Extracting facts and opinions from court decisions

Ivan Liao and **Kaifei Peng**
University of California, Berkeley Extension
{ciliao, pkf123} @ berkeley.edu.tw

1 Motivation (Question Formulation)

Precedents, or prior court decisions that are related to the legal cases at issue, are crucial for both parties, not only in the countries with common law systems, but also in those with civil law systems. To prevail in the legal disputes or litigation, it is necessary and universal for legal professionals to search for precedents that are in favor of their clients' standings. However, legal cases have become more complicated nowadays, and attorneys need to spend more time seeking the right precedents that are both strongly relevant or similar to the current case at hand and that are advantageous to their clients. Sometimes they need to spend hours digging into the law databases before finding something truly useful, which, nevertheless, has already considerably cost clients a fortune.

Thus, if the automatic NLP techniques can aid in speeding up the treasure hunt process, i.e. specifically showing relevant and essential information for each court opinions in search results, along with which group the case belongs to, i.e. which party wins the case, much of legal experts' time can be spared. By achieving aforementioned goals, only most important information is shown in front of each case, hence the user of the implemented system can soon realize whether the case is useful or not. For instance, for a patent litigation case, if the party names, patent involved, specific events with temporal and spatial information, and which party wins the case, even the deciding court reasoning sentences, can be shown before the full text of each court decision, patent experts can soon find if the case is meaningful to them. To our knowledge, this implementation has not done by others, including proprietary and popular legal databases.

To implement such systems, it is necessary to have a review of current methods on information extraction (IE) and clustering, along with other relevant NLP techniques, to see which methodology is feasible and can be applied to meet our objectives:

- Extract events (facts about the cases).
- Extract plaintiff's, defendant's and court's opinions respectively.
- Use our documents with the verdict result discarded to train a clusterer/classifier, which can be used for court decision predicting given the plaintiff's and defendant's proposition documents beforehand.

2 Literature Review

Jurafsky and Martin (2018) provided as a reading material in ANLP class, is a basic introduction of task of information extraction and lists the general subtasks of information extraction, namely NER (named entity recognition), relation extraction, time extraction, events extraction and template filling, supplemented with ways of implementation, ranging from relatively old rule-based ways to recent ML- and NN-based approaches.

Research on information extraction (IE) has been made during past decades. Applying pattern recognition on syntactic and semantic analyzed texts has been a prevalent way for IE. Hobbs et al. (1997) implements this by building finite-state transducer cascades, which tracks the patterns of words or more sophisticated word group, to extract information. Friburger and Maurel (2004) presents a finite-state transducer cascades to extract named entities in texts with the help of the INTEX system. Aside from rule-based approach, Freitag and McCallum (1999) explores information extraction with Hidden Markov Model (HMM) sequential model and extracts the text corresponding to predefined target states.

As for IE on legal texts and court documents, a hybrid of methods are often combined to achieve the purpose. Moens et al. (2007) detects arguments in legal texts by viewing it as a text classification task, which is solved by selecting rigorous text feature and applying multinomial naive Bayes classifier and maximum entropy model. Biagioli et al. (2005) recognizes provisions mentioned in law documents by text classification using Multiclass Support Vector Machine (MSVM). It then extracts further information relevant to the specific detected provision type by frame-filling based on shallow syntactic parsing and semantic annotation with IRC finite-state compiler.

While above works focus on extracting the topic of a piece of legal texts, Lee (1998) is more close to our intended work in that it extracts specific criminal events with detailed information. It also adopts template filling as previous works.

Brüninghaus and Ashley (2001) was another work toward abstracting legal texts with IE methods to extract useful features for future case result classification on a specific legal domain. As for the actual procedure, first, a "squib," or the summary, of a court decision, as well as the manual establishment of essential "Factors" for specific legal domains, have to be created first. Second, specific names and products were replaced with general roles and trade secret-related general information terms, and actions of roles were captured. Third, transformed texts were used as the input to classification rules, and see if a specific Factors can be matched as the binary output.

3 Proposed Methods

High-level idea: Since most relevant works apply rule-based pattern matching methods, we decide to adopt similar methodology, with currently existing POS tagging, semantic annotation algorithms. We will also try automatic method like sequential model and classification.

Also, since different legal domains can contain different set of rules, we decided to focus on a specific domain first: patent law. Since court decisions from the US Court of Appeals for the Federal Circuit are the main source of important precedents for patent litigation cases, we decided to use some 325 cases as our original dataset.

We plan to use NLTK and SpaCy to handle POS tagging and to conduct IE on NER, relation, time, and events, to see which tools yield more sensible outcome. We then may move forward to use Scikit Learn to train NER classifiers and see if better classification rules can be learned automatically to reach better result of precision and recall. If time permitted, we may implement our system without these pre-implemented models and create our own system with our own codes, whereby we can have more flexibility.

Once the prototype of our system is established, we may expand our dataset to include more cases, say 1000 cases, and see if more specific and robust classifier can be learned.

References

Carlo Biagioli, Enrico Francesconi, Andrea Passerini, Simonetta Montemagni, and Claudia Soria. 2005, June. *Automatic semantics extraction in law documents*. In Proceedings of the 10th international conference on Artificial intelligence and law (pp. 133-140). ACM.

Stefanie Brüninghaus and Kevin D. Ashley. 2001, May. *Improving the representation of legal case texts with information extraction methods*. In Proceedings of the 8th international conference on Artificial intelligence and law (pp. 42-51). ACM.

Dayne Freitag and Andrew McCallum. 1999, July. *Information extraction with HMMs and shrinkage*. In Proceedings of the AAAI-99 workshop on machine learning for information extraction (pp. 31-36).

Nathalie Friburger and Denis Maurel. 2004. *Finite-state transducer cascades to extract named entities in texts*. Theoretical Computer Science, 313(1), 93-104.

Jerry R. Hobbs, Douglas Appelt, John Bear, David Israel, Megumi Kameyama, Mark Stickel, and Mabry Tyson. 1997. *FASTUS: A cascaded finite-state transducer for extracting information from natural-language text*. Finite-state language processing, 383-406.

Dan Jurafsky and James H. Martin. 2018. *Speech and language processing*. London: Pearson.

Richard Lee. 1998, October. *Automatic information extraction from documents: A tool for intelligence and law enforcement analysts*. In Proceedings of 1998 AAAI Fall Symposium on Artificial Intelligence and Link Analysis (Vol. 23). AAAI Press Menlo Park, CA.

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