Automatic Query Generation from Legal Texts for Case Law Retrieval

Daniel Locke(✉), Guido Zuccon, and Harrisen Scells

Queensland University of Technology, Brisbane, Australia
{daniel.locke,harrisen.scells}@hdr.qut.edu.au, g.zuccon@qut.edu.au

Abstract. This paper investigates automatic query generation from legal decisions, along with contributing a test collection for the evaluation of case law retrieval. For a sentence or paragraph within a legal decision that cites another decision, queries were automatically generated from a proportion of the terms in that sentence or paragraph. Manually generated queries were also created as a ground to empirically compare automatic methods. Automatically generated queries were found to be more effective than the average Boolean queries from experts. However, the best keyword and Boolean queries from experts significantly outperformed automatic queries.

1 Introduction

In common law jurisdictions, such as the United Kingdom, United States, and Australia, where the doctrine of precedent (*stare decisis*) is applicable, finding relevant and therefore binding legal principles is crucially important so that lawyers can discharge their duties to the court.[1]

Finding factually or legally applicable case law forms a large part of a lawyer's work. Studies have found that lawyers spend roughly 15 h per week finding case law, or as much as 28% of their yearly working hours [12,14]. With research playing a fundamental role in a lawyer's work, increasing the quality of legal research tools is of great importance. Despite this, little research has been conducted as to retrieval of case law, previous research has typically focused on either argumentation retrieval, ontological frameworks and case-based reasoning or retrieval in a discovery setting.[2]

This paper explores the use of automatic methods for the generation of queries for case law retrieval. These methods could be integrated: (i) in a contextual suggestion system that provides lawyers with relevant cases as they write; or (ii) within search functionalities, to support lawyers in the formulation of effective queries.

[1] The doctrine of precedent requires, broadly speaking, that like circumstances are considered in a like fashion; a case that considers a certain set of factual circumstances therefore must be followed for any future circumstances that are analogous.

[2] The obligation of parties to litigation to disclose all documents relevant to issues between them.

W.-K. Sung et al. (Eds.): AIRS 2017, LNCS 10648, pp. 181–193, 2017.
https://doi.org/10.1007/978-3-319-70145-5_14

As automated methods, we evaluated a number of common keyword extraction methods, namely proportional inverse document frequency (IDF-r) [10], Kullback-Leibler divergence for informativeness (KLI) [17], and parsimonious language models (PLM) [6]. These methods aim to select appropriate terms as candidate queries from portions of legal documents.

We compared the investigated automatic methods to the original sentences and paragraphs that reference a previous legal decision. As an additional comparison, we also collected a number of queries (Boolean and best-match based) manually built by a legal expert. A new test collection was created to empirically evaluate the methods and investigation, and relevance assessments were made by a legal expert. This collection, which comprises of 63,916 documents, 100 topics, 248 manually created queries and a total of 2645 relevance assessments, is an additional contribution of this paper to the research in case law retrieval.

The paper continues as follows. In Sect. 2 we describe related work, including highlighting the lack of an adequate test collection for evaluating retrieval and query generation methods for case law retrieval. In Sect. 3 we describe the query generation methods investigated in this work, along with details on the creation of the test collection. In Sect. 4 we describe the experimental settings and the results of the empirical evaluation. Section 5 concludes this paper and provides an account of future work.

2 Related Work

Legal Information Retrieval. Early work in case law retrieval has focused on inference networks and comparing natural language and Boolean queries for retrieval of case law [18]. More broadly, and more recently, legal information retrieval (IR) has focused on: (1) question answering (Q&A) (the ResPubliQA task, which considered Q&A tasks involving approximately 10,000 legislative texts of the European Union [13], and the COLIEEE collection, which involved Q&A over a smaller number of Articles of the Japanese Civil Code [7], are both recent examples of this); (2) discovery, with TREC conducting from 2006, the Legal Track [2]; (3) argument retrieval (for instance see the work of Grabmair et al. [5]); and (4) network analysis of citations (for instance see van Opijnen [20]).

Lack of a test collection for Legal IR. The most notable gap in the area of Legal IR is the lack of an existing standardised test collection. While legal decisions are now more easily accessible and more freely available, no standard corpus exists for testing information retrieval of case law. This can, perhaps, be contrasted with the test collections used in the TREC legal discovery tracks.

Only two collections that contain case law have been created for legal IR [8,9], both of which were used for analysis of diversification. Koniaris et al. [8], who also acknowledged the lack of “standard testing data”, created the largest collection, which contains 63,000 decisions of the United States Supreme Court, 300 queries and automatically generated relevance judgments (see below for details). This collection takes as its queries, a subset of the areas of law found in the Westlaw

Digest.[3] These queries, however, are unsuitable for the task investigated in this paper. In Table 1, we have compared queries used by Koniaris et al. [8] with those of Turtle [18], which represent queries created by an expert searcher (a lawyer), rather than extracted from the Westlaw Digest; this shows the artificial nature and the broad scope of the queries from Koniaris et al. Note that the collection compiled by Turtle is not publicly available. The table also shows for comparison a query from our collection that was created by a domain expert.

Table 1. Queries previously employed in legal information retrieval studies and an example of our queries. Koniaris et al.'s [8] queries were artificially created from Westlaw Digest topics. Turtle's [18] queries were created by lawyers.

Source	Query	Generation method
Koniaris et al. [8]	Products liability	Topic in Westlaw Laws of America Digest
Turtle [18]	(741 +3 824) FACTOR ELEMENT STATUS FACT /P VESSEL SHIP BOAT /p (46 +3 688) "JONES ACT" /P INJUR! /S SEAMAN CREWMAN WORKER	Manually created by expert searcher
This work	"sovereign immunity" AND (immunity OR indemnif!) AND state AND suit AND (surrend! OR exist!) AND (tribe OR tribal OR "indian trib!")	Manually created by expert searcher

Furthermore, the method used by Koniaris et al. [8] to generate relevance judgments does not lend itself to the task investigated in this paper. In that work, relevance judgments were created automatically from an LDA topic model created over the top-n results of each query, after which an acceptance threshold of 20% was taken to obtain a relevance decision for a given query. An alternative, realistic, determination of query relevance would be preferable: in this paper we use a domain expert to provide relevance assessments.

Queries in legal IR. Schweighofer and Geist [16] note that, unlike in other tasks, the effectiveness of Boolean queries in the legal domain might not be inferior to that of queries in best match retrieval. This is because, they argue, lawyers are domain experts and will necessarily have knowledge of synonyms, without which effectiveness may suffer. However, as they note, domain knowledge has its limits, and one cannot reasonably know all other possible choices for a word. Turtle makes similar comments [18,19], suggesting that the larger the collection searched on, the greater the difficulty in creating an effective Boolean

[3] A keynumber system of categorised areas and subareas of law. Areas of law can be searched or browsed by number.

query. Despite this, Poje [14] found that lawyers in practice for 2 years or less favoured natural language queries, whereas lawyers with more than 2 years practice favoured Boolean queries. For this reason, the collection that we contribute contains both Boolean and keyword queries.

Keyword extraction and query reduction. Keyword extraction (or key terms selection) consists of identifying appropriate terms "capable of representing information content" [15]. The goal of a keyword extraction method is to extract and rank keywords from an information object (a sentence, document or collection) [21]. Verberne et al. have investigated six unsupervised keyword extraction methods across a number of tasks, including from news retrieved for Boolean queries and for which keywords were extracted for the purpose of query suggestion. Keyword extraction methods are relevant to our work because they can be used to select appropriate keywords to form queries that retrieve relevant case law. In legal IR, the task of keyword extraction has been generally referred to as catchphrase or catchword identification; automatic methods for this task have been shown effective for legal document summarisation [4].

Query reduction consists of identifying one or more subsets of an original verbose query that allows for the better retrieval of relevant information. Query reduction is akin, in practice, to keyword extraction in that verbose queries that are to be reduced can be used in place of information objects as inputs of keyword extraction methods. Kumaran and Carvalho's [11] methods rely on the generation of shorter subqueries from an initial query, for which a classifier is used to predict the quality of a given subquery. Bendersky and Croft [3] developed an unsupervised method for extracting key concepts from verbose queries. Both the methods of Kumaran and Carvalho [11] and Bendersky and Croft [3] rely on the generation of permutations of sub-queries. This renders the methods not feasible for (very) long text, like the sentences and paragraphs used in our work (generating all possible sub-queries for text of n terms requires $n!$ combinations). Our work shares some similarities with the recent study by Koopman et al. [10], who investigated the generation of clinical queries from medical narratives. The similarities are that, like us, they also used proportional IDF (IDF-r) for query term selection, and they also studied the automatically generated queries with respect to queries issued by domain experts.

No prior work has examined the application of keyword extraction or query reduction methods for the automatic generation of queries for case law retrieval.

3 Methods

3.1 Creation of Test Collection

The collection contained 63,916 decisions (cases) of the United States Supreme Court (USSC)[4]. For each document, we included the title, the plain text, the HTML, the date the decision was filed, and a list of cited opinions. The HTML

[4] Decisions were downloaded from http://courtlistener.com.

was the whole of the text of a document, generally scanned from a pdf. We created a plain text representation by removing html tags. The average document length (for the plain text field) was 1,918 words.

To create topics for evaluation, we selected 50 cases from the collection, among the most recent in the collection, so that each case referred to two or more topically separate decisions. From these, we manually selected citations of 100 decisions, and built a corresponding set of 100 topics. For each topic, we included the sentence from the original case that cited the decision to be found, and the paragraph in which the sentence was contained. A topic was a sentence in a case (decision) of the United States Supreme Court. A case and therefore 2 topics were included if the 2 decisions that became separate topics:

1. were cited in the opinion of the court rather than the syllabus (court added headnote summary);
2. did not quote the cited case for a significant portion of the sentence;
3. cited a case that was within the collection (i.e. from the USSC);
4. were not for the same case;
5. were not two citations for the same proposition on separate occasions;
6. were not a citation where the court says it granted "*certiorari*"[5];
7. were not in the arguments advanced by a party to the case, unless it was an argument that became stated as a proposition of law;
8. were, if contained in a sentence that cited more than one case for one proposition, included as one citation instance with all other citations for the propostion, provided they met the above criteria;
9. were, if a sentence contained multiple citations for separate clauses, i.e. for different legal propositions, included as separate topics with the separate clauses as the cited case.

We describe the process to generate queries from the topics in the next subsections.

To select the evaluation measures to be used in the empirical experiments we further considered the task at hand. The query generation methods studied here are intended to be used to automatically retrieve relevant cases when lawyers write. In this case, the methods would retrieve relevant prior cases, which would be served as contextual recommendations to the user. These methods can also be used to help lawyers construct queries or suggest queries when exploring a collection of prior legal cases and decisions. This, and the availability of citation indices through which cited decisions can be traced, means in our view, in both scenarios users would only be interested in and inspect a handful of search results/suggestions. Thus, we identify precision at rank 1 and 5 (P@1 and P@5) as suitable evaluation measures for this task. We also consider average precision at rank 5 (AP@5) so as to attribute more importance to ranking relevant decisions early on. In addition, we calculate mean reciprocal rank (MRR), as it indicates, on average, at what rank the first relevant decision is identified. Note

[5] A statement by the Court as to whether it would grant review of a lower court's decision.

that given the way we built our collection, there is always at least one relevant decision given a case (the decision that is cited).

Relevance assessments were created by pooling the manually generated queries, the baseline sentences and paragraphs (used in a standard BM25 system, see Sect. 4.1) and the automatic query generation methods investigated in this work. Pooling was performed to guarantee that runs had nearly complete assessments for the target precision evaluation measures. Assessments were provided by the first author of this paper (a lawyer, but not a legal practitioner) using a purposely created web interface. A total of 2,593 assessments were made.

Relevance was determined on the basis of the extracted sentence for the topic. If the decision being assessed was the cited decision, it was determined to be relevant automatically. If the decision cited the same case as the sentence, and it was for the same or a legally similar proposition it was relevant. Otherwise the first couple of paragraphs of the decision were read to determine the issue of the case, and a keyword search of the decision was made to determine relevance. If the decision was on the topic broadly, being that it would be useful for a lawyer to read or at least skim, then it was classified as relevant. We did not take the full list of cited decisions available for a case as relevant as a case may cite a number of different decisions for any number of different topics.

3.2 Manual Query Generation

As a baseline, for each topic, we evaluated both the topic sentence, and the paragraph that contained the topic sentence, with any of the cited case name removed. On average, sentences contained 47.17 terms, while paragraphs contained 148.13 terms. Sentences and paragraphs were also the information objects given as input to the automatic query generation methods.

For each topic, we also created one to three Boolean queries. These queries were created by the first author of this paper. Boolean queries were used because lawyers commonly use Boolean queries to search for case law in the existing legal systems (see Poje [14], and Sect. 2). A total of 248 Boolean queries were manually created (on average, 2.48 queries per topic) by identifying important keywords from each citation (topic) and from introducing keywords relevant to an area of law based on the first author's domain knowledge.

Boolean queries were also transcribed by removing the Boolean operators to be used in best-match retrieval systems (a match query in Elasticsearch).

Finally, keyword-based queries were created by the legal expert by manually listing keywords that were relevant to the topic or area of law, from a skim read of the issues in the case or from the paragraph or sentence, and included other words where a topic might usually use such words as synonyms. This was done at the same time as the creation of the Boolean queries.

3.3 Automatic Query Generation

For both topic sentences and paragraphs (information objects), we investigated the following automatic query generation methods: (i) the IDF-r method [10],

(ii) the Kullback-Liebler informativeness (KLI) [17], and (iii) the parsimonious language model (PLM) [6].

For all methods, we considered the list of terms T contained in an information object (a sentence or paragraph). The methods are used to produce a subset list T'. This list is produced by ranking the terms according to the specific method used (e.g., by IDF score for the IDF-r method), where scores indicate how relevant a term is for describing the information object, then selecting the top n terms from this ranking, according to a rank cut-off (or proportion parameter). For example, IDF-r selects the $\lceil \frac{|T|}{r} \rceil$ terms in T with the highest IDF score.

For the KLI method, we used statistics for terms in our collection to compute the probability of a term t in an information object D (sentence or paragraph in our case), $P(t|D)$. Conversely, to compute $P(t|C)$ we used statistics for terms in a background language model, computed using a general purpose collection. With these statistics we were able to compute the KLI score for a term; formally:

$$KLI(t) = P(t|D) log \frac{P(t|D)}{P(t|C)} \tag{1}$$

For the PLM method, we used our collection as the background language model, $P(t|C)$, and the information object, D (the sentence or paragraph) as the foreground language model. Probabilities were estimated using the expectation maximization algorithm, with the steps defined as:

$$E - step : \quad e_t = tf(t, D) \frac{\lambda P(t|D)}{(1 - \lambda) P(t|C) + \lambda P(t|D)} \tag{2}$$

$$M - step : \quad P(t|D) = \frac{e_t}{\sum_{t' \in D} e_{t'}} \tag{3}$$

where $\lambda \in [0, 1]$ is a smoothing parameter that controls the influence of statistics from the collection (C) over the statistics from the information object (D).

4 Experiments

4.1 Experimental Settings

We indexed the collection (plain text part) using Elasticsearch version 5.4.2.[6] When indexing, we used Porter stemming and no stop words. As the retrieval function we used the default in Elasticsearch (BM25 with $b = 0.75$, $k1 = 1.2$). We did not tune these parameters as no ground truth was available for tuning at retrieval time (relevance assessments were performed once runs were pooled).

Queries were processed also with Elasticsearch and using the Porter stemmer; stop words were removed from queries. For the Boolean queries, these were formatted using the Elasticsearch query syntax for Boolean queries; all other queries were treated as *match* queries in Elasticsearch (i.e. standard best match).

[6] https://www.elastic.co/.

Sentences (s) and paragraphs (p) were used as information objects from which to generate queries with the automatic methods studied in this paper. Before using the automatic methods, information objects were stripped of stop words. Results were analysed with respect to these different information sources.

For PLM, we explored settings of the smoothing parameter λ[7] between 0.1 (PLM is dominated by the statistics of the collection language model) to 1 (no smoothing: all statistics are computed from the topic sentence or paragraph), with step 0.1. These results are analysed separately (see below), and only the best settings for PLM are compared with the other methods. The background collection was Clueweb12B, while our collection was used as the foreground.

All results were analysed for different levels of the proportion parameter r, which dictates how many of the original terms (from the sentences or the paragraphs) were required to be selected to generate the queries. For example, $r = 0.1$ indicates that 10% of the total number of terms in the information objects were selected (rounded to the upwards integer). Note, $r = 0$ corresponds to selecting one term only.

4.2 Results for Manual Queries

Table 2 reports the retrieval effectiveness for the manually generated queries, along with the sentence (`S`) and paragraph (`P`) baselines. For the Boolean queries, we report both the mean average effectiveness achieved by the different query variations of the topics (`Bavg` – recall that, for every topic, between 1 and 3 Boolean variations were obtained) and the mean of the best effectiveness achieved by Boolean queries for each topic (`Bbest`). The corresponding values are also reported for the Boolean queries from which we removed the Boolean operands (`NBavg` and `NBbest`). Finally, we also report the effectiveness achieved when manually selecting keyword queries from the sentences (`K`).

The results highlight that Boolean queries are largely outperformed by non Boolean queries; specifically `NBbest` outperforms all other manual queries, with `K` also performing above all methods, but `NBbest`. Interestingly, the `Bavg` and `NBavg` are outperformed by querying using the whole of a topic sentence or paragraph. With respect to `Bavg` and `Bbest`, their performance may be hindered as a result of some queries being too specific or restrictive: a total of 36 queries returned no results, and a total of 9 topics returned no results. Further, this, in combination with the performance of `Bbest` and `NBbest` compared to `Bavg` and `NBavg` shows that manually created queries do not perform consistently: the same expert user formulated queries that largely varied in effectiveness. This finding is in line with previous research on query variations in other domains [1,22].

4.3 Tuning of PLM

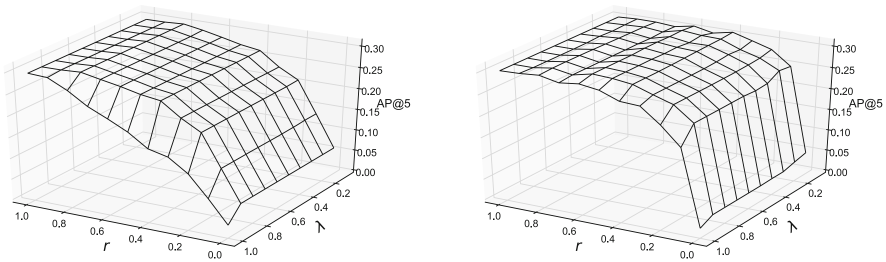
The Parsimonious language model method is characterised by a smoothing parameter that controls the influence of statistics from the collection over the statis-

[7] λ is responsible for smoothing between the background language model (the legal collection), and the foreground language model (the sentence or paragraph).

Table 2. Effectiveness of manual queries and baselines on the 100 topics in the collection. For Boolean queries, multiple queries were devised for each topic: `Bavg` and `NBavg` refer to the mean average effectiveness over all query variations for all topics (with Boolean queries, and with the keywords in the Boolean queries but no Boolean operators); `Bbest` and `NBbest` refer to the average effectiveness for the best query of each topic. Statistical significant differences (paired t-test, $p < 0.05$) compared to the baselines are reported with * (for topic sentences, `S`) and † (for topic paragraphs, `P`).

	P@1	P@5	AP@5	MRR
`Bavg`	$0.4400^{*\dagger}$	$0.3140^{*\dagger}$	$0.1528^{*\dagger}$	$0.4844^{*\dagger}$
`Bbest`	0.7012	0.6500	0.2530	0.4840
`NBavg`	$0.5733^{*\dagger}$	0.3913^{*}	$0.2393^{*\dagger}$	$0.6362^{*\dagger}$
`NBbest`	0.8377^{*}	$0.7800^{*\dagger}$	0.5520^{*}	0.3441^{*}
`K`	0.7300	$0.5020^{\dagger}$	0.3121	0.8033
`S`	0.6800	0.4540	0.2958	0.7520
`P`	0.6800	0.4440	0.3015	0.7656

tics from the information object (the topic sentence or paragraph). We studied the impact of this parameter on effectiveness (AP@5). Figure 1 reports AP@5 for varying levels of λ and the proportional cutoff value r.

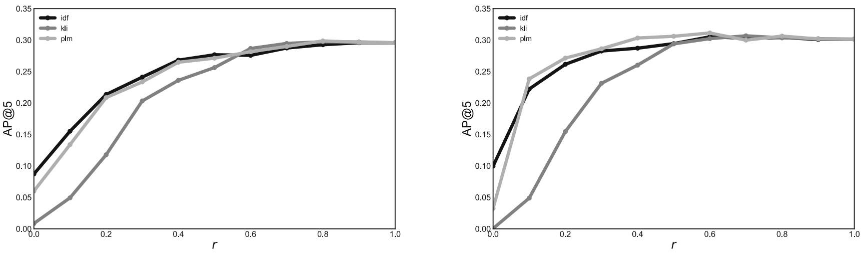


(a) PLM for topic sentences.

(b) PLM for topic paragraphs.

Fig. 1. AP@5 for PLM for varying values of the smoothing parameter λ and the proportion cutoff r, applied to topic sentences (left) and paragraphs (right).

While results are heavily affected by the proportion of terms selected when generating a query (r), there appear to be no differences in effectiveness due to different settings of the smoothing parameter λ, when $\lambda \leq 0.8$ and sentences are considered as input. The situation is similar when applying PLM on paragraphs. This may be due to the fact that smoothing only affects a limited number of not relevant documents. Because of these results, we fix $\lambda = 0.5$ when comparing PLM to the other query generation methods.



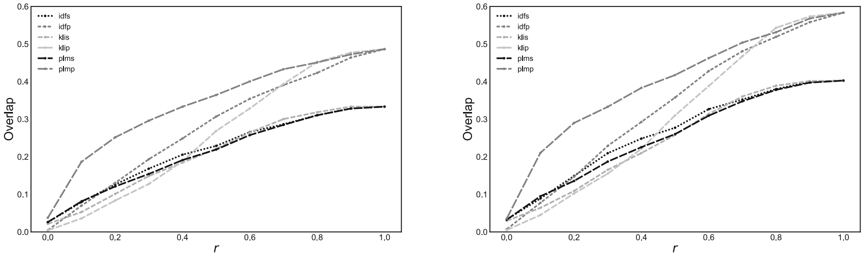
(a) Effectiveness of automatic generation methods for topic sentences.

(b) Effectiveness of automatic generation methods for topic paragraphs.

Fig. 2. AP@5 for the automatic generation methods for varying values of the proportion cutoff r, when applied to topic sentences (left) and paragraphs (right).

Table 3. Effectiveness of automatically generated queries using sentences (`s`) and paragraphs (`p`) as input information objects, as well of baselines. The methods are reported using the configuration of r (proportion of query terms) that provided the highest effectiveness in terms of AP@5. Statistical significant differences (paired t-test, $p < 0.05$) compared to the baselines are reported with † (for topic paragraphs, `P`).

	r	P@1	P@5	AP@5	MRR
`S`	-	0.6800	0.4540	0.2958	0.7520
`P`	-	0.6800	0.4440	0.3015	0.7656
`idf(s)`	0.9	0.6800	0.4540	0.2958	0.7522
`kli(s)`	0.8	0.6900	0.4540	0.2974	0.7591
`plm(s)`	0.8	0.6900	0.4560	0.2987	0.7593
`idf(p)`	0.6	0.6800	0.4480	0.3053	0.7556
`kli(p)`	0.7	0.7000	0.4420	0.3068	0.7671
`plm(p)`	0.6	0.6900	0.4640	0.3114	$0.7629^\dagger$



(a) Overlap between automatic generation methods and `Bbest`.

(b) Overlap between automatic generation methods and `K`.

Fig. 3. Number of query terms in common (overlap) between Boolean queries and keywords, and automatically generated queries, for increasing levels of r.

4.4 Results for Automatically Generated Queries

We now analyse the effectiveness of automatic query generation methods with respect to r, the number of query terms: this is reported in Fig. 2. Methods `plm(p)` (PLM on paragraphs) and `plm(s)` (on sentences) both outperform baselines `S` and `P` (see Table 3) and display the best performance for automatic reduction methods, except `kli(p)` for P@1 and MRR when the proportion that takes the maximum AP@5 is found. Methods `plm(p)`, `plm(s)` and `idf(p)` achieve the highest results at earlier proportions compared to `kli(p)` and the other methods of reduction from `S`. The effectiveness of the PLM methods are also evidenced through the higher percentage of overlap with the manually created queries, as shown in Fig. 3. While this is the case, all methods of reduction from `S` only achieve their best results at very high proportions (nearing the whole of the sentence), and over half of the paragraph for reduction methods from `P`. Nevertheless, these results are sensibly lower than some of the manual queries devised by the legal expert, e.g., `Bbest` and `NBbest`. Perhaps therefore, methods to introduce new terms into a query (i.e. query expansion), rather than reduce a query to distill terms are an appropriate area of investigation.

5 Conclusion

In this work we have considered automatic query generation methods for case law retrieval, and compared these methods with manual queries akin to those prepared by legal experts. We found that existing keyword extraction methods are as effective as average Boolean queries issued by experts. However, we also found that automatic methods are substantially inferior to keyword queries, and the best Boolean queries from experts. These effective queries often use terms not mentioned in the cases for which the queries are designed. The query generation methods we considered only select terms from portions of the cases at hand, thus not inferring additional relevant terms not mentioned in the cases. Methods that introduce new terms into a query (i.e. query expansion) are an appropriate area for future investigation.

As part of this research, we have also contributed a test collection for case law retrieval. While our collection contains a number of manual relevance assessments that allows us to reliably evaluate the methods considered here, more assessments are required to evaluate other methods. Future work will extend this collection to make it reusable for evaluation for other case law retrieval methods. Our collection, retrieval runs and analysis are available online at https://github.com/ielab/ussc-caselaw-collection.

References

1. Bailey, P., Moffat, A., Scholer, F., Thomas, P.: User variability and ir system evaluation. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 625–634. ACM (2015)

2. Baron, J.R., Lewis, D.D., Oard, D.W.: Trec 2006 legal track overview. In: The Fifteenth Text REtrieval Conference (TREC 2006) Proceedings (2006)
3. Bendersky, M., Croft, W.B.: Discovering key concepts in verbose queries. In: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 491–498. ACM (2008)
4. Galgani, F., Compton, P., Hoffmann, A.: Combining different summarization techniques for legal text. In: Proceedings of the Workshop on Innovative Hybrid Approaches to the Processing of Textual Data, pp. 115–123. Association for Computational Linguistics (2012)
5. Grabmair, M., Ashley, K.D., Chen, R., Sureshkumar, P., Wang, C., Nyberg, E., Walker, V.R.: Introducing luima: an experiment in legal conceptual retrieval of vaccine injury decisions using a uima type system and tools. In: Proceedings of the 15th International Conference on Artificial Intelligence and Law, ICAIL 2015, pp. 69–78 (2015)
6. Hiemstra, D., Robertson, S., Zaragoza, H.: Parsimonious language models for information retrieval. In: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 178–185. ACM (2004)
7. Kim, M.-Y., Xu, Y., Lu, Y., Goebel, R.: Legal question answering using paraphrasing and entailment analysis. In: Tenth International Workshop on Juris-Informatics (JURISIN) (2016)
8. Koniaris, M., Anagnostopoulos, I., Vassiliou, Y.: Multi-dimension diversification in legal information retrieval. In: Cellary, W., Mokbel, M.F., Wang, J., Wang, H., Zhou, R., Zhang, Y. (eds.) WISE 2016. LNCS, vol. 10041, pp. 174–189. Springer, Cham (2016). doi:10.1007/978-3-319-48740-3_12
9. Koniaris, M., Anagnostopoulos, I., Vassiliou, Y.: Evaluation of diversification techniques for legal information retrieval. Algorithms **10**(1), 22 (2017)
10. Koopman, B., Cripwell, L., Zuccon, G.: Generating clinical queries from patient narratives. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (to appear, 2017)
11. Kumaran, G., Carvalho, V.R.: Reducing long queries using query quality predictors. In: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 564–571. ACM (2009)
12. Lastres, S.A.: Rebooting legal research in a digital age. Technical report, LexisNexis (2013)
13. Peñas, A., et al.: Overview of ResPubliQA 2009: question answering evaluation over european legislation. In: Peters, C., Di Nunzio, G.M., Kurimo, M., Mandl, T., Mostefa, D., Peñas, A., Roda, G. (eds.) CLEF 2009. LNCS, vol. 6241, pp. 174–196. Springer, Heidelberg (2010). doi:10.1007/978-3-642-15754-7_21
14. Poje, J.: Legal research. American Bar Association Techreport 2014 (2014)
15. Salton, G.: Automatic Information Organization and Retrieval. McGraw Hill Text, New York (1968)
16. Schweighofer, E., Geist, A.: Legal query expansion using ontologies and relevance feedback. In: CEUR Workshop Proceedings, vol. 321, pp. 149–160 (2007)
17. Tomokiyo, T., Hurst, M.: A language model approach to keyphrase extraction. In: Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment, vol. 18, pp. 33–40. Association for Computational Linguistics (2003)
18. Turtle, H.: Natural language vs. boolean query evaluation: a comparison of retrieval performance. In: Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 212–220 (1994)

19. Turtle, H.: Text retrieval in the legal world. Artif. Intell. Law **3**(1), 5–54 (1995)
20. van Opijnen, M.: Citation analysis and beyond: in search of indicators measuring case law importance. In: JURIX, vol. 250, pp. 95–104 (2012)
21. Verberne, S., Sappelli, M., Hiemstra, D., Kraaij, W.: Evaluation and analysis of term scoring methods for term extraction. Inform. Retrieval J. **19**(5), 510–545 (2016)
22. Zuccon, G., Palotti, J., Hanbury, A.: Query variations and their effect on comparing information retrieval systems. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, pp. 691–700. ACM (2016)