Automatic cited decision retrieval: Working notes of Ielab for FIRE Legal Track Precedence Retrieval Task

Daniel Locke
Queensland University of Technology
Brisbane, QLD, AUS
daniel.locke@hdr.qut.edu.au

Guido Zuccon
Queensland University of Technology
Brisbane, QLD, AUS
guido.zuccon@qut.edu.au

ABSTRACT

Case law, in a common law domain, provides for statements of principle that bind courts as to the manner in which like cases are to be decided. Citations of previous cases are as a result, frequent in judicial writing. The FIRE Information Retrieval from Legal Documents Precedence Retrieval task concerned finding cited documents, where the name of the cited decision had been removed [4]. In this paper, we discuss our three automated methods to find cited cases which achieved the highest scores in all reported measures.

CCS CONCEPTS

• **Information systems → Information retrieval**; **Specialized information retrieval**;

KEYWORDS

Information retrieval, expert systems, legal information retrieval, automatic query reduction

1 INTRODUCTION

Previous legal decisions establish binding precedent for factually similar matters. Finding these decisions is important so that lawyers can properly discharge their duties to the Court. The FIRE Information Retrieval from Legal Documents Precedence Retrieval task concerned finding cited documents where the name of the cited decision had been removed. The object of the task was, given a decision that cited previous decisions the name of which had been removed, to find and rank these cited decisions higher than decisions that were not cited.

The task at hand is different to a traditional legal citation extractor. Mowbray [5] for instance involves lexical parsing of text to identify citations, and linking citations to decisions. In this task we are not provided with the citation, but instead we are only provided with surrounding text. Accordingly, such methods are unsuitable; our task is merely a retrieval task.

We utilise automated methods for the identification of these cited decisions from the surrounding text. Namely, we use three simple methods each based on the text surrounding a citation being: (I) the text itself as a baseline; (ii) proportional inverse document frequency (IDF-r) [2]; and (iii) parsimonious language models (PLM) [1]. Our methods, are in essence an application of our earlier work in [3]. To the same extent, our methods again are similar to those used in the recent study by Koopman et al. [2], which investigated generating clinical queries from patient narratives, in that they also used proportional IDF (IDF-r) for query term selections.

In our earlier work in [3], we explored the performance of these methods on our own collection, in addition to the effect of different text lengths for queries. We concluded that: (i) longer queries led to better performance; (ii) of the automated methods evaluated, proportional PLM and IDF outperformed KLI; and (iii) the impact of the smoothing parameter, λ in PLM had little effect within a certain range. In line with these conclusions, we chose to evaluate PLM and IDF, as well as a large amount of surrounding text as a baseline.

The paper continues as follows. In Section 2 we describe our methods and empirical setup, and in Section 3 we briefly describe our results in the task.

2 METHODOLOGY

We submitted three runs: (i) `flt_ielab_para`; (ii) `flt_ielab_idf`; and (iii) `flt_ielab_plm`. Each of these methods is fully automatic.

For each method, we start with the following. We take the position in the text of the removed citation by finding '[?CITATION?]'. We then take surrounding text by finding, from either side of the citation, 40 spaces, 5 periods or 2 carriage returns. We chose to take such a large amount of surrounding text as a result of our findings in [3] that longer queries led to better performance. While in our earlier work the average length of a sentence was 47 words, and paragraph was 148 words, we chose a smaller number than this. In our earlier work the paragraphs were manually selected, and as a general observation, paragraphs of decisions of the United States Supreme Court appeared to be longer. In this task we included measures such as the number of carriage returns and number of periods to ensure that we should be obtaining the text from a paragraph.

Following this, we cleaned the surrounding text for each citation by removing the [?CITATION?] text, removing all punctuation and removing stopwords. We used as our stopword list the standard list provided in Elasticsearch. We keep any numbers found in this text for the reason that ad decision may refer to sections of legislative texts. These queries are then parsed through each method (as described below), and then evaluated in Elasticsearch.[1] As our retrieval function we used BM25, with 'b' set to 0.75 and 'k' set to 1.2.

For each topic, we evaluated each query as a standard best match query. We return the top 1000 documents for each query. For each topic, if more than one cited case was to be found, i.e. there was more than one [?CITATION?] present in the text, for each citation we retrieved 1000 documents, and then sorted the documents by score to return the top 1000 unique documents for the topic. Where

[1] We used version 5.4.2, available at www.elastic.co.

the same document was returned by multiple queries, we kept only its highest score.

For flt_ielab_para, as a baseline, we took all terms that remain in the surrounding text after removal of stopwords and punctuation as the query.

For flt_ielab_idf we ranked each term in the surrounding text by its IDF score. We then took the 50% of the terms with the highest rank as our query.

For flt_ielab_plm, as with flt_ielab_idf, we ranked each term in the surrounding text by its probability from a parsimonious langauge model. Again, we took the 50% of the terms with the highest probability as the query. Probabilities were estimated using the expectation maximization algorithm, with the steps being:

$$E - step: \quad e_t = tf(t, D)\frac{\lambda P(t|D)}{(1-\lambda)P(t|C) + \lambda P(t|D)} \tag{1}$$

$$M - step: \quad P(t|D) = \frac{e_t}{\sum_{t' \in D} e_{t'}} \tag{2}$$

We set $\lambda \in [0, 1]$ at 0.5, as per our earlier findings in [3] that the paramater had little effect in a similar task. We used the 2000 prior cases as the background language model, $P(t|C)$, and the surrounding text as the foreground language model, $P(t|D)$.

3 RESULTS

	MAP	MRR	P@10	Recall@10
flt_ielab_para	0.3637	0.7017	0.2211	0.7487
flt_ielab_idf	0.3902	0.7193	0.2362	0.7809
flt_ielab_plm	0.3859	0.7097	0.2367	0.7709
next best	0.3291	0.6325	0.218	0.681

Table 1: Effectiveness of automatically generated queries for our runs, including the next best result

Our runs performed the best for the task in all measures, with IDF being the highest result in MAP, MRR and Recall@10, and PLM being the highest in P@10. The next best run was the next best in all evaluation measures with the exception of R@10, where one other run also achieved a score of 0.681.

Our results are interesting in so far as PLM is outperformed by IDF in all measures except P@10. This is in contrast to our earlier findings in a similar task in [3]. While we did not measure P@10 nor Recall@10 in our earlier work, we saw that PLM outperformed IDF in all measures, including P@5 and MRR where a longer text input was considered. The measures we chose in our work were different in so far as we did not view the task in that work as a recall orientated; we were concerned with finding only a small number of decisions, and thus we were concerned with measures such as P@1 and P@5.

In line with our earlier findings, we also see that large information objects leads to decent performance. While we do not know the length of other queries used by other teams, from the high performance of our baseline para run we infer that other teams evaluated shorter queries.

REFERENCES

[1] D. Hiemstra, S. Robertson, and H. Zaragoza. Parsimonious language models for information retrieval. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 178–185. ACM, 2004.

[2] B. Koopman, L. Cripwell, and G. Zuccon. Generating clinical queries from patient narratives. In *Proceedings of the 40th international ACM SIGIR conference on Research and development in information retrieval*, 2017 (to appear).

[3] D. Locke, G. Zuccon, and H. Scells. Automatic Query Generation from Legal Texts for Case Law Retrieval. In *Information Retrieval Technology: 13th Asia Information Retrieval Societies Conference, AIRS 2017, Jeju, Korea, November 22 – November 25, 2017, Proceedings*, LNCS. Springer International Publishing AG, December 2017.

[4] A. Mandal, K. Ghosh, A. Bhattacharya, A. Pal, and S. Ghosh. Overview of the FIRE 2017 track: Information Retrieval from Legal Documents (IRLeD). In *Working notes of FIRE 2017 - Forum for Information Retrieval Evaluation*, CEUR Workshop Proceedings. CEUR-WS.org, December 2017.

[5] A. Mowbray, P. Chung, and G. Greenleaf. A free access, automated law citator with international scope: the lawcite project. *European Journal of Law and Technology*, 7(3), 2016.