Semantic Search and Summarization of Judgments Using Topic Modeling

Tien-Hsuan WU\textsuperscript{a}, Ben KAO\textsuperscript{a}, Felix CHAN\textsuperscript{b}, Anne SY CHEUNG\textsuperscript{b}, Michael MK CHEUNG\textsuperscript{b}, Guowen YUAN\textsuperscript{a}, and Yongxi CHEN\textsuperscript{b}
\textsuperscript{a}Department of Computer Science, The University of Hong Kong
\textsuperscript{b}Faculty of Law, The University of Hong Kong

Abstract. Online legal document libraries, such as WorldLII, are indispensable tools for legal professionals to conduct legal research. We study how topic modeling techniques can be applied to such platforms to facilitate searching of court judgments. Specifically, we improve search effectiveness by matching judgments to queries at semantics level rather than at keyword level. Also, we design a system that summarizes a retrieved judgment by highlighting a small number of paragraphs that are semantically most relevant to the user query. This summary serves two purposes: (1) It explains to the user why the machine finds the retrieved judgment relevant to the user’s query, and (2) it helps the user quickly grasp the most salient points of the judgment, which significantly reduces the amount of time needed by the user to go through the returned search results. We further enhance our system by integrating domain knowledge provided by legal experts. The knowledge includes the features and aspects that are most important for a given category of judgments. Users can then view a judgement’s summary focusing on particular aspects only. We illustrate the effectiveness of our techniques with a user evaluation experiment on the HKLII platform. The results show that our methods are highly effective.

Keywords. Topic modeling, Semantic search, Judgment summarization

1. Introduction

In common law jurisdictions, prior judgments (a.k.a. precedents) are important parts of the law. Retrieving relevant judgments is an important task in legal research. To find existing judgments, one may resort to legal databases such as the World Legal Information Institute (WorldLII) \cite{1}. Although existing legal database systems provide search functions that facilitate judgment retrieval, they are mostly limited to simple keyword search. It is well known that keyword-based search suffers from poor query expressiveness.

In this paper we address the judgment retrieval problem by applying topic modeling techniques to perform semantic search and judgment summarization. Specifically, our approach consists of the following three components.

[Semantic Search] Existing search engines deployed in legal database systems such as HKLII mostly retrieve judgments based on keyword matching. This is ineffective especially when the search intent involves abstract concepts that can be expressed in various wordings or in technical terms that the query issuer is not familiar with. We achieve semantic search by applying topic modeling. In a nutshell, topic modeling is a
A statistical framework that analyzes a document corpus to identify distinguishing words that have strong associations (e.g., based on co-occurrences of words in documents). A “topic” can be considered as a cluster of words based on their associations, which, collectively, express certain abstract concept. Figure 1 shows an example word cluster that expresses the concept (or topic) of “finger injury”. Our method of semantic search is to first analyze court judgments to discover topics (in topic-modeling sense) and identify the topics that are covered by each specific judgment. When given a query, the topic that is most relevant to the query is found and the judgments that have the best coverage of the topic are retrieved. Figure 2 illustrates our idea.

[Query-driven Summarization] A search engine often returns search results as a list of hypertext links, each with the corresponding document title displayed. It is difficult for the user to determine if a document is really relevant to his/her query by inspecting the title only. As we observed in analyzing the HKLII search log, very often a user would click and read many returned documents that turned out to be irrelevant to the search intent. This is a major source of inefficiency in judgment search as users toil through long and complex judgments. Our approach to ameliorating unproductive search results filtering is to perform automatic judgment summarization. Specifically, a small fraction (such as 5%) of a judgment’s paragraphs are selected by the machine based on their relevancy to the given user query. These paragraphs are highlighted in the judgment and they serve as a query-specific summary of the judgment. By reading this small (5%) summary, the query issuer gets to know why the machine thinks that the judgment is relevant to the query, obtains a basic understanding of the judgment’s content, and thus is able to quickly determine if the judgment should be filtered or collected.

[Aspect-driven Summarization] After a user accepts a judgment as relevant with the help of a query-specific summary, the user typically needs to know more about the judgment with respect to different aspects of the case concerned. To address this, our system provides aspect-specific summaries of judgments. Our approach is to first consult legal experts on the most important features and aspects of each judgment category. For example, for personal injury cases, aspects of interests include a plaintiff’s background, treatments, losses, and compensations. The machine would then summarize a judgment based on each aspect by finding a small number of paragraphs in the judgment that are most relevant to the chosen aspect. Figure 3 shows an example.

2. Algorithms

In this section we discuss how topics are generated (Section 2.1) and how we use the generated topics in semantic search and judgment summarization (Section 2.2).
2.1. Topic Generation

We adopt Latent Dirichlet Allocation (LDA) [2] as the topic modeling method. Given a collection of documents, LDA generates topics by computing two sets of distributions:

1. A word distribution for each topic. The word distribution captures the words that express the topic. An example is shown in Figure 1. (2) A topic distribution for each document. The topic distribution reflects the probability of each topic occurring in the document. In the following discussion, we use personal injury compensation cases in Hong Kong to illustrate our methods. We consider three ways of applying LDA to judgment data. They differ in how judgment documents are processed and whether expert knowledge on specific judgment category is taken into account. Figure 4 illustrates the three approaches. Next, we give details of the approaches.

[No Domain Knowledge (NoDK)] Judgments are first preprocessed by removing numbers and stop words. Then, we run LDA on the judgments using MALLET’s implementation [3] to generate topics. This is illustrated in Figure 4(a).

[Feature Domain Knowledge (DK-F)] We consult legal experts to obtain a list of features that are important for the specific category of judgments in the corpus. For example, for PI judgments, these features include "age at the time of incident" and "whether the injury is permanent". Judgments are then manually annotated to identify spans of text that contain information related to the features. We call these spans of text "labeled text". We strip each judgment of its unlabeled text; Only labeled text is retained to which we apply LDA. The idea is to remove unimportant details so that the topics generated are related to the more important contents of judgments. Figure 4(b) illustrates this approach.

[Aspect Domain Knowledge (DK-A)] We further consult legal experts to group features into aspects. For example, for PI cases, four aspects are given, namely, (plaintiff’s) background, injury, treatment, and loss. We perform topic modeling on the labeled text under each aspect separately. For example, with respect to the background aspect, we...
retain only the labeled text for background-related features in judgments before applying LDA. The idea is to generate aspect-specific topics so that judgments can be summarized based on a desired aspect. Figure 4(c) illustrates this approach.

2.2. Applications

In this section we discuss how we make use of the generated topics to perform semantic search and judgment summarization.

2.2.1. Semantic Search

Let $T = \{ T_1, \ldots, T_N \}$ be the set of $N$ topics obtained from LDA. Given a query $q$ and a judgment $J$, we evaluate the relevancy of $q$ and $J$ w.r.t. the $N$ topics. The similarity of $q$ and $J$ is then measured by their overlapping topics. Specifically, a topic $T_i$ is represented by a word vector $[w_{i,1}, \ldots, w_{i,k}, \ldots]$, where each $w_{i,k}$ is a word with probability $p_{i,k}$ of being relevant to topic $T_i$. For each word $w_{i,k}$, we apply word2vec [4] to obtain its word embedding vector $w_{i,k}$. We then compute the average of all word embedding vectors $w_{i,k}$’s weighted by their probabilities $p_{i,k}$’s. We call the resulting embedding vector the topic semantic vector $v_{T_i}$ of topic $T_i$. Next, we process the query $q$ in a similar fashion: we first obtain the word embedding vector of each word in $q$ and then compute the vectors’ average. We call the resulting embedding vector the query semantic vector $v_q$ of query $q$. The query-topic similarity score, $s_{q,i}$, between query $q$ and topic $T_i$ is measured by the cosine similarity of the semantic vectors, i.e., $v_{T_i} \cdot v_q / \|v_{T_i}\| \|v_q\|$. We collect the similarity scores over all topics into a query-topic probability vector $p_q = [s_{q,1}, \ldots, s_{q,i}, \ldots]$ of query $q$. This vector summaries the relevancy of each topic to the query $q$. For a judgment $J$, LDA produces a judgment-topic probability vector $p_J = [t_{J,1}, \ldots, t_{J,i}, \ldots]$ where $t_{J,i}$ is the probability that judgment $J$ is relevant to topic $T_i$. Finally, we compute the similarity between query $q$ and judgment $J$ by taking the dot product $p_q \cdot p_J$. Given a query $q$, we return the judgments with the highest similarities as the search results.
2.2.2. Query-Driven Summarization

Given a query \( q \) and a judgment \( J \), our objective is to find a small fraction (e.g., 5\%) of the paragraphs in \( J \) that are the most relevant to \( q \). These selected paragraphs serve as a query-specific summary of \( J \) to \( q \), which helps the user understand whether \( J \) is indeed desired. To achieve that, we first find the most relevant topic \( T_0 \), which is the topic that gives the highest query-topic similarity score, i.e., \( T_0 = \arg \max_T\left(q, t_i\right) \). Next, for each paragraph \( G \) in \( J \), we compute a paragraph semantic vector \( v_G \) by averaging the word2vec embedding vectors of the words in \( G \). The similarity between paragraph \( G \) and topic \( T_0 \) is then measured by the cosine similarity of their semantic vectors, i.e., \( \frac{v_{T_0} \cdot v_G}{\|v_{T_0}\|\|v_G\|} \). Paragraphs with the highest similarities are selected as the summary.

2.2.3. Aspect-Driven Summarization

In Section 2.1 we discussed three ways of generating topics. In particular, with DK-A, topics are grouped into aspects (see Figure 4(c)). Given a judgment \( J \) and an aspect \( A \), our objective is to find a small number of paragraphs in \( J \) that best describe the case w.r.t. aspect \( A \). For simplicity, we explain our approach assuming that plaintiff’s background is the aspect of interest. Our method can be generalized to cover any other given aspect. Let \( T_B = \{T_1, \ldots, T_M\} \) be a set of \( M \) topics generated by the DK-A model under the “plaintiff’s background” aspect. For a judgment \( J \), we consider its judgment-topic probability vector \( p_J \) (see Section 2.2.1) and find the topic in \( T_B \) that gives the highest probability among those in \( p_J \). We denote this top-ranked topic \( T_J \). Formally, \( T_J = \arg \max_{T \in T_B} t_{J, T} \). Next, we measure the similarity between each paragraph \( G \) in judgment \( J \) and the topic \( T_J \) in the same way as we did in query-driven summarization, i.e., by the cosine similarity of \( v_G \) and \( v_{T_J} \). Paragraphs with the highest similarities are selected as the summary.

3. Evaluation

In this section, we present the evaluation of our topic modeling approaches. In particular, we give experimental results comparing different topic generation methods using query-driven summarization as the target application.

We collected 832 judgments on personal injury (PI) compensation cases handed down in Hong Kong from 1999 to 2021. The judgments contain 606 to 11,257 words each, with an average length of 4,552 words. Our legal experts suggested 79 features for PI cases, among which 48 are of interests to this study. These 48 features are grouped into four aspects, namely, "background" (11 features), "injury" (9 features), "treatment" (8 features), and "loss" (20 features). We hired 10 law students to manually label these features in the judgments. The data is used to derive topic models under the NoDK, DK-F, and DK-A approaches (see Figure 4).

To evaluate the quality of the query-specific summaries provided by each method, we prepared a set of 20 test queries that search for PI judgments. These queries are real user queries extracted from the HKLII search log. We submitted each query \( q \) to HKLII search engine and retrieved the top-ranked PI judgment \( J \) after filtering out those in the search results that were irrelevant to \( q \). This gave us a query-judgment \((q, J)\) pair. We then applied query-driven summarization (Section 2.2.2) to determine a similarity score of each paragraph in \( J \) w.r.t. the query \( q \). The paragraphs were then ranked based on their
similarity scores and the top 5% of the paragraphs were selected as the summary of \( J \).
We considered 3 approaches to generate topics, namely, NoDK, DK-F, and DK-A. Each of them resulted in a summary, which might differ from those of others. Hence, we got three summaries for each \( q-J \) pair corresponding to the three methods.

We recruited 8 legal experts (who either have the Postgraduate Certificate in Laws qualification or are currently practising law) to evaluate the summaries. Given a \( q-J \) pair, we merged the three summaries obtained from the methods into a single collection and presented the paragraphs to an expert for “grading”. The expert was asked to read the query and the judgment, and then assign a score of ‘0’ (not relevant), ‘1’ (somewhat relevant), or ‘2’ (relevant) to each paragraph in the collection. During the process, the expert was totally blind to which method was used to select the paragraphs. Each \( q-J \) pair was graded by 1 to 3 experts.

We evaluate a summary’s quality by comparing it with the optimal summary using the normalized discounted cumulative gain (nDCG) metric [5]. Specifically, let \( S_X = \{ G_1, G_2, \ldots, G_k \} \) be a summary of \( k \) paragraphs taken from a judgment \( J \) by method \( X \) (\( X = \text{NoDK}, \text{DK-F}, \text{or DK-A} \)), such that the paragraphs \( G_i \)’s are sorted in decreasing order of their similarities with the identified topic (i.e., \( \frac{v_T \cdot v_{G_i}}{\| v_T \| \| v_{G_i} \|} \), see Section 2.2.2). Let \( s(G_i) \) be the average relevancy score of \( G_i \) given by the human assessors. The DCG score of summary \( S_X \) is given by \( \text{DCG}(S_X) = \sum_{i=1}^{k} \frac{s(G_i)}{\log_2(i+1)} \). The (theoretical) optimal summary, denoted by \( \hat{S} \), is constructed by collecting paragraphs in \( J \) that are given the highest average relevancy scores by the assessors until \( k \) paragraphs are collected. The nDCG score of summary \( S_X \) is then given by \( \frac{\text{DCG}(S_X)}{\text{DCG}(\hat{S})} \). Note that nDCG scores of summaries range from 0 to 1, with 1 indicating that the summary matches the optimal one perfectly in selecting paragraphs and assessing their relevancy to the query.

Table 1 shows the average nDCG scores of the summaries obtained by the three different methods. Moreover, we consider summaries with nDCG \( \geq 0.75 \) (\(< 0.5 \)) to be of good (poor) quality. Table 1 shows the number of good/poor summaries for each method. From the table, we see that DK-A, which considers domain knowledge of different PI aspects, significantly outperforms NoDK and DK-F. First, DK-A has a very high average nDCG score (0.86) compared with NoDK (0.66) and DK-F (0.64). Secondly, DK-A produces 18 good summaries for the 20 query-judgment pairs and no poor summaries. The reason for DK-A’s excellent performance is that it generates topics with respect to different aspects. That allows DK-A to generate more topics than other methods and the topics are more precise and focused. Table 1 further shows that NoDK and DK-F have comparable performance in terms of average nDCG scores. On closer inspection, we find that there is not a clear advantage of one over the other; For some \( q-J \) pairs, NoDK gets better scores, while DK-F is better for other \( q-J \) pairs. As we mentioned in Section 2.1, DK-F ignores unlabeled text in generating topics. That helps remove unimportant content in judgments and improve topic modeling. Occasionally, however, DK-F is too aggressive and some content that is useful in generating topics is inadvertently removed, resulting in poor summary quality.

Figure 5 shows a screenshot of our query-specific summary design. A user types a query in a search box (top of left panel). The retrieved judgment is shown in the right panel with paragraphs in the summary highlighted. Excerpts of the summary paragraphs
are collected and displayed in the lower part of the left panel. The user can read the paragraphs excerpts in the summary to determine if the retrieved judgment is relevant to his/her search intent. By clicking on a paragraph excerpt, the system will display the corresponding paragraph in the judgment in the right panel. This allows the user to read the context of the summary paragraphs for further details.

4. Conclusion

In this paper we studied the problem of effective semantic search and judgment summarization in digital legal library systems. We proposed a general framework to achieve the tasks through topic modeling. We considered three approaches (NoDK, DK-F, and DK-A) of generating topics. We also proposed algorithms for generating query-specific and aspect-specific judgment summaries, and algorithms for performing semantic search.

Acknowledgement

This work is supported by HKU-TCL Joint Research Center for AI and Innovation and Technology Fund (Grant ITS/234/20).

References

[1] World Legal Information Institute. WorldLII Website; 2021. https://www.worldlii.org/.
[2] Blei DM, Ng AY, Jordan MI. Latent dirichlet allocation. JMLR. 2003;3:993–1022.
[3] McCallum AK. MALLET: A Machine Learning for Language Toolkit; 2002.
[4] Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J. Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems; 2013. p. 3111–3119.
[5] Järvelin K, Kekäläinen J. Cumulated gain-based evaluation of IR techniques. ACM Transactions on Information Systems (TOIS). 2002;20(4):422–446.