On the importance of legal catchphrases in precedence retrieval*

Edwin Thuma†
University of Botswana
Department of Computer Science
Gaborone, Botswana
thumae@mopipi.ub.bw

Nkwebi P. Motlogelwa‡
University of Botswana
Department of Computer Science
Gaborone, Botswana
motlogel@mopipi.ub.bw

ABSTRACT

This paper presents our working notes for FIRE 2017, Information Retrieval from Legal documents -Task 2 (Precedence retrieval). Common Law Systems around the world recognize the importance of precedence in Law. In making decisions, Judges are obliged to consult prior cases that had already been decided to ensure that there is no divergence in treatment of similar situations in different cases. Our approach was to investigate the effectiveness of using legal catchphrases in precedence retrieval. To improve retrieval performance, we incorporated term dependency in our retrieval. In addition, we investigate the effects of deploying query expansion on the retrieval performance. Our results show an improvement in the retrieval performance when we incorporate term dependence in scoring and ranking prior cases. However, we see a degradation in the retrieval performance when we deploy query expansion.

KEYWORDS

Precedent retrieval, term dependency, query expansion, legal catchphrases

1 INTRODUCTION

Common Law Systems around the world recognize the importance of precedence in Law. In making decisions, Judges are obliged to align their decisions to relevant prior cases. Thus, when lawyers prepare for cases, they research extensively on prior cases. In addition, Judges also consult prior cases that had already been decided to ensure that a similar situation is treated similarly in every case [3]. This can be overwhelming due to the enormous number of prior cases and length of each. Task 2 of the Information retrieval in Legal Documents track (precedence retrieval), explores techniques and tools that could ease this task [3]. In general, precedence retrieval will retrieve a ranked list of prior cases that are related to a certain current case.

In this work we investigate the importance of legal catchphrases as queries in precedent retrieval. These legal catchphrases are extracted from current cases. To achieve this, we used a training set of documents provided for Task 1 (catchphrase extraction) where case documents have corresponding gold standard catchphrase. We used Term Frequency-Inverse Document Frequency (TF-IDF) term weighting model to identify similarity between documents in the training set and current cases. Queries were formulated using legal catchphrases from the most relevant documents in the training set.

For retrieval, we deployed the parameter-free DPH term weighting model to score and rank prior cases. Moreover investigate whether taking the dependence of query terms in to consideration when ranking and scoring prior cases could improve thr retrieval performance.Previous work has shown that incorporating term dependency in scoring and ranking documents could significantly improve the retrieval performance [4]. In addition we deployed query expansion where the original queries are reformulated by adding new terms to investigate its impact on retrieval performance. Previous research has shown that query expansion could improve retrieval effectiveness [1].

This paper is structured as follows. Section 2 contains a background on algorithms used. Section 3 describes the experimental setup. In Section 4, we describe the methodologies used for the 3 runs submitted by team UB_Botswana_Legal for Task 2. Section 5 presents results and discussions.

2 BACKGROUND

In this section, we begin by presenting a brief but essential background on the different algorithms used in our experimental investigation and evaluation. We start by describing the TF-IDF term weighting model, in Section 2.1. We then describe DPH term weighting model in Section 2.2, Lastly we describe the Bose-Einstein 1 (Bo1) model for query expansion in Section 2.3.

2.1 TF-IDF term weighting model

In our experimental setup, we used *TF-IDF* [5] to score and rank documents. Generally, *TF-IDF* calculates the weight of each term t as the product of its term frequency (tf) weight in document d and its inverse document frequency (idf_t).

$$score_{TF\text{-}IDF}(d, Q) = \sum_{t \in Q} 1 + \log(tf) * \log \frac{N}{df_t} \quad (1)$$

*On the importance of legal catchphrases in precedence retrieval
†Lecturer, Department of Computer Science, University of Botswana
‡Lecturer, Department of Computer Science, University of Botswana

Where:

- tf is the term frequency of term t in document d.
- df_t is the document frequency of term t - the number of documents in the collection that the term t occurs in.
- $idf = \log \frac{N}{df_t}$ is the inverse document frequency of term t in a collection of N documents

2.2 DPH Term Weighting Model

Our baseline system used the parameter-free DPH term weighting model from the Divergence from Randomness (DFR) framework [2]. The DPH term weighting model calculates the score of a document d for a given query Q as follows:

$$score_{DPH}(d, Q) = \sum_{t \in Q} qtf \cdot norm \cdot \left(tf \cdot \log((tf \cdot \frac{avg\_l}{l}) \cdot (\frac{N}{tfc})) + 0.5 \cdot \log(2 \cdot \pi \cdot tf \cdot (1 - t_{MLE})) \right) \quad (2)$$

where qtf, tf and tfc are the frequencies of the term t in the query Q, in the document d and in the collection C respectively. N is number of documents in the collection C, avg_l is the average length of documents in the collection C and l is the length of the document d. $t_{MLE} = \frac{tf}{l}$ and $norm = \frac{(1-t_{MLE})^2}{tf+1}$.

2.3 Bose-Einstein 1 (Bo1) model for Query Expansion

In our experimental investigation and evaluation, we used the Terrier-4.0 Divergence from Randomness (DFR) Bose-Einstein 1 (Bo1) model to select the most informative terms from the topmost documents after a first pass document ranking. The DFR Bo1 model calculates the information content of a term t in the top-ranked documents as follows [1]:

$$w(t) = tfx \cdot \log_2 \frac{1 + P_n(t)}{P_n(t)} + \log_2(1 + P_n(t)) \quad (3)$$

$$P_n(t) = \frac{tfc}{N} \quad (4)$$

where $P_n(t)$ is the probability of t in the whole collection, tfx is the frequency of the query term in the top x ranked documents, tfc is the frequency of the term t in the collection, and N is the number of documents in the collection.

3 EXPERIMENTAL SETUP

3.1 Document Collection

In this work we use the document collection provided by the Information Retrieval in Legal Documents track organizers. It comprised 200 documents representing current cases and 2000 documents representing prior cases [3]. For each current case, the objective is to retrieve relevant ranked prior cases such that the most relevant appear at the top of the list and the least relevant at the bottom together with scores for prior case.

3.2 Precedence Retrieval Experimental Platform

For all our experimental evaluation, we used Terrier-4.2, an open source Information Retrieval (IR) platform. Documents were preprocessed before indexing: tokenising text, stemming each token using the full Potter stemming algorithm, and stopword removal using terrier stopword list.

4 METHODOLOGY

4.1 query formulation

Query Generation For the different Runs

For all the runs in this task, we indexed the 100 case documents provided in task1, which had the corresponding catchphrases using Terrier-4.2 IR platform. During indexing, each case document was first tokenised and stopwords were removed using the Terrier stopword list. Each token was then stemmed using the full Porter stemming algorithm.

For each current case provided in task 2, We used the TF-IDF term weighting model in Terrier 4.2 to score and rank the indexed case documents. Each case document was first pre-processed using the same pre-processing steps undertaken during indexing. After retrieving the top 40 case documents, we formulated queries for each current case using the gold standard catchphrases that appear in these ranked case documents and also in the current case document used for retrieval.

4.2 UB_Botswana_Legal_Task2_R1

Using the formulated queries, we deployed the parameter-free DPH Divergence from Randomness term weighting model in Terrier-4.2 IR platform as our baseline system to score and rank the prior cases.

4.3 UB_Botswana_Legal_Task2_R2

We used UB_Botswana_Legal_Task2_R1 as the baseline system. In addition, we deployed the Sequential Dependence (SD) variant of the Markov Random Fields for term dependence. Sequential Dependence only assumes a dependence between neighbouring query terms [4, 6]. In this work, we used a default window size of 2 as provided in Terrier-4.2.

4.4 UB_Botswana_Legal_Task2_R3

We used UB_Botswana_Legal_Task2_R1 as the baseline system. In addition, we deployed a simple pseudo-relevance feedback on the local collection. We used the Bo1 model for query expansion to select the 10 most informative terms from the top 3 ranked documents after the first pass retrieval (on the local collection) [6]. We then performed a second pass retrieval on this local collection with the new expanded query.

5 RESULTS AND DISCUSSION

This work set out to investigate the importance of legal catchphrases in precedence retrieval. The results of our submission in Table 1 were evaluated by the organizing committee of this task. Since most of the catchphrases were bi-grams and tri-grams, our exploitation of sequential term dependency variant for the Markov Random Fields for term dependence led to improvements in retrieval performance in terms of Mean Average Precision and Precision @ 10. Our attempt to improve retrieval performance using query expansion resulted in degradation in the retrieval performance. We suspect this might have been to due to query drift.

Table 1: Fire 2017 UB-Botswana Legal Run Evaluation results for Task 2

Run ID	Mean Average Precision	Mean reciprocal Rank	Precision@10	Recall@100
UB_Botswana_Legal_Task2_R3	0.1671	0.3478	0.1225	0.559
UB_Botswana_Legal_Task2_R1	0.1487	0.3506	0.112	0.546
UB_Botswana_Legal_Task2_R2	0.1078	0.3017	0.0785	0.43

REFERENCES

[1] G. Amati. 2003. Probabilistic Models for Information Retrieval based on Divergence from Randomness. *University of Glasgow,UK, PhD Thesis* (June 2003), 1 – 198.

[2] G. Amati, E. Ambrosi, M. Bianchi, C. Gaibisso, and G. Gambosi. 2007. FUB, IASI-CNR and University of Tor Vergata at TREC 2007 Blog Track. In *Proceedings of the 16th Text REtrieval Conference (TREC-2007)*. Text REtrieval Conference (TREC), Gaithersburg, Md., USA., 1–10.

[3] Arpan Mandal, Kripabandhu Ghosh, Arnab Bhattacharya, Arindam Pal, and Saptarshi Ghosh. 2017. 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.

[4] Donald Metzler and W. Bruce Croft. 2005. A Markov Random Field Model for Term Dependencies. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '05)*. ACM, New York, NY, USA, 472–479. https://doi.org/10.1145/1076034.1076115

[5] Juan Ramos. 1999. Using TF-IDF to Determine Word Relevance in Document Queries. (1999).

[6] Edwin Thuma, Nkwebi Peace Motlogelwa, and Tebo Leburu-Dingalo. 2017. UB-Botswana Participation to CLEF eHealth IR Challenge 2017: Task 3 (IRTask1 : Ad-hoc Search). In *Working Notes of CLEF 2017 - Conference and Labs of the Evaluation Forum, Dublin, Ireland, September 11-14, 2017*. http://ceur-ws.org/Vol-1866/paper_73.pdf