RMIT University at TREC 2008: Legal Track

Ying Zhang Falk Scholer Andrew Turpin

School of Computer Science and IT
RMIT University, GPO Box 2476V
Melbourne 3001, Australia

1 Introduction

This paper reports on the participation of RMIT university in the 2008 TREC Legal Track Ad Hoc task. OCR errors can corrupt the document view formed by an information retrieval system, and substantially hinder the successful retrieval of relevant documents for user queries. In previous research, the presence of errors in OCR text was observed to lead to unstable and unpredictable retrieval effectiveness. In this study, we investigate the effects of OCR error minimization — through de-hyphenation of terms, and the removal of corrupted or "noise" terms — on retrieval performance. Our results indicate that removing noise terms can lead to significant savings in terms of index size.

2 OCR Error Minimization

Printed hard-copy documents can be converted into electronically-editable form using Optical Character Recognition (OCR) technology. However, this technology generally does not achieve an accuracy of 100%. That is, errors are commonly introduced as a result of the conversion process, and any hyphenation of words occurring at the end of lines will be transferred directly from the source to the target format.

Noise term removal

A major problem that arises when using OCR text with keyword-based information retrieval systems is that the OCR process introduces errors. That is, previously valid terms may become corrupted, introducing noise into the collection. The presence of many such "noise terms" within the text, and subsequently in the index, has two detrimental effects: first, keyword searches based on correctly formulated terms will not match with the corrupted term, even if the original version of the term would have been a valid match. Second, the presence of many noise terms may lead to unreliable term weighting in documents, which may have detrimental effects on the similarity functions used in the retrieval system.

The key challenge is to determine which terms are "noise". In our experiments below, we treat those terms with collection frequency ≤ 2 and document frequency ≤ 2 as noise terms.

Once noise terms are identified, there are two ways of dealing with them: first, error-correcting algorithms may be applied, to attempt to transform the corrupted version of the word back to its

source. We investigated OCR-spell,[1] an an OCR-based spell-checking tool that extends ispell, proposed by Taghva and Stofsky (2001). However, initial experiments indicated some limitations of this approach for the TREC Legal Track environment: the tool did not correct some commonly-occurring errors (for example, most errors that incorporated an erroneous punctuation symbol in the middle of a term were not identified). Moreover, the data used in the Legal Track collection includes many proper names (for example, names of companies and individuals), and technical terms (for example, from chemical or medical analyses), which were not identified. Due to these problems, we did not investigate OCR-spell further.

A second approach is to simply remove the noise terms from the collection. While this does not help the "missed match" problem, where a query keyword no longer matches the corrupted version of a term that would originally have been a match, we hypothesise that this process should remove some of the noise that is introduced into the collection term statistics, and therefore lead to better behaviour of ranking functions.

Text de-hyphenation

Hyphenated words that span lines in the printed source document will be converted into the target electronically-editable format in the same way. Such word occurrences will therefore not match standard keyword searches for the non-hyphenated form of the word. One steps to reduce term mismatch due to the OCR process is therefore to remove the end-of-line hyphens, and re-assemble the word fragments.

When a term token that occurs at the end of a line ends with a hyphen, we remove the hyphen and join this token with the first token that occurs on the subsequent line. This new joined token is checked against the *ispell* dictionary.[2] If the term is found in the dictionary, the original two fragments are replaced with the new, joined term. If the term is not found, the original fragments are retained in the collection.

3 Experiments

The collection used for the 2008 TREC Legal Track is the IIT Complex Document Processing Information (CDIP) Test Collection (Tomlinson et al., 2007). It contains $6,910,192$ metadata records from US tobacco companies; $6,794,895$ of the records included document text of varying quality from an optical character reader. 45 new topics were created for the 2008 track; each topic contains a `RequestText` (a natural language description of the request, typically one-sentence), a `ProposalByDefendant` (an initial boolean query proposed by the defendant), a `RejoinderByPlaintiff` (a rejoinder boolean query from the plaintiff, and a `FinalQuery` (the final boolean query from the negotiations).

For our experiments, we used the *Zettair* search engine developed by the Search Engine Group at RMIT University.[3] The similarity function is based on a Dirichlet-smoothed language model (Zhai and Lafferty, 2004). In line with the track guidelines, each submitted retrieval run consisted of up to 100,000 documents.

[1] We used version 1.0 of the OCR-spell software, available from `http://www.isri.unlv.edu/ISRI/Software`

[2] `http://www.gnu.org/software/ispell/ispell.html`

[3] Zettair is available under a BSD license from `http://www.seg.rmit.edu.au/zettair`

	# of unique terms	# of total terms	index size (MB)
Original text ($p1$)	130,531,969	8,183,835,310	24,573
De-hyphened text ($p2$)	156,548,835	9,446,100,292	23,534
Noise-removed text ($p3$)	26,001,570	6,325,243,280	14,506

Table 1: *Number of unique indexed terms, number of total indexed terms, and the size of index for different pre-processing approaches.*

3.1 Indexing

We created three separate indexes as follows:

Original text ($p1$): For each record in the collection, we indexed the following fields from the metadata and the OCR document: `<au>`, `<ca>`, `<no>`, `<cr>`, `<np>`, `<rc>`, `<pc>`, and `<tp>`. HTML entities were converted into their characters.

De-hyphened text ($p2$): Hyphens occurring in line breaks have been removed from the original text, and the trailing part of the word has been joined to the preceding line, if it was found in the ispell lexicon. This de-hyphened text collection is re-indexed as $p2$.

Noise-removed text ($p3$): We removed the "noise terms" from the de-hyphened text collection and re-indexed the new collection as $p3$.

3.2 Run descriptions

Our retrieval experiments consist of six official runs on three different indexes using the search request as stated in the `RequestedText` and `FinalQuery` fields, respectively.

For `RequestedText` runs (RMITRP1, RMITRP2, RMITRP3), all query terms are used to conduct a bag-of-words ranked search. The matching algorithm used was a Dirichlet-smoothed language model.

`FinalQuery` runs (RMITBP1, RMITBP2, RMITBP3) were also run as bag-of-words searches. However, special string-matching and Boolean operators that are not natively supported in the Zettair search engine were expanded as follows:

- Parentheses or proximity operators were removed from the query text.
- Wildcards were expanded to all the possible variations that appear in the ispell lexicon.

4 Results and Discussion

Table 1 shows the number of unique terms and total that were indexed by our search engine for each of the described pre-processing approaches. The index size is also shown. De-hyphenation, while leading to an increase in the number of unique terms, actually results in a reduced index size overall. Removing noise terms ($p3$) significantly reduces the index size: $p3$ is only 59% of the size of the original collection, $p1$.

Run	EST_K-F1	EST_R-F1	EST_RB	EST_P5	EST_R100000
Median	0.0702	0.1109	0.4073	0.1959	0.2805
RMITRP1	0.1578	0.2158	0.2622	0.5615	0.4337
RMITRP2	0.1586	0.2173	0.2628	0.5808	0.4472
RMITRP3	0.1129	0.1777	0.2172	0.3846	0.3500
RMITBP1	0.0704	0.1481	0.2148	0.4038	0.4016
RMITBP2	0.0646	0.1367	0.2071	0.3962	0.3766
RMITBP3	0.0681	0.1583	0.2186	0.4692	0.4182

Table 2: *Results for the Legal Track 2008.*

The main effectiveness measure for the 2008 Legal Track is F1@K ("est_K-F1")[4], defined as:

$$F1@K = \frac{2 * Precision@K * Recall@K}{(Precision@K + Recall@K)}$$

where K is an integer between 0 and $100,000$ inclusive, representing the threshold at which the system believes the competing demands of recall and precision are best balanced. For each topic, we determined the value of K as the total number of documents with a similarity value greater than 0.5. Secondary measures for the track are F1@R ("est_R-F1"). For comparison with previous years, Recall@B ("est_RB") is also reported. Reportedly, the sampling approach favored depths 5 and 100000, so P@5 ("est_P5") and R@100000 ("est_R100000") are shown.

Results for our *ad hoc* runs are shown in Table 2. The row labelled "Median" shows the median results of all 2008 Legal Track participants. Our baseline ranked retrieval approach RMITRP1, using `RequestedText` with no pre-processing of the collection, performed well. De-hyphenation (RMITRP2) let to marginal improvements for the F1@K, F1@R, P@5 and R@100000 measures. The improvements are not statistically significant at the 95% confidence level based on the Wilcoxon signed-rank test.

Removing noise terms, on the other hand, led to a slight drop in performance for all effectiveness measures (RMITRP3). However, again, none of these differences are significant at the 95% confidence level. Our techniques can be used to decrease the size of the index by over 40%, with no significant fall in retrieval effectiveness.

Our runs based on the final boolean query from the negotiations, `FinalQuery`, perform much worse than those using the `RequestedText`. We believe that this may be in part due to over-expansion of wildcard matches when transforming the Boolean requests into ranked requests. In contrast to the previous results, for the `FinalQuery` runs our de-hyphenation approach harms performance for all effectiveness measures (RMITBP2). Noise-term removal, on the other hand, leads to improvements on all measures except F1@K compared to using the original collection (RMITBP3).

5 Conclusions

In this paper we have investigated two simple collection pre-processing approaches, aiming to overcome some of the errors introduced into the TREC Legal Track collection by OCR processes.

[4]The evaluation measures are defined in the TREC 2008 Legal Track: Ad Hoc and Relevance Feedback Task Guidelines, available at `http://trec-legal.umiacs.umd.edu/adhocRF08b.html`

Our results show that, for a standard ranking approach based on using natural language request text as a query, de-hyphenation can offer some small increments in retrieval performance.

We hypothesised that the removal of noise terms might improve retrieval performance by dampening interference in the term distribution statistics that are used to calculated ranked retrieval similarity scores. However, our approach of simply removing noise terms, defined by scarcity of occurrence in individual documents and the collection as a whole, did not improve retrieval performance (the changes in effectiveness were not statistically significant). However, noise term removal led to a significant saving in terms of resources: the inverted index for the collection with noise terms removed takes up only 60% of the space of the original index. We therefore recommend the removal of noise terms.

In future work, we plan to investigate other OCR-based error-correction algorithms, so that instead of simply removing noise terms, these can be mapped back to their original form.

References

Taghva, K. and Stofsky, E. (2001), "Ocrspell: an interactive spelling correction system for ocr errors in text", *International Journal on Document Analysis and Recognition* **3**(3), 125–137.

Tomlinson, S., Oard, D. W., Baron, J. R. and Thompson, P. (2007), Overview of the TREC 2007 legal track, *in* "The Sixteenth Text REtrieval Conference (TREC 2007)", National Institute of Standards and Technology, Gaithersburg, MD.

Zhai, C. and Lafferty, J. (2004), "A study of smoothing methods for language models applied to information retrieval", *ACM Transactions On Information Systems* **22**(2), 179–214.