Interactive Knowledge Graph for Case Law Analysis

Zichen Bai, Yun-Hsuan Chen, Tien-Chi Hsin, Ching-Chieh Huang, Ming-Ying Li, and Ting Liao*
**Listed in alphabetical order. All team members have contributed a similar amount of effort.*

Introduction

Law is a field that preserves the human wisdom of reasoning and logical thinking via language. Language is based on interpretation and contains ambiguity, which makes it difficult to retrieve legal knowledge via machine. Many analyses on legal textual information rely on labor- and expertise-intensive annotation work, which creates a "knowledge acquisition bottleneck" (Liebwald, 2007; Moreno and Redondo, 2016). Although there are programs like IBM's Watson and Debater, there is still a lack of structural knowledge base in the field (Ashley, 2017). Legal professionals today have to search for legal concepts and sort out facts by reading a huge amount of pure texts. A method that can extract the message from a legal document and present it in a direct and informative way will provide immense benefits in saving time and resources for legal professionals and the public alike.

Problem Definition

Legal analytics are still under development. Although many studies and programs have successfully predicted legal outcomes or perform categorization, not many have built the ontology and knowledge graph (KG) to actually structuralize the knowledge of individual legal documents. Thus, the purpose of our project is to (1) cluster cases and identify their topics (2) construct legal KGs for case laws. This would save time for legal professionals in scrutinizing each and every case, as well as deliver legal information to the general public. The purpose of this project is to display a hierarchical tree clustering results for users, and the KGs for each case.

Survey

Since the 1980s, many have applied artificial intelligence techniques in the legal field, including legal ontology, entity recognition, modeling legal reasoning, legal outcome prediction, and case categorization, to list just a few (Ashley, 2017; Zhong, 2020). To identify reasons in court decisions and predict outcomes, Raghupathi et al. (2018) applies Hadoop MapReduce and other algorithms to handle large amounts of unstructured text data. Using support vector machines model (SVM), Xu et al. (2019) builds a classifier to predict categories of lawsuits. Combining aRecurrent Neural Network (RNN) with Long Short-Term Memory units (LSTM), Polo et al. (2020) classifies legal proceedings, which are composed of sequential texts. Using the conditional random fields (CMF), Lin et al. (2012) presents an approach to label legal documents automatically while taking textual context into account, which can further help classify and predict sentences for similar types of cases. Focusing on legal citation, Sadeghian et al. (2016) extracts the citation text from legal documents and then classified them using k-means.

Several studies have developed ontologies and build KGs to structuralize the relations between entities in texts KG is a structured knowledge base that can be used in knowledge representation, management, and application. KGs are built based on ontology, which defines entities, relations, and attributes in our domain (Bellomarini et al., 2020). Yu et al. (2017) uses multiple layers of LSTM units to find the important context between one question, surrounding relation, and the context of relation pair. Szekely et al. (2015) crawls data from webpage and dealt with the noises to build a searchable KG to combat human trafficking. Specific for legal analytics, Mommers (2001) defines ontological terms and concepts including factual knowledge to transform concepts in law into knowledge-based domain and range. Cavar et al. (2018) develops their own relation extraction structures for case laws to construct KGs. He et al. (2019) presents an example of an interactive visualization of KGs through user-centered development iteration. In addition, to build the KG, the machine needs to understand the relations between subjects. Zhu et al. (2015) uses textual content and semantic location and time to recognize human activities; Van Hee et al. (2018) builds a binary classifier to detect cyberbullying via social media text among youngsters, both using SVM for the classification.

Method

To achieve the goal of this project, we adopted four steps: (1) conduct web crawling to collect cases from the case law database: casetext.com, (2) cluster cases of similar contents via Latent Dirichlet Allocation (LDA), (3) extract relations in each cases, and (4) build a visualization and correction interface for our KG based on relation extraction results.

(1) Data Collection

The cases in legal databases such as "casetext.com" are pure text, and there is no existing dataset for us to use directly. At the beginning, we manually collected around 500 cases for our first keyword "Samsung v. Apple," which was time consuming and may contain human errors. Thus, we developed a web crawler to collect each search result as an individual .txt file. Since the database required users to login, we used the Python package *selenium webdriver* to set up a user-agent to access the database. Then, we collected all search results, i.e. cases, by identifying headings with the <a> tag and stored all their URLs. We further assessed each search result via the saved URL to collect the texts of each case. For each case, we stored the full legal citation[1], e.g. "Samsung Elecs. Co. v. Apple Inc., 137 S. Ct. 429 (2016)" under the <h4 _ngcontent-sc30> tag, and stored all contents under the <p> tags as the full text of each case. We use full text for the case clustering model (LDA). For the KG, we focused on the background section of each case. However, not all cases have such a section. Thus, we first confirmed if "Background" is contained in any of the <h3> tags. If yes, we would retrieve the texts under the "Background" heading. To determine the end of the "Background" section, we identified the next <h3> heading that did not contain a dot (.)[2]. In this way, we stored the background section of each case (if available) as a .txt file. In this project, three sets of keywords were used: "Samsung v. Apple," "Microsoft v. United States," and "Oracle v. Google." A total of 1,179 cases (Avg. 6,012.73 words each) were collected for topic classification (LDA), and the background section of a total of 510 cases (Avg. 25,803.31 words each) were collected[3]. The cases collected can be found in https://reurl.cc/3LAQ9L for LDA cases and https://reurl.cc/VXZoQN for KG cases.

(2) Case Clustering and Topic Identification

We implemented Latent Dirichlet Allocation (LDA) for case clustering and topic identification. LDA is an unsupervised statistical model that can identify unobserved groups among a set of documents which share similar contents. Before applying the LDA model, we preprocessed the 1,179 cases by removing the stop words (eg. I, him, on) and retaining only adjectives and nouns. We used grid search to compute the optimum values of hyperparameters, and decided to cluster into three groups based on the model perplexity as well as the interpretability of the results. When conducting the LDA model, the algorithm used the matching rate to classify the cases into the highest matching topic. After applying the model, key words of each topic were identified, and the probability of belonging to that topic for each case was acquired. We found three topics with 141, 685, and 353 cases, respectively. Figure 1 demonstrates the word clouds for the three topics found. We therefore named the three topics as Competition Disputes, Procedural Disputes, and Substantive Disputes (Algorithm, Patent).

[1] Legal citations are references to specific legal sources, and each case law has its unique citation.
[2] This is because the naming of the heading in cases laws, if there are subsections in "section," the heading will be, for example, "A. The Technology," "B. Google's Accused Product: Android," … etc. Thus, The next section heading, for example, "Discussion," would be without the dot (.).
[3] The average word count of text files containing 'background' is more than regular text files. There are two main reasons. The first is that many text files without a background section are short. The second reason might be the exception of our rule to scrape the background section, so it unexpectedly scrapes the text to the end of the page.

Figure 1. Word clouds for LDA clustering results

(3) Relation Extraction

The relation extraction system is designed as below.[4] We develop our own grammar matching rules to process the sentences in legal documents. The matching rules we have embodied include: (1) subject, relation, and object triple extraction for a basic sentence, (2) assumption 'that'-clause pattern matching, (3) description 'that'-clause pattern matching (4) coordinate conjunction branching, and (5) participial construction restoration. The main idea is to leverage Rules (2)-(5) to make the sentences simpler, and then apply Rule (1) to precisely extract the triple. The technical details of relation extraction have been explained in the progress report, and we do not repeat them here.

Rule (1) is the main goal of relation extraction, which is to find the subject, relation and objects of each sentence. We processed each sentence via spaCy dependency trees based on multi-layer CNN. We used parentheses to mark the main object word and list other object words together without parentheses. We also applied spaCy neuralcoref based on a Transformer-based model, to perform coreference restoration, which helps us better restore right subjects or objects before the rules mentioned earlier. The background section of a case law usually includes the facts sorted out by the court, the claims of the plaintiff and defendant, and the decisions or opinions made by court(s) in previous trial(s). Therefore, we roughly classified these sentences into two categories: assumptions or description sentences by applying Rules (2) and (3). We hierarchically processed the independent clause and 'that'-clause sentences, and recognized a sentence as an assumption sentence if it contains words "will" or "would," and as a descriptive sentence if it does not contain such words. Since sentences with several complex facts usually contain coordinate conjunction or a participial construction, we recognize such sentences and restructure them into sentences that Rule (1) can process by applying Rules (4) and (5). With Rule (4), if a sentence contains coordinate conjunction, we split the sentence into two sentences. With Rule (5), we restored participial construction by finding the subjects that were omitted, and split such a sentence into two sentences.

(4) Visualization

LDA and relation extraction results were stored in JSON format files, which were then used to construct a visualization demo including 274 cases that are easier to understand for users. We considered the hierarchical structure of case clustering from LDA was best preserved in the form of a collapsible tree. Thus, we created an expand-collapse feature upon mouse click for users to access every case for a certain topic. The topic tree shares the same zoom/pan feature seen in the KG. Links to the KG were created for leaf cases of the tree.

[4] Cavar et al. (2018) claim that they utilize Lexical Functional Grammar (FLG) but do not show the detailed methods in their workshop paper. The main difference of our project is that we leverage the dependency parsing in python spacy and develop our own grammar matching rules to process the sentences.

The construction of the KG was based on the concept of force graph from CSE 6242 HW2, each subject, object, and relation will be assigned as a unique set of nodes in the graph, marked by different colors. Edges between the nodes should be accompanied with arrows to indicate the relationship between the nodes, assumed direction being subject pointing to the relation pointing to the object. The original case text is shown to the right of the graph for convenience purposes. To provide the optimal amount of information for every user, the graph adapted and built upon the interactive features of the force graph: (1) every node and their connected nodes and edges can be dragged around or pinned to the canvas for convenient organization of the information. The graph also supports zoom and pan for better user experience. (2) Upon clicking on relation nodes, the original sentence will be highlighted on the full text to the right. This can help the users to swiftly switch between looking at nodes from the KG and reading the text sentence. (3) Filters are provided for users to search for certain nodes, allowing them to quickly obtain information needed. (4) Addition and deletion tools allow users to add subject-relation-object pairs from the KG. This provides means of editing the graph based on the needs of each user.

To provide a consistent user experience, we created a guide page to combine the two visualizations. A user may start with the topic tree to locate a case of interest, and clicking on the case would bring the user to the KG of the case. An example of the visualization interface can be found in the experiment section of the report.

Experiment and Evaluation

(1) Case Clustering

To perform and evaluate LDA model we use, we leverage gridsearch ({'n_components': [3, 5], 'learning_decay': [.5, .7], 'learning_method': ['online','batch']}) to find the setting with the smallest model perplexity. The perplexity is 2677.43, which is reasonable as our model copes with 1,179 cases (Avg. 6,012.73 words). In human evaluation, we record the probability of the classified topic for each document and observe whether the probability and document are reasonable. The keywords in each topic word cloud in Figure 1 are highly-related in human perspective; thus, we can define the name of the topics. In the later visualization part, the collapsible tree can show the clustering results intuitively. Thus, the LDA model can help the construction of the collapsible tree.

(2) Relation Extraction

We developed four metrics to examine our current result for relation extraction: correctness, completeness, readability, and functionality. The relation extraction result is compared with that of TextRazor, a natural language processing API. It should be noted that our relation extraction unifies deep learning with our defined rules, which is unlikely to be perfect.

We evaluated the correctness and completeness of our relation extraction result, i.e. if our system can extract the correct subject, relations and objects, and how complete we can capture the meaning of a sentence. For example, for the Apple-Inc.-v.-Samsung-Elecs.-Co.-727-F.3d-1214-Fed.-Cir.-2013 case, around 85% are completely or partially correct for subject extraction. For relations, our current system is able to extract a majority of them correctly, especially after identifying ‘that’ clause. On the other hand, objects usually contain more complex information. Unlike TextRazor which simply lump everything after the predicate (the main verb of a sentence), we are able to identify and extract the main object word and some other information. However, not all important information is extracted, and the readability can be further improved. For complex sentences, as mentioned in the Proposed Methods section, we are able to split sentences connected by coordinate conjunctions or restore those with participial construction. (eg. Sentence 1 in the Apple v. Samsung Case).

Figure 2. Interactive Topic Tree and the Corresponding Cases

Evaluation is also performed on the readability of the results, i.e. how readable they are for the users. Sentences that were not part of the background information of the case text were excluded from our analysis, such as sentences for references. We also replaced pronouns with the original nouns they are indicating so it makes sense when we present them as edges and nodes, and replaced "we"with "the court" to indicate the sentences that are representing the court's opinion.

Lastly, we evaluated the functionality of our results, i.e. the functions that we can provide from our extraction result to assist users, such as categorizing different types of information in a case law. Assumption sentences were differentiated from description sentences to help legal professionals sorting out claims asserted by the parties and decisions made by the court. As dates are critical information in legal cases, we also identified dates so users can align facts in the chronological order.

(3) Visualization

The goal of this project eventually is to provide an alternative way of getting information from case text with efficiency, as such we evaluated our visualization result based on user experience. Similarly to the evaluation above, we used TextRazor as a comparison. The designed use flow is that when users enter the interface, the case topic tree (Figure 2) is first presented. Users may explore the topics and cases underneath. When an interesting case is found, users can click on the case to bring up the KG for that case. In the KG, users will see the extracted relation triples as well as the full case text.

Compared to TextRazor our interface showed advantages in interaction ability and information presentation. Our interface contains about 1,000 law cases categorized into several topics for users to browse through and explore. This way of processing and clustering law cases from plain text is relatively new in the practice. For individual cases, a sample KG is shown in Figure 3 below. Compared with the table charts visualization achievable through the process of TextRazor, our KG presents the relations with nodes and edges. Users can start with a node and gradually tour through the entire graph following the edges, which may be a more natural way of obtaining relevant information than reading the table row by row. Besides, we believed that a subject with relationships to multiple objects was most effectively presented in a node-edge format than in tables. With the D3 framework, the KG provides functionalities such as zoom, pan, and drag. Combined with other features such as text highlighting and relation searching, case text reading experience is separated into refined modules connected by mouse clicks.

To provide an objective evaluation on user experience, we tested and rated our visualization interface within the group members. LDA visualization (topic tree) and relation extraction visualization (KG) were rated separately on convenience and usefulness, and relation extraction was additionally rated for readability. The scale ranged from 1 to 5 with 1 being not helpful and 5 being very helpful. Out of the 5 dimensions, our team rated the convenience of KG the highest (4.5), followed by usefulness (4.3) and convenience (4) of topic tree. The usefulness (3.8) and readability (3.4) of KG received relatively low

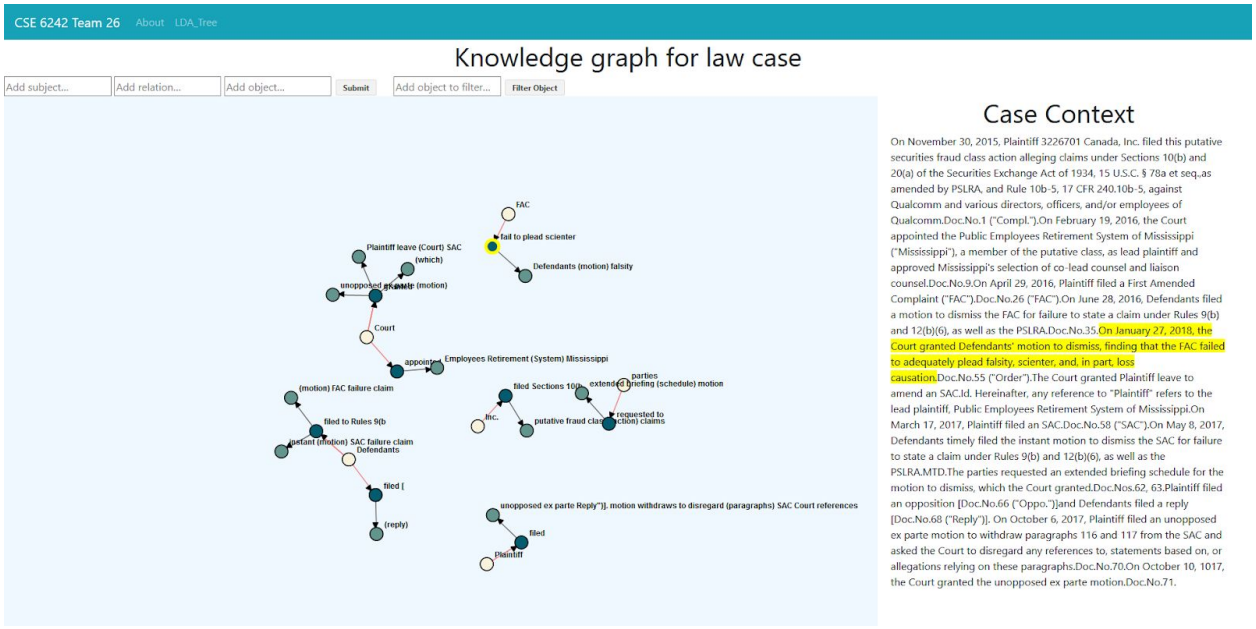

Figure 3. Interactive Knowledge Graph and the Corresponding Case Context

scores. We also collected improvement suggestions within the team members, most of which were related to the UI of topic tree and KG. Some suggested that additional features such as importance ranking and user upload tool be added in future versions.

Discussion and Conclusion

In this project, we created a web crawler to obtain case text from webpages as our data. Cases were organized into 3 topics identified by LDA. We then implemented new rule-based algorithms for relation extraction to build up our KG of each case background. We mainly evaluate both performances in human evaluation. We perform gridsearch and perplexity to obtain a reasonable LDA model. The extracted results suggested a 35% improvement from a simple information extraction method. The word clouds show highly-related words in each topic. Collapsible topic trees and relation KGs were constructed for law case visualization. An interactive interface website was created to access the case information. The interface also allows users to customize the KG by adding nodes and edges, as well as applying filters.

References

Ashley, K. D. (2017). Artificial Intelligence and Legal Analytics. doi:10.101hj7/9781316761380

Bellomarini, L., Sallinger, E., & Vahdati, S. (2020). Knowledge Graphs: The Layered Perspective. In *Knowledge Graphs and Big Data Processing* (pp. 20-34). Springer, Cham.

Cavar, D., Herring, J., & Meyer, A. (2018). Law Analysis using Deep NLP and Knowledge Graphs.

Crotti Junior, A., Orlandi, F., O'Sullivan, D., Dirschl, C., Reul, Q. (2019). Using mapping languages for building legal knowledge graphs from XML files. In *Workshop on Contextualized Knowledge Graphs co-located with the 18th International Semantic Web Conference (ISWC)*, 2019.

He, X., Zhang, R., Rizvi, R., Vasilakes, J., Yang, X., Guo, Y., . . . Bian, J. (2019). ALOHA: Developing an interactive graph-based visualization for dietary supplement knowledge graph through user-centered design. *BMC Medical Informatics and Decision Making, 19*(S4). doi:10.1186/s12911-019-0857-1

Liebwald, Doris. (2007). Semantic Spaces and Multilingualism in the Law: The Challenge of Legal Knowledge Management.. 321. 131-148.

Lin, S.d. (2012). Exploiting Machine Learning Models for Chinese Legal Documents Labeling, Case Classification, and Sentencing Prediction [In Traditional Chinese]. In International Journal of Computational Linguistics & Chinese Language Processing, Volume 17, Number 4, December 2012-Special Issue on Selected Papers from ROCLING XXIV.

Marzouk, M., & Enaba, M. (2019). Text analytics to analyze and monitor construction project contract and correspondence. *Automation in Construction, 98*, 265-274. doi:10.1016/j.autcon.2018.11.018

Mommers, L.. (2003). Application of a knowledge-based ontology of the legal domain in collaborative workspaces. , 70 - 76 (2003). 10.1145/1047788.1047799.

Moreno, A., & Redondo, T. (2016). Text Analytics: The convergence of Big Data and Artificial Intelligence. *International Journal of Interactive Multimedia and Artificial Intelligence, 3*(6), 57. doi:10.9781/ijimai.2016.369

Polo, F. M., Ciochetti, I., & Bertolo, E. (2020). Predicting Legal Proceedings Status: an Approach Based on Sequential Text Data. *arXiv preprint arXiv:2003.11561.*

Raghupathi, V., Zhou, Y., & Raghupathi, W. (2018). Legal Decision Support: Exploring Big Data Analytics Approach to Modeling Pharma Patent Validity Cases. *IEEE Access, 6*, 41518-41528. doi:10.1109/access.2018.2859052

Sadeghian, A., Sundaram, L., Wang, D. Z., Hamilton, W. F., Branting, K., & Pfeifer, C. (2018). Automatic semantic edge labeling over legal citation graphs. *Artificial Intelligence and Law, 26*(2), 127-144. doi:10.1007/s10506-018-9217-1

Szekely, P., Knoblock, C. A., Slepicka, J., Philpot, A., Singh, A., Yin, C., . . . Ferreira, L. (2015). Building and Using a Knowledge Graph to Combat Human Trafficking. In *International Semantic Web Conference,* 2015

Van Hee, Cynthia & Jacobs, Gilles & Emmery, Chris & Desmet, Bart & Lefever, Els & Verhoeven, Ben & Pauw, Guy & Daelemans, Walter & Hoste, Véronique. (2018). Automatic Detection of Cyberbullying in Social Media Text.

Xu, Y., Zhang, M., Wu, S., & Hu, J. (2019). Lawsuit category prediction based on machine learning. *2019 IEEE International Conference on Intelligence and Security Informatics (ISI)*. doi:10.1109/isi.2019.8823328

Yu, M., Yin, W., Hasan, K. S., Santos, C. D., Xiang, B., & Zhou, B. (2017). Improved Neural Relation Detection for Knowledge Base Question Answering. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. doi:10.18653/v1/p17-1053

Zhong, H., Xiao, C., Tu, C., Zhang, T., Liu, Z., and Sun, M. (2020). How does NLP benefit legal system: A summary of legal artificial intelligence. In *Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL)*, Jul. 2020.

Zhu, Z., Blanke, U., & Tröster, G. (2016). Recognizing composite daily activities from crowd-labelled social media data. *Pervasive and Mobile Computing, 26*, 103-120. doi:10.1016/j.pmcj.2015.10.007