31st March 2018

Reviewers of "Advances in Knowledge Discovery and Management", Vol. 8 (AKDM-8)

Dear Reviewers,

Thank you very much for the time and effort you put into reviewing our paper. We deeply appreciate your comments and suggestions.

In the aim of improving our paper, we have revised the manuscript by introducing the requested modifications.

Please find attached a point-by-point response to your comments. We trust that you'll find this improved version of our work to your satisfaction.

Sincerely yours,

Gildas Tagny Ngompe, Sébastien Harispe, Jacky Montmain, Guillaume Zambrano and Stéphane Mussard

Reviewer 1:

We greatly appreciate the reviewer's comments. Please find below point-by-point responses to your comments (Reviewer's comments are provided in italics and colored dark gray).

This is a very interesting and comprehensive work.

In the introduction, I was surprise that you don't explicitly mention the use of your approach to manage the problem of the setting of legal precedent (the act of "faire jurisprudence") while I imagine it could be a strong case of decision analysis.

- **Response:** We thank the reviewer for this suggestion. A specific comment about this particular use case has been added to the last paragraph of Section 2.1. We fully agree with you regarding this relevant case of decision analysis. One of our goals has in fact been to use extracted information for the purpose of studying the definition of models that could serve to predict court decision outcomes. Through document similarity estimation, we anticipate detecting commonalities between decisions that could subsequently be used to predict the results of similar unsolved cases.

In the broadest part of the related work section, authors could also mention the work on deontic logics and defeasible logics applied to normative requirements and legal rule formalization (e.g. LegalRuleML implementations).

- **Response:** We have actually considered that detecting norms (among other information) could help identify practical cases where legal rules have been applied. The Information Extraction techniques studied in our proposal might therefore indirectly contribute to improving the accuracy of rule-based reasoning. In this context, it is therefore definitely worthwhile to study how case-based reasoning can be combined with rule-based reasoning approaches, such as deontic and defeasible logics. We mentioned this consideration in the first paragraph of Section 2.1.

This also made me think that, if it is not already the case, the authors should participate (and submit) to the JURIX conference

- **Response:** We thank the reviewer for this suggestion. Conferences such as JURIX and ICAIL definitely attract the relevant audiences we are targeting.

You wrote "These elements typically appear in RTF, DOCX, or DOC documents for text formatting. They give no indication of the start of sections or other information. Someone might find interesting to exploit this rich formatting of DOC format to extract information but we didn't find it useful and we decided to work only at a plain text level to deal with less variations among texts."
I was surprised to read that you did not find formatting/styling indications useful for a task of sectioning. I was expecting this to be backed up by showing there uselessness in the feature selection experiment.

- **Response:** Formatting information could indeed be useful for several of the Information Extraction tasks. However, as we mentioned in the last paragraph of Section 4.1, unfortunately no formatting standard applies across all courts. The formatting style of a given document merely depends on the author or on the legal search engine used to retrieve the decisions. In addition, the documents retrieved from some search engines such as www.legifrance.gouv.fr are not provided with enhanced formatting (the reviewers are invited to refer to examples here: https://www.legifrance.gouv.fr/rechJuriJudi.do?reprise=true&page=1). Moreover, various document formats exist. It is therefore very difficult to establish a common pattern for formats and styles. More importantly, we sought to prevent defining models that would be highly dependent on information that was not always available.

In section "4.2 Create a training dataset" and then in section "5.1.1 Dataset" I was expecting some metrics on the training set and its quality e.g. size, evidences of inter-annotator agreement (inter-rater reliability, inter-rater agreement) as Cohen's Kappa, Krippendorff's Alpha.

- **Response:** We agree that information was lacking as regards the specific dataset aspects mentioned. In accordance with the reviewer's remarks, we have added the number of mentions for each entity type in Table 1, as well as more information on the number of lines and tokens for the dataset in Section 5.1.1. We also performed careful annotator inter-agreement evaluations to determine an agreement rate using Cohen's Kappa measure (please refer to Section 5.1.1).

In section "4.3 Define candidate features" I cannot help but wonder why these features are selected and more importantly among which set of possible features i.e. how the potential features can be identified as comprehensively as possible?

- **Response:** This is a complex question, and we are unsure whether a definitive answer can easily be provided. No *per se* standard and finite set of features actually exist; the candidate features may be either handcrafted or learned using an unsupervised method. In practice, features are typically defined based on state-of-the-art, trial and error, and modeler intuition through reliance on an analysis of patterns related to the entities of interest. In this context, it is difficult to claim that any definitive set of relevant features has been identified (as the number of potential features is indeed infinite). Nevertheless, in this vein, since we agree that questioning the relevance of incorporated features is crucial, a substantial portion of the paper covers the key topic of feature selection, i.e. how to select the best subset of features from a collection of candidate relevant features.

For instance why is the page number (number of the page where a token appears) not a feature?

- **Response:** The page number was not a feature because most entities not tied to a specific decision area unfortunately do not tend to appear on a specific page (based on prior experiments, even when the page number has been modified to a relative quantity, e.g. page_number / number_pages). As regards entities that are commonly located in specific decision areas (e.g. date, lawyers), our strategy relies on the use of dedicated extractors applied to relevant decision areas. Such relevant areas are provided by models dedicated to the sectioning task. By segmenting areas of interest first (where specific information tends to appear), we have sought, in some sense, to assist training highly accurate task-specific models.

Why did you ignore styling/formatting features?

- **Response:** As explained above, we prefer that our models avoid reliance on information inherent in writing style-related patterns since all future decision extractions may not contain this specific information.

How can we minimize the risk of missing a relevant feature?

- **Response:** Representational learning is a good way to capture the various aspects of the tokens. We therefore feel it important to use unsupervised methods (e.g. topic modeling, word2vec) in order to minimize the risk of missing relevant, hard-to-handcraft aspects of tokens. However, no guarantee exists that all relevant features or aspects will be captured. Moreover, handcrafted features are more easily interpretable than learned ones. Complementary information on this topic has been added to the latest version of the paper.

"Due to the number of experiments to run, we did a simple split of the data set into two subsets."
Does that mean you did not perform cross-validation like k-folding validation?

- **Response:** We conducted a random train-test split for the experiments discussed in Sections 5.2, 5.3 and 5.4. The detailed results (Section 5.5) however were obtained through 5-fold cross-validations.

Suggestions:

(Have the English proofread).

- **Response:** The latest version of the paper has been proofread by a professional English copy-editor.

"that a significant amount of decisions is available" -> "that a significant number of decisions are available"

- **Response:** We have rephrased to read "that many decisions are available online".

"(article 700 du code de procédure civile), or abbreviated (article 700 CPC), or along with some others (article 700 et 699 du code de procédure civile). »
You should translate into English or at least provide an English translation in addition.

- **Response:** All French phrases appearing are now translated.

"Authors Suppressed Due to Excessive Length" appears in the page headers.

- **Response:** Now corrected.

"that the more there is labels and entities to label" -> "that the more there ARE labels and entities to label"

- **Response:** Now corrected.

"the selection process still very long for both algorithms" -> "the selection process is still very long for both algorithms"

- **Response:** Now corrected.

"for testing. we tested only" -> "for testing. We tested only"

- **Response:** Now corrected (rephrased).

Reviewer 2:

We greatly appreciate the reviewer's comments. Below are our point-by-point responses (Reviewer's comments are provided in italics and colored dark gray).

This article presents the results of a study in machine learning applied to the processing of legal text (French court decisions). In particular, two tasks are approached at the same time: entity detection and section classification, using two well-established sequence-based learning mechanisms (CRF and HMM).

While it is clear that solving these tasks is an important step towards the automatic processing of legat text, there are a number of open issues in this work, that I will try to address in the following.

First and foremost, I don't get why two unrelated tasks are tackled using similar methods, features and evaluation procedures. It would make sense to solve the two problems of entity detection and section classification at once, if somehow they were mutually informative, which they probably are, but this information is not used here (e.g. disambiguated entities are not used as features for section classification, unless I missed something in the paper). Since instead the two tasks are solved separately, it seems a bit forced to fit them in the same "box".

- **Response:** This is indeed a very interesting point that we discussed at great length among our team. Based on our discussions, we decided not to force studied tasks to fit into the same box but rather apply them sequentially in a "pipeline" approach. The sectionning that we suggest may not only be useful for the tagging entity. Sectioning can indeed be useful on its own for indexing tasks related to the project that this research pursues (e.g. answering user queries by targeting a specific section). However, the reviewer is right, and we agree that both tasks may be mutually informative since all entities except norms are located in the header. We have separately trained a model to tag header entities with just the header contents and another model to tag norms appearing in other sections. Thanks to these advances, we might have obtained the same results using a joint inference model, in which section labeling and entity labeling are learned together. Nevertheless, we still need to compare the pipeline and joint-inference approaches in order to reach a definitive conclusion; such an interesting experiment is scheduled to be performed, as mentioned in the latest version of the paper.

For instance, the use of line-based features is counter-intuitive (it may work, but is it backed by relevant literature?) Entities are recognized at the word or multi-word expression level, while sections comprise fundamentally different atomic elements. Therefore, I don't see any parallel between the two tasks besides originating from the same kind of data.

- **Response:** For section detection purposes, we did not use word-based features in order to avoid having two words of the same line classified in different sections. In addition, we opted not to use sentence-based features because the sentences have not been clearly indicated (punctuation). The documents are composed of more statements than the actual number of sentences, and some statements are not always clearly separated. We therefore used line-based features given that the sections are more readily structured into lines than anything else. In proceeding, we were inspired by “*Pinto, David, et al. "Table extraction using conditional random fields." Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 2003”*. In this work, the authors used line-based features to extract tables from documents since tables are devoid of sentences. The proposed approach was a two-phase pipeline, in which the sections are detected first and entities labeled afterwards. A labeling model is set up for each task (i.e. we trained 3 labeling models). In this manner, it is possible to handle the two tasks using different atomic elements.

Regarding the methods, HMM and CRF are both valid ways of dealing with sequential data. This could definitely be a plus when dealing with sections, where the problem can be cast as detecting the transitions between subsequent sections, but they might not be the best tool for entity detection and classification. Moreover, there is a huge literature about named entity recognition and classification, entity linking, word sense disambiguation and related tasks, which is only marginally touched in the related work section.

- **Response:** This is a very interesting remark. Our paper focuses on using either HMM or CRF for entity and section detection. The purpose herein was to propose a baseline with established models like HMM & CRF. It is indeed true that a good deal of literature introduces methods that could be used for the tasks studied in our paper, e.g. hierarchical labeling or joint inference (to simultaneously train multiple mutually informative tasks), deep learning (LSTM, GRU, CNN) and representation learning, ensemble sequence labeling, model combination (mixed LSTM-CRF), and joint conditional probabilistic log-models, to cite just a few. We plan on studying the use of some of these models later on. Such models would indeed be candidates for close study and analysis subsequently, by considering: i) the datasets provided, and ii) CRF-based approaches as baselines within a broader survey context. The main purpose of this paper is in fact to provide a preliminary study that focuses on the various aspects related to the design of traditional yet robust text labeling models. In this context, it was important for us to first make use of widely used and approved techniques like HMM and CRF. We'll ensure that this study is extended with a review of more recent and currently popular methods and techniques.

The selection of features is kind of arbitrary. Undoubtedly a great deal of work went into this part, but at the same time, again, there is a lot of published work in computational linguistic and NLP venues on what features work best for tasks such as the ones at hand. There exists readily available feature extractors and processing pipelines, and methodologies for automatic feature engineering. What seem to be the most popular word-based feature nowadays is missing, that is, some kind of work embedding (e.g., LSA or word2vec vectors). Those are easy to extract from unannotated text and have been employed in a variety of tasks. Topic modeling is a similar concept, but is perhaps better suited for document classification than word representation.

- **Response:** We thank the reviewer for this suggestion. We'll be experimenting with these features during future work. The list of features tested in any single paper cannot be exhaustive; hence, the proposed approach is meant to be compared with other solutions and extended by incorporating other tagging algorithms and features.

The lack of comparison with established methods is also a problem for the evaluation. Since there is no context, not even a comparison against a trivial method to use as a baseline, it is difficult to interpret the numeric results of the evaluation. An upper bound is also missing, due to the fact that almost no information is provided about the gold standard dataset manually created for evaluation purposes. How big is the data set in term of words and sentences?

- **Response:** We added the average number of lines and words per document in the section describing the dataset (Section 5.1.1). The labeling step has been performed on the entire document for the sectioning task and on the entire section content for the NER task. We did not proceed at the sentence level since the sentences are not clearly separated (especially in the header part of the document). A discussion related to the use of other methods has been inserted above. Regarding the evaluation of the upper bound, as mentioned when answering the other reviewer's remark, we also performed careful annotator inter-agreement evaluations in order to provide an agreement rate using Cohen's Kappa measure (please refer to Section 5.1.1).

How many entities and sections are there?

- **Response:** We have extended Table 1 by adding the median and total numbers of mentions in the dataset. Regarding section analysis, each document was manually split into 3 sections (Header, Body and Conclusion).

What was the annotation procedure like and what was the agreement between annotators?

- **Response:**
    - Our annotation procedure was quite simple and traditional, in accordance with standards practiced in the field. The annotator must simply label any mention of the entities using the suitable tag, as described in Table 1.
    - A second annotator performed another annotation of a subset of our dataset. Based on the two subset annotations, we evaluated the inter-annotator agreement rate using Cohen's Kappa measure (Section 5.1.1). Moreover, we added detailed information regarding the dataset in both Table 1 and Section 5.1.1.

Some of the tabled results are a bit confusing, such as "entity-level F1" scores incredibly low. Perhaps reporting the full figures, i.e., including precision and recall scores, would help understanding the performance of each configuration.

- **Response:** We included precision and recall scores in Tables 3 and 4. The low entity-level F1 scores you are referring to might be those obtained with the HMM for the sectionning task. These scores are low because the HMM generally misclassifies a few lines at the transition between sections. Due to this observation, many sections are not fully detected; consequently, the entity-level measures wind up being low.

*The presentation is generally good, with some exception. Section 2.2 is sometimes a bit vague (e.g. "Machine learning based systems..." with no reference) and there are occasional English mistakes, e.g. "such list can be crafted by experts or *learn[ed] from ..." or "rules-based systems".*

- **Response:** We enriched the literature of Section 2.2 and corrected numerous mistakes. Also please note that the latest version of the paper has been proofread by a professional English copy-editor.

In conclusion, this is a good paper as far as the effort towards legal informatics is concerned, with the big caveat of a somewhat weak experimental evaluation. The other principal issue is that this work is not on par with the modern standards in NLP, the field from which the method is taken. In case of rejection, the author could consider updating their method by including more recent techniques, while keeping the focus on the same data and tasks.

- **Response:** We thank the reviewer for this suggestion; this work will certainly be followed by studies on the use of recent models and findings in NLP involving datasets and meaningful CRF baselines provided by the proposed paper.

Detecting sections and entities in court decisions using HMM and CRF graphical models

Gildas Tagny Ngompé, Sébastien Harispe, Guillaume Zambrano, Jacky Montmain, Stéphane Mussard

Abstract Court decisions are legal documents that undergo careful analysis by lawyers in order to understand how judges make decisions. Such analyses can indeed provide invaluable insight into application of the law for the purpose of conducting many types of studies. As an example, a decision analysis may facilitate the handling of future cases and detect variations in judicial decision-making with respect to specific variables, like court location. This paper presents a set of results and lessons learned during a project intended to address a number of challenges related to searching and analyzing a large body of French court decisions. In particular, this paper focuses on a concrete and detailed application of the HMM and CRF sequence labeling models for the tasks of: i) sectioning decisions, and ii) detecting entities of interest in their content (e.g. locations, dates, participants, rules of law). The effect of several key design and fine-tuning features is studied for both task categories. Moreover, the present study covers steps that often receive little discussion yet remain critical to the practical application of sequence labeling models, i.e.: candidate feature definition, selection of good feature subsets, segment representations, and impact of the training dataset size on model performance.

1 Introduction

A court or judicial decision may be defined as either the judges' final decision at the end of a trial or a document containing the case description, i.e. judges' decision and motivations. The latter definition will be considered herein. This article

Gildas Tagny Ngompé, Sébastien Harispe, Jacky Montmain
LGI2P, Ecole des mines d'Alès, e-mail: gildas.tagny-ngompe@mines-ales.fr, sebastien.harispe@mines-ales.fr, jacky.montmain@mines-ales.fr

Guillaume Zambrano, Stéphane Mussard
CHROME EA 7352, Université de Nîmes e-mail: guillaume.zambrano@unimes.fr, stephane.mussard@unimes.fr

will discuss the detection of sections and entities in French court decisions. These decisions are semi-structured digital documents that share the same overall format as defined by three sections: the *header*, the *body*, and the *conclusion*. Each section encompasses specific information regarding a case: i) the header contains numerous metadata (e.g. date, court location, names of involved persons); ii) the body details facts, previous legal proceedings, parties' arguments and judges' arguments; and lastly iii) the conclusion summarizes judges' final decisions. Even though all decisions follow such a general layout, the format inside the individual sections may differ. Since information appears in a section according to its type, our initial aim is to detect each section, by means of segmenting the decision. We assume here that sectioning the decision would simplify the entity extraction process. We also expect that sectioning will aid in other tasks, such as extracting claim-related information. This work focuses in particular on detecting entities like the date when the decision was pronounced, the type of court, its location, and the names of the judges, parties and their lawyers. Table 1 lists the entities being targeted and provides some examples of how they appear in French court decisions.

Entities	**Tags**	**Examples**	**#mentions**[a]	
			Median[b]	**Total**[c]
Registry Number	**rg**	"10/02324", "60/JAF/09"	3	1318
City	**ville**	"NÎMES", "Agen", "Toulouse"	3	1304
Type of court	**juridiction**	"COUR D'APPEL"	3	1308
Division in the court	**formation**	"1re chambre", "Chambre économique"	2	1245
Date	**date**	"01 MARS 2012", "15/04/2014"	3	1590
Appellant	**appelant**	"SARL K.", "Syndicat ...", "Mme X ..."	2	1336
Respondent	**intime**	- // -	3	1933
Intervenor	**intervenant**	- // -	0	51
Lawyer	**avocat**	"Me Dominique A., avocat au barreau de Papeete"	3	2313
Judge	**juge**	"Monsieur André R.", "Mme BOUSQUEL"	4	2089
Judge's Function	**fonction**	"Conseiller", "Président"	4	2062
Norm or legal rule	**norme**	"l' article 700 NCPC", "articles 901 et 903"	12	7641
non-entity	**O**	*words outside any targeted entity*	-	-

[a] number of entity mentions in the labeled dataset we used for experiments
[b] median number of occurrences or mentions per document in the dataset
[c] total number of occurrences in the dataset

Table 1: Entities and corresponding tags used to label their words.

This study analyzes the application of two labeling graphical models, namely HMM (Hidden Markov Model) and CRF (Conditional Random Fields), for the tasks of detecting sections as well as legal named entity mentions. Both these tasks are handled by undertaking the information extraction challenge known as sequence labeling. The idea herein is to split a text into tokens, in such a way that the object of interest (section or entity in our case) contains one or multiple tokens. Next, a labeling model labels the tokens using the suitable entity tag.

Over the remainder of this paper, Section § 2 complements our work by introducing several challenges that rely on an analysis of court decisions; it will also be demonstrated how information extraction can help address these issues. Afterwards, Section § 3 will discuss the two graphical models studied in depth within this paper, i.e. HMM and CRF, while Section § 4 will review technical details relative to the detection of sections and entities in French court decisions. Section § 5 will present the empirical evaluations performed and share the set of results recorded. Lastly, Section § 6 will conclude the paper by highlighting our main findings and offering an outlook for future research.

2 Court decisions analysis: challenges and existing work

2.1 Challenges associated with French court decisions analysis

Judicial decisions are essential for legal practitioners. More specifically, lawyers are accustomed to researching and analyzing decisions in order to solve the problems at hand or to advise their clients. Decision analysis can indeed provide invaluable insight into potential applications and studies. As an example, a decision analysis may be conducted for the purpose of handling future cases, mainly because justice is a complex matter and its language is barely understandable to non-lawyers (Cretin, 2014), hence allowing them to assess the legal risk of their actions without requiring the assistance of an expert. Such an analysis might also help to detect variations in judicial decision-making in considering specific variables such as time and location. A critical need therefore exists for automatic tools that can exhaustively analyze application of the law. The next step pertains to leveraging the current body of decisions so as to evaluate and even predict judicial decision-making? This capability is of great interest to several companies, such as LexisNexis with its Lex-Machina[1] system. New French startups like Predictice[2] and Case Law Analytics[3] are also investigating these avenues. The manual analysis of an exhaustive body of decisions is a very demanding task, maybe even impossible, given that courts issue many decisions (over 2 million in France every year[4]). Legal experts typically encounter two main obstacles: (i) identifying a collection of decisions of interest regarding a specific topic; and (ii) analyzing the targeted collection of documents. Despite the fact that many decisions are available online, searching for them from a large pool remains difficult due to the limitations of current legal search engines, which merely propose simple search criteria like keywords. Extracting useful in-

[1] https://lexmachina.com

[2] http://predictice.com

[3] Http://caselawanalytics.com

[4] http://www.justice.gouv.fr/budget-et-statistiques-10054/chiffres-cles-de-la-justice-10303/

formation from decisions would improve document description and organization. Based on such information, it would also be possible to extend these search criteria with simple ones (e.g. judges' names, rules) or semantic ones (e.g. type of case or claim). The extracted information might therefore be helpful for both the identification and analysis of a body of decisions of interest. Note the potential for many other applications as well. For example, extracting legal entities is not only very useful for enriching text content or constituting a legal knowledge base, but also for "anonymizing" legal texts in order to ensure confidentiality (Plamondon et al., 2004). Moreover, detecting norms is useful to the identification of practical cases where rules had been applied, which in turn may make rule-based reasoning more accurate. It is definitely worthwhile to study how case-based reasoning can be combined with rule-based reasoning approaches, including the application of deontic and modal defeasible logics (Lam et al., 2016).

Natural language processing and text mining techniques enable an automatic document analysis that mitigates the barriers of data quantity, domain complexity and language. For example, McCallum et al. (2000) designed a system for entity recognition and text classification in structuring a large collection of scientific articles to facilitate their search. As for the legal domain, we are currently designing an automated approach that gives rise to an exhaustive, descriptive and predictive analysis of the jurisprudence. This analysis requires structuring the corpus of decisions first according to their characteristic information: registration number in the general directory (RG), court, city, date, judges, legal rules (norms), parties' claims and the requested amounts involved (e.g. damages, length of prison sentence), the corresponding response from judges (*accept* or *reject*), and the amounts actually awarded. The formalization of information and relations (e.g. a norm supporting a claim) serves to semantically describe and organize decisions into a knowledge base. The fundamental objective of our project is to extract information from court decisions, with such information needing to be formalized in order to build a jurisprudence knowledge base. Many useful applications rely on this kind of knowledge base: understanding how laws are applied, anticipating the decision-making of courts, searching similar decisions, analyzing and comparing the legal risks for given time periods and locations, identifying the factors correlated with judges' decisions, and identifying those decisions to be considered as a reference for a particular type of case (i.e. establishing the legal precedent). The construction of such a judicial knowledge base requires a description of the individual decisions. These documents are freely written texts yet with a certain level of structure. The various types of information of interest they contain entail different knowledge discovery tasks. For example, the extraction of locations, dates, individual names and legal rules (norms) is similar to named-entity recognition (NER), a task widely studied in natural language processing (Marrero et al., 2013) through several competitions, such as CoNLL NER shared tasks (Tjong Kim Sang and De Meulder, 2003) and the Ester 2 information extraction task (Galliano et al., 2009). Many works also exist for different languages, including Chinese NER (Wu et al., 2003) and French NER (Tellier et al., 2012). Other tasks however, such as claims information extraction, require other methods. Since this article is focusing on information detection from court decisions, the previous works addressing similar tasks will be discussed in the next subsection.

2.2 Information detection in court decisions

Four distinguishable entity detection approaches have been identified (Chau et al., 2002):

- Lexical lookup systems are designed based on a list of previously known entities, along with their synonyms within the domain of interest. For instance, in the legal domain, a lexicon may contain the legal rules and judges' names. The list of entities may be handwritten by experts or learned from a labeled dataset (training phase); however, it proves to be very difficult to maintain such a list because the domain might be changing regularly (new laws). Moreover, entity mentions may have several variants. For example, the same rule "Article 700 of the Civil Procedure Code" might appear alone fully cited (*article 700 du code de procédure civile*), abbreviated (*article 700 CPC*), or combined with other rules like in "Articles 700 and 699 of the Civil Procedure Code" (*articles 700 et 699 du code de procédure civile*). Such issues, including ambiguities (e.g. different entities using the same words), had limited early systems (Palmer and Day, 1997).

- Rule-based systems are built on domain-specific rules that sufficiently describe contextually, structurally or lexically the diversity of entity mentions. These are advantageous because their errors are easily explained, yet manually defining the rules involved requires considerable effort, in particular for a large body of decisions. Furthermore, a given set of rules may not always be reused in other domains. However, a number of adaptive rule-based approaches serve to overcome these issues, while still benefiting from the "explicability" of rule-based systems (Siniakov, 2008; Chiticariu et al., 2010).

- Statistical systems adapt statistical language models, typically from text compression methods in order to detect entities. For instance, Witten et al. (1999) adapted the Prediction by Partial Matching compression schemes for NER.

- Machine learning-based systems run text segment multi-class classifiers. For example, the traditional Naive Bayes text classifier was trained to detect gene mentions (Persson, 2012) by classifying tokens, given a manually-defined feature set. Sequence labeling algorithms, such as the CRF (Finkel et al., 2005) also classify text segments by modeling the transitions between token labels. More recently, deep learning architectures are achieving the best results on multiple information extraction tasks, including NER (Lample et al., 2016).

Some works have combined various approaches to extract entities from legal texts, e.g. by describing contextual information using rules to address the ambiguity issue of the lexical lookup method (Mikheev et al., 1999; Hanisch et al., 2005). Moreover, after segmenting the documents with a CRF-based model, Dozier et al.

(2010) combined multiple approaches in order to recognize entities in U.S. Supreme Court decisions. They defined separate rule-based detectors to identify the jurisdiction (geographical area), type of document, and judges' names, in addition to introducing a lexical lookup for detecting the court and a trained classifier for the title. These detectors showed promising results albeit with limited recalls of between 72% and 87%.

The HMM and CRF models studied in this paper have also been used for purposes of legal entity recognition. As an example, the HMM were compared with the Perceptron Algorithm with Uneven Margins (PAUM) (Li et al., 2002) for the task of recognizing institutions and references in other decisions and judicial act mentions (law, contract, etc.) in Czech court decisions (Kríž et al., 2014). Both models yield good results, with F1-scores of 89% and 97% for the HMM using trigrams as features and F1-scores of 87% and 97% for the PAUM using the 5-gram lemmas and words part-of-speech. (Cardellino et al., 2017), on the other hand, used CRF and neural networks for legal named entity recognition. The poor results they reported for the recognition in rulings confirmed that legal NER is indeed a difficult task. Nevertheless, the entity-linking approach they proposed might be quite powerful in disambiguating entities for our study. The work herein focuses on flat HMM and linear chain CRF models, as described in the following section. Although flat CRF or HMM models are generally trained to detect entities, hierarchical methods might also be worth studying since they can jointly learn to detect sections and entities with a multi-layer model (Surdeanu et al., 2010).

3 Labeling text using HMM and CRF models

Let's now consider a text (decision) T as a sequence of observations t1:n, with each ti being a segment of text (word, line, sentence, etc.). In considering a collection of labels, labeling T consists of assigning the appropriate labels to each ti. A segmentation task of T entails splitting T into non-overlapping groups (i.e. partitions), such that the elements of a group necessarily constitute a subsequence of T. In other words, segmenting T corresponds to labeling it in considering a specific constraint.

3.1 Hidden Markov Models (HMM)

An HMM is a finite-state machine with a set of states $\{s_1, s_2, ..., s_m\}$ that intends to assign a joint probability $P(T,L) = \prod_i P(l_i|l_{i-1})P(T|l_i)$ to pairs of observation sequences $T = t_{1:n}$ and labels $L = l_{1:n}$. Since an HMM is a generative model, each label l_i corresponds to the state s_j in which the machine has generated observation t_i. There are as many possible labels as there are states. The labeling process of T consists of determining the best label sequence L^* that maximizes the joint probability ($L^* = \underset{L}{\operatorname{argmax}} P(T,L)$). An evaluation of all possible label sequences is necessary to determine the one that best fits T. To avoid the exponential complexity $O(m^n)$ of this approach, with n being the sequence size and m the number of possible labels, the labeling process generally uses the Viterbi decoding algorithm (Viterbi, 1967), which is based on dynamic programming. This algorithm browses the text from t_1 to t_n while searching for the state path (label sequence) with the best score at each position i of T (i.e. the highest probability $P(t_{1:i}, l_{1:i})$). This algorithm employs HMM parameters that have been estimated from a training sample of annotated texts:

- A set of states $\{s_1, s_2, ..., s_m\}$ and an alphabet $\{o_1, o_2, ..., o_k\}$
- The probability that s_j generates the first observation $\pi(s_j), \forall j \in [1..m]$
- The transition probability distribution $P(s_i|s_j), \forall i, j \in [1..m]$
- The emission probability distribution $P(o_i|s_j), \forall i \in [1..k], \forall j \in [1..m]$

The transition and emission probabilities can both be inferred using a maximum likelihood estimation method, such as the expectation maximization algorithm. The Baum-Welch algorithm (Welch, 2003) is a specification designed especially for HMM. The advantage of HMM lies in its simplicity and training speed. On the other hand, it is difficult with HMM to represent multiple interactive features for text elements as well as to model the level of dependence between distant observations because the hypothesis of independence between observations is highly restrictive (i.e. the current state depends solely on the previous states and the current observation). Rabiner (1989) provided further details about HMM for interested readers.

3.2 Conditional random fields (CRF)

Even though the Viterbi algorithm is also used to apply CRF to text labeling, the CRF and HMM structures still differ. Rather than maximizing the joint probability P(L, T) like in HMM models, a CRF (Lafferty et al., 2001) searches for the sequence of labels L^* that maximizes the following conditional probability:

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

where Z is a normalization factor. The potential functions $f(\cdot)$ are the features handled by CRF models. Two types of feature functions can be identified: transition features, which depend on the labels at the previous and current positions (l_{i-1} and l_i respectively) and on T; and state features, which are functions of l_i and T. These functions $f(\cdot)$ are defined with either binary or real-valued functions $b(T, i)$ that combine the descriptors of a position i within T (Wallach, 2004). In order to label legal rules, a CRF model may include, for example, the following potential functions for labeling "*700*" in this context "... *l'article 700 du code de procédure civile...*" (i.e. "... Article 700 of the Civil Procedure Code ..."):

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

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

with

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

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

with t_i being an observation in T, POS the part-of-speech of t_i (NUM = numerical value, NOM=noun), and where NP1 and NS1 denote the lemma of nouns before and after t_i, respectively. The symbols <*unknown*> and @*card*@ stand for encoding unknown lemmas and lemmas of numbers, respectively. Since the two functions f_1 and f_2 can be activated at the same time, they define overlapping features. With multiple activated functions, the belief in $l_i = \text{NORME}$ is boosted by the sum of the weights of the activated functions $(\lambda_1 + \lambda_2)$ (Zhu, 2010). A CRF model employs a function $f_j(\cdot)$ when its conditions are met and $\lambda_j > 0$. The various weighted features $f(\cdot)$ are defined with the descriptors characterizing the text and the labels from the training dataset. The training phase consists mainly of estimating the parameters vector $\lambda = (\lambda_1, ..., \lambda_F)$ from previously annotated texts $\{(T_1, L_1), ..., (T_M, L_M)\}$ where T_k is a text and L_k the corresponding label sequence. The optimal λ value maximizing the conditional likelihood of the objective function $\sum_{k=1}^{M} \log P(L_k|T_k)$ on the training data is retained. In general, this estimation strategy is based on the gradient of the objective function and uses it in an optimization algorithm such as L-BFGS (Liu and Nocedal, 1989).

The following section will discuss how we handled the different particularities of the documents treated using descriptors in order to design a tagging system.

4 Detecting sections and entities in French court decisions

4.1 Specificities of court decisions

The analysis of court decisions reveals a structure with three sections presented in a specific order, namely: the header metadata (*entête*); the body of the decision, which comprises the litigation details and the motivation behind judges' decision (*corps*); and lastly a brief conclusion of their decision (*dispositif*). The division of decisions into sections might serve to better organize the information extraction tasks. An intuitive approach would call for defining an algorithm capable of recognizing the transitions between sections through the use of regular patterns. However, transition markers are not standardized and have many variants; in some cases, they are either titles or symbols (asterisks, hyphens, etc.) or else nothing at all. Also, the explicit transitions remain quite heterogeneous. For example, the transition from the header to the body can be indicated by the headings "*Exposé*", "*FAITS ET PROCÉDURES*", "*Exposé de l'affaire*", etc. As regards the conclusion, it usually begins with the keyword *PAR CES MOTIFS* (On these grounds), sometimes with simple variants (e.g. "*Par Ces Motifs*") or more exceptional ones ("*P A R C E S M O T I F S :*"). Other expressions can also be found in decisions ("*DECISION*", "*DISPOSITIF*", "*LA COUR*", etc.). The same patterns of special characters, such as "*" or "-", often separate the sections and subdivide a section within the same document.

The same types of variability appear for entities. Parties and lawyers are often placed after a particular keyword, like "*APPELANTS*" or "*DEMANDEUR*" for appellants, "*INTIMES*" for respondents, and "*INTERVENANTS*" for intervenors. The names of individuals, companies and cities begin with a capital letter or are entirely in uppercase. Yet other common words may appear in uppercase as well, for instance the titles of certain fields (e.g. *APPELANTS*, *DÉBATS*, *ORDONNANCE DE CLÔTURE*). They could contain numbers, such as registry numbers and dates, and often include punctuation marks (e.g. "/"), initials and abbreviations. The lines containing entities are usually observed in the same order (i.e. *appellants* before *respondents*, *respondents* before *intervenors*). However, many types of entities appear in headers, unlike the other two sections, in which norms are the only entities of interest. The header is more structured than the other sections, although its structure may differ between any two decisions.

When collecting court decisions, documents are available in various formats, including .rtf on www.legifrance.gouv.fr, .doc(x) and .txt on the LexisNexis website, where we retrieved the dataset documents used in this study. Each document downloaded from LexisNexis contains one or more decisions. Their textual content has been extracted by removing unnecessary elements like continuous invisible characters and blank rows. These elements typically appear in .rtf or .doc(x) documents for text formatting purposes; they provide no indication of the beginning of sections or any other information for that matter. Enhanced formatting may be targeted to extract information, but no formatting standard has been established from one court jurisdiction to the next. We have decided to concentrate on plain text in order to cope with fewer variations among texts while applying the same processing procedure on documents regardless of their origin or formatting. A simple architecture for section and entity detection system has been designed (Figure 1) based on these observations. The documents are first collected and preprocessed according to their format. Then, after sectioning the decisions, the entities are identified by the structure of the sections where they were mentioned. The following subsections will discuss some design aspects to take into account in order to generate good results from such a system.

4.2 Training dataset creation

Since HMM and CRF are both supervised models, they should be trained on examples in order to estimate their parameters. A sufficient set of decisions must therefore

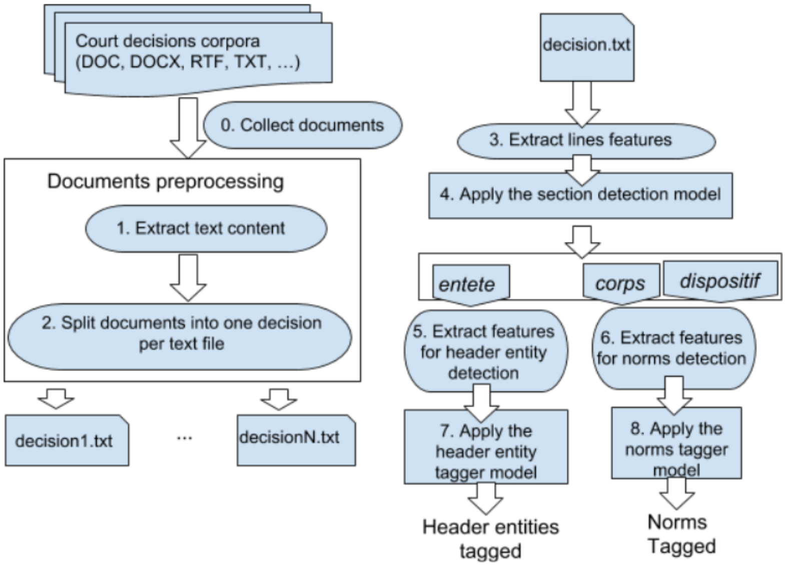

Court decisions corpora
(DOC, DOCX, RTF, TXT, ...)
0. Collect documents
Documents preprocessing
1. Extract text content
2. Split documents into one decision per text file
decision1.txt
...
decisionN.txt
decision.txt
3. Extract lines features
4. Apply the section detection model
entete
corps
dispositif
5. Extract features for header entity detection
6. Extract features for norms detection
7. Apply the header entity tagger model
8. Apply the norms tagger model
Header entities tagged
Norms Tagged

Fig. 1: Applying trained taggers: After collecting and preprocessing the documents, the section line tagger is firstly applied then the named entity taggers can be applied simultaneously in the different sections.

be selected and annotated by labeling their sections and entities. In the present case, annotations are provided in XML format. The objects of interest are annotated manually, a step that requires considerable human effort and precision. To speed and improve the work of human annotators, the annotation protocol presented in Quaero (Rosset et al., 2011) has been defined with a set of specific guidelines that notify: the text type to be chosen, the labels to be used (and when to use them), and the treatment to be applied in special cases (Petrillo and Baycroft, 2010). Software tools are also available to assist with the annotation process by using a mouse to highlight segments of interest instead of the manual typing of tags. As an example, GATE Teamware (Bontcheva et al., 2013) has been involved in a collaborative annotation process of a body of legal work (Wyner and Peters, 2012).

4.3 Candidate features definition

4.3.1 Candidate features of lines for section detection

Let's now consider the line to be labeled during section detection. We have avoided word-based features in order to prevent words from the same line to be classified in different sections. We chose not to proceed at the sentence level given the lack of clear sentence separation (especially in the header part of the document). Several criteria may be used to differentiate sections, i.e.: the length of the lines (longer in the body, shorter in the header), the first terms of certain lines (typical to each section), and the total number of lines. An HMM only accommodates one descriptor assimilated with the element to be labeled. Other descriptors might be the position of the element to be labeled (line number) or the beginning of the line. A feature capturing the line length may be either absolute (the exact number of words in the line) or relative, depending on a line length categorization. Based on the line length distribution quantiles over a body of decisions, we have defined three categories: *LQ1* ($length \leq 5$), *LQ2* ($5 < length \leq 12$), and *LQ2* ($12 < length \leq 14$). We have also categorized the parts of documents in order to capture a relative line position. During the feature extraction, the document is considered to be split into N parts (10 in our experiments). The relative position of a line is thus the number of the part containing the particular line. In sum, the features are described as follows (with their labels in parentheses):

- line shape: the entire line (*token*), its first words (*t0, t1, t2*), absolute length (*absLength*), and relative length (*relLength*);
- line context: the line number (*absNum*) and number of the document part containing the line (*relNum*), the first two words of the previous (*p0, p1*) and subsequent lines respectively (*n0, n1*), and their respective absolute and relative lengths (*pLength, pRelLength, nLength, nRelLength*).

4.3.2 Candidate features for entity mentions detection

Entity detection consists of training either a CRF or an HMM to label the various entities (word, punctuation, number, identifier). Both models necessitate certain features, some of which may be handcrafted based on patterns observable in the texts. It is also possible to obtain other features from the output of other text analysis tasks.

Handcrafted features based on observations: Based on decision observations, we have defined the following spelling-based features for words of both norms and entities in the headers (with their names in parentheses):

- word shape: the word (token), its lemma (*lemma_W0*), "Does it begin with a capital letter?" (*startsWithCAP*), "is it entirely capitalized?" (*isAllCAP*), "is it a lone initial?" like for instance "B." (*isLONELYINITIAL*), "does it contain a punctuation character?" (*PUN-IN*), "is it all punctuation?" (*isALLPUN*), "does it contain a digit character?" (*DIGIT-IN*), "are there just all digits?" (*isALLDIGIT*);
- word context: the previous and subsequent words, i.e. the 4 neighbors (*w-2, w-1,w1,w2*) and their lemmas(*lemmaW$_i$*),

The lemmatization step homogenizes variants of the same word. The adjacent words are chosen to emphasize those words commonly used to mention entities.

Most notably for headers, we have defined additional features to capture the word context: line number (*lineNum*), position of the element in the line (*numInLine*), "does the text contain the keyword intervenant?" (*intervenantInText*), does the text come after the keyword "APPELANT" (*isAfterAPPELANT*), "INTIME" (*isAfterINTIME*), "INTERVENANT" (*isAfterINTERVENANT*). We also considered the last lines, where the token was previously encountered in the text (*lastSeenAt*), and the number of times it was found (*nbTimesPrevSeen*), because the parties' names are often repeated at different locations. We also defined a special feature for norms: "is the token a keyword of legal rules?" (*isKEYWORD*). For this latter descriptor, we drew up a short list of keywords typically used to cite legal rules (*article, code, loi, contrat, décret, convention, civil, pénal*, etc.).

Extending features: The notion here is to use the labels from other tasks as features in our models. Let's consider the part-of-speech and word topic:

Part-of-speech tagging: The part-of-speech (POS) tagging identifies the part-of-speech of given words. Some works often use POS tags as features when some entities tend to contain particular parts-of-speeches. For example, the names of individuals are composed of proper nouns (Chang and Sung, 2005). We extracted the POS tag of the current token (*POS*) as well as that of its neighbors (*POSW-2, POSW-1, POSW1, POSW2*).

Topic modeling: Like Polifroni and Mairesse (2011) and Nallapati et al. (2010), we employ word-topic associations in order to describe our words. The basic idea here is to model a set of N_{topics} of topics and use their IDs as features. It might be worthwhile to make use of the probability inferred from the topic model, but the inference underlying the LDA model (Blei et al., 2003) is not deterministic (the probability distribution changes for the same word when running several inferences). Nevertheless, since the topic order does not significantly change, we used the ID of the more relevant word topic (*topic0*) as well as that of its neighbors (*w-2topic0, w-1topic0, w1topic0, w2topic0*).

4.4 Selecting the most relevant feature subset

CRF features: After defining a number of candidate features, nothing will ensure that combining all of them leads to the most optimal performance. The aim then is to compose the smaller feature subset leading to the best result. We studied two wrapper strategies that always seem to converge and that do not require manually defining the size of the target subset.

The bidirectional search (BDS) (Liu and Motoda, 2012) runs the Sequential Forward Selection (SFS) and the Sequential Backward Selection (SBS) in parallel. The SFS seeks an optimal subset, in beginning with an empty set and adding a feature at each iteration as it increases the information criteria (i.e. the objective function) of the selected subset. The objective function is the macro-averaged F1-measure at the token level. In the same manner as SBS, it begins with the full set of features and removes the worst one that serves to decrease the macro-averaged F1-measure. The features being either added or removed must not be those that SBS removed or added before.

Algorithm 1: Bidirectional search algorithm

```
   Data: annotated dataset, $X$ list of all the candidate features
   Result: optimal feature subset
1  Start SFS with $Y_{F_0} = \emptyset$;
2  Start SBS with $Y_{B_0} = X$;
3  $k = 0$;
4  while $Y_{F_k} \neq Y_{B_k}$ do
5      $x^+ = \underset{x \in Y_{B_k} \setminus Y_{F_k}}{\text{argmax}}\ F1measure(Y_{F_k} + x)$;
6      $Y_{F_{k+1}} = Y_{F_k} + x^+$;
7      $x^- = \underset{x \in Y_{B_k} \setminus Y_{F_{k+1}}}{\text{argmax}}\ F1measure(Y_{F_k} - x)$;
8      $Y_{B_{k+1}} = Y_{B_k} - x^-$; $k = k + 1$;
9  return $Y_{F_k}$;
```

Instead of running SFS and SBS in parallel, the Sequential Floating Forward and Backward Selection models (SFFS and SFBS) (Pudil et al., 1994) correct their limitations separately. To overcome the inability of SBS to reevaluate the utility of a feature after being discarded, the SFBS performs forward steps as the objective function improves upon each backward step, while the SFFS performs backward steps upon each forward step. The SFFS has been experimented with in this study.

Algorithm 2: Sequential Floating Forward Selection

```
    Data: annotated dataset, $X$ list of all the candidate features
    Result: optimal feature subset
1   Start SFS with $Y_0 = \emptyset$;
2   $k = 0$;
3   repeat
4       $x^+ = \underset{x \notin Y_k}{\text{argmax}}\, F1measure(Y_k + x)$;
5       $Y_k = Y_k + x^+$;
6       $x^- = \underset{x \in Y_k}{\text{argmax}}\, F1measure(Y_k - x)$;
7       if $F1measure(Y_k - x^-) > F1measure(Y_k)$ then
8           $Y_{k+1} = Y_k - x^-$; $X = X - x^-$; $k = k + 1$;
9           ; Go to step 6;
10      else
11          Go to step 4;
12  until $X = \emptyset$ or $X = Y_k$;
13  return $Y_k$;
```

Algorithms 1 and 2 describe BDS and SFFS implemented for this study.

HMM features: To select the best features for the HMM models, we tested the various candidates one after the other. The feature yielding the best result on the annotated dataset is thus selected.

4.5 Select the segment representation

We are dealing herein with many multi-word entities of various kinds (e.g. l'article 700 du code de procédure civile). To increase the performance of a tagger model, some parts of the entities could be emphasized through a suitable segment representation. We have studied the effects of some of the segment representations described in Konkol and Konopík (2015). The IO model does not emphasize any particular part of the entity and assigns the same label to all entity tokens. Other models distinguish either the first token of the entity (BIO), or the last one (IEO) or both (BIEO). Figure 2 illustrates these tagging models on a test text segment. The best segment representation is associated with the best F1-measure.

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

Fig. 2: Example of text labeling using different segment representations

5 Experiments and results

This section will describe the experimental protocol and discuss the results. More specifically, these results pertain to: the selection of segment representations, feature subsets, and the assessment of an expected improved performance through annotating more training data. Moreover, the results will be evaluated for each type of section and entity.

5.1 Experiment settings

5.1.1 Dataset

To evaluate natural language processing methods, Xiao (2010) suggested a sufficient sample dataset to be chosen, in ensuring a balance given the variety of data and representativeness of the language. We preprocessed and manually annotated a set of 505 court decisions. Averages of 262.257 lines and 3,955.215 tokens were found per document. To simulate the representativeness of this body, the decisions were randomly chosen by varying both the city and year. The last two columns in Table 1 show the distribution of the labeled entities in the dataset. Based on a subset of 13 documents labeled by 2 different annotators, the inter-agreement rates were computed using Cohen's Kappa statistic. These inter-agreement rates were computed at the character level because some words might be cut by incorrect annotations (e.g. *<juridiction>cour d'appe</juridiction>l* vs. *<juridiction>cour d'appel</juridiction>*), or the annotator might not agree on whether or not an apostrophe needs to be included (e.g. *l'<norme>article 700* vs. *<norme>l'article 700*). Kappa rates of 0.705 and 0.974 were recorded for the entity and section labeling, respectively. According to Viera et al. (2005), the level of agreement is *substantial* for the entities (0.61 - 0.80) and *almost perfect* for the sections (0.81 - 0.99).

5.1.2 Evaluation protocol

The norm detection could be evaluated with the annotated examples of both the *corps* and *dispositif* sections. Our focus now turns to the F1-measure of each entity type, i.e. how well the tagger model is able to tag every entity token with the right label (token-level), in addition to detecting the entities entirely (entity-level). On both levels, the F1-measure formula is: $F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$. Both precision and recall values are reported in percentage terms.

Evaluation at token-level: Precision and Recall values were computed over test sets for each label l as follows:

$$Precision_l = \frac{\text{number of tokens correctly labeled by the model with } l}{\text{number of tokens labeled by the model with } l}$$

$$Recall_l = \frac{\text{number of tokens correctly labeled by the model with } l}{\text{number of tokens manually labeled with } l}$$

Evaluation at entity-level: Precision and Recall values were computed over test sets for each entity class e as follows (note: an entity mention is "correctly detected" if the model correctly labels all its tokens):

$$Precision_e = \frac{\text{number of entities of type } e \text{ correctly detected by the model}}{\text{number of entities detected and classified in } e \text{ by the model}}$$

$$Recall_e = \frac{\text{number of entities of type } e \text{ correctly detected by the model}}{\text{number of entities manually classified in } e}$$

Overall evaluation: the overall evaluation measures were computed independently of the token label or the entity type but for both levels as follows:

$$Precision = \frac{\text{number of entities (resp. entity tokens) correctly detected by the model}}{\text{number of entities (resp. entity tokens) detected by the model}}$$

$$Recall = \frac{\text{number of entities (resp. entity tokens) correctly detected by the model}}{\text{number of entities (resp. entity tokens) manually annotated}}$$

We next present the overall evaluation for the sectioning and detection entities in the header at both the token-level (Table 5) and entity-level (Table 6).

5.1.3 Software tools

We have used the HMM and CRF models as implemented in the Mallet Library (McCallum, 2002). The HMM-based models were trained by the maximum likelihood method and the CRF-based models by the L-BFGS method, since it runs faster with multiple processes in parallel. For entity detection, the *tokenization* of section contents into words and the extraction of their lemma and parts-of-speech were conducted using the French part-of-speech functionality from TreeTagger[5] (Schmid, 2013). The LDA implementation provided by Mallet was then used to extract certain topics. More precisely, a corpus of some 6,000 decisions was employed to train the LDA for the purpose of modeling 100 topics. These topics were modeled with lemmas of words of entire text content for decisions with neither punctuation nor French stop words. Table 2 presents some of the representative words found in the initial topics. The extraction of other handcrafted features was coded from scratch for this experiment. The precisions, recalls and F1-measures were all computed with the evaluation script supplied for the CoNLL-2002 shared tasks[6].

Topic ID	Representative words
0	prejudice damages sum undergo reparation title fault pay interest responsibility
1	society wage-earner group Mirabeau power claim article dismissal court title
2	harassment work wage-earner moral employer fact certificate do health behavior
3	sale act price seller buyer notary condition clause sell building
4	work post reclassification employer doctor dismissal wage-earner unfitness visit

Table 2: Representative words of the first topics out of 100 (translated into English)

5.2 Selecton of the segment representation

In order to evaluate how the segment representation may affect the results, we implemented four representations (IO, IEO2, BIO2, BIEO). IEO2 and BIO2 are variants of the IEO and BIO representations, respectively. Both use the "E-" and "B-" prefix to tag words of one-word entities, unlike IEO1 and BIO1, which instead use "I-". A simple split of the dataset yields two subsets: 25% for training HMM and CRF models, and 75% for testing. The performances reported in Table 3 are the average F1-measures over the test set entities. For both CRF and HMM, only the feature *token* is used. Training time may be very long, especially for header entity detection with CRF. It seems obvious that the greater the number of entities to label, the slower the pace of training. The same number of labels does not always lead to the same training time, and IOE2 helps CRF converge a bit faster than BIO2. It is also worth noting that some representations are more helpful for certain tasks than others. For instance, more complex representations do not improve the results for

[5] http://www.cis.uni-muenchen.de/ schmid/tools/TreeTagger

[6] http://www.cnts.ua.ac.be/conll2002/ner/bin/conlleval.txt

CRF-based section detection, yet do improve results in other tasks (e.g. IEO2 for entities in headers and for norms). Unfortunately, this improvement is insignificant even though complex representations are more likely to help detect all the entity words. As another example, at the token-level, the difference between the F1-score of the IO tagging model is always very close to the best score (usually less than 2%).

Table 3: Results of different segment representations for the segmentation task.

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

[a] Results on a simple dataset split into 25% for training and 75% for testing with HMM and CRF training iterations limited to 100
[b] Duration in seconds before training converges or reaches 100 iterations

5.3 *Feature subset selection*

To compare the BDS and SFFS methods, we relied on just the IO tagging model. Further study would compare the various combinations of segment representations and feature selection methods. Due to the large number of feature subsets that both algorithms must compare, testing all these combinations would take many days. During our experiments, the SFFS performs 185 training runs of the CRF for sections. The BDS method lasted more than 15 hours for 600 training sessions. Even though we stored some F1-measures in order to avoid running training for the same feature subset multiple times, the selection process was still very long for both algorithms. We tested each of the candidate features for the HMM-based models.

The selected combinations are unexpected because some of the special features of neighbor tokens have been chosen. For instance, in the case of section detection, the next line seems to be very important yet not the previous one. It is also interesting to note that the features, especially those based on our observations, occur in the final selected subsets for entity detection (e.g. isAfterIntervenant, isKEYWORD). Let's also point out that the absolute length (absLength) of the line plays a major role in detecting sections since it has been selected for both the HMM and CRF models (BDS selection). With these selected subsets (see Table 4), the models perform better than with either the token alone or all the extracted features combined together. The improvement in their quality remains insignificant when considering the time required to run both algorithms. Hence, a better and faster algorithm should be used instead of SFFS and BDS.

5.4 Increase of learning with experience

Some experiments were conducted to assess the quality improvement in the models as expected with more annotated training data. Their findings yielded information on how our tagger models behave depending on the size of the training dataset. Instead of splitting the data 25%-75%, the dataset was split 75% for training and 25% for testing. Only 20 fractional rates of the training sets were actually tested (from 5% to 100%). At each training-testing trial, the same test dataset was used for the various training set fractions. The CRF and HMM learning curves are depicted in Figures 3a and 3b. It is obvious that the F1-scores increase with more training data for the CRF-based and HMM-based models, but the improvement does not seem to be very significant, with over 60% of the training dataset for any given task. It is possible that the additional examples share the same structure compared to most of the others. Hence, this study can be extended by selecting the most useful examples in the training dataset. Raman and Ioerger (2003) demonstrated the benefits of example selection algorithms combined with feature selection for classification aims. These same methods may be applied to sequence labeling.

5.5 Detailed results for each entity type and section type

We detail herein a number of results for each entity and section type. The tests were conducted with all the features for the CRF-based models. Only *absLength* and *token* were used for the HMM models. The IO tagging scheme was introduced for segment representation. The maximum number of training iterations was set at 500 in order to ensure training convergence (even though HMM-based models never converge after 500 training iterations). Tables 5 and 6 display the results of 5-fold

Table 4: Effects of selected feature subsets on results.

Detection Task	Tagger	Token-level[a]			Entity-level[a]			Features subset
		Precision	Recall	F1	Precision	Recall	F1	
Sections	CRF	99.31	99.31	99.31	90.28	90.68	90.48	BDS[b1]
		99.55	99.55	**99.55**	85.69	85.84	85.76	**SFFS**[b2]
		99.36	99.36	99.36	88.16	88.39	88.27	ALL*
		91.75	91.75	91.75	64.49	56.55	60.26	token
	HMM	90.99	90.99	**90.99**	4.18	3.63	3.89	**absLength**
		86.97	86.97	86.97	4.08	3.30	3.65	relLength
		37.59	37.59	37.59	18.81	18.81	18.81	token
Header entities	CRF	94.00	91.42	92.69	92.26	88.76	90.47	BDS[c1]
		94.10	91.93	**93.00**	92.64	88.96	90.76	**SFFS**[c2]
		94.20	91.86	93.02	93.05	89.59	91.28	ALL
		86.86	78.96	82.73	80.84	65.17	72.17	token
	HMM	76.90	80.41	**78.61**	62.66	52.16	56.93	**token**
		66.48	69.67	68.04	39.34	28.36	32.96	lemma_W0
		39.63	37.50	38.54	15.49	5.35	7.95	POS
Norms	CRF	95.91	96.72	**96.31**	91.14	90.45	90.80	**BDS**[d1]
		95.68	95.45	95.57	90.34	88.27	89.29	SFFS[d2]
		95.07	96.69	95.87	90.87	90.64	90.76	ALL
		95.60	92.96	**94.26**	88.06	83.50	85.72	token
	HMM	89.21	94.25	91.66	72.67	77.28	74.90	**token**
		90.31	92.81	91.54	69.24	69.46	69.35	lemma_W0

[a] Results on a simple dataset split into 25% for training and 75% for testing with 100 maximum training iterations for CRF, and 80% for training and 20% for testing with 50 maximum training iterations for HMM

[b1] BDS selection for sections : [p0, n0, relNum, absLength, t0, t1, t2]

[b2] SFFS selection for sections: [n0, nRelLength, relNum, t0, t1, t2]

[c1] BDS selection for entities in headers [POSW1, isAfterAPPELANT, numInLine, w-2topic0, POSW2, isAfterINTERVENANT, isAfterINTIME, POSW-2, isLONELYINITIAL, token, lemma_W0, lemmaW-2, isALLPUN, w-1, w1, w2, isALLCAP]

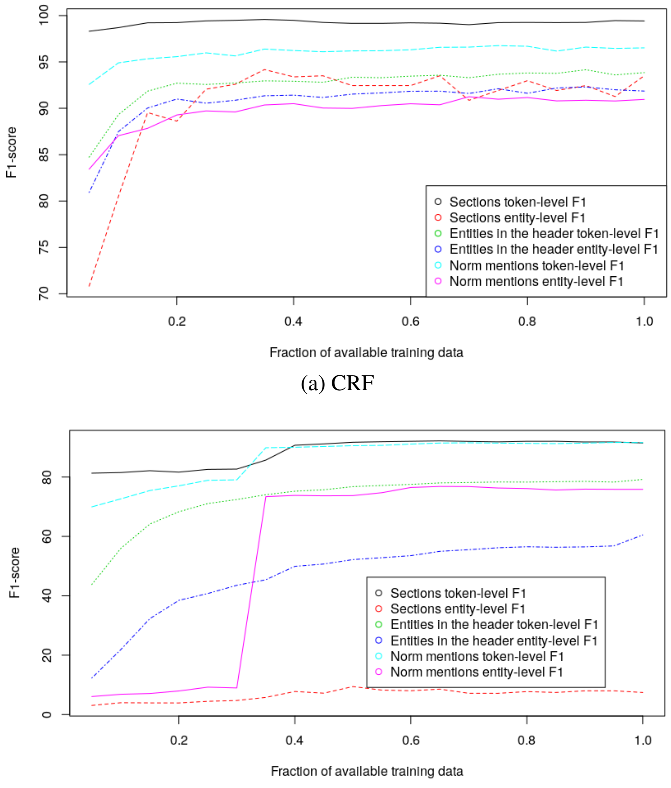
[c2] SFFS selection for entities in headers [numInLine, w-2topic0, lemmaW-2, isAfterINTERVENANT, isAfterINTIME, w-1, w1, w2, isALLCAP, token]

[d1] BDS selection for norms [POSW1, w-2topic0, isKEYWORD, lemmaW2, DIGIT-IN, token, lemmaW1, lemmaW-2, POS, isALLPUN, w-1, w2, PUN-IN, w-2]

[d2] SFFS selection for norms [POSW1, lemmaW-2, w-1, DIGIT-IN]

cross-validations at the token-level and entity-level, respectively. From a general standpoint, HMM-based models perform quite well at the token-level with only one feature, especially in detecting sections and norms. An HMM is capable of labeling the norms, in light of the common rules typically mentioned, and moreover tends to conform to a standard syntax (*article* [IDENTIFIER] [ORIGIN]). The HMM model however is not as effective in detecting entities entirely. As for CRF-based models, their results are good at both the token and entity-levels for all tasks, despite limitations in detecting party mentions.

Some labeling errors are possibly due to the proximity and similarity between entities of different types. For example, *intervenor* mentions are typically erroneously



(a) CRF

(b) HMM

Fig. 3: Learning curves at token and entity levels

classified as *respondent* or *lawyer*, maybe because all three types are names of individuals and mentioned near one another (*intervenors* are usually mentioned just after respondents' lawyers). Some *appellant* mentions are also classified as *respondent* in many instances. Similarly, misclassifications are only made between successive sections during sectioning, i.e. *Header* and *Body*, or *Body* and *Conclusion*. It would appear that such misclassifications occur on transition lines between the given sections. Another interesting remark is that some entities tend to recur multiple times in the text. For example, the parties are mentioned before any of the details about them and their lawyers; moreover, some norms are mentioned repeatedly and quite often in abbreviated forms. Although these multiple occurrences are not always exactly identical, such redundancy may help reduce the risk of missing some entities. This aspect could be utilized to correct model imperfections.

	Precision	Recall	F1
I-corps	92.46	95.25	93.83
I-dispositif	53.44	48.46	50.83
I-entete	97.91	91.93	94.83
Overall	90.63	90.63	90.63
I-appelant	34.46	16.87	22.65
I-avocat	85.17	98.75	91.46
I-date	75.67	72.45	74.02
I-fonction	88.81	64.46	74.70
I-formation	79.38	94.38	86.23
I-intervenant	82.07	38.04	51.98
I-intime	50.40	68.09	57.93
I-juge	73.40	88.73	80.34
I-juridiction	85.15	98.37	91.28
I-rg	68.53	22.14	33.47
I-ville	91.50	82.41	86.72
Overall	76.21	82.26	79.12
I-norme	88.23	93.70	90.89

(a) HMM with *absLength* and *token* as feature for sections and entities resp. and with the IO segment representation

	Precision	Recall	F1
I-corps	99.57	99.69	99.63
I-dispositif	98.63	97.59	98.11
I-entete	99.51	99.55	99.53
Overall	99.48	99.48	99.48
I-appelant	84.34	76.27	80.10
I-avocat	98.02	98.15	98.09
I-date	98.00	96.60	97.30
I-fonction	95.23	95.13	95.18
I-formation	98.80	99.45	99.12
I-intervenant	83.38	68.26	75.07
I-intime	82.54	83.33	82.93
I-juge	97.55	97.23	97.39
I-juridiction	98.91	99.69	99.30
I-rg	97.81	97.44	97.62
I-ville	98.94	99.15	99.04
Overall	95.13	94.51	94.82
I-norme	97.14	96.09	96.62

(b) CRF with all features and with the IO segment representation

Table 5: Precision, Recall, F1 measures at token-level.

6 Conclusion

Applying the HMM and CRF models in the aim of detecting sections and entities in court decisions is a difficult task. This paper has discussed the effects of various design aspects on result quality. In sum, the improvement derived seems to be quite insignificant when selecting the segment representation and feature subset separately. However, opting for the right configuration by comparing the feature subset selection with various segment representations might offer a better method. Due to the long time period required to search for the optimal feature subset, it would be preferable to use a very fast feature selection algorithm. Moreover, even though results improve as the training sample grows, still the overall F1-measure seems to reach a limit very quickly. Since some entities are not very well detected, it may be beneficial to add suitable examples in order to address these specific issues.

Two major difficulties arise in the way the models are being applied, namely: the annotation of a sufficient number of examples, and the definition of compatible features (i.e. capable of being combined to improve results). The annotation effort can be reduced with a system whose actual performance shows the ability to properly label most entities. It would then be sufficient to manually verify those annotations in order to correct any errors committed by the system on new decisions using annotative frameworks. As for the definition of features, since we define handcrafted features by observing our chosen learning set, these features might not

	Precision	Recall	F1
corps	0.99	0.99	0.99
dispositif	12.05	7.33	9.11
entete	10.47	10.50	10.48
Overall	7.22	6.27	6.71
appelant	17.84	5.60	8.52
avocat	44.29	39.15	41.56
date	66.87	62.15	64.43
fonction	89.84	64.13	74.84
formation	61.50	65.86	63.61
intervenant	14.29	4.00	6.25
intime	30.28	27.47	28.80
juge	73.54	83.21	78.07
juridiction	81.31	87.66	84.37
rg	68.53	22.41	33.77
ville	89.52	84.70	87.05
Overall	64.59	54.56	59.15
norme	71.94	78.45	75.05

(a) HMM with *absLength* and *token* as feature for sections and entities resp. and with the IO segment representation

	Precision	Recall	F1
corps	89.57	90.10	89.83
dispositif	98.02	97.82	97.92
entete	92.11	92.48	92.29
Overall	93.22	93.47	93.34
appelant	84.05	77.29	80.53
avocat	90.97	90.30	90.63
date	97.96	96.60	97.27
fonction	96.89	96.94	96.92
formation	98.40	98.95	98.68
intervenant	62.50	40.00	48.78
intime	79.31	78.93	79.12
juge	96.58	96.35	96.47
juridiction	98.86	99.54	99.20
rg	97.57	98.02	97.79
ville	98.85	99.15	99.00
Overall	93.77	92.93	93.35
norme	92.66	91.38	92.01

(b) CRF with all features and with the IO segment representation

Table 6: Precision, Recall, F1 measures at entity-level.

fit very well on a different dataset (different countries, different languages, different jurisdictions). To avoid the huge effort required to define features manually, it would be preferable to use features automatically learned from labeled or unlabeled corpora, like word embeddings.

In future works, we intend to explore other types of approaches, including deep learning. Some deep learning techniques actually perform quite well under these circumstances by combining CRF and deep learning representations (Huang et al., 2015; Ma and Hovy, 2016). It would also be worthwhile to complete the named entity recognition task. In building a knowledge base, it is indeed essential to define disambiguation and resolution approaches for entities with multiple occurrences, in addition to matching the extracted entities with reference entities, as in Dozier et al. (2010) and Cardellino et al. (2017). These entities could then be processed in order to extract more complex information, such as parties' claims and the judges' corresponding answers.

References

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent Dirichlet allocation. *the Journal of Machine Learning Research*, 3:993–1022.

Bontcheva, K., Cunningham, H., Roberts, I., Roberts, A., Tablan, V., Aswani, N., and Gorrell, G. (2013). Gate teamware: a web-based, collaborative text annotation framework. *Language Resources and Evaluation*, 47(4):1007–1029.

Cardellino, C., Teruel, M., Alemany, L. A., and Villata, S. (2017). A low-cost, high-coverage legal named entity recognizer, classifier and linker. In *Proceedings of the 16th edition of the International Conference on Articial Intelligence and Law*, pages 9–18. ACM.

Chang, Y.-s. and Sung, Y.-H. (2005). *Applying name entity recognition to informal text*. Stanford CS224N/Ling237 Final Project Report.

Chau, M., Xu, J. J., and Chen, H. (2002). Extracting meaningful entities from police narrative reports. In *Proceedings of the 2002 annual national conference on Digital government research*, pages 1–5. Digital Government Society of North America.

Chiticariu, L., Krishnamurthy, R., Li, Y., Reiss, F., and Vaithyanathan, S. (2010). Domain adaptation of rule-based annotators for named-entity recognition tasks. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1002–1012. Association for Computational Linguistics.

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

Dozier, C., Kondadadi, R., Light, M., Vachher, A., Veeramachaneni, S., and Wudali, R. (2010). Named entity recognition and resolution in legal text. In *Semantic Processing of Legal Texts*, pages 27–43. Springer.

Finkel, J. R., Grenager, T., and Manning, C. (2005). Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 363–370. Association for Computational Linguistics.

Galliano, S., Gravier, G., and Chaubard, L. (2009). The ester 2 evaluation campaign for the rich transcription of French radio broadcasts. In *Tenth Annual Conference of the International Speech Communication Association*.

Hanisch, D., Fundel, K., Mevissen, H.-T., Zimmer, R., and Fluck, J. (2005). Prominer: rule-based protein and gene entity recognition. *BMC bioinformatics*, 6(1):S14.

Huang, Z., Xu, W., and Yu, K. (2015). Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Konkol, M. and Konopík, M. (2015). Segment representations in named entity recognition. In *International Conference on Text, Speech, and Dialogue*, pages 61–70. Springer.

Kríž, V., Hladká, B., Dědek, J., and Nečaský, M. (2014). *Statistical Recognition of References in Czech Court Decisions*, pages 51–61. Springer International Publishing, Cham.

Lafferty, J., McCallum, A., and Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *International Conference on Machine Learning*.

Lam, H.-P., Hashmi, M., and Scofield, B. (2016). Enabling reasoning with legalruleml. In *International Symposium on Rules and Rule Markup Languages for the Semantic Web*, pages 241–257. Springer.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., and Dyer, C. (2016). Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Li, Y., Zaragoza, H., Herbrich, R., Shawe-Taylor, J., and Kandola, J. (2002). The perceptron algorithm with uneven margins. In *ICML*, volume 2, pages 379–386.

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

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

Ma, X. and Hovy, E. (2016). End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. *arXiv preprint arXiv:1603.01354*.

Marrero, M., Urbano, J., Sánchez-Cuadrado, S., Morato, J., and Gómez-Berbís, J. M. (2013). Named entity recognition: fallacies, challenges and opportunities. *Computer Standards & Interfaces*, 35(5):482–489.

McCallum, A. K. (2002). *MALLET: A Machine Learning for Language Toolkit*. http://mallet.cs.umass.edu/.

McCallum, A. K., Nigam, K., Rennie, J., and Seymore, K. (2000). Automating the construction of internet portals with machine learning. *Information Retrieval*, 3(2):127–163.

Mikheev, A., Moens, M., and Grover, C. (1999). Named entity recognition without gazetteers. In *Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics*, pages 1–8. Association for Computational Linguistics.

Nallapati, R., Surdeanu, M., and Manning, C. (2010). Blind domain transfer for named entity recognition using generative latent topic models. In *Proceedings of the NIPS 2010 Workshop on Transfer Learning Via Rich Generative Models*, pages 281–289.

Palmer, D. D. and Day, D. S. (1997). A statistical profile of the named entity task. In *Proceedings of the fifth conference on Applied natural language processing*, pages 190–193. Association for Computational Linguistics.

Persson, C. (2012). Machine learning for tagging of biomedical literature.

Petrillo, M. and Baycroft, J. (2010). Introduction to manual annotation. *Fairview Research*.

Plamondon, L., Lapalme, G., and Pelletier, F. (2004). Anonymisation de décisions de justice. In *XIe Conférence sur le Traitement Automatique des Langues Naturelles (TALN 2004)*, pages 367–376.

Polifroni, J. and Mairesse, F. (2011). Using latent topic features for named entity extraction in search queries. In *INTERSPEECH*, pages 2129–2132.

Pudil, P., Novovičová, J., and Kittler, J. (1994). Floating search methods in feature selection. *Pattern recognition letters*, 15(11):1119–1125.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

Raman, B. and Ioerger, T. R. (2003). Enhancing learning using feature and example selection. *Texas A&M University, College Station, TX, USA*.

Rosset, S., Grouin, C., and Zweigenbaum, P. (2011). *Entités nommées structurées: guide d'annotation Quaero*. LIMSI-Centre national de la recherche scientifique.

Schmid, H. (2013). Probabilistic part-of-speech tagging using decision trees. In *New methods in language processing*, page 154. Routledge.

Siniakov, P. (2008). *GROPUS an Adaptive Rule-based Algorithm for Information Extraction*. PhD thesis, Freie Universität Berlin.

Surdeanu, M., Nallapati, R., and Manning, C. (2010). Legal claim identification: Information extraction with hierarchically labeled data. In *Proceedings of the LREC 2010 Workshop on the Semantic Processing of Legal Texts*.

Tellier, I., Dupont, Y., and Courmet, A. (2012). Un segmenteur-étiqueteur et un chunker pour le français. *JEP-TALN-RECITAL 2012*, page 7.

Tjong Kim Sang, E. F. and De Meulder, F. (2003). Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, pages 142–147, Stroudsburg, PA, USA. Association for Computational Linguistics.

Viera, A. J., Garrett, J. M., et al. (2005). Understanding interobserver agreement: the kappa statistic. *Fam Med*, 37(5):360–363.

Viterbi, A. J. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory*, 13(2):260–269.

Wallach, H. M. (2004). Conditional Random Fields: An Introduction. *University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-04-21*.

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

Witten, I. H., Bray, Z., Mahoui, M., and Teahan, W. J. (1999). Using language models for generic entity extraction. In *Proceedings of the ICML Workshop on Text Mining*.

Wu, Y., Zhao, J., and Xu, B. (2003). Chinese named entity recognition combining a statistical model with human knowledge. In *Proceedings of the ACL 2003 workshop on Multilingual and mixed-language named entity recognition-Volume 15*, pages 65–72. Association for Computational Linguistics.

Wyner, A. and Peters, W. (2012). Semantic annotations for legal text processing using GATE Teamware. In *Semantic Processing of Legal Texts (SPLeT-2012) Workshop Programme*, page 34.

Xiao, R. (2010). *Handbook of Natural Language Processing*, chapter 7 - Corpus Creation, page 146–165. Chapman and Hall, second edition.

Zhu, X. (2010). *Conditional Random Fields*. CS769 Spring 2010 Advanced Natural Language Processing. http://pages.cs.wisc.edu/ jerryzhu/cs769/CRF.pdf.