

UB at FIRE 2020 Precedent and Statute Retrieval

Tebo Leburu-Dingalo^a, Nkwebi Peace Motlogelwa^a, Edwin Thuma^a and
Monkgogi Modungo^a

^aDepartment of Computer Science, University of Botswana

Abstract

In this paper we explore several retrieval strategies in an attempt to identify relevant statutes and prior cases using a description of a current situation (current case). In particular, we investigate whether we can improve the retrieval performance of a precedent retrieval system by indexing only the key concepts in the prior case documents. In addition, we investigate whether we could improve the retrieval performance by expanding the original queries and performing retrieval on a summarized document collection. The results suggest that expanding the current case can improve the retrieval performance when the retrieval is performed on a summarised document collection of prior cases. For statute retrieval, we investigate whether the retrieval performance could be improved by extracting only the key concepts from the queries or by expanding the queries without summarising the statute documents. The results of this study suggest that summarising the current case can improve the retrieval performance of a statute retrieval system.

Keywords

Precedent Retrieval, Statute Retrieval, Text Summarization

1. Introduction

The development of tools that enable users access to digitized legal content has been a major focus area of research since the 1960's [1]. Central to this has been the provision of search and retrieval strategies that are effective in returning the most relevant information given the vast amount of digital content being generated daily in the legal field. To this end a notable number of Information Retrieval(IR) systems have been developed for the legal field that borrow heavily from the Natural Language Processing (NLP) and Machine Learning (ML) techniques already proven effective in general search tasks. Activities that have benefited in this regard include the retrieval of relevant statutes and precedents that can be used as references for an ongoing case. Statutes are written laws referred to by judges in support of judicial decisions [2]. Precedents on the other hand are meant to support the principle that obliges courts to follow decision reached in historical cases when making a ruling on a similar case [3]. However as noted by Carvalho et al. [4], Maxwell and Schaefer [5] and Yoshioka et al. [6], majority of systems developed for the legal field inclusive of precedence and statute retrieval systems are not able to achieve expected level of performance even when implemented with approaches proved effective in general search tasks. This general lack of performance by IR systems in the legal domain has been attributed to several factors including the inherent complex syntax and domain specificity of language used in law documents [4].

Forum for Information Retrieval Evaluation 2020, December 16-20, 2020, Hyderabad, India


✉ leburut@ub.ac.bw (T. Leburu-Dingalo); motlogel@ub.ac.bw (N.P. Motlogelwa); thamae@ub.ac.bw (E. Thuma); mudongom@ub.ac.bw (M. Modungo)

🌐 <https://www.ub.bw> (T. Leburu-Dingalo); <https://www.ub.bw> (N.P. Motlogelwa); <https://www.ub.bw> (E. Thuma); <https://www.ub.bw> (M. Modungo)



© 2020 Copyright for this paper by its authors.

Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

Another factor identified in this regard is the tendency of law documents to be long and wordy, which could impact retrieval performance when used as queries (current case). As several studies have shown, longer or verbose queries are more difficult to process by IR systems when compared to their shorter version. Bendersky and Croft [7] illustrate this in their exploration of a probabilistic model for verbose queries using the newswire and web collections. The effectiveness of shorter queries against longer queries is further confirmed by Huston and Croft [8] in their evaluation of query processing techniques against data drawn from Yahoo! Answers CQA service. Research efforts towards improving effectiveness of IR systems in the legal domain are currently being supported by several international initiatives such as the Forum for Information Retrieval Evaluation (FIRE)¹ platform. To achieve this, the platform avails datasets against, which researchers can develop and evaluate comparable IR systems. The datasets are availed through a series of tasks that address different aspects of legal information retrieval.

In this paper, we present our work that we submitted for participation at the Artificial Intelligence for Legal Assistance (AILA)², which is a series of shared tasks aimed at developing datasets and methods for solving variety of legal informatics problem [9]. In particular, we participated in Task 1, which focuses on precedent and statute retrieval. Precedent retrieval Task 1A focuses on the identification of relevant prior cases for a given a legal situation representing a current case. Statute retrieval Task 1B focuses on the identification of the most relevant statutes for a given legal situation. Our approach explores the effectiveness of using shortened versions of both the query and document texts as opposed to their original versions. To this end we deploy text summarization to find the most informative terms to act as representatives for queries and documents in both retrieval tasks. The remainder of this paper is organized as follows in Section 2 we present related work. Section 3 describes the methods used in this study. In Section 4 we discuss our experiments. Section 5 discusses our results and discussion.

2. Related Work

2.1. Statute and Precedence Retrieval Systems

Statute laws and precedents serve an important role in countries that follow the common law systems. Statutes enable judges to apply legal principles when handling a case while precedents or prior decided cases allow them to reach similar decisions for subsequent cases with similar issues or circumstances. Additionally lawyers are able to use the resources as references when preparing for a case. Several statute and precedence retrieval systems have since been proposed aimed at enabling judges and lawyers timely access to these resources. Zhao [10] use a combination of IDF and improved BM25 to implement a competitive method for precedence retrieval. The BM25 model is enhanced by using relevance scores of both the original and filtered case. The query case is filtered by selecting the top ranking scored query terms based on IDF scores. Thenmozhi [11] deploy the use of Parts of Speech (POS) tagging and a vector space model to implement a model for precedent retrieval. The method uses both concepts and relationships from text as features. A feature vector is constructed for each document based on TF-IDF scores. Prior cases are then retrieved and ranked for each current case based on a cosine similarity measure. Shao et al [12] obtain relative success with a vector space based model for statute retrieval. The authors use both the original query and summary of the query generated using TextRank. Candidate statutes are constructed using both the title and the description

¹<http://fire.irsi.res.in/fire/2020/home>

²<https://sites.google.com/view/aila-2020>

of the statutes.

2.2. Query Reduction in Legal Retrieval

Reduction of verbose or lengthy queries in an effort to improve retrieval performance is an approach that has been adopted by many in the literature. Driving this research is the fact that many studies show that systems tend to perform better for shorter versions of queries as illustrated by [7] and [8]. Many strategies advanced towards legal retrieval thus deploy summarization techniques that seek to represent a document with a subset of the most informative terms or key concepts from the document. Thuma et. al [13] demonstrate the efficacy of this approach in a statute retrieval task. The authors observe notable improvement in system performance when TagCrowd is used to generate query terms using key concepts derived from a longer description of a query case. A degradation in performance is further observed if the summarized query is expanded with informative terms from the corpus. Rossi and Kanoulas [14] combine text summarization and a generalized language model BERT to measure pairwise similarity between documents in a legal retrieval task. Text in this work is summarized using a graph-based algorithm TextRank. Sandeep and Bharadwaj [15] obtain summarized versions of case documents by filtering out insignificant terms based on a predefined threshold. The significance of a term is determined by a linear combination of the term's frequency and its POS tag weight. A nearest neighbour approach is then used to determine similarity between the query and candidate documents.

3. Description of Methods

In our experiments, we used *TF-IDF* as the main technique for both document ranking and retrieval and text summarization. A brief description of approaches used is outlined below.

3.1. Term Weighting Model

Our first proposed approach uses the *TF-IDF* term weighting model to rank and retrieve documents (prior cases/ statutes). *TF-IDF* is a numerical statistic that is calculated by taking the product of two components; term frequency (*tf*) and inverse document frequency (*idf*). *tf* refers to the number of times term *t* occurs in document *d* [16]. The basic *idf* calculation is as follows:

$$idf(t) = \log \frac{N}{n_i} \quad (1)$$

where *N* is the total number of documents in collection *C*, and *n_i* is the number of documents the term *t* occurs in.

3.2. Text summarization algorithm with *TF-IDF*

The text summarization algorithm³ we used run on the Python Natural Language ToolKit (NLTK)⁴ and uses the *TF-IDF* algorithm. The algorithm computes a score for each sentence as the sum of *TF-IDF* scores of each word in the sentence as shown below:

$$sentence_{score} = \sum_{i=first\ word}^{last\ word} TF-IDF_{word_i}$$

³<https://towardsdatascience.com/text-summarization-using-tf-idf-e64a0644ace3>

⁴<https://www.nltk.org/>

The algorithm summarizes only those sentences with a sentence score greater than the threshold. The threshold is computed as the average score for sentences as follows:

$$threshold = \left(\sum_{i=first\ sentence}^{last\ sentence} sentence_i score \right) / (number\ of\ sentences)$$

We used the training queries to select the optimal threshold to use. In particular, we conducted several experiments in which we varied the threshold, then performing actual retrieval, and lastly, evaluating the retrieval performance. The most effective threshold of 0.35 was then used with the test datasets to perform the actual retrieval.

4. Experimental Setting

FAQ Retrieval Platform: For all our experiments, we used Terrier-4.2⁵ [17], an open source Information Retrieval (IR) platform. All the documents used in this study were first pre-processed before indexing and this involved tokenising the text and stemming each token using the full Porter stemming algorithm [18]. A comprehensive description of the test collection used in this study can be found in Bhattacharya et. al [9].

4.1. Task 1A: Precedent Retrieval

A baseline retrieval was conducted using Terrier 4.2, the original prior case documents and the original test queries using *TF-IDF* as the term weighting model (**UB-1**). The second experiment used summarised prior case documents to improve retrieval effectiveness by extracting only key concepts from the prior case documents (**UB-2**). In the final run, we investigate whether we could improve retrieval effectiveness by expanding the original queries with the top 10 terms selected from the top 3 ranked documents after the first pass retrieval (**UB-3**). We performed the retrieval on the summarised prior case documents. For query expansion we used the Terrier 4.2 Bo1 model for query expansion to select the expansion terms.

4.2. Task 1B: Statute Retrieval

A baseline retrieval was conducted using Terrier 4.2, the original test corpus and the original test queries using *TF-IDF* as the term weighting model (**UB-1**). The second experiment used summarised queries to improve retrieval effectiveness by extracting only key concepts from the queries (**UB-2**). In the final run, we investigate whether we could improve retrieval effectiveness by expanding the summarised queries with the top 10 terms selected from the top 3 ranked documents after the first pass retrieval (**UB-3**). For query expansion we used the Terrier 4.2 Bo1 model for query expansion to select the expansion terms.

5. Results and Discussion

Our results from both experiments were submitted to the AILA 2020 competition for evaluation by the organizers. The evaluation for Task 1A and Task1B uses MAP, BPREF, recip_rank and P@10. The results of Task 1A and Task1B based on the aforementioned evaluation measures are shown in

⁵<http://terrier.org/>

Table 1

Task 1A: Precedent Retrieval

Run ID	MAP	BPREF	recip_rank	P@10
UB-1	0.1229	0.07	0.2033	0.09
UB-2	0.1168	0.0798	0.1967	0.07
UB-3	0.1573	0.1128	0.238	0.08

Table 2

Task 1B: Statute Retrieval

Run ID	MAP	BPREF	recip_rank	P@10
UB-1	0.3085	0.2633	0.573	0.14
UB-2	0.3134	0.2633	0.5787	0.15
UB-3	0.1876	0.1502	0.2468	0.09

Table 1 and Table 2 respectively. Our third approach (UB-3) in Task 1A, which uses expanded queries for retrieval on a summarised document collection of prior cases performed better than all the other systems that participated in the task across three evaluation measures (MAP, BREF and recip_rank, Table 1) . For statute retrieval (Task 1B), an improvement in the retrieval performance was attained after summarising the current case (Table 2). Our approach performed better than all other teams that participated in the task when systems were only evaluated using recip_rank, which was **0.5787** (Table 2). Overall, the results of this study and also previous works suggest that document and query summarisation can improve the retrieval performance for both statute and precedent retrieval. To develop a full picture of the effects of summarization, additional studies will be needed that empirically evaluate and develop novel document summarisation techniques tailored for the legal domain.

References

- [1] J. Bing, Performance of legal text retrieval systems: The curse of boole, *Law. Libr. J.* 79 (1987) 187.
- [2] P. Bhattacharya, K. Ghosh, S. Ghosh, A. Pal, P. Mehta, A. Bhattacharya, P. Majumder, Fire 2019 aila track: Artificial intelligence for legal assistance, in: *Proceedings of the 11th Forum for Information Retrieval Evaluation, FIRE '19*, Association for Computing Machinery, New York, NY, USA, 2019, p. 4–6. URL: <https://doi.org/10.1145/3368567.3368587>. doi:10.1145/3368567.3368587.
- [3] L. K. Branting, A reduction-graph model of precedent in legal analysis, *Artificial Intelligence* 150 (2003) 59–95.
- [4] D. S. Carvalho, V. D. Tran, V.-K. Tran, L.-M. Nguyen, Improving legal information retrieval by distributional composition with term order probabilities., in: *COLIEE@ ICAIL, 2017*, pp. 43–56.
- [5] K. T. Maxwell, B. Schafer, Concept and context in legal information retrieval, in: *Proceedings of the 2008 Conference on Legal Knowledge and Information Systems: JURIX 2008: The Twenty-First Annual Conference*, IOS Press, NLD, 2008, p. 63–72.
- [6] M. Yoshioka, Y. Kano, N. Kiyota, K. Satoh, Overview of japanese statute law retrieval and en-

- tailment task at coliee-2018, in: Twelfth International Workshop on Juris-informatics (JURISIN 2018), 2018.
- [7] M. Bendersky, W. B. Croft, Discovering key concepts in verbose queries, in: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '08, Association for Computing Machinery, New York, NY, USA, 2008, p. 491–498. URL: <https://doi.org/10.1145/1390334.1390419>. doi:10.1145/1390334.1390419.
 - [8] S. Huston, W. B. Croft, Evaluating verbose query processing techniques, in: In Proc. of SIGIR, SIGIR '10, 2010, pp. 291–298.
 - [9] P. Bhattacharya, P. Mehta, K. Ghosh, S. Ghosh, A. Pal, A. Bhattacharya, P. Majumder, Overview of the FIRE 2020 AILA track: Artificial Intelligence for Legal Assistance, in: Proceedings of FIRE 2020 - Forum for Information Retrieval Evaluation, 2020.
 - [10] Z. Zhao, H. Ning, L. Liu, C. Huang, L. Kong, Y. Han, Z. Han, Fire2019@aila: Legal information retrieval using improved BM25, in: FIRE (Working Notes), volume 2517 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2019, pp. 40–45.
 - [11] D. Thenmozhi, K. Kannan, C. Aravindan, A text similarity approach for precedence retrieval from legal documents., in: FIRE (Working Notes), 2017, pp. 90–91.
 - [12] Y. Shao, Z. Ye, Thuir@aila 2019: Information retrieval approaches for identifying relevant precedents and statutes, in: FIRE (Working Notes), volume 2517 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2019, pp. 46–51.
 - [13] E. Thuma, N. P. Motlogelwa, T. Leburu-Dingalo, M. Mudongo, Query reduction for an effective japanese statute law retrieval, in: 2019 Conference on Next Generation Computing Applications (NextComp), 2019, pp. 1–4. doi:10.1109/NEXTCOMP.2019.8883643.
 - [14] J. Rossi, E. Kanoulas, Legal information retrieval with generalized language models, Proceedings of the 6th Competition on Legal Information Extraction/Entailment. COLIEE (2019).
 - [15] G. Sandeep, S. Bharadwaj, An extraction based approach to keyword generation and precedence retrieval: Bits pilani-hyderabad., in: FIRE (Working Notes), 2017, pp. 74–77.
 - [16] S. Robertson, Understanding inverse document frequency: on theoretical arguments for idf, *J. Documentation* 60 (2004) 503–520.
 - [17] I. Ounis, G. Amati, P. V., B. He, C. Macdonald, Johnson, Terrier Information Retrieval Platform, in: Proceedings of the 27th European Conference on IR Research, volume 3408 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, Heidelberg, 2005, pp. 517–519.
 - [18] M. Porter, An Algorithm for Suffix Stripping, *Readings in Information Retrieval* 14 (1997) 313–316.