and Practice of Digital Libraries, Springer (2000) 59–68

[7] Şulea, O.M., Zampieri, M., Vela, M., van Genabith, J.: Predicting the law area and decisions of french supreme court cases. In: Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017. (2017) 716–722

[8] Sulea, O.M., Zampieri, M., Malmasi, S., Vela, M., Dinu, L.P., van Genabith, J.: Exploring the use of text classification in the legal domain. In: Proceedings of the 2017 ASAIL Workshop on Automatic Semantic Analysis of Information in Legal Text, London, UK, to appear. (2017)

[9] Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Information processing & management **24**(5) (1988) 513–523

[10] Manning, C.D., Raghavan, P., Schütze, H.: Scoring, term weighting and the vector space model. In: Introduction to information retrieval. Cambridge university press, Cambridge (2008) 109–133

[11] Tagny Ngompé, G., Harispe, S., Zambrano, G., Montmain, J., Mussard, S.: Reconnaissance de sections et d'entités dans les décisions de justice: application des modèles probabilistes HMM et CRF. In: Procedings of Extraction et Gestion des Connaissances EGC 2017, Grenoble, France - Revue des Nouvelles Technologies de l'Information. (2017) 201–212

[12] Frank, E., Hall, M., Witten, I.: The weka workbench. Data mining: Practical machine learning tools and techniques. Burlington: Morgan Kaufmann (2016)

[13] Palm, R.B., Hovy, D., Laws, F., Winther, O.: End-to-end information extraction without token-level supervision. arXiv preprint arXiv:1707.04913 (2017)