University of Glasgow | School of Computing Science

Documents as Queries for the Identification of Public Domain Knowledge

Kelvin Fowler (2083905f)

February 22, 2019

Note: Joint Honours - 40 Credits

ABSTRACT

Government documents must be reviewed for sensitivities before they are released to the public. If the information within a document is already in the public domain then this document can be released to the public without going through a more rigorous sensitivity review process, as no new information will be being released. The cross checking of document contents with public domain documents is currently undertaken manually in an insecure and cumbersome way, involving human reviewers entering manual queries into public web search engines. An automatic system could feasibly formulate effective queries from documents and retrieve public domain documents for cross referencing. It is, however, difficult to replicate the way a human might form an effective query from reading a document, as these documents are potentially very long and contain much information. We propose to use the extraction of named entities and temporal entities as key features in an automatic query generation. We propose to use these extracted entities in a complex query with a learning to rank approach to learn which sections of this query are the most important for this task. We evaluated this approach using a proxy test collection representative of the type of documents encountered in this task and found that none of the features we defined showed any improvement against a defined baseline. We did however find that results were improved by simply using the top 100 terms weighted under TF-IDF to form a query. We evaluated our methods on a collection of government documents, with similar findings of reduced precision upon applying a learned model. This leads us to the conclusion that stronger features need to be investigated for this task and more analysis could have been done to predict the failure of our features.

1. INTRODUCTION

Government departments have a responsibility to prevent sensitive information from being released to the public. Some information in government documents already exists in the public domain. If a document contains only that information which is already in the public domain then the full sensitivity review process may be skipped. This type of review can occur when a government document must be transferred to the public domain either due to the time limitations imposed by The Public Records Act [2] or because of a Freedom of Information Request, in line with The Freedom of Information Act [1]. We want this identification of public domain information to be as efficient and effective as possible to reduce the need for in-depth sensitivity review.

As it stands the process for public domain comparison is cumbersome and insecure with reviewers having to manually enter queries into public search engines in order to find public domain documents to cross reference with the document under review. This is a slow process and exposes potentially sensitive information in queries to external search services. Automation could help improve this process with faster and more effective retrieval of results. In addition, the use of an offline and local corpus of public domain documents removes the security concerns surrounding the inadvertent release of sensitive information. To compound the problem the number of documents which require review is ever increasing due to new technologies [3], further motivating the introduction of assistive technology.

The task of retrieving public domain documents relevant to the contents of a given document can be seen from an Information Retrieval (IR) perspective. We wish to formulate a query from a government document which can effectively and efficiently retrieve related public domain documents, in order that a manual reviewer can compare the retrieved documents to the government document. Entering a full document as a query can be very slow, which is a potential problem for this type of assisted review [11]. The question then is how to formulate an effective query from a given government document.

In this paper we describe techniques for document analysis which produce such queries for a given government document and issue them to search a set of public domain documents. These government documents are often reporting events to others, and so contain references to people, places and organisations. They also generally focus on a specific time. Intuitively these are important aspects of any document and give the reader a clear idea of who and what a document is referring to.

In Table 1 we see a short sample of a document and an example of a query a human might form in order to find related public domain documents. You will notice the use of temporal and named entities in this query. This type of query generation is what we wish to automate.

Following this intuition, our approach focusses on named entities (people, places and organisations) and temporal entities (specific text references to dates), while also considering the other important terms in a document which provide valuable context. These terms are split into a system of subqueries in order that a learning to rank approach can learn with supervision which sections are most important in the retrieval of relevant public domain documents. Learning to rank requires a large amount of training data to learn appropriate weightings for query features. We do not have a large collection of government documents with relevance judged public domain documents and so we produced and use a proxy test collection generated from an existing TREC adhoc test collection[1]. This proxy test collection serves the purpose of providing training data and preliminary test queries however our approach is then evaluated on a small collection of government documents with known relevant public domain documents. This allows us to verify that our general approach works in the specific task we aim to solve.

The goal of this paper is to provide a retrieval technique that can assist the manual process of public domain comparison in sensitivity review. We do not seek to replace the manual aspect of this task and in fact archivists have made clear that they would be reluctant to trust technology alone [12].

Throughout this paper we will refer to documents which are being reviewed as **source documents** and public domain documents which are being retrieved as **target documents.** Thus, queries will be formed from **source documents**.

Our contributions are as follows:

[1] `https://trec.nist.gov/`

Document Text	Potential Manual Query
A federal appeals court has rejected Democratic presidential candidate Michael Dukakis' attempt to use his power as governor to block a Massachusetts National Guard training mission in Central America. The 1st U.S. Circuit Court of Appeals issued a one-sentence order Tuesday affirming a May 6 decision by U.S. District Judge Robert Keeton, who upheld federal supremacy over the National Guard. "Having examined the briefs of the parties ... and having had the benefit of oral argument, we affirm the judgement on the basis stated in the district court's well-reasoned opinion," the appeals court said. Dukakis sued in January to block the assignment of 13 public relations specialists in the Massachusetts Guard to Honduras and Panama for two weeks in May because of his opposition to Reagan administration policies in the region. In his presidential campaign, Dukakis has denounced the administration for its "failed and illegal" policy of supporting Nicaraguan rebels, and he said sending Guard troops to the region was an attempt to intimidate Nicaragua. Dukakis, who was campaigning Tuesday in California and Colorado, had no immediate comment on the appellate decision. A press aide, Steven Crawford, said the governor was disappointed that the court chose not to uphold the "important principle" of state authority. ...	Michael Dukakis Massachusetts National Guard 1988

Table 1: An example of manually forming a query to retrieve public domain documents.

- We propose a novel method for using complex query operators and learning to rank to implicitly learn the most important sections of queries generated from source documents in order to retrieve target documents.
- We generate a proxy test collection using existing relevance judgements in order to obtain data for learning a model.
- We contribute to the ongoing research surrounding the improvement of document review in government.
- We identify that more research is needed to effectively define features for use in learning to rank to improve queries in this research area.

The remainder of this paper is structured as follows. In Section 2 we discuss the background of the project including related literature and limitations of existing approaches. In Section 3 we provide a general overview of the approach we take. Sections 4 and 5 discuss in more detail our techniques for the extraction of query components and the subsequent creation of complex queries for learning to rank respectively. Details on our experimental set-up are given in Section 6, with our experimental results on our proxy collection following in Section 7. We then evaluate our approach on a source collection of government documents in Section 8. Finally, we conclude with a discussion and avenues for future work in Section 9.

2. BACKGROUND

This section will discuss related literature to our research. We will discuss other approaches to sensitivity review and technology assisted review. Temporal entities and their existing uses will be covered, as well as complex query formulations and learning to rank approaches. We will explain why this task cannot be completed to a sufficient standard at the moment with the discussed existing techniques and identify specific limitations that we aim to address.

2.1 Sensitivity and Other Review

Applying information retrieval to assist sensitivity review is an ongoing problem and several distinct tasks have been approached. For instance, the use of automatic classification of sensitivities can be seen in work by McDonald et al [23] through analysis of document sentiment and other features. More recently McDonald et al have included active learning in the sensitivity review process [22], finding that this active feedback from human reviewers can be very helpful in automatically identifying previously unknown types of sensitivities.

More broadly, assisted sensitivity review comes under the banner of technology assisted review (TAR). This is mainly used in the realm of E-Discovery where opposing legal parties attempt to uncover as much information about each other as is possible [25]. E-Discovery seeks to obtain maximum recall (the retrieval of all relevant documents) so that neither party misses any details. In our case we are more concerned with precision in that archivists are unlikely to read all related documents during the public domain comparison task.

In fact, this type of TAR is becoming increasingly prevalent, featuring in news articles surrounding the 2016 U.S. Presidential Election [4]. TAR allowed the F.B.I. to review 650,000 emails in a week through automatic classification and duplicate deletion.

Identifying public domain knowledge in government documents through IR has been addressed in part by my 4th year project at the University of Glasgow [11], which lays the ground work for the more advanced research described in this paper. From this point the previous project will be referred to as **the L4 project**. Our previous method relied only on the use of named entities for the generation of bag-of-words queries and made no use of temporal entities or any learning to rank techniques, limitations which we address in this paper. We found that using only named entities in a basic bag of words query performed only slightly worse than using all documents terms as a query, however the named entity query was far faster to perform. In addition to the limitations of not using temporal entities or learning to rank, we did not make correct use of named entities. Any named entity longer than a single term was split into its individual terms, which meant that we lost any multi-term relationships in these named entities. In this work we amend this limitation by retaining multi-word named entities during indexing and retrieval.

The L4 project also created a UI for this assisted sensitivity review, which could be used by government departments in the public domain comparison task.

To our knowledge, this is the only other work which has attempted this task.

2.2 Named Entities

In addition to the performance of the named entity query in the L4 project there is other literature which suggests that named entities can be helpful in information retrieval tasks.

For example, named entities have been used in new event detection by Kumaran et al [15], who discuss when to use named entities in reference to news articles, finding that careful use of named entities in different situations can improve new event detection.

Khalid et al [14] also see improvement in the domain of question answering when identifying named entities.

Clearly then, named entities can be an important addition to information retrieval systems in various tasks.

2.3 Time in Information Retrieval

Time can be an important factor to consider in information retrieval, dependent on the task at hand. There are several technologies available for identification and resolution of temporal entities in documents such as Heideltime [29] and SUTime [9]. These allow temporal expressions to be represented in a normalised format, such as TimeX3 from TimeML [2], ideal for further manipulation and matching. These temporal extraction techniques are used on both source and target documents to identify temporal entities.

Strötgen et al [28] divide queries into a temporal part and textual part and examine how best to identify the most relevant temporal expressions in a document, both in general and with respect to a query. Of features they identify, value frequency of temporal entities is particularly relevant to us. Value frequency simply counts the occurrences of different temporal entities, assuming those mentioned most are more important. We use the value frequency approach of [28] to calculate one variant of focus time when analysing documents.

Jatowt et al [13] describe a method for approximating the focus time of a given document, however this requires connection to a knowledge base and as such is beyond the scope of this project. Although we do not replicate these specific focus time approaches, we make some attempt at estimating the focus times of our source and target documents.

Given the literature surrounding time in information retrieval, the omission of this from our approach in the L4 Project was a clear limitation.

2.4 Complex Queries

In IR, we are not limited to classic bag-of-words queries and often, beyond what is visible to the user, queries are represented in a more complicated manner. For example, query expansion is described by Xu et al [30] which adds additional terms to the query by examining documents returned by the original query. Additionally, Metzler and Croft [24] discuss identifying if adjacent query terms have relationships and factoring this into the retrieval model. Further, Bendersky et al [5] learn concept importance in queries using markov random fields, altering the query accordingly.

Macdonald et al [19] discuss rewriting user queries in order to improve efficiency, which other approaches often degrade.

Most relevant to us is a way of representing queries as consisting of several distinct sections. This segmentation is possible with tagging which is available in Terrier [17]. In addition to tagging, there are other operations which complex query languages, such as the Indri Query Language [27], support. An example is query term proximity limitations. There are many such operators and a list of examples can be seen on the Indri website [3].

Lee et al [16] use complex query operators in order to create queries from large arbitrary sections of text. Specifically Lee et al use these complex query operators to weight sections of text they have extracted from the selected text. The features Lee et al use to weight sections of text are calculated before retrieval and include examples such as frequency of occurrence in query logs. These features are not specific to the retrieval task being performed.

The novelty of our work is the scoring of subqueries using the implicit weights learned during learning to rank, rather than a predetermined assignment as in [16].

[2] `http://www.timeml.org/`

[3] `http://lemurproject.org/lemur/IndriQueryLanguage.php`

2.5 Learning to Rank

Learning to rank (L2R) is a well used machine learning technique in information retrieval. Macdonald et al [18] give a clear explanation of the various methods and techniques used in learning to rank.

In summary, learning to rank uses existing relevance judgements on a collection of queries and documents to learn a ranking strategy based on defined features. It works by reranking a pool of documents based on various defined features.

Features defined for learning to rank are usually separated into 3 distinct categories, query dependent, query independent and query features. Query independent features could be PageRank [6] or any other score calculated independent of the issued query. Query features do not depend at all on potentially retrieved documents and seek to learn how to alter retrieval methods based purely on the type of query being issued. In this work we define only query dependent features, which are calculated based on information in both the query and a given document.

These features are calculated on a sample of documents which is generally retrieved using a query under a known, well performing, simple retrieval model. This initial sample of retrieved documents is cut off at a value k and the feature scores are calculated for these top k documents [18]. The size of this sample can be altered. Dang et al [10] discuss the sampling step of learning to rank, where we seek to retrieve as many relevant documents as possible (high recall) so that the second, learned reranking step can move as many relevant documents to high ranks as possible. Using this intuition we investigate which sampling set is the most effective for our task.

For example, Qin et al [26] define query dependent features such as calculating the BM25 score of only the abstract of a document with respect to a query.

Clearly then, learning to rank can be used in a wide variety of information retrieval tasks to improve performance and the lack of learning to rank in our previous approach is a clear limitation.

2.6 Limitations of Existing Approaches

We have shown that previous approaches contain limitations, reinforced by findings in related literature. Here we list and label these limitations for reference.

Limitation 1 In the L4 Project [11] named entities were not represented in way which maintained multi-term relationships.

Limitation 2 The L4 project [11] did not seek to use temporal information to assist retrieval, while the literature clearly suggests that this could provide an improvement to performance.

Limitation 3 Lee et al [16] is perhaps the closest work to what we seek to achieve here, however their approach to generating queries from arbitrary sections of text is dependent on features which seek to find a general importance of concepts from the given text. They do not use learning to rank in order to allow the system to decide which features are most important for the given task.

Having provided examples of previous relevant approaches and their limitations, we now move on to give an overview of the approach we use, with reference to the limitations stated above.

3. OVERVIEW OF APPROACH

In this section we give an overview of the approach we take in this paper. We begin in Section 3.1 by formally defining the task we aim to address and then give a summary of our method in Section 3.2.

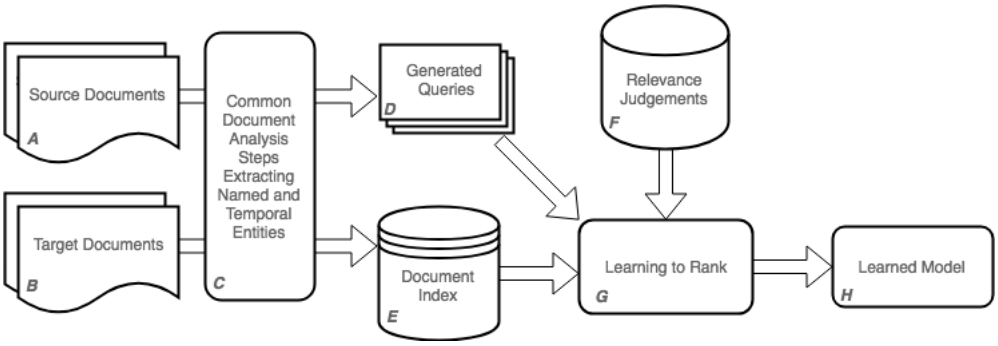

Figure 1: The Stages Involved in Producing A Learned Model

3.1 Problem Definition

The task we aim to complete is effective ranking of relevant public domain documents based on a given source government document.

The inputs are a source government document and set of target public domain documents. The desired output is a ranking of these target public domain documents in terms of relevance to the government document. These retrieved target documents will be used for comparison in the public domain comparison task prior to sensitivity review.

Given a source document $s \in S_C$ in a set of source documents, and a set of public domain target documents T_C, our goal is to produce a process $p(s) = Q$ which creates a query leading to an effective ranking of T_C for the public domain comparison task. An effective ranking is characterised by relevant documents being retrieved above non relevant documents. This is therefore a high precision task, as reviewers are unlikely to read all retrieved documents. Thus the measures we associate with success of the final ranking are Mean Average Precision (MAP) , Mean Reciprocal Rank (MRR) and Precision at 5 (P5). The remainder of this section will give an overview of how we approach this task.

3.2 Approach

In summary, our approach involves document analysis on source and target documents to produce queries and indexes. Following this, we use existing relevance judgements to learn a model using learning to rank based on existing relevance judgements. This learned model can then be used for ranking target documents in response to unseen source documents. Figure 1 summarises our approach displaying the various steps needed to produce the learned model.

Given a source document and set of target documents we must analyse these in order to extract the information we are interested in. There is a common analysis step for both types of documents. The source and target documents are analysed in bulk. We can see this in the parts labelled ***A, B*** and ***C*** in Figure 1.

We extract named entities from documents. We are careful to ensure our technique does not split multi-word named entities into separate terms, addressing **Limitation 1**. Specifically, any named entity which is longer than 1 word will be saved as such in both the index and query, meaning there will be far less ambiguity when attempting to match named entities.

We also extract and resolve temporal entities to specific dates. In addition to these terms we calculate various types of focus time and include these in queries and the index. Finally, we extract the remaining terms for queries and indexing.

For source documents, these extracted terms are used to form different queries, seen in ***D*** of Figure 1. For target documents, these terms go into an index for matching during retrieval, seen in seen in ***E*** of Figure 1.

The queries we produce from source documents have multiple parts and we call these subqueries. There is a sample subquery which is used to obtain the pool of documents for reranking during learning to rank. There is also the various subqueries consisting of named entities, temporal entities and other terms which are used for defining the learning to rank features for reranking of the sample documents. These features include matches of various types of named entities and features which use temporal entities in different ways, addressing **Limitation 2**. An example of a complex query can be seen in Table 3.

The features we define are all distinctly based on individual subqueries within our generated queries. This means that the learned model will effectively be a weighting model for the distinct query sections. This addresses **Limitation 3** by learning a model based on the specific task rather than pre-calculated features.

Due to a lack of a sufficient of amount relevance judgements on a collection of government source documents we produce a proxy test collection from an existing TREC ad-hoc test collection. This provides us with relevance judgements which we use for training, validation and testing of the learning to rank approach. These are labelled ***F*** in Figure 1.

The produced queries and relevance judgements are passed to a learning to rank model as labelled ***G*** in Figure 1. The result is a learned model which can be applied to future retrieval.

Following the learning of this model from the training data we evaluate our learned approach on a small collection of government documents with associated relevance judgements for public domain documents.

Having given an overview of the method we use in this paper we now give a more concrete explanation of our steps. First we describe the document analysis process used for indexing and query production.

4. DOCUMENT ANALYSIS

In this section we describe the components that we extract from documents in order to formulate queries, including our use of named and temporal entities. These are analysis steps which are common to both source and target documents, however the former produces queries and the latter produces an index of documents for retrieval. This distinction is shown in Figure 1.

The reason for this common analysis is so that terms in queries and indexed documents match during querying. So, if we wish to use some artefact to form a subquery we must ensure the same analysis has taken place on all target documents.

We now describe the individual components we extract from documents.

4.1 Named Entities

We have seen from Section 2 that named entities are a valuable artefact to extract from documents for query formulation. Encouraged by the results from [11] we continue to use named entities in queries

In this project named entities are extracted using the Stanford NLP NE Recogniser [20]. We extract 3 types of named entities: people, locations and organisations. We are careful to maintain multi word named entities to avoid loss of information addressing **Limitation 1**. For example, there are likely many mentions of

"Hillary" and "Clinton", but far fewer mentions of "Hillary Clinton".

There may be many different named entities in a document and some may be mentioned more than others. As such we use TF-IDF to calculate which terms in our document are most important.

4.2 Temporal Entities

Temporal Entity	Resolution
Last Thursday	14/04/1988
Today	18/04/1988
1988	1988
January	01/1988
Sunday	13/04/1988
Tonight	18/04/1988
This Year	1988
Median Date	13/04/1988
VF_{year}	1988
VF_{month}	04/1988
VF_{day}	18/04/1988

Table 2: Examples of Resolved Temporal Entities and Subsequent Calculated Focus Times For A Document Created on Monday 18/04/1988

As discussed in Section 2, [28] and [13] show that inclusion of temporal entities is beneficial in other tasks. Further, we specifically list the lack of use of temporal entities in our previous approach as **Limitation 2**.

Following this motivation, we extract and analyse temporal entities in documents to produce two variants of focus time, one based on median time and the other based on value frequency. We do not expect that untreated temporal entities will be of great use for retrieval as they will be very numerous and varied.

Temporal entities were resolved using the Heideltime temporal tagger [29]. This allows us to identify temporal entities in text such as "last Thursday" and resolve this reference the specific date in question. In order to do this resolution it is necessary to know the document creation date. Examples can be seen in Table 2.

We only consider 3 granularities of temporal entities: year references (***YY***), month references(***YY-MM***) and day references(***YY-MM-DD***).

Median focus time is calculated by taking all references to time in a document and ordering them from earliest to latest. We then take the median of this set of dates and resolve this median to its day granularity. We store this median focus date in queries and index as 3 separate terms, representing day (MED_{day}), month (MED_{month}) and year(MED_{year}). For a source and target document to match on median focus year, only the MED_{year} terms must match. For a month match both MED_{month} and MED_{year} must match and we require all 3 terms to match for an exact median focus day match.

Value frequency focus time is calculated by counting references to specific years, months and days. This is inspired by the use of value frequency in [28]. For our 3 granularities of temporal entities, all mention a year and so we count the year mentioned in each temporal entity. We take the most frequently occurring year references to be the VF_{year}. Only temporal entities of type ***YY-MM*** and ***YY-MM-DD*** can be used to count references to months and so these are used to calculate VF_{month}. Similarly, we use only ***YY-MM-DD*** references to produce VF_{day}.

We can see in Table 2 examples of these focus times for an example set of temporal entities.

4.3 Additional Terms

In addition to named and temporal entities there are many other terms in a document. These provide valuable context for the reader, and as we have seen from our example in Table 1 would generally be necessary in a query to produce high quality results.

After identifying named entities and temporal entities we can simply identify these other terms as those which are left over. We can tag these as belonging to the "other" set.

It is worth noting that not all of these terms will be valuable to us and so we use TF-IDF in order to identify which terms are most important from a given document.

Having explained our steps for the analysis of documents for the creation of queries and indexing, we now move on to explain our use of learning to rank in combination with complex queries.

5. LEARNING WITH COMPLEX QUERIES

This section gives details on our method of using learning to rank to implicitly learn the importance of sections of queries formed from documents. The queries we formulate contain many distinct parts, which we refer to as subqueries. Unfortunately we do not know which parts of the query are most important for our task. Unlike Lee et al [16] we wish to calculate this importance within the context of our specific task and datasets.

Learning to rank uses relevance judgements and so, we can know that the learner will be trying to optimize retrieval performance rather than any other metric. If we were to use precalculated features we could be inferring relationships which are not helpful in our task. Using learning to rank prevents any incorrect assumptions being factored into the model.

We specifically wish to use learning to rank as a method for understanding the implicit importance of our subqueries and so we must define features which represent this. This is not possible using the standard bag of words model and so we reformulate our query into specific subqueries. This is accomplished by using complex query operators available in Terrier, as mentioned in Section 2.

We label terms with specific tags in order to form these subqueries. For example, we separate and label person named entities and organisation named entities and have tags for our various focus time calculations.

An example of a complex query composed of various subqueries is shown in Table 3

With these subqueries we can define features which are only calculated with respect to individual subqueries. Applying this learned model on subsequent queries is in effect applying a weighting scheme to the individual subqueries.

Table 4 defines all of the features used in our learning to rank approach.

Features $pBiL_{MEDmonth}$ and $pBiL_{MEDday}$ use pBiL [21] because they use the unordered window proximity complex query operator to enforce that the median focus terms must appear immediately next to each other in documents. The indexing step ensure that this is the case for all documents, and so this proximity rule just ensures that both terms appear and match the source document.

5.1 Sampling

The first step of learning to rank is to produce a sample of documents which can be re-ranked according to the defined features [18].

Although our end goal is a high precision task, the retrieval of the sample of documents for reranking is a high recall task. We wish to have the maximum number of relevant documents in this sample as possible so that when feature scores are calculated they are being calculated on the maximum number of relevant documents. The more irrelevant documents in the sample, the more likely a non relevant documents is to end up at a high position in the results.

We have control over the size of the sample of documents, and this is set automatically by Terrier to 1000. It is possible to achieve the best possible recall (100%) by returning every document in the target collection during the sampling step. Although this means that all relevant documents will have their feature scores calculated and the overhead of calculating these scores for all other documents will be extremely high. There is therefore a balance to be struck between sample size and recall score.

Document Text	Example Complex Query
Dukakis Loses Appeal Challenging Federal Authority Over National Guard A federal appeals court has rejected **Democratic** presidential candidate **Michael Dukakis'** attempt to use his power as governor to block a **Massachusetts** National Guard training mission in **Central America**. The **1st U.S. Circuit Court of Appeals** issued a one-sentence order **Tuesday** affirming a May 6 decision by **U.S. District** Judge **Robert Keeton**, who upheld federal supremacy over the National Guard. "Having examined the briefs of the parties ... and having had the benefit of oral argument, we affirm the judgement on the basis stated in the district court's well-reasoned opinion," the appeals court said. **Dukakis** sued in January to block the assignment of 13 public relations specialists in the **Massachusetts** Guard to **Honduras** and **Panama** for two weeks in **May** because of his opposition to **Reagan** administration policies in the region. In his presidential campaign, **Dukakis** has denounced the administration for its "failed and illegal" policy of supporting **Nicaraguan** rebels, and he said sending Guard troops to the region was an attempt to intimidate **Nicaragua**. **Dukakis**, who was campaigning **Tuesday** in **California** and **Colorado**, had no immediate comment on the appellate decision. A press aide, **Steven Crawford**, said the governor was disappointed that the court chose not to uphold the "important principle" of state authority. ...	*#sample*{ Michael Dukakis, Robert Keeton, federal, court, ... } *#persons*{ Michael Dukakis, Robert Keeton, Dukakis, Reagan, Steven Crawford } *#locations*{ Massachusetts, Central America, U.S., Massachusetts, Honduras, Panama, California, Colorado} *#organisations*{ 1st U.S. Circuit Court of Appeals, National Guard} *#times*{ May, Tuesday} *#median focus time*{ 01/01/1988 } *#vf focus time*{ 01/01/1988 } *#other terms*{ federal, appeals, court, reject, presidential, candidate, attempt, power, governor, block, training, ...}

Table 3: An Example Representation of a Complex Query Created From a Document

We now move on to describe our experimental set-up and experiments conducted using the above explained method.

6. EXPERIMENTAL SET-UP

This section will describe the steps taken to evaluate our system. We describe the test collections we use as well as the learning to rank configuration.

6.1 Research Questions

We begin by defining formally the research questions which we seek to answer through this evaluation.

RQ.1 How can we maximise recall at the sampling step of retrieval to achieve the best results using learning to rank?

RQ.2 What information in source documents is most important for the retrieval of public domain documents containing information in the source document?

RQ.3 Can we combine these results to create a comprehensive retrieval model in order to retrieve public domain documents helpful to the sensitivity review task?

6.2 Test Collections

Collection	Documents
AP88	79919
Reuters 2008	101253
Govt. Source	2742
Govt. Source w/ Judgements	20

Table 5: Test Collection Statistics

In the previous L4 Project [11], a small test collection was manually created. The topics were queries formed from 20 government documents, each of which had an associated 20 relevant public domain news articles from a collection of Reuters stories from 2008. The government documents came from a corpus of government documents. The target documents were from a corpus of Reuters news stories from 2008. Because our approach in this research involves the use of learning to rank, we require a much larger collection of relevance judgements than the 20 that we have from the L4 project. 20 source documents with 20 relevance judgements each is not enough to infer relationships needed for learning to rank. It was not feasible create more manual relevance judgements as this would have been a very time consuming task. Thus it was necessary to produce a test collection by some other means.

Having analysed the government documents collection of source documents it became clear that these documents generally followed a similar structure of reporting events from the viewpoint of various embassies. These reports exist to inform a reader about events happening at specific times, involving people, organisations and locations.

News articles are very similar in nature to the government documents in our source collection, as they also seek to report on events, the difference being that the content of a news articles is necessarily already in the public domain. This similarity suggests that it may be possible to use news articles as representative source documents for the generation of queries.

In Table 6 we can see that government documents and our chosen corpus of proxy source documents follow a similar structure. Both example documents take a clear tone of reporting an event to another person, mentioning specific named entities and times.

Unfortunately, we did not have access to relevance judgements which defined news document as being relevant to one another. Instead we chose to use the implicit relevance of documents to one another which are relevant for a common topic.

For the TREC ad-hoc topics 51-100 there exists relevance judge-

Feature Ref	Feature Name	Description
DPH_{sample}	Sample Terms DPH	DPH Score for Terms used in Sample Query
TF_{person}	Person TF	TF Score for Person Named Entities
TF_{loc}	Location TF	TF Score for Location Named Entities
TF_{org}	Organisation TF	TF Score for Organisation Named Entities
BM_{time}	Time Terms Constant	Constant WM Matching on Temporal Terms
Bin_{VFyear}	Vf Focus Year	Binary Matching on most frequently mentioned year.
$Bin_{VFmonth}$	Vf Focus Month	Binary Matching on most frequently mentioned month (YY-MM).
Bin_{VFday}	Vf Focus Day	Binary Matching on most frequently mentioned day (YY-MM-DD).
$Bin_{MEDyear}$	Median Focus Year	Binary Matching on Median Calculated Focus Year
$pBiL_{MEDmonth}$	Median Focus Month	Matches of Same Adjacent Median Year, Month, Day Focus Terms (pBiL [21])
$pBiL_{MEDday}$	Median Focus Day	Matches of Same Adjacent Median Year, Month, Day Focus Terms (pBiL [21])

Table 4: Features Used in this Paper

Govt. Source Document	Proxy Source Document
Meeting with the Ambassador on March 3, President Correa expressed outrage at President Uribe for bombing his country, saying he expected the international community to condemn unprovoked aggression. He said the GOE had no relationship with the FARC, and accepted the Ambassador's statement that the Manta FOL had not supported the Colombian military operation. Correa plans to visit all countries bordering Colombia in the coming days to seek solidarity for its position. The Foreign Minister departs March 3 to attend OAS sessions on the incident scheduled for March 4.	More than 150 former officers of the overthrown South Vietnamese government have been released from a re-education camp after 13 years of detention, the official Vietnam News Agency reported Saturday. The report from Hanoi, monitored in Bangkok, did not give specific figures, but said those freed Friday included an ex-Cabinet minister, a deputy minister, 10 generals, 115 field-grade officers and 25 chaplains.

Table 6: Comparing Government Documents and Proxy Source Documents

ments for a collection of public domain news articles. These are a collection of associated press documents from 1988 which can be found in TIPSTER disk 2. We will refer to this collection as the AP88 collection. Given an existing TREC ad-hoc topic there exists a set of relevant public domain news articles. If two articles are relevant for the same topic, it suggests there is some information overlap in these documents.

Thus we made the assumption that if documents were relevant for the same topic then they were relevant to each other in our public domain comparison task.

The method described in Algorithm 1. This method for generating a test collection from existing topics and query relevance judgements is inspired by a method described by Lee et al [16] for choosing text segments from documents to generate queries from.

Having explained the formation of our proxy test collection we now give statistics on the test collections we use in this paper. We use several different collections of documents and queries in this work. Our proxy test collection is generated from 1988 Associated Press (AP) news articles found in TIPSTER disk 2. TREC-1 ad hoc topics 51-100 have relevance judgements for TIPSTER disks 1&2. We ignore all relevance judgements for documents outside of the 1988 AP Collection.

The 1988 AP collection contains 79919 documents.

Our method of producing source documents from this collection provided us with 2742 source documents. When creating a query from one of these source documents, that document is removed from the target collection and so retrieval using this collection

Result: Proxy Test Collection
T_C, the set of target public domain documents
Q, the set of TREC topics
$S_C = \emptyset$, the set of source documents
$Qrels = \emptyset$, the set of relevance judgements for S_C
for *each query q in topic set* Q **do**
  get documents with judgements for $q, J_q \subseteq T_C$
  get set of documents which are relevant $R_q \subseteq J_q$
  for *each doc d in relevant docs* R_q **do**
    add judgements of $J_q \setminus d$ to judgements $Qrels$
  end
  add R_q to S_C,
  If we form a query Q_d from a document d in in S_C
    the we assume judgements of $J_q \setminus d$ apply
end

Algorithm 1: Generating a Proxy Test Collection From Existing Relevance Judgements

only ever takes place with 79918 documents.

In addition to this proxy test collection we have a small collection of government documents with relevance judgements for public domain documents.

This set of public domain documents are Reuters in 2008 and there are 101253 of them.

We create queries from 20 government documents however we index 1018 documents for the purpose of calculating TF-IDF scores for term importance.

These statistics are summarised in Table 5.

6.3 Learning to Rank

Our learning to rank model is the Jforests [4] implementation of LambdaMART [8] which uses multiple additive regression trees and is a combination of pointwise and listwise learning to rank approaches. [18].

In order to prevent over-fitting to training data during learning to rank we partitioned our test collection into training, validation and test sets. This was done with a 60/20/20 ratio resulting in 1371 training source documents, 685 validation and 686 test. This follows suggestion from Macdonald et al [18].

7. EXPERIMENTAL RESULTS

In this section we address **RQ1** and **RQ2** which are evaluated on the AP88 collection with proxy relevance judgements.

7.1 RQ1 - Recall During Sampling Step

This section covers **RQ1** by investigating how we can build a sample query which gives us high recall. We need a high recall sample of documents for re-ranking based on various features. Additionally, we are concerned with the time it takes to perform this initial query step, as users are only willing to wait for search results for so long [7].

Using the system of subqueries discussed in Section 5, we now have complete control over the terms which will be used to obtain this sample pool of documents for reranking. We will call this subquery the **sample subquery**. The experiments in this section deal with altering the contents of this sample subquery to obtain the best recall scores possible while maintaining efficiency. As discussed in Section 5, we require a high recall sample in order to calculate feature scores on as many relevant documents as possible. We alter the number of terms in the sample subquery and we also use TF-IDF weights on each term. The number of terms is altered by taking the top k terms when ordered by TF-IDF score calculated over the AP88 collection. In addition, we alter the retrieval model used when obtaining this sample to ensure we have chosen the model which gives the highest recall.

We expect that having more terms in the sample subquery will improve recall scores, however will slow down results. We expect that query terms being weighted with TF-IDF scores will improve recall further.

We report recall scores averaged over 1371 source documents in the training partition of the proxy test collection. Recall is calculated using the associated proxy relevance judgements. We also report the average time taken per query. In all of the experiments in this section we return 1000 documents per query.

First, in Table 7 we report the recall scores of using the top 100 terms weighted with TF-IDF using 5 different common retrieval models.

Model	Recall
DPH	**0.6928**
TF-IDF	0.6853
TF	0.6627
PL2	0.6324
BM25	0.5758

Table 7: Recall with Different Retrieval Models, Query Size 100, Terms Weighted with Tf-Idf

[4] `https://github.com/yasserg/jforests/`

We see from Table 7 that DPH provides the highest recall score of the 5 retrieval models we test. Given this result, we use DPH as the retrieval model for the following experiments which alter the terms used in the sample subquery.

We now investigate which terms we should use in our sample subquery in order to achieve maximum recall within a reasonable time. We score each term in the source document with its TF-IDF score with respect to the entire collection and sort these from highest to lowest. We take the top k terms under this scoring system and report the results in Table 8. We also use *all* terms in a subquery, this means we do not cut off the ordered terms at any k. Across 1371 queries, when using all terms, the average query length was 221, the smallest query was 52 terms and the largest was 540. We see that using the top 100 terms gives us the highest recall score when the terms are added to the sample subquery with equal weights. As expected, the subquery with the smallest size performs the fastest on average.

Size	Recall	Mean Query Processing Time (s)
All	0.6082	0.4400
250	0.6163	0.3792
100	**0.6534**	0.1216
50	0.6480	0.0557
25	0.6243	0.0481
10	0.5708	**0.0357**

Table 8: Recall and Average Processing Time with Different Sample Subquery Sizes, Unweighted Queries w/ DPH. Results over 1371 queries.

When using DPH, we also have the ability to weight the terms in our sample subquery with a given weight. We can assign to each term its calculated TF-IDF score.

We repeat the previous experiment with these per term TF-IDF weights applied and report the results in Table 9.

Size	Recall	Mean Query Processing Time (s)
All	0.6904	0.4856
250	**0.6908**	0.4277
100	0.6824	0.1203
50	0.6619	0.0673
25	0.6324	0.0598
10	0.5750	**0.0273**

Table 9: Recall and Average Processing Time with Different Sample Subquery Sizes, Weighted Queries w/ DPH. Results over 1371 queries.

We see from Table 9, that this markedly improves the results of various sizes of sample subquery, with the 250 term subquery performing best in terms of recall. Again, as expected the smallest query is performed fastest on average.

We notice, however that although the 100 term sample subquery is produces a slightly worst recall than *All* and *250*, it is completed in less that half of the average time. This is important to us, as sampling is only the first step in learning to rank, after which we must calculate feature scores over all retrieved documents. The decrease in recall between the weighted 250 term subquery and the 100 term subquery is not statistically significant

under the student's t-test for p-value of 0.05. This indicates that using this much faster query does not significantly effect recall.

The conclusions we draw from the data reported with regards to **RQ1** is that more terms in the sample subquery seems to produce improved recall scores, up to around 250 terms. We also see significant increases in recall by weighting terms in our sample subquery using TF-IDF scores. Although this 250 term weighted query produces the highest recall out of all of our experiments, it takes more the double the average time to execute compared to a 100 term query. For this reason we conclude that a 100 term weighted sample subquery issued under DPH produces sufficient recall in an acceptable time for the sample section of learning to rank.

We now move on to report our findings regarding feature importance during the reranking of this sample, addressing **RQ2**.

7.2 RQ2 - Feature Importance

We now investigate feature importance, in response to **RQ2**. In order to gain a clear understanding of which features are important we perform several experiments using learning to rank with different combinations of the features defined in Table 4. In addition we analyse the distributions of our various features in order to predict what performance we might expect.

For all of these experiments, we fix the sampling subquery to be the top 100 terms based on their TF-IDF score. These terms are weighted with this TF-IDF score and the retrieval model for the sample subquery is DPH. This is based on the findings from **RQ1**, which suggests that this approach is effective and efficient.

Our experiments will examine the effect on precision that reranking based on these features has. Learning to rank will be trained on the set of training source documents from the generated proxy AP88 source collection. We also use a validation set of source documents from this collection to prevent over-fitting. The results reported are here are the retrieval performance scores from applying the learned model to retrieval on the test partition of the AP88 source document collection.

We report all of these results in Table 10.

All documents in our test collection for training are likely to have a similar focus year, since they are all documents from 1988. This means that our VF_{year} feature is likely perform poorly in comparison to a collection which references many different years.

Our baseline for the following experiments will be the performance of the sample subquery which is obtained before reranking. This allows us to see the impact on performance the re-ranking based on different features has. In all of the following experiments, the recall reported does not change between the reported baseline and the re-ranked results, indicating clearly that the set of retrieved documents is the same, however the specific ranking order has been altered due to the features.

These baseline results can be seen in the first row of Table 10, denoted by an empty feature set, $\emptyset$.

We can use the sample subquery to define a feature, DPH_{sample}, which is the score of a given document calculated via DPH on the sample subquery.

The results of using named entities based features are not promising on the proxy test collection. Rather than improve precision, the results are actually degraded. We take an individual type of named entity and for this subset of terms we calculate the TF (or other measure) score. Our intuition is that documents with a higher tf score for the given subquery are more likely to be relevant to the source document.

We can see an example of a proxy source document which exhibits this relationship in Figure 2. This is the distribution of TF (term frequency) score for the person named entity subquery for all target documents versus relevant target documents. We can see that there is a clear indication that relevant documents are more likely to have a high TF score for this subquery.

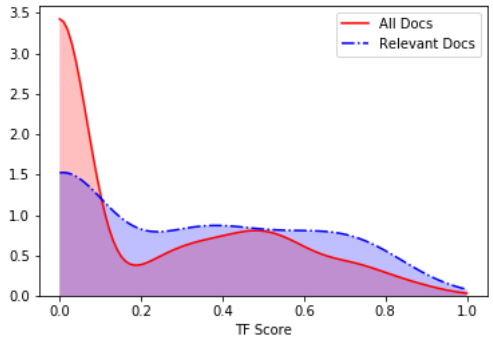

Figure 2: Feature Distribution over All Documents vs Relevant Documents for a Single Source Document, TF scoring for Person Named Entity Subquery

What is unfortunate is that this distribution which suggest these features would be successful does not carry over when we plot the same distribution over all of the training documents as can be seen in Figure 3. Both Figures 2 and 3 are only plotted on the persons named entity subquery, however a similar picture can be seen for organisations and locations.

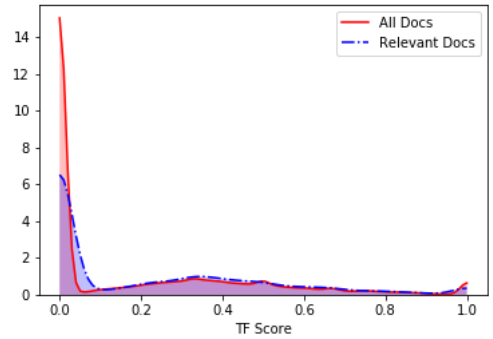

Figure 3: Feature Distribution over All Documents vs Relevant Documents

Considering the distribution seen in this graph, we do not expect that our named entity features will show any improvement over our baseline.

The distribution graphs shown in Figures 2 and 3 are produced using kernel density estimate from Seaborn [5]. They show an estimated probability density function for the given data. The area under the graph between any two x values is the probability of a TF score in that range of x values.

We expect that binary matching on individual temporal terms will not be a helpful feature as the times are very numerous and varied.

We suspect that to make use out of temporal entities we must use the focus time features which we calculated. Unfortunately, since the training collection is only from AP88, it is highly likely that the majority of focus years will 1988, meaning this feature may not be particularly helpful either. In fact, 1987 and 1988 composed 80% of the median focus years and 87% of the value frequency focus years.

[5] `https://seaborn.pydata.org/index.html`

Feature Set	MAP	MRR	P@5
$\emptyset$	**0.2013**	**0.6229**	**0.4338**
DPH_{sample}	0.1898	0.6048	0.4131
$DPH_{sample}, TF_{person}$	0.1840	0.6025	0.4044
DPH_{sample}, TF_{org}	0.1855	0.6063	0.4015
DPH_{sample}, TF_{loc}	0.1859	0.6005	0.4020
$DPH_{sample}, TF_{person}, TF_{org}$	0.1784	0.5989	0.4041
$DPH_{sample}, TF_{person}, TF_{loc}$	0.1634	0.5939	0.3974
$DPH_{sample}, TF_{loc}, TF_{org}$	0.1843	0.6139	0.4023
$DPH_{sample}, TF_{person}, TF_{org}, TF_{loc}$	0.1553	0.6004	0.3948
DPH_{sample}, Bin_{time}	0.1885	0.6213	0.4134
$DPH_{sample}, Bin_{VFyear}, Bin_{VFmonth}, Bin_{VFday}$	0.1878	0.6143	0.4140
$DPH_{sample}, Bin_{VFmonth}, Bin_{VFday}$	0.1827	0.6123	0.4096
$DPH_{sample}, Bin_{VFmonth}$	0.1886	0.6199	0.4125
$DPH_{sample}, Bin_{MEDyear}, pBiL_{MEDmonth}, pBiL_{MEDday}$	0.1763	0.5929	0.3956
$DPH_{sample}, TF_{person}, TF_{org}, TF_{loc}, Bin_{VFmonth}$	0.1780	0.5995	0.3962

Table 10: Learning to Rank with Various Feature Combinations, Evaluated on Proxy Test Collection formed from AP88

In examining per query results we found that some queries were improved with the addition of features. An example is a query which had a P@5 score of 0.2 when using no features for reranking. When using only DPH_{sample} feature this increased to 0.6 and when using both $DPH_{sample}, TF_{person}$ as features this increased to 0.8. Let us examine this document and the returned results in greater detail. The source document AP881130-0167 is describes environmental promises in the 1988 presidential campaign. The person named entities include George Bush and Michael Dukakis. Comparing the top 5 results of DPH_{sample} and $DPH_{sample}, TF_{person}$ only one new document is added. This document is concerned with the environment however none of the person named entities overlap. This leads us to believe that this named entity feature does not contribute to effective reranking is merely adds noise. Although the added document was relevant the addition was clearly due to any effectiveness of our feature.

Our results do not show any increase versus the baseline results, which indicates that reranking based on the features is not helpful in this instance.

In summary, the features we define all degrade precision on the retrieved target documents. As such we cannot confidently decide which query component is most important as it appears that all of the features we defined exhibit no clear relationship to relevance.

8. EVALUATION ON GOVERNMENT DOCUMENTS

In this section we show our results when evaluating our methods on a test collection composed of government documents with relevant public domain documents.

With no clear combination of features that performs exceedingly well we repeat some of the above experiments.

In the L4 Project [11] we reported the results shown in Table 11. Both queries in Table 11 are simple bag of words queries. The all terms query was created by using all terms in the source document. The named entities query was created by only using the people, location and organisation named entities as a bag of words query. We reported P.4 as the produced UI showed 4 documents in the initial search results. Our conclusion in the L4 Project was that named entities were clearly an important factor in this retrieval task, because the although the results were degraded when not using other terms, this was only by a small amount. Using only the named entities as a query was also much more efficient. The performance of the bag of words named entity query in the L4 Project was a strong contributor to the selection of named entities to form subqueries for features in learning to rank.

We cannot use the results from Table 11 as a baseline when performing the evaluation in this section using our new techniques. This is because we use a subset of the source and target collection due to time constraints. The test collection statistics are reported in Table 5. As such, we report the results of using bag of words

Query Type	MAP	MRR	P.4
All Terms	0.4554	0.8500	0.6750
Named Entities	0.3979	0.7526	0.5875

Table 11: Results of Evaluation from L4 Project, Adapted from [11]

queries on these test collections in Table 12. Our use of named entities has, however, changed slightly as we now maintain multi word named entities, as discussed in Section 4.

Query Type	MAP	MRR	P.5	Recall
All Terms	0.4337	0.7988	0.5700	0.8299
Named Entities	0.3863	0.7052	0.4600	0.9746

Table 12: The same evaluation, but on this years data

We define the results of Table 12 as a baseline because they represent our previous understanding of the task.

Our experiments will follow a similar structure to those in Section 7.2, reporting the results of the sample subquery followed by the changes observed when reranking based on various features during learning to rank. We use the same learned models from the AP88 data as in Section 7.2. Again, this is because we do not have sufficient training data in the government document test collection.

From analysing the result of Section 7.2 we do not expect to see any significant increase in results when using our new features. We are unsure how the sample subquery will perform, as it may prove to be a better alternative bag of words query than those investigated in the L4 Project.

In Table 13 we report the results of using our learned models on the government test collection. Again the first row represents the performance of the sample subquery alone with no feature based reranking.

The sample subquery used is again the 100 term TF-IDF weighted query as tested in Section 7.1. This produces 100% recall for all source documents in the government document collection, which is an improvement compared to the baseline results in Table 12.

Feature Set	MAP	MRR	P@5
None	0.5842	0.8375	0.6500
DPH_{sample}	0.5547	0.8667	0.6100
$DPH_{sample}, TF_{person}, TF_{org}, TF_{loc}$	0.5013	0.8083	0.5700
$DPH_{sample}, Bin_{VFmonth}$	0.5541	0.8417	0.6300
$DPH_{sample}, TF_{person}, TF_{org}, TF_{loc}, Bin_{VFmonth}$	0.5339	0.7517	0.6100

Table 13: Learning to Rank with Various Feature Combinations on Real Government Documents Collection

Inspecting Table 13 we see that, in comparison to the bag of words queries tested in Table 12, there is an increase over all reported measures when simply performing the sample subquery. Unfortunately, using any features for reranking degrades the performance compared to this sample subquery. We suspect this is because the features chosen do not have a relationship in the training data necessary for identifying relevant documents.

We are specifically interested in the performance of the P@5 measure. This is the precision of the top 5 documents returned, which are likely the first that manual reviewer would examine. Comparing the all terms bag of words query with the sample subquery we find that this increase is not statistically significant when using the two tailed students t-test with a p-value of 0.05.

For the sample subquery map does increase significantly when using the two tailed students t-test with a p-value of 0.05 when compared with the named entity bag of words query.

To summarise the results of this section, our approach of using a learned model did not improve precision when reranking our sample of documents. This is in line with our findings in Section 7.2. We do, however, find that using a 100 term sample subquery weighted with TF-IDF provides an increase in precision and recall over baselines. This means that for future defined features we have a well performing sample subquery to use for the first stage of learning to rank.

9. CONCLUSIONS

Our aim in this research was discover if learning to rank could be used to improve the performance of retrieving public domain documents from queries by implicitly calculating the importance of specific subqueries generated from a source document.

Of the features we defined none showed any improvement over the baseline of the learning to rank sample subquery. Having examined the data we conclude that our feature selection was poor and distributions graphs extracted from the training data could have predicted this lack of performance for the named entity features, however this relationship was not recognised until very late in the project.

We found that we can produce a high recall sample of documents by using the top 100 terms from a source document under a TF-IDF weighting scheme. This sample subquery can be used as the basis for further research involving learning to rank with generated queries from source documents.

In future work we would investigate the performance of additional defined features. Especially we would like to investigate the use of document creation time more heavily. Complex query operators also offer interesting routes for defining features which focus specifically on term proximities. We could also use synonyms as part of the complex query framework. Additionally, more analysis could have been performed on the training test collection to identify which features showed promise. The production of much larger test collection using existing government documents would also be highly beneficial in identifying appropriate features for learning to rank.

Of topics not investigated, we could also investigate the use of knowledge bases for the improving the use of named entities.

Acknowledgements. I would like to thank my supervisors, Craig Macdonald and Graham McDonald for their guidance and patience during this project.

10. REFERENCES

[1] Freedom of information act, 2000 (UK).

[2] Public records act, 1958, as amended (UK).

[3] S. A. Allan. Record review. `https://www.gov.uk/government/publications/records-review-by-sir-alex-allan`.

[4] T. Barrett. FBI worked 'around the clock' to review emails in Clinton server probe. `http://edition.cnn.com/2016/11/06/politics/fbi-review-hillary-clinton-emails-comey/`.

[5] M. Bendersky, D. Metzler, and W. B. Croft. Learning concept importance using a weighted dependence model. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 31–40. ACM, 2010.

[6] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117, 1998.

[7] J. Brutlag. Speed matters for google web search, 2009. `http://bit.ly/IhsoTN`.

[8] C. J. Burges. From ranknet to lambdarank to lambdamart: An overview. Technical report, June 2010.

[9] A. X. Chang and C. D. Manning. Sutime: A library for recognizing and normalizing time expressions. In *LREC*, volume 2012, pages 3735–3740, 2012.

[10] V. Dang, M. Bendersky, and W. B. Croft. Two-stage learning to rank for information retrieval. In *Proceedings of the 35th European Conference on Advances in Information Retrieval*, ECIR'13, pages 423–434, Berlin, Heidelberg, 2013. Springer-Verlag.

[11] K. Fowler. Public vs. private: Identifying public domain knowledge. 2017.

[12] T. Gollins, G. McDonald, C. Macdonald, and I. Ounis. On using information retrieval for the selection and sensitivity review of digital public records. In *Proceedings of PIR@ SIGIR*, pages 39–40, 2014.

[13] A. Jatowt, C.-M. Au Yeung, and K. Tanaka. Estimating document focus time. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 2273–2278. ACM, 2013.

[14] M. A. Khalid, V. Jijkoun, and M. De Rijke. The impact of named entity normalization on information retrieval for question answering. In *Proceedings of the IR Research, 30th European Conference on Advances in Information Retrieval*, ECIR'08, pages 705–710, Berlin, Heidelberg, 2008. Springer-Verlag.

[15] G. Kumaran and J. Allan. Text classification and named entities for new event detection. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, pages 297–304, New York, NY, USA, 2004. ACM.

[16] C.-J. Lee and W. B. Croft. Generating queries from user-selected text. In *Proceedings of the 4th Information Interaction in Context Symposium*, pages 100–109. ACM, 2012.

[17] C. Macdonald, R. McCreadie, R. L. Santos, and I. Ounis. From puppy to maturity: Experiences in developing terrier. *Proc. of OSIR at SIGIR*, pages 60–63, 2012.

[18] C. Macdonald, R. L. T. Santos, and I. Ounis. The whens and hows of learning to rank for web search. *Information Retrieval*, 16(5):584–628, Oct 2013.

[19] C. Macdonald, N. Tonellotto, and I. Ounis. Efficient & effective selective query rewriting with efficiency predictions. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, pages 495–504, New York, NY, USA, 2017. ACM.

[20] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky. The stanford corenlp natural language processing toolkit. In *ACL (System Demonstrations)*, pages 55–60, 2014.

[21] R. McCreadie, C. Macdonald, I. Ounis, J. Peng, and R. L. Santos. University of Glasgow at TREC 2009: Experiments with Terrier. Technical report, University of Glasgow, United Kingdom, 2009.

[22] G. McDonald, C. Macdonald, and I. Ounis. Active learning strategies for technology assisted sensitivity review. In G. Pasi, B. Piwowarski, L. Azzopardi, and A. Hanbury, editors, *Advances in Information Retrieval*, pages 439–453, Cham, 2018. Springer International Publishing.

[23] G. McDonald, C. Macdonald, I. Ounis, and T. Gollins. Towards a classifier for digital sensitivity review. In *Proceedings of the European Conference on Information Retrieval*, pages 500–506. Springer, 2014.

[24] D. Metzler and W. B. Croft. 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, pages 472–479, New York, NY, USA, 2005. ACM.

[25] D. W. Oard, W. Webber, et al. Information retrieval for e-discovery. *Foundations and Trends® in Information Retrieval*, 7(2–3):99–237, 2013. `http://w.codalism.com/research/papers/ow13fntir.pdf`.

[26] T. Qin, T.-Y. Liu, J. Xu, and H. Li. Letor: A benchmark collection for research on learning to rank for information retrieval. *Information Retrieval*, 13(4):346–374, Aug 2010.

[27] T. Strohman, D. Metzler, H. Turtle, and W. B. Croft.

Indri: A language model-based search engine for complex queries. In *Proceedings of the International Conference on Intelligent Analysis*, volume 2, pages 2–6. Amherst, MA, USA, 2005.

[28] J. Strötgen, O. Alonso, and M. Gertz. Identification of top relevant temporal expressions in documents. In *Proceedings of the 2nd Temporal Web Analytics Workshop*, pages 33–40. ACM, 2012.

[29] J. Strötgen and M. Gertz. Heideltime: High quality rule-based extraction and normalization of temporal expressions. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 321–324. Association for Computational Linguistics, 2010.

[30] J. Xu and W. B. Croft. Query expansion using local and global document analysis. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '96, pages 4–11, New York, NY, USA, 1996. ACM.