Extracting information about claims from judgments: problem, evaluation and a first baseline approach with text classification and term weighting

Gildas Tagny Ngompé[1,2], Sébastien Harispe[1], Jacky Montmain[1], Guillaume Zambrano[2], and Stéphane Mussard[2]

[1] Ecole des mines d'Alès, Laboratoire de Génie Informatique et d'Ingénierie de Production (LGI2P), site de Croupillac, 7 rue Jules Renard, Alès, France, {gildas.tagny-ngompe,sebastien.harispe,jacky.montmain}@mines-ales.fr,
[2] Université de Nîmes, CHROME EA7352, site Vauban, Rue du Dr Georges Salan, 30021 Nîmes Cedex 01, France, {guillaume.zambrano,stephane.mussard}@unimes.fr

Abstract. The analysis of judgments is very important in the work of legal practitioners to understand how judges rule. Gathering information about the claims i.e. find out what have been ask, what have been decided and why, is the key to understand a case. We introduce the task of extracting the claimed quantum, the result polarity, and the granted quantum for each individual claim present in a court decisions. We address the problem by categories of claims since an expert are generally interested in a particular type of claim. we propose also a baseline approach that uses term weighting schemes and classification techniques to learn useful features related to a category in order to locate the targeted information inside 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 reasoning of judges of a legal case. A legal practitioners should analyze a set of court decisions in order to understand how courts make decisions on particular types of cases, and what factors make judges to accept or to reject a claim. First of all, from each decision, experts should find answers to some semantic questions such as: "what was claimed?", "Was the claim granted?", "How much was ranted to the claimant?" and "why was this solution pronounced?". A comprehensive analysis of a type of case should implies to collect and analyses a large amount of documents, and that requires a lot of time and money if it is done manually. Legal experts need to be assisted automatically for the analysis of comprehensive corpora to do their work faster. We are addressing the problem of extracting automatically what was claimed and granted from judgment

documents. More precisely, we are interested in 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 quantum).

Our aim is to enrich a knowledge base of judgments structured as shown on Table 1. This table is an extract of the annotation data we used in this work and it shows some claims (a claim per line) for damages under Article 700 of the Code of Civil Procedure (*article 700 du code de procédure civile*) extracted from the court decision number 14/06911 of the appeal court (CA) of Lyon (the city where the court is located); only one is accepted. There are to claims from the same document; the last one was accepted and the defendant was order to pay to the claimant 1500 euros, but the second claim was rejected.

Each column describes an information on the claim. The first three columns identify the court decision (document) where the claim was extracted from (resp. the type of court, the city, the registry identifier number of the document at the level of the court). The next three columns describe the claim made by a party . The claims are organized into categories defined by two informations : an object (e.g. damages, financial penalty) and a norm (the legal basis). The last two columns describe the corresponding answer that the judges gave; it is defined by the result polarity (reject / accept) and the quantum granted (*quantum resultat*).

Table 1. Structure and examples of claims

	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 €

We define the problem of extracting information about claims on a set of categories defined by an expert annotator by giving the object and the norm. Claims are understandable according to their category. Since there is a lot of categories (at least 500), we define the task by category to ease the annotation process of the evaluation data.

Except from the quantity of existing categories, there are other challenges to face when dealing with this task. For example, there are multiple claims of same category or of different categories in a document. All the informations of a claim are not usually in the same sentence or phrase, and often not even in the same section (even if court decisions doesn't have a standard structure template shared by all the courts). The claimed quantum is usually in a region that describe all the claims and the arguments of the claimant (following the facts and previous procedures or judgments). As for the results, they usually

located at the bottom of the document in the section of the dispositions i.e. the judges' rulings (*dispositif*). However the section of dispositions expresses the rulings in a very contracted way, and more detailed are to be found in the section above; that is the section of the reasons (*motifs de la decision*) where the judges justify their dispositions. The ruling is usually more explicit there.

Moreover, the texts are unstructured and written in a natural and domain-specific language with multiple complex aspects such as ambiguities, aggregated statement (e.g. a result that answer multiple claims), references to previous judgments or previous claims. Some examples are given by Figure 1 and Figure 2.

La société A. conclut à la confirmation du jugement entrepris sauf à former appel incident sur la disposition du jugement l'ayant déboutée de sa demande de dommages intérêts pour abus de procédure et elle demande à la cour de condamner l'appelante à lui payer la somme de 20 000 euros à titre de dommages intérêts ...

Fig. 1. An unstructured expression of a claim

La cour, ...
Confirme la décision entreprise en toutes ses dispositions,

Fig. 2. An expression of result in a decision of a court of appeal, with a reference to the decision of the previously-judged litigation that is examined and an aggregation over all the dispositions of that previous judgment.

We present also a way towards a solution to solve the task of extracting claims. Because there are a lot of categories and it is very difficult to labeled the data, we define a two-stage or a pipeline based approach that is adaptable to a specific category of interest. The pipeline starts with a classification of the document to be sure it contains the category and if that is the case, the other stage, the recognition stage leverages the lexicon commonly used to expressed claims of that category, to locate the informations of the different claims.

Before we give more details about that pipeline, we first discuss the evaluation process in the following section.

2 Evaluation protocol and metrics

2.1 When is an information correctly extracted?

The task is defined according to a category, so is the evaluation. Let's denote the category of interest by $c_i \in C$, with C the set of all the existing categories. To evaluate the extraction process, we should first have a standard way to know when the an extracted information is correct or not. We need a testing dataset

labeled manually and structured as the Table 1 with some claims of reference comprehensively extracted from a given corpus D_{c_i}. We also need a corpus $D_{\overline{c_i}}$ of court decisions that do not contain any claim of the category to verify that the model does not extract claims from documents that do not contain the category.

To compare extracted information with information of reference, we consider quanta as numbers since they are amounts of money. It is enough to convert them into real numbers to compare them. For our experiments we just consider digits and commas (decimal separator in French) in the mention of a quantum. As for the result polarity, it is categorical (*accepte / rejette / sursis à statuer*) and thus it is very easy to evaluate its extraction. Moreover, a claim has 3 informations, and the extractor system should be able to return the 3 informations jointly in order to consider the claim to have been correctly extracted. We can also evaluate the system on the extraction of just one, two or three informations. Finally, a document might contain multiple claims of the same category, thus the extracted information are matched successively with the claim of reference.

2.2 Measure the success

To evaluate how good a model can extract claims of a category $c_i \in C$, we need a labeled validation corpus $D = D_{c_i} \cup D_{\overline{c_i}} = \{D_j\}_{j \in [1,|D|]}$.

Let's note I the set of types information that are to be extracted about a claim. I might be a subset of $\{Q\_DMD, S\_RST, Q\_RST\}$ where Q_DMD, S_RST, Q_RST are resp. the requested quantum, the meaning of the result, and the granted quantum.

During the evaluation, we are going to match successively tuples extracted from D_j with those in the gold standard annotation of D_j. A tuple is well extracted from $D_j \in D_{c_i}$ if there is still a non identified tuple in the manual annotation of D_j that matches exactly with the extracted one.

We use the F-score metric because it is commonly used to evaluate information extraction tasks. We first define the basic metrics number of true positives (TP_{c_i,I,D_j}), of false positives (FP), and of false negatives (FN) at the level of the document:

Given a category, there are two levels of evaluation depending on whether we want to evaluate the capacity of a system to extract all the claims of a document (document level), or its capacity to extract all the claims in the corpus $D_{c_i} \cup D_{\overline{c_i}}$ (claim level).

2.3 Document level evaluation With precision, recall and F1-measure

:

First of all, we compute the precision (P_{c_i,I,D_j}) and the recall (R_{c_i,I,D_j}) at the level of a document D_j.

$$P_{c_i,I,D_j} = \frac{\#d_{j,k} \in c_i \text{ correctly extracted from } D_j}{\#d_{j,k} \in c_i \text{ extracted from } D_j} = \frac{TP_{c_i,I,D_j}}{TP_{c_i,I,D_j} + FP_{c_i,I,D_j}}$$

$$R_{c_i,I,D_j} = \frac{\#d_{j,k} \in c_i \text{ correctly extracted from } D_j}{\#d_{j,k} \in c_i \text{ present in } D_j} = \frac{TP_{c_i,I,D_j}}{TP_{c_i,I,D_j} + FN_{c_i,I,D_j}}$$

Then the final result score is obtained with the mean of the precisions and recalls over the number of documents. With respect to those formula, the document level evaluation is computable only over the corpus D_{c_i}.

$$Precision_{c_i,I,D_{c_i}} = \frac{\sum_{j=1}^{|D_{c_i}|} P_{c_i,I,D_j}}{|D_{c_i}|} \qquad Recall_{c_i,I,D_{c_i}} = \frac{\sum_{j=1}^{|D_{c_i}|} R_{c_i,I,D_j}}{|D_{c_i}|}$$

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

Corpus or overall level evaluation With precision, recall and F1-measure : We want to take into account the errors made by a model that identifies a claim inside a document of $D_{\overline{c_i}}$. Hence, we focus on the precision and the recall at the level of the corpus $D_{c_i} \cup D_{\overline{c_i}}$. They are defined as usually:

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

and the F1-measure is written similarly to the claim-level one:

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

The numbers of true positives (TP), false positives (FP), and false negatives (FN) are defined as follow:

- $TP_{c_i,I,D}$ is the number of information of type I correctly extracted from D: $TP_{c_i,I,D} = \sum_{j=1}^{|D|} TP_{c_i,I,D_j} = TP_{c_i,I,D_{c_i}}$
- $FP_{c_i,I,D}$ is the number of information extracted of type I from $D_{\overline{c_i}}$ and wrongly classified as c_i (it is equal to $FP_{c_i,I,D_{\overline{c_i}}}$)
- $FN_{c_i,I,D}$ is the number of information of type I missed from D (it is equal to $FN_{c_i,I,D_{c_i}}$)

Since a court decision contains multiple claims of similar or different categories, our intuition is that categories are distinguishable from others by their language or more precisely by a particular vocabulary used to expressed them. Hence, our idea is that once it is certain that a category is present in the document, it might be easier to locate the the targeted informations because they are probably not far from the terms usually used to express a claim or an order.

3 Term weighting and supervised document categorization

3.1 Global term weighting schemes

Term weighting metrics are commonly studied feature selection for text categorization and information retrieval. The supervised metrics compute a global

score of a term over a labeled corpus by comparing the probability of the appearance term occurrence in a document of a given category c_i (i.e. in D_{c_i}) and its appearance in a document out of c_i (i.e. in $D_{\overline{c_i}}$). On the other side, the unsupervised methods compute a global score of a term all over the corpus independently of the category of the document where the term appears. Even if supervised schemes compute directly a correlation score between a category and a term, we expect that if a category is really discriminative over all the corpus of decisions, the terms that differentiate D_{c_i} and $D_{\overline{c_i}}$ should be those terms that are related to c_i. Thus even unsupervised weighting schemes such as inverse document frequency (idf)[1], the probabilistic idf(pidf) [2], BM 25 idf ($bidf$) [3] are interesting to experiment. We study some existing schemes (Table 2).

3.2 Text classification using term weighting

Extracting features or Representing document as vector :

Term-weighting based vector space model or Bag-of-words representation :

the Bag-of-words is a popular and simplified representation through which each document is represented as a vector whose dimensions are defined by a set of terms that are learned from a training corpus. The number of occurrences of the terms is consider while their ordering is ignored. So, given the term t_i within the dimensions $T = t_1, t_2, ..., t_{|T|}$, and a document D_j, the weight score $w(t_i, D_j)$ of t_i for D_j is the product of a local weight $lw(t_i, D_j)$ of t_i computed inside D_j, a global weight $gw(t_i)$ of t_i computed over the training corpus, and a normalization factor $nf(D_j)$ computed over the terms in D_j to avoid issues of different document lengths [6]:

$$w(t_i, D_j) = lw(t_i, D_j) \times gw(t_i) \times nf(D_j)$$

Semantic or Topic -based vector space model :

Dimensionality reduction :

The vector space model is subject to high dimensionality and sparsity problems. The dimensionality or data reduction aims to reduce the representation to a minimum set of discriminative features. For text categorization, There are two main approaches to reduce the dimensions.

First, the feature selection method consists of defining a threshold value a on a unsupervised term-weighting scheme like the , or a supervised term weighting scheme such as information gain (ig), chi-square ($chi2$), the NGL correlation coefficient (ngl), ...

The second approach is called feature transformation. It consists of transforming or projecting an original vector into a new space model usual with less dimension than the original but more discriminative.

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 [1]: 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" [4] :	$ngl(t_k, c_i) = \frac{\sqrt{N}((N_{t_k,c_i} N_{\overline{t_k},\overline{c_i}}) - (N_{t_k,\overline{c_i}} N_{\overline{t_k},c_i}))}{\sqrt{N_{t_k} N_{\overline{t_k}} N_{c_i} N_{\overline{c_i}}}}$
Coefficient of "Galavotti, Sebastiani, and Simi" [5]	$gss(t_k, c_i) = (N_{t_k,c_i} N_{\overline{t_k},\overline{c_i}}) - (N_{t_k,\overline{c_i}} N_{\overline{t_k},c_i})$
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	$dbidf(t_k, c_i) = \log(\frac{(N_{\overline{c_i}} - N_{t_k,\overline{c_i}} + 0.5) * (N_{t_k,c_i} + 0.5)}{(N_{c_i} - N_{t_k,c_i} + 0.5) * N_{t_k,\overline{c_i}} + 0.5)}$

3.3 Classifiers algorithms

Using the vector space model representation, any vector based off-the-shelf classifier model can be used for text categorization. Traditional algorithms such as the

Table 3. Local weighting metrics studied in this work

Description	Metric formula
Raw term frequency [6] : number of times t_i occurs in D_j	$tf(t_i, D_j) = occ(t_i, D_j)$
Term presence [6] : all the terms weight the same inside a document	$tp(t_i, D_j) = \begin{cases} 1 \text{ , if } tf(t_i, D_j) > 0 \\ 0 \text{ , otherwise} \end{cases}$
Logarithm of the term frequency :	$logtf(t_i, D_j) = 1 + \log\left(tf(t_i, D_j)\right)$
Augmented normalized term frequency [6]: k is set such that the normalized weight should lie in $[0.5, 1]$	$atf(t_i, D_j) = k + (1-k)\frac{tf(t_i, D_j)}{\max\limits_{t_k \in T} tf(t_k, D_j)}$
Averaged term frequency based normalization [7]	$logave(t_i, D_j) = \frac{1+\log tf(t_i, D_j)}{1+\log \operatorname{avg}\limits_{t_k \in T} tf(t_k, D_j)}$

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

Naive Bayesian (NB), support vector machines (SVM), the K nearest neighbor (KNN), decision tree are very popular and strong baselines for text categorization when they are associated with the appropriate document representation.

Even if the LDA topic modeling based vector representation might outperform all the other representations [8, 9], the state-of-the-art methods are based on deep learning algorithms and word embedding representations.

Even in the legal domain, SVM [10] and Ensemble learning methods [11] behave very well for predicting the law area and the decision of judges of the French Supreme Court.

http://arxiv-sanity.com/search?q=text+classification

4 A first approach based on supervised term weighting

Since all the categories are not known in advance, the only definition we consider for a category and a claim is given by an expert. Legal experts define a category by combining a legal norm (article 700) to an object of claim (damages). They define what are consider as claim for a category with a sample of examples of claims, that are extracted from a sample of decisions. But we can say that in general, any claimed quantum defined a claim and has an associated result. Our approach extracts the claimed quanta, the result polarity the obtained quantum, and matches them through the pipeline (Figure 3) described in the following subsections.

4.1 Stage 1 : preprocessing court decisions

When it comes to collect court decisions, the documents are available in different format (DOC, RTF, PDF, text, ...). The first convert them to the plain

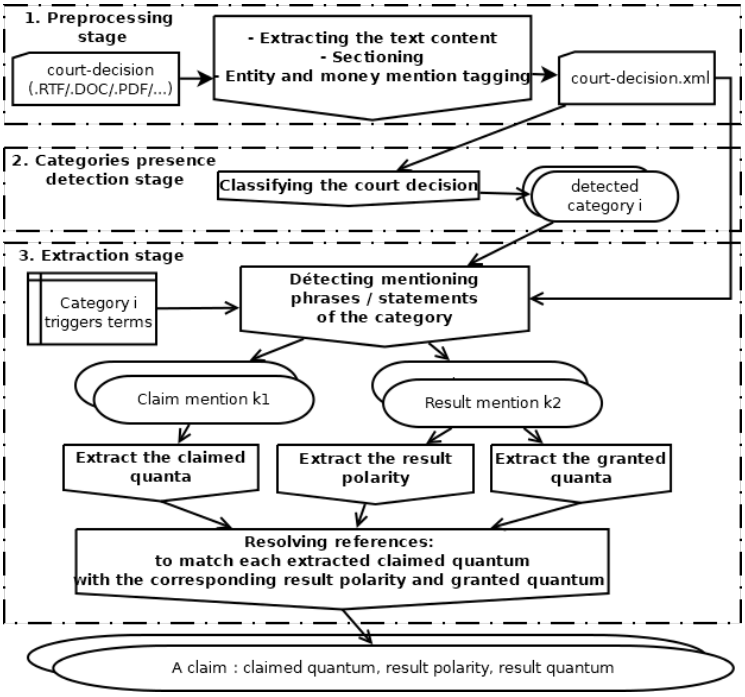

Fig. 3. Claim extraction proposed approach

text format to enable the same processing for every documents. Then, the input document is preprocessed using a similar Conditional Random Fields [12] based approach suggested by [13]. In fact, the document is sectioned and then, the entities and money mentions are tagged. But instead of 3 sections as in [13], we split documents into 5 sections that are displayed in this order : the Header section (*entête*) that contains meta-data, the Litigation section (*litige*) containing facts, previous procedures, claims and arguments of the parties, the Reasons section (*motifs*) where judges give details about their arguments and results, the Dispositions section that summarizes the result of the cases (*dispositif*), the signature foot part that is separated from the dispositions to avoid noise.

4.2 Stage 2 : categories presence detection

Since our approach depends on the predefined categories, we should make sure that a category exists in the decision before running the extraction process. The purpose of this sub-task is to avoid the waste of time and the risk of noise that might occur when trying to extract claims from a document that does not contain the targeted category.

Application process: classify the main content of the document To detect whether a document contains a claim of a given category or not, the category detection stage use a binary classifier that is trained particularly for the category. So, the main part of the document (i.e. the 3 main sections Litigation, Reasons, and Dispositions) is represented as a vector and presented at the input of the binary classifier of the category. This method make us have as many classifiers as categories but it is a suitable and simple solution to this sub-task. Since a decision has multiple claims of similar or different categories, a unique multi-class and multi-label classifier seem a better solution but not all the categories are precisely defined and the datasets are labeled per category.

Training process: learning the best classifier To learn the best detector for a category c_i, the system first learn the best meta-parameters that are suitable for the category: the global weighting metric, the suitable global weight threshold to select features, the local weighting metric and the suitable classifier algorithm. The meta-parameters are learned with a cross-validation by splitting the training dataset. At the input of the cross-validation process, we supply a list of global weighting metrics (Table 2), a list of local weighting metrics (Table 3), a list of thresholds values, and a list of algorithms. Finally, the documents of the training dataset $D^{train} = D_{c_i}^{train} \cup D_{\overline{c_i}}^{train}$ are represented as vectors using the best weighting local metric and global metric with the best threshold to train the best found algorithm.

4.3 Stage 3 : claim-result information extraction

Application phase: zoning, locating and matching This stage first uses some trigger terms associated with the category to locate claim and result mentions resp. in section Litigation and Dispositions in order to extract the informations. We define two strategies to detect mentions of claims and results:

1. Triggers presence based strategy: the idea is to tag the trigger terms in the input document, and then to zone the phrase inside which each term is located. Finally the claim or result mentions are simply those that contain a trigger term
2. Zone or phrase weighting based strategy: since the quanta are money amounts, the idea here is to zone around all the tagged money mentions first and compute a weight of the zone by summing the weight of the triggers that it contains. The zones that are selected, to extract the information, are those that have a weight greater than a learned threshold weight. The system learned a threshold for claim phrases and another threshold for result phrases.

The process of zoning is quite easy. A zone is a substring going from the statement introducing word before the pivot (money or trigger) and stopping before the next introducing word or the next point. An example is given by the Figure 4. We define two lexicons: one for the claims and the other for the results (Table 4).

" ... 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 en application de l' <norm> article 32-1 du code de procédure civile </norm>
- le**</claim>** condamner à payer la somme ..."

Fig. 4. Example of a claim statement zoned aroung the trigger *amende civile*

Table 4. Some introducing words of statements with quantum

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*

Finally, the quanta are extracted by taking the money mentions that are the closest to a trigger. We search the quanta before the trigger first, and if there is none available, we search the money mention that is after the trigger. As for the result, the polarity is interpreted according to the first word that introduce a result. As shown in Table 4, there are word introducing the result polarity "accept" (*accepte*) and the polarity "stay of proceedings" (*sursis à statuer*). The others indicate that the claim was rejected.

The next phase is to match each claimed quantum to a result quantum. In our approach the strategy is simple: after matching the claim statement to the corresponding result statement, we match the quanta 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.

Training phase: Learning the best triggers that indicate statements or zones of interest Although we don't have token-level labeled dataset and labeled triggers as given usually in event extraction tasks such as in the ACE campaign [14], we still need to learn the triggers terms that indicates mentions of claims and results. Our aim is to find some terms that are usually close to the quanta of claims of a category of interest.

We use a cross-validation phase to determine the best trigger terms (and the claim and result zones weight thresholds if the zone weighting based strategy is used to select the statements). For each train-test phase, we used the training set documents of that phaseto rank the terms that occur more in statements of c_i than in other statements:

1. zone the claim statements around money mentions in the Litigation section: the phrases we mark up have the pattern : *claim introducing word + ... + money + ... (ending before the next introducing word or a point)*
2. zone the result statements around money mentions in the Dispositions section: the phrases we mark up have the pattern : *result introducing word + ... + money + ... (ending before the next introducing word or a point)*
3. mark up the not null quanta of the ground truth dataset (i.e. the tagged money mentions that are equal to that quanta).
4. combine the text content of the phrases containing a quantum in a new document to create a new corpus of the category c_i
5. combine the text content of the phrases that do not contain any quantum in a new document to create a new corpus of $\overline{c_i}$
6. rank the terms of the new corpus using a global weighting metric.

After ranking the terms, the learning approach then selects the terms by selecting them successively from the most important term to the less one.

Algorithm 1: Parameters learning algorithm for the extraction stage

Data: D^{train}, D^{test}, 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 SFS 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^{train};
  wt_{claim} = minimal weight of the quanta zones in the Litigation section of D^{train};
  wt_{result} = minimal weight of the quanta zones in the Dispositions section of D^{train};
  if *maxF1-score* $<$ *F1-score*$(Y_k + x, wt_{claim}, wt_{result}, D^{test})$ **then**
    $Y_{k+1} = Y_k + x_k$;
    $optThreshold_{claim} = wt_{claim}$;
    $optThreshold_{result} = wt_{result}$;
    max_F1-score = F1-score$(Y_k + x, wt_{claim}, wt_{result}, D^{test})$;
  else
    $Y_{k+1} = Y_k$;
until $X = \emptyset$ *or convergence of the F1-score*;
return Y_k, $optThreshold_{claim}$, $optThreshold_{result}$;

Then optimal term lists Y_k are merged over the cross-validation phases and the weights of the terms are averaged. The threshold weights are also averaged.

5 Experiments

5.1 Data

The data we used for experiments were labeled manually by a legal scholar. The dataset contains 610 claims extracted from 431 decisions listed per category as following:

1. category *doris* (*dommages-intérêts pour trouble de voisinage*): 103 claims extracted from 64 documents among which only 18 contain more than one claim (2 dcouments of 3 claims each, et 12 documents of 2 claims, et the remaining documents have resp. 12, 7, 5, 4 claims). Thus more than 90% documents have a single claim.
2. category *danais* (*dommages-intérêts pour procédure abusive*): 208 claims were extracted from 198 documents among which only 19 containing more than a claim (2 documents of 6, 3 documents of 4, 5 documents of 3 and 9 documents of 2 claims). Thus
3. category acpa (*amende civile pour procédure abusive*): 23 claims were extracted from 23 court decisions, thus a claim per document (100%).
4. category *concdel* (*dommages-intérêts pour concurrence déloyale*): 58 claims were extracted from 30 documents with 19 containing one claim (about 63 %), 5 documents with 2, 2 with 3, and the remaining documents with resp. 4, 5, 6, 8 claims.
5. category *styx* (*dommages-intérêts sur le fondement de l'article 700 du code de procédure civile*: 89 claimswere extracted from 50 court decisions with 12 containing a single claim (24%), 37 documents of 2 each et a document with 3 claim.
6. category *dcppc* (*déclaration de créance au passif de la procédure collective*): 129 claims were extracted from 91 court decisions with only 19 containing more than a claim (9 documents of 2 claims each, et 5 documents of 3, 3 of 4 et 2 documents of 6 claims).

5.2 Results

Learning relevant terms

c_i *vs.* $\overline{c_i}$

c_i *vs. Big set of unlabeled documents*

Quantum zones vs. non-quantum zones

A relevant list per section

Decision binary classification framework : category presence detection

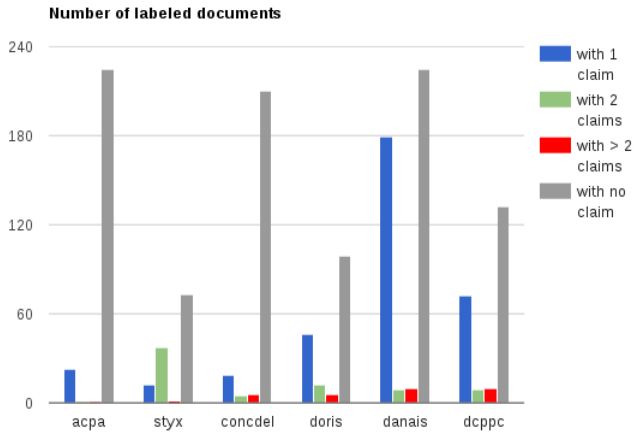

Fig. 5. Claims distribution over the labeled documents

Table 5. 5-fold cross-validation results (avg-F1) of the category presence detection phase

Category	Evaluation Metrics		
	Precision	Recall	F1-score
acpa	1.0	1.0	1.0
concdel	1.0	1.0	1.0
danais	0.993	0.992	0.992
dcppc	1.0	1.0	1.0
doris	1.0	1.0	1.0
styx	0.984	0.983	0.983

Information recognition stage

5.3 Errors analysis

6 Related works: event extraction from real world data

An event is an entity that occurs (in time) and that is complex by its informations (arguments or participants and attributes) [14]. Then, a legal claim may be considered as a legal event involving the claimants and the defendants as participants, the requested, the money requested, the money obtained, and the result's meaning as attributes. The shape of the training/testing dataset may enforce a certain kind of method. For example, event extraction problems are usually defined with a dataset of texts were informations are tagged directly in the sentences [14]. That suggests usually methods like sequence labeling [15, 16, 17].

Table 6. 5-fold cross-validation results (avg-F1) of the category presence detection phase

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

Table 7. 5-fold cross-validation results (avg-F1) of the recognition stage

	acpa		concdel		danais		dcppc		doris		styx	
Weight	R	F1	R	F1	R	F1	R	F1	R	F1	R	F1
CHI2	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
Δ DF	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

But, since labeling data might cost a lot of effort, time and money, it is better to do the minimum annotations possible. So, labeled information of a dataset might be separated from the source text (in a database or a Csv file for example), to make the labeling easier an. That is our case for claims information extraction. Moreover, the expert annotator might need to standardize the final information. This kind of data is more difficult to process since it becomes difficult to asses the location in the text of the information and the association of the extracted arguments with the suited event occurrence. [18] suggests a deep neural network based model to deal with this kind of dataset. Although the approach is still limited to the case of a single event per text, the attention mechanism is able to capture the arguments. So it remains to deal with the matching to solve the

Table 8. Recognition stage: 5-fold cross-validation results on some tuples of information with the best global weight metric of each category

	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

case of multiple events per text. Thus end-to-end approaches like [18] seem to be more suitable for our problem.

7 Conclusion

References

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

[2] Wu, H., Salton, G.: A comparison of search term weighting: term relevance vs. inverse document frequency. In: ACM SIGIR Forum. Volume 16., ACM (1981) 30–39

[3] Jones, K.S., Walker, S., Robertson, S.E.: A probabilistic model of information retrieval: development and comparative experiments: Part 2. Information processing & management **36**(6) (2000) 809–840

[4] 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

[5] 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

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

[7] 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

[8] Liu, Z., Li, M., Liu, Y., Ponraj, M.: Performance evaluation of latent dirichlet allocation in text mining. In: Fuzzy Systems and Knowledge Discovery (FSKD), 2011 Eighth International Conference on. Volume 4., IEEE (2011) 2695–2698

[9] Onan, A., Korukoglu, S., Bulut, H.: Lda-based topic modelling in text sentiment classification: An empirical analysis. Int. J. Comput. Linguistics Appl. **7**(1) (2016) 101–119

[10] 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
[11] 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)
[12] Lafferty, J., McCallum, A., Pereira, F.C.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. International Conference on Machine Learning (2001)
[13] 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
[14] Linguistic Data Consortium: ACE (Automatic Content Extraction) English Annotation Guidelines for Events. 5.4.3 edn. (2005)
[15] Yang, B., Mitchell, T.: Joint extraction of events and entities within a document context. arXiv preprint arXiv:1609.03632 (2016)
[16] Judea, A., Strube, M.: Event extraction as frame-semantic parsing. In: Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics (* SEM@ NAACL-HLT). (2015) 159–164
[17] Nguyen, T.H., Cho, K., Grishman, R.: Joint event extraction via recurrent neural networks. In: HLT-NAACL. (2016) 300–309
[18] Palm, R.B., Hovy, D., Laws, F., Winther, O.: End-to-end information extraction without token-level supervision. arXiv preprint arXiv:1707.04913 (2017)