A Study of Retrieval Models for Long Documents and Queries in Information Retrieval

Ronan Cummins
The Computer Laboratory
University of Cambridge, UK
ronan.cummins@cl.cam.ac.uk

ABSTRACT

Recent research has shown that long documents are unfairly penalised by a number of current retrieval methods. In this paper, we formally analyse two important but distinct reasons for normalising documents with respect to length, namely *verbosity* and *scope*, and discuss the practical implications of not normalising accordingly. We review a number of language modelling approaches and a range of recently developed retrieval methods, and show that most do not correctly model both phenomena, thus limiting their retrieval effectiveness in certain situations. Furthermore, the retrieval characteristics of long natural language queries have not traditionally had the same attention as short keyword queries. We develop a new discriminative query language modelling approach that demonstrates improved performance on long verbose queries by appropriately weighting salient aspects of the query. When combined with query expansion, we show that our new approach yields *state-of-the-art* performance for long verbose queries.

1. INTRODUCTION

With the increased variety in the forms of queries (e.g. *copy-and-paste*, spoken, expanded queries) and the increased variation in what constitutes a *document* (e.g. micro-blogs, poetry, lyrics, video lectures, parliamentary debates, books) length normalisation is an increasingly important aspect of any retrieval method. Recent research [12, 13, 14] has shown that long documents are unfairly treated by many retrieval methods.

Robertson and Walker [19] hypothesised two main reasons for varying document lengths, namely *verbosity* and *scope*. Verbosity relates to the tendency of an author to use more words than is necessary when writing on a particular topic. It captures an aspect of document length that is independent of *relevance*. For example, consider the following excerpt from the book *"Green Eggs and Ham"* by Dr. Seuss which contains 28 tokens but only 15 word types (terms).[1]

I would not like them here or there.
I would not like them anywhere.
I do not like green eggs and ham.
I do not like them, Sam-I-am

The entire book *"Green Eggs and Ham"* contains approximately 750 tokens but contains only 50 word types and would typically be considered verbose. Therefore, controlling for verbosity usually involves normalising the term-frequency counts in some way, as if we have two documents where the *only* difference between them is that one document is more verbose (e.g. concatenating a document with itself), the verbose document would have higher term-frequencies simply due to its length.

On the other hand, the scope hypothesis captures the alternative intuition that documents may be longer because they cover a variety of topics. Documents that cover a variety of non-relevant topics need to be penalised. One can imagine a hypothetical document that is created by the concatenation of all documents in the collection. Although this hypothetical document may contain information relevant to a query, its scope is so broad that its utility is negligible. Therefore, it makes sense to normalise the document according to the scope of the document. However, while scope has traditionally been seen as document specific, we argue that the level of scope normalisation to apply to a document is also dependent on the query. The intuition behind this is that the likelihood of a document matching any aspect of the query, is also dependent on the scope of the query. As a result, for long queries the level of scope normalisation needs to be greater as we do not wish to over-promote documents with a broad scope.

Although short keyword queries have traditionally been the focus of experimental evaluations, longer natural language queries have received more attention of late [1, 2, 17]. The most common way of estimating a query model in the language modelling framework [23] is by using maximum-likelihood (ML) estimates, which essentially assumes that all tokens generated by the query model are equally important. However, for long natural language queries it is

[1] The distinction between *word types* (i.e. lexical signifiers that represents certain concepts) and *word tokens* (i.e. the occurrences of certain types in situation) is a fundamentally important property of language [22]. It is unfortunate that the word *term* in the information retrieval literature has sometimes been used to identify either *word types* or *word tokens*.

WWW 2016, April 11–15, 2016, Montréal, Québec, Canada.
ACM 978-1-4503-4143-1/16/04.
http://dx.doi.org/10.1145/2872427.2883009.

highly likely that many *noisy* tokens are generated from a *background* query language model. This suggests correctly modelling long verbose queries would lead to improvement in retrieval effectiveness.

In this paper, we formally describe the verbosity hypotheses and the interaction between document scope and the query. Subsequently we analyse a representative selection of modern retrieval methods and in particular, we aim to discover how changes in document verbosity affect the retrieval effectiveness of these methods. To analyse the interaction of document and query scope, we perform a simulation that studies the change in effectiveness of the different retrieval methods for varying query lengths.

Focusing our attention on modelling longer natural language queries, we develop a new principled discriminative query language model (DQM). By determining the probability that a particular query token is *topical*, the model aims to correctly weight the salient aspects of the query. We incorporate this new query model into a number of document language models and demonstrate improved performance on longer queries. Finally, we show that pseudo-relevance feedback can be used to further improve performance on longer queries, outperforming the current *state-of-the-art* for verbose queries.

The remainder of the paper is as follows: Section 2 outlines related work in the area of document normalisation and language modelling. Section 3 introduces three approaches to smoothing language models and a means of determining the topical terms in documents. Section 4 introduces a number of formal constraints regarding verbosity and scope. An analysis and simulation of a number of retrieval methods is presented in Section 5. In Section 6 we develop our discriminative query model, while Section 7 presents our experimental results. Finally, Section 8 concludes with a discussion and conclusion.

2. RELATED WORK

Table 1 outlines the notation used throughout the paper. Most retrieval functions outlined in the literature can be seen as scoring a document d in a collection c with respect to a query q as follows:

$$f(q, d) = \sum_{t \in q} dw(d, t, c) \cdot qw(q, t, c) \tag{1}$$

where $dw(d, t, c)$ weights the term in the document and $qw(q, t, c)$ weights the term in the query.

2.1 Length Normalisation

Document length normalisation is a well-studied area in information retrieval [18, 21, 9, 12, 5]. Early work [18] introduced both the scope hypothesis and the verbosity hypothesis, on which there has been substantial subsequent research and comment [15, 26, 3]. Recent work [13, 12] has recognised that a number of state-of-the-art retrieval methods over-penalise long documents, and while they have developed a number of constraint-based modifications of the retrieval methods to overcome this problem, there lacks a formal treatment of document length normalisation with regard to both verbosity and scope. Table 2 outlines the modifications to BM25 [19] (BM25+), and the multinomial language model with a Dirichlet prior [25] (Dir+) that are reported to fairly retrieve long documents.[2] Paik develops a highly effective retrieval method [16] which is reported as incorporating aspects of verbosity and scope. This *Multi Aspect TF* (MATF) retrieval method is shown in Table 2 but has not been extensively tested for longer queries. Even more recently [14], a more formal treatment of scope and verbosity has been proposed which leads to modifications of BM25 and a language modelling approach that incorporate both kinds of normalisation. Interestingly, the modification to the language model results in a very similar retrieval method to that developed in a recent language modelling approach based on the multivariate Pólya distribution [6]. This Pólya document language model captures word burstiness and reportedly contains both verbosity and scope normalisation. This document language model is referred to as SPUD in Table 2, and will be used as the retrieval method upon which we will develop further improvements for longer queries. The study of verbosity and scope in this paper includes several of these new retrieval functions (including BM25+, Dir+, MATF, SPUD).

Table 1: Feature Notation

Key	Description
$c(t,d)$	frequency of word type t in document d
$c(t,q)$	frequency of word type t in query q
$\|d\|$	# of tokens in document d
$\|q\|$	# of tokens in query q
$\|\vec{d}\|$	# of types in document d
$\|\vec{q}\|$	# of types in query d
cf_t	frequency of type t in the entire collection
df_t	document frequency of type t
$\|c\|$	# of tokens in the collection c
$\|v\|$	# of types in the collection (vocabulary)
N	number of documents in the collection
$\|d\|_{avg}$	average # of tokens in a document

In the language modelling approach to information retrieval, it has been shown that the hyper-parameters in the document models should be tuned to different settings for optimal retrieval when using queries of different length (i.e. higher μ in Dir and higher b in BM25). However, this is theoretically anomalous as the document model is a model of document generation and should remain static when the collection is static. If different weightings are needed for different kinds of query for whatever reason, then this is a strong indication that it is the query model that needs to account for this. In an attempt to address this problem, Zhai and Lafferty introduced the two-stage language model [24] that applies both Dirichlet smoothing and Jelinek-Mercer smoothing. We use this approach as a baseline against which we compare the discriminative query models developed in this paper.

3. LANGUAGE MODELLING

In this section, we briefly outline three document language models that we further analyse in this work. In addition,

[2] The original version of BM25 and the Dirichlet-prior language model can be found by setting the lower-bounding parameter $\delta = 0$.

Table 2: State-of-the-art retrieval functions with different document normalisation characteristics. The parameter settings are those suggested in the literature for use with longer queries.

Method	$dw(d,t,c)$	$qw(q,t,c)$	Default Hyper-parameters
MATF	$[w \cdot \frac{log_2(1+c(t,d))}{log_2(1+c(t,d)+log_2(1+(\|d\|/\|\vec{d}\|))} + (1-w)\frac{c(t,d) \cdot log_2(1+\frac{\|d\|avg}{\|d\|})}{(c(t,d) \cdot log_2(1+\frac{\|d\|avg}{\|d\|})+1)}] \cdot \frac{log(\frac{N+1}{df_t}) \cdot \frac{cf_t}{df_t}}{\frac{cf_t}{df_t}+1}$	$c(t,q)$	$w = \frac{2}{1+log_2(1+\|q\|)}$
BM25+	$(\frac{(k_1+1) \cdot c(t,d)}{c(t,d)+k_1 \cdot ((1-b)+b \cdot \|d\|/\|d\|_{avg})} + \delta) \cdot log(\frac{N}{df_t})$	$c(t,q)$	$k_1 = 1.2, b = 0.75,$ $\delta = 1.0$
JM	$log((1-\lambda) \cdot \frac{c(t,d)}{\|d\|} + \lambda \cdot \frac{cf_t}{\|c\|})$	$c(t,q)/\|q\|$	$\lambda = 0.7$
Dir+	$log(\frac{c(t,d)}{\|d\|+\mu} + \frac{\mu \cdot cf_t/\|c\|}{\|d\|+\mu}) + log(1 + \frac{\delta}{\mu \cdot cf_t/\|c\|})$	$c(t,q)/\|q\|$	$\mu = 2000, \delta = 0.05$
SPUD	$log((1-\omega) \cdot \frac{m_d \cdot c(t,d)}{\|d\|} + \omega \cdot \frac{m_c \cdot df_t}{\sum_{t'} df_{t'}})$	$c(t,q)/\|q\|$	$\omega = 0.8, m_d = \|\vec{d}\|$

for each particular language model we determine, via Bayes' rule, the probability that a particular term is topical in a document.

3.1 Smoothed Document Models

In the language modelling approach each document is assumed to have been drawn from a smoothed document model $\boldsymbol{\mathcal{M}}_d$ as follows:

$$p(t|\boldsymbol{\mathcal{M}}_d) = (1-\lambda) \cdot p(t|\boldsymbol{\mathcal{M}}_\tau) + \lambda \cdot p(t|\boldsymbol{\mathcal{M}}_c) \qquad (2)$$

where $\boldsymbol{\mathcal{M}}_\tau$ models the topical aspect of the document and $\boldsymbol{\mathcal{M}}_c$ is a background model of language. When using a multinomial language model ($\boldsymbol{\mathcal{M}} = \boldsymbol{\theta}$), the probability of a term t given the topic model $p(t|\hat{\boldsymbol{\theta}}_\tau)$ is estimated as $c(t,d)/|d|$ and the probability of t given the background model $p(t|\hat{\boldsymbol{\theta}}_c)$ is estimated as $cf_t/|c|$. The multinomial model with Jelinek-Mercer smoothing (JM) is instantiated by setting λ to a constant value, while the multinomial with Dirichlet-prior smoothing (Dir) is instantiated by setting $\lambda = \mu/(\mu + |d|)$ where μ is a constant value and has been shown to be more stable than Jelinek-Mercer smoothing [25].

More recently a document language model (called SPUD) [6] captures word burstiness by assuming that documents are generated by a multivariate Pólya distribution ($\boldsymbol{\mathcal{M}} = \boldsymbol{\alpha}$). In this model the hyper-parameter, denoted ω instead of λ, controls the smoothing and is stable at $\omega \approx 0.8$. The parameters of the model can be interpreted as assuming a Dirichlet prior over a multinomial topical model and a multinomial background model parameterised as follows:

$$\hat{\boldsymbol{\alpha}}_\tau = \{m_d \cdot \frac{c(t,d)}{|d|} : t \in d\} \quad \hat{\boldsymbol{\alpha}}_c = \{m_c \cdot \frac{df_t}{\sum_{t'} df_{t'}} : t \in c\} \qquad (3)$$

where m_d is a parameter which is inversely related to the burstiness of the individual document model, and m_c is a background mass parameter that can be estimated once via numerical methods as outlined in the original work [6]. Both scale parameters m_d and m_c can be interpreted as beliefs in the parameters $\frac{c(t,d)}{|d|}$ and $\frac{df_t}{\sum_{t'} df_{t'}}$ respectively. However, while the interpretation of both of these parameters is understood, m_d has no single objective estimate as there is a lack of samples available from the document model to be able to estimate $\boldsymbol{\alpha}_\tau$. Appealing to an argument of parsimony, m_d was recommended to be set to the number of non-zero dimensions in d (i.e. $|\vec{d}|$) which also corresponds to the mass of the maximum entropy Dirichlet and is often referred to as Bayes' prior probability.

3.2 Query Models and KL-Divergence

The query-likelihood approach is often used to rank document models (JM, Dir, and SPUD) with respect to a query string. An alternative approach, and one which is adopted for all language models used in this paper, is to assume that the query has been drawn from a query model [23] and to rank document models with respect to the query model using the negative of the KL divergence. For all the language models in this paper, we rank according to following function:

$$-KL(q,d) \propto \sum_{t \in q} log\ p(t|\boldsymbol{\mathcal{M}}_d) \cdot p(t|\boldsymbol{\mathcal{M}}_{q\tau}) \qquad (4)$$

where $\boldsymbol{\mathcal{M}}_{q\tau}$ is a model of the topic of the query. Eq. 4 can be seen as a slightly more constrained version of the general ranking function in Eq. 1. In Section 6, we introduce a new discriminative query model (DQM) to define $\boldsymbol{\mathcal{M}}_{q\tau}$ for use with long natural language queries in the KL divergence framework.

3.3 Probability of Topicality

In general, the language modelling approach is a generative ranking approach, whereby we can generate documents from a document model if needed. However, when given a particular term t occurring in a document d, one could ask the more discriminative question, *what is the probability that term t was drawn from the topical part of the model?* This is related to Harter's idea of eliteness [8] and other descriptions of *aboutness* [20]. Using Bayes' theorem with Eq. (2), the probability of t being generated by the topical model of d is as follows:

$$p(\boldsymbol{\mathcal{M}}_\tau|t) = \frac{(1-\lambda) \cdot p(t|\boldsymbol{\mathcal{M}}_\tau)}{(1-\lambda) \cdot p(t|\boldsymbol{\mathcal{M}}_\tau) + \lambda \cdot p(t|\boldsymbol{\mathcal{M}}_c)} \qquad (5)$$

which reduces to the following in the case of Jelinek-Mercer smoothing:

$$p_{JM}(\boldsymbol{\theta}_{\tau}|t) = \frac{\frac{c(t,d)}{|d|}}{\frac{c(t,d)}{|d|} + \frac{\lambda}{1-\lambda} \cdot \frac{cf_t}{|c|}} \tag{6}$$

and

$$p_{Dir}(\boldsymbol{\theta}_{\tau}|t) = \frac{c(t,d)}{c(t,d) + \mu \cdot \frac{cf_t}{|c|}} \tag{7}$$

in the case of Dirichlet-prior smoothing. It can also be shown that the probability of t being topical in d in the SPUD language model is:

$$p(\boldsymbol{\alpha}_{\tau}|t) = \frac{\frac{c(t,d)}{|d|/|\vec{d}|}}{\frac{c(t,d)}{|d|/|\vec{d}|} + \frac{\omega}{1-\omega} \cdot \frac{df_t}{\sum_{t'} df_{t'}} \cdot m_c)} \tag{8}$$

Firstly, it is interesting to note the general shape of these functions is similar to that of BM25 as the probability of t being topical is asymptotic, and in the case of JM and SPUD uses some length normalisation. Furthermore, this probability (i.e $p(\boldsymbol{\mathcal{M}}_{\tau}|t)$) can be used to compare terms across documents and therefore can be used to analyse verbosity normalisation in the language model (i.e. scope normalisation has been eliminated from these equations due to the re-formulation). We use these formula for analysis in Section 5 and again in Section 6 to define a discriminative query model useful for longer queries.

4. VERBOSITY AND SCOPE

Recent work [14] has formulated the relationship between the scope $s(d)$ and the verbosity $v(d)$ of a document d as follows:

$$v(d) = \frac{|d|}{s(d)} \tag{9}$$

such that one only needs to specify a measure of scope $s(d)$ to be able to determine verbosity, or vice versa. As previously mentioned verbosity normalises within-document term-frequencies (i.e. $c(t,d)/v(d)$), while scope is some measure of the breath of the information in a document. Before outlining some constraints on $s(d)$, we review a formal constraint [15, 6] that the verbosity hypothesis implies, and which we will use in a subsequent simulation. For any particular document d, a more verbose document can be constructed by concatenating the document with itself until it is k times its original size. This hypothetical verbose document does not cover more topics and so given any query, the relevance of this more verbose document should be equal to the relevance of the original document to the query. The following constrains retrieval functions so that they adhere to the verbosity hypothesis:

Constraint 1 (LNC2*). *If document d and d' are two documents, where d' is constructed by concatenating d with itself until it is k times its original length, and if $f(q,d)$ is the score returned from a retrieval function f which is used to rank d with respect to q, then $f(q,d) = f(q,d')$.*

Furthermore, the following constraint captures the intuition regarding scope and its interaction with the query:

Constraint 2 (SQLNC). *Let q be a query and assume that d_1 and d_2 are two documents such that $q \subset \{d_1\}$, $q \subset \{d_2\}$. Furthermore, let us assume that $f(q,d_1) = f(q,d_2)$ and $s(d_2) > s(d_1)$. If we create a new query q' by adding to q a previously unseen query term t (i.e. $t \notin q$) such that $s(q') > s(q)$, and if $t \notin \{d_1\}$ and $t \notin \{d_2\}$, then $f(q,d_1) > f(q',d_1)$, $f(q,d_2) > f(q',d_2)$, and $f(q',d_1) > f(q',d_2)$.*

Firstly, this constraint ensures that documents that mismatch query terms get penalised. When a term t that does not appear in d_1 or d_2 is added to the original query q, the score of both documents should decrease (i.e. $f(q,d_1) > f(q',d_1)$ and $f(q,d_2) > f(q',d_2)$). Furthermore, the constraint also ensures that documents that have a greater scope get penalised more when they mismatch query information ($f(q',d_1) > f(q',d_2)$). The penalisation is query sensitive because documents with a greater scope are more likely to contain each distinct query term. For a single query, this constraint does not necessarily[3] lead to a re-ranking of documents. However, the constraint regulates scope normalisation over a set of queries and is likely to be important for session-based IR [10]. In a practical setting, it is very likely that queries relating to the same information need are reformulated and re-submitted to an IR system during a single session and therefore both q and q' could appear in the same user session. We note that a related, but more limited, constraint has previously been proposed [5].

As yet we have not specified how one might measure scope. A previous constraint (SC1) states that $s(d)$ is a non-decreasing function of document length $|d|$. We now outline two further constraints on any measure of scope.[4] If $s(d)$ is some measure of the scope of a document then:

Constraint 3 (SC3). *If document d and d' are two documents, where d' is constructed by concatenating d with itself, then $s(d) = s(d')$.*

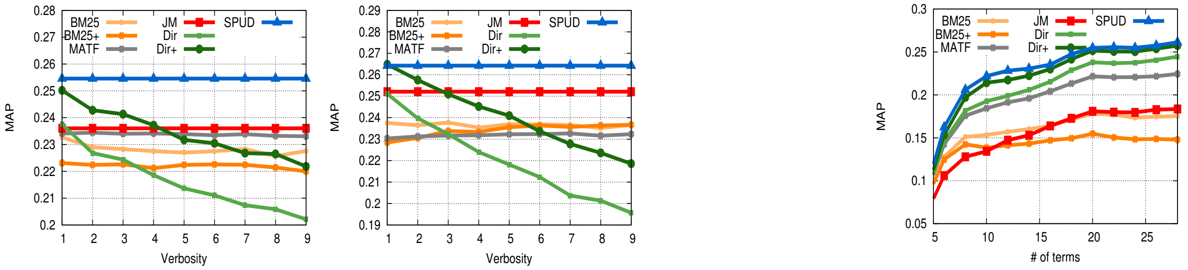
which follows from LNC2* previously and

Constraint 4 (SC4). *If t is a term and d is a document such that $t \notin d$, and if we construct a new document d' by adding to d an occurrence of t, then $s(d') > s(d)$.*

The first of these two formalisms constrains the measure of scope such that $s(d)$ must also be invariant to the hypothetical kind of document self-concatenation and follows intuitively from the definition of scope. This rules out many the measures of scope formulated in previous work [14] and in particular, it rules out measures of scope that use a strictly increasing monotonic transformation of $|d|$. The second of these two constraints states that as new word types appear in the document, scope increases. In a *bag-of-words* framework, this second constraint follows intuitively from what is meant by scope, as authors tend to introduce new word types when increasing the scope of a document. However, when considering semantic matching functions, SC4 might

[3] Often hyper-parameters can be tuned to optimise performance for one query (e.g. tuning b in BM25)

[4] We note that in previous work [14] another constraint SC2 states that verbosity ($v(d)$) is also a non-decreasing function of the document length $|d|$. We believe this to be too strong an assumption.



(a) Change in performance (MAP) as verbosity (n) increases on robust-04 for *description only* queries (left) and for *description and narrative* queries (right).

(b) Change in Performance (MAP) as query scope ($|\vec{q}|$) increases on robust-04 by increasing number of terms used in *description and narrative* queries.

Figure 1: Simulation of changing (a) document verbosity and (b) query scope independent of relevance

be relaxed as the appearance of new terms that are synonymous with previously seen terms may not necessarily increase scope. We will analyse the retrieval functions with respect to these constraints in the next section.

5. ANALYSIS AND SIMULATIONS

In this section we analyse both scope and verbosity hypotheses of a number of ranking functions via simulation.

5.1 Verbosity Analysis

We now analyse all of the retrieval methods outlined in Table 2 with regard to the LNC2* verbosity constraint in the previous section. We can formally analyse all retrieval methods by ensuring that $f(q, d) = f(q, k \times d)$. When concatenating a document with itself, the frequency of t in d becomes $k \times c(t, d)$ and the document length in tokens becomes $k \times |d|$. For BM25 it can be shown that the constraint leads to the following equality:

$$\frac{c(t,d)}{(1-b) + b \cdot |d|/|d|_{avg}} = \frac{k \cdot c(t,d)}{(1-b) + b \cdot k \cdot |d|/|d|_{avg}} \quad (10)$$

which is only true when $b = 1$ and is usually not an effective setting for BM25 [18]. Furthermore, for all three standard language modelling approaches (JM, Dir, and SPUD), we can determine adherence to LNC2* by examining the probability of topicality (i.e. $p(\boldsymbol{\mathcal{M}}_\tau|t)$ for each of the methods. From equations 6 to 8, we can see that the term-frequency aspect is normalised for verbosity using $v(d) = |d|$ in JM, by $v(d) = 1$ in Dir (i.e. no verbosity normalisation), and by $v(d) = |d|/|\vec{d}|$ in SPUD. Therefore, the document score of JM and SPUD is invariant when a document is concatenated with itself as $c(t, d)/v(d)$ is invariant for both of these functions, thus adhering to LNC2*. However, Dir (equation 7) does not adhere to the constraint as there is no term-frequency normalisation. It also follows that Dir+ does not adhere to the verbosity constraint. This implies that the measure of verbosity $v(d)$, and consequently the scope $s(d)$, is different for the three main language models (JM, Dir, and SPUD).

Adherence to this constraint has potential implications for *adversarial* search. For example, if an author wishes their document to be retrieved for specific query-terms, they can artificially increase the score that their document will receive by artificially concatenating the document with itself numerous times. In order to determine the practical repercussions of failing to adhere to LNC2*, we next conduct a simulation using TREC data.

5.2 Verbosity Simulation

In order to analyse document verbosity independent of other effects, we simulate changing the verbosity of documents in a collection. We use the Robust-04 collection as it contains homogeneous documents (Newswire articles) whose lengths do not vary substantially (see Table 6). We then change the verbosity of a selection of documents by concatenating each selected document with itself. Specifically for $k = \{1 : n\}$ we select $1/n$ documents and self-concatenate each document until they reach k times their original length. For example, if $n = 3$ we divide the collection into three, where the documents in the first third remain their original length ($k = 1$), for the second third the documents grow to twice their size ($k = 2$), and for the final third the documents are three times longer ($k = 3$) than their original length in tokens. Therefore, as n grows the degree of document verbosity increases for different documents in different amounts.

In particular, we simulate the above scenario by changing the term-frequency ($c(t, d)$) and relevant length measures ($|d|$ and $|d|_{avg}$) in Table 1 prior to using a particular retrieval method. We did not alter the relevant collection-wide statistics (df_t, cf_t, $|c|$, or $\sum_t df_t$) so that the measures of term importance remain unaltered. However, we did alter the average document length in tokens used in MATF and BM25 (BM25+) because the average length aspect $|d|_{avg}$ is used as a document length pivot in those functions.

Fig. 1a shows the results of this simulation for both *description only* queries and for combined *description and narrative* queries. The hyper-parameters of each retrieval method were tuned to the initial unmodified collection. The effectiveness in terms of mean average precision (MAP) for both types of queries for the seven different retrieval methods is shown. We can see that Dir and Dir+ are particular sensitive to a change in verbosity as the performance changes dramatically as verbosity increases. This occurs because the mixture parameter $\frac{|d|}{\mu+|d|}$ is sensitive to the number of tokens in the document. If the number of tokens in a document $|d|$ increases dramatically, the Dir and Dir+ retrieval methods place more credence on the within-document features, and therefore many documents are being retrieved increasingly based on their within-document information. BM25, BM25+, and MATF are less sensitive to changes in verbosity but are not invariant. In summary, and consistent with our analysis, JM and SPUD are the only functions that correctly model verbosity.

5.3 Scope Analysis

We now analyse all of the retrieval methods outlined in Table 2 with regard to the SQLNC constraint. Firstly, the BM25 (and BM25+) document score does not decrease if a new query-term is absent in a document. Therefore, they cannot adhere to SQLNC. The JM language model can also be written such that it *only* involves a summation of query-term matches [25] and so does not decrease when a new query-term mismatches. Given that the measure of verbosity in JM is $v(d) = |d|$, if follows from Eq. 9 that the scope is $s(d) = 1$ (i.e. no scope normalisation).

Conversely, the Dir (and Dir+) language model contains a document penalisation factor of $log(\mu/(\mu + |d|)$ for every query token, and so the document score decreases for every new query-term does not appear in the document. SPUD's penalisation factor is $log(\mu/(\mu + |\vec{d}|)$ and so also decreases when a new query-term mismatches. From the previous section we discovered that for Dir (and Dir+), $v(d) = 1$ which implies that $s(d) = |d|$ from Eq. 9, while for SPUD $v(d) = |d|/|\vec{d}|$ which implies that $s(d) = |\vec{d}|$. From this analysis, we can see that JM has only verbosity normalisation, while Dir (Dir+) has only scope normalisation. However, the scope normalisation employed by Dir does not adhere to SC2. The only language model that adheres SQLNC and SC2 is SPUD.

5.4 Scope Simulation

In order to analyse the effect that the query scope has on the effectiveness of different retrieval methods, we measure the effectiveness of each retrieval method as queries grow in length. In particular, we again used the description and narrative fields of each topic in the Robust-04 dataset. We created initial queries by extracting all tokens appearing in natural order up to the first 5 word-types. We tuned all retrieval methods to these limited queries. Subsequently, we issued a new query for each topic each time a new word type appeared in the description and narrative. Fig 1b shows the change in effectiveness for each retrieval method as new word types are encountered in the topics. The first thing to note is that many of the retrieval methods perform similarly when tuned for shorter 5-term queries. However, as queries grow in length their performance differs greatly, although in general for all retrieval methods, longer queries are more effective.

As is well-known, BM25 (and BM25+) needs to be tuned for queries of different lengths, and therefore performs poorly as queries grow in length. Similarly JM does not adhere to SQLNC and also performs poorly for longer queries compared to the other language models. The best performing approaches are those that adhere to SQLNC as their scope normalisation adapts automatically to queries of different length.

5.5 Summary

Given the analysis in this section, we now summarise our findings in Table 3. Although our analysis has not specifically focused on MATF, we have included the results here.

Table 3: Adherence to Constraints

	Constraint				
Method	LNC2*	SQLNC	SC1	SC3	SC4
MATF	no	cond.	no	no	no
BM25	if $b = 1.0$	no	no	no	no
BM25+	if $b = 1.0$	no	no	no	no
JM	yes	no	no	no	no
Dir	no	yes	yes	no	no
Dir+	no	yes	yes	no	no
SPUD	yes	yes	yes	yes	yes

6. VERBOSE QUERY MODELLING

In this section, we outline a new discriminative query modelling (DQM) approach for long natural language queries.

6.1 Discriminative Query Model (DQM)

When a user formulates a short keyword query (e.g. *black, bear, attacks*), it is usually assumed that they have already distilled the topical aspect of the information need. Consequently, one may assume that the probability that a particular query token is topical is 1.0 and this can be normalised accordingly to estimate the maximum likelihood query model (i.e. $\{\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\}$). This is the standard method of estimating query models for use with KL-Divergence.

However, when dealing with natural language queries (e.g. *A relevant document would discuss the frequency of vicious black bear attacks worldwide and the possible causes for this savage behavior*) it is likely that many terms are generated by a background query language model. Therefore, we assume that long natural language queries are generated by drawing terms from a query model $\boldsymbol{\mathcal{M}}_q$ which consists of both a topical language model $\boldsymbol{\mathcal{M}}_{q\tau}$ and a background query language model $\boldsymbol{\mathcal{M}}_{qc}$. The topical query model describes the topical information that the user requires, while the background query model describes the syntactic glue of the general query language. Examples of fragments that can be explained by the background query language model are tokens such as *"I, am, looking, for,"*, and *"A, relevant, document, may, include,"* (a stereotypical TREC construct). Therefore, our new query model is defined as follows:

$$\boldsymbol{\mathcal{M}}_q = (1 - \lambda_q) \cdot \boldsymbol{\mathcal{M}}_{q\tau} + (\lambda_q) \cdot \boldsymbol{\mathcal{M}}_{qc} \qquad (11)$$

where λ_q is the probability mass of the background query language model. Although the background query language model is likely to contain some structural clues regarding relevance, in this paper we simple regard this model as generating noise tokens, and therefore aim to extract the topical part of each query. This can be achieved in a similar manner to before (Section 3.3) by determining the probability that a particular query term t was generated by the topical query model using Bayes' theorem as follows:

$$p(\boldsymbol{\mathcal{M}}_{q\tau}|t) = \frac{(1 - \lambda_q) \cdot p(t|\boldsymbol{\mathcal{M}}_{q\tau})}{(1 - \lambda_q) \cdot p(t|\boldsymbol{\mathcal{M}}_{q\tau}) + (\lambda_q) \cdot p(t|\boldsymbol{\mathcal{M}}_{qc})} \qquad (12)$$

The final step involves determining the distribution of terms in the topical query model $\boldsymbol{\mathcal{M}}_{q\tau}$ by normalising over the tokens in q as follows:

Table 4: Standard Query Model and Discriminative Query Model for TREC Topic 336

A relevant document would discuss the frequency of vicious black bear attacks worldwide and the possible causes for this savage behavior.								
	Standard		TA [17]		DQM$_c$		DQM$_q$	
Rank	term	$\boldsymbol{\mathcal{M}}_{q\tau}$	term	$\boldsymbol{\mathcal{M}}_{q\tau}$	term	$\boldsymbol{\mathcal{M}}_{q\tau}$	term	$\boldsymbol{\mathcal{M}}_{q\tau}$
1	relev	0.083	attack	0.131	viciou	0.100	bear	0.118
2	document	0.083	bear	0.128	savag	0.100	savag	0.118
3	discuss	0.083	behavior	0.125	frequenc	0.097	viciou	0.118
4	frequenc	0.083	frequenc	0.114	behavior	0.095	frequenc	0.118
5	viciou	0.083	caus	0.102	worldwid	0.092	black	0.114
6	black	0.083	document	0.081	relev	0.088	attack	0.106
7	bear	0.083	relev	0.075	bear	0.084	behavior	0.106
8	attack	0.083	viciou	0.041	black	0.078	worldwid	0.103
9	worldwid	0.083	black	0.054	attack	0.073	caus	0.054
10	caus	0.083	savag	0.039	document	0.070	discuss	0.023
11	savag	0.083	discuss	0.026	caus	0.061	document	0.009
12	behavior	0.083	worldwid	0.021	discuss	0.057	relev	0.008

$$p(t|\boldsymbol{\mathcal{M}}_{q\tau}) = \frac{c(t,q)\cdot p(\boldsymbol{\mathcal{M}}_{q\tau}|t)}{\sum_{t'\in q}(c(t',q)\cdot p(\boldsymbol{\mathcal{M}}_{q\tau}|t'))} \qquad (13)$$

which we call the discriminative query model (DQM). For the specific instantiation of the model using multivariate Pòlya distributions ($\boldsymbol{\mathcal{M}}_{q\tau} = \boldsymbol{\alpha}_{q\tau}$), the probability that a particular term t came from the topical part of the query model, when assuming that the background query model is the general collection, is as follows:

$$p(\boldsymbol{\alpha}_{q\tau}|t) = \frac{c(t,q)}{c(t,q) + \frac{(1-\omega_q)}{\omega_q}\frac{df_t}{\sum_{t'} df_{t'}}\frac{m_c\cdot|q|}{|\vec{q}|}} \qquad (14)$$

which can be plugged into Eq. 13 to yield the DQM using the multivariate Pòlya. The one free parameter in this specific query model is ω_q which determines the belief in the background query model. Table 4 (DQM_c) shows an example of the output of DQM using this model, where we can see more plausible topical probabilities for the natural language query. Furthermore, we can also formulate corresponding DQMs using JM smoothing and Dirichlet smoothing in a similar manner by plugging equations 6 or 7 into Eq. 13. These new query models can replace the maximum-likelihood query model in Eq. 4 and be used to rank documents accordingly.

6.2 Query Background Models

In an online setting, a large background query language model could be built using a query-log consisting of long natural language queries. However, given that such resources are often unavailable, we simulate this ideal scenario in two different ways. As a first approach, we simply use the same background model as used by the document models, as per Eq. 14. This essentially means that terms that appear frequently in the collection will be weighted less in the query. As a second alternative approach, we use a large set of TREC topics as a background query model. In particular, we use 500 topics (350-550, 600-850) and use the description and narrative fields to build the background model. Table 5 shows the top 10 most frequent terms in the background query models that are built from the two different approaches. The most frequent query terms appearing in the collection are quite different from those appearing across all queries. It is likely that terms that appear in many long queries are those that users tend to use when requesting documents (e.g. *document*, *provide*, *specify*, *identify*) and that contribute little topical information. Experiments that test these two approaches are outlined in Section 7.3. In order to distinguish a retrieval method that uses DQM with the document collection to a DQM that uses the set of 500 topics, we use the notation DQM$_c$ and DQM$_q$ respectively.

Table 5: 10 most probable terms in background query model

	WT2g docs		500 topics	
Rank	term	$p(t|\boldsymbol{\theta}_c)$	term	$p(t|\boldsymbol{\theta}_{qc})$
1	inform	0.00054	relev	0.061
2	home	0.00054	document	0.049
3	time	0.00052	discuss	0.016
4	page	0.00050	inform	0.013
5	includ	0.00046	includ	0.008
6	provide	0.00042	describ	0.008
7	public	0.00040	contain	0.008
8	servic	0.00040	specif	0.007
9	gener	0.00038	provid	0.005
10	nation	0.00037	identifi	0.005

6.3 Hyper-parameter Sharing

Now that we have outlined the new discriminative query model, we briefly specify how we pair these with specific document modelling approaches. To specify a complete retrieval method, we assume one type of smoothing for both the document model and the discriminate query model. For example, if we assume that text is generated using a multinomial with Dirichlet smoothing (Dir), we use this for both the document model and the discriminative query model. This pairing of models results in four new retrieval methods that use DQM, which we call DQMJM, DQMDir, DQMDir+, and DQMSPUD.[5]

Each of the standard document language models outlined in Section 3 has one free hyper-parameter (i.e. λ, μ, or ω). Furthermore, each of the discriminative query models

[5] Exploring all possible pairs of combinations is left for future work.

Table 6: Test collection characteristics

document statistics							query statistics	
Collection	Genre	# docs	avg. # types per doc	avg. # tokens per doc	dev. # tokens per doc	$\hat{m}_c$	avg. # tokens per desc query	avg. # tokens per desc+narr query
WT2g	Web	247,491	289	742	1993	352	13	47
Robust-04	News	528,155	193	343	704	240	15	54
WT10g	Web	1,691,000	193	457	2158	257	11	35
Gov2	Web	25,205,179	210	682	1798	201	11	57

also contains one free parameter (i.e. λ_q, μ_q, ω_q) which corresponds in meaning to the parameters in the document model. In order to avoid excessive parameter tuning, we allow these parameters to be shared such that $\lambda = \lambda_q$, $\mu = \mu_q$[6], and $\omega = \omega_q$. This results in DQMJM, DQMDir, DQMDir+, and DQMSPUD containing one free-parameter and allows a fairer comparison with the original models (JM, Dir, Dir+, SPUD).[7] Fundamentally, this type of hyper-parameter sharing constrains the document and query models, and means that we have the same generative assumptions for both the document and query. Without any evidence to suggest that verbose queries and documents are generated differently, we appeal to Occam's razor.

7. EXPERIMENTS

In this section we present the test collections, baselines, and experimental results. Firstly, we aim to evaluate the performance of the discriminative query models (DQM) for all of the language models outlined in this work. Secondly, we aim to compare the performance of our best approach combined with query expansion to the *state-of-the-art* approach to verbose queries [17].

7.1 Test Collections

For ease of comparison with existing work [1, 2, 17], we use similar test collections and sets of queries. As a first set of queries, we used the *description* part of TREC topics which usually consists of about 8-15 terms. As a second set of even longer queries we use both the *description and narrative* of the TREC topics[8]. Some characteristics of the TREC collections and queries are outlined in Table 6. We preprocessed documents and queries by removing standard stopwords[9] and stemmed using Porter's algorithm.

7.2 Tuning Baselines

The aim of the experiments is to evaluate the effectiveness of the DQM approach for long natural language queries for all of the language models outlined in this paper (JM, Dir, Dir+, and SPUD). We include four reference baseline approaches (MATF, BM25, BM25+, and the two-stage language model (TLM) [25]).

As we are evaluating unsupervised retrieval methods, we choose to simulate a more realistic retrieval setting. In particular, we assume that we have no relevance judgements available[10] to tune on prior to deploying a retrieval method on a particular collection. Therefore, we tune all retrieval methods on the WT2g collection using all verbose queries (both *description only* and *description and narrative* queries combined into one set) and apply the tuned settings to the remaining collections and queries. Specifically we tuned the one free-parameter in JM and SPUD (and their DQM counterparts) by conducting a parameter sweep over the range $(0.0 - 1.0)$ in increments of 0.05. We tuned the one free-parameter in Dir and Dir+ (and their DQM counterparts) similarly by sweeping over the range $(500 - 8000)$ in increments of 500. We tuned BM25 and BM25+ using a grid search for k_1 (in the range $(0.5 - 4.0)$ in steps of 0.5) and b (in the range $(0.1 - 1.0)$ in steps of 0.1). For both Dir+ and BM25+, we set δ to its default value (0.05 and 1.0 respectively). For the two-stage language model (TLM), we performed a grid search for μ (in the range $500 - 8000$ in increments of 500) and λ (in the range $0.1 - 1.0$ in steps of 0.05). More effort was put into tuning BM25 (BM25+) and TLM as they have multiple parameters. Table 7 shows the optimal hyper-parameter settings for natural language queries on the WT2g tuning collection. All settings are consistent with previous research using these approaches.

The current reported *state-of-the-art* method for verbose queries used the Dir language model incorporated with a feedback step which re-weights salient aspects of the query [17]. This method (labelled TA) with a default parameter setting of $c = 10$ is used with a tuned Dir language model.

Figure 2 shows the performance of the four language models during tuning on the WT2g collection. We note that the DQM versions of the language models are never lower than the original versions and are substantially higher for the longer queries, with DQM$_q$ being the best query model for the four language models on the tuning collection. This suggests that the DQM can provide useful weighting for queries and improves performance without introducing any new hyper-parameters to the model (i.e. the hyper-parameters are shared).

[6] As the length of the topics are much smaller than documents, we actually fixed μ_q to $\mu/10$ for Dir and Dir+.

[7] Adding extra hyper-parameters to a model will often improve performance simply because the model is more flexible. For unsupervised ranking, parameter tuning should be kept to a minimum, as it is the model and its parameter estimates that should explain the data, and not the hyper-parameters.

[8] Some studies have used all three fields in their queries. However, we only include aspects of the topics that are written in natural language (i.e. fully formed sentences) as we wish to study retrieval scenarios where no *keywords* have already been manually annotated. We have noted in preliminary experiments that including the title field artificially boosts overall performance by boosting the query signal.

[9] www.lextek.com/manuals/onix/stopwords1.html

[10] This simulates the first running of a new track in TREC where we lack relevance assessments for a new collection.

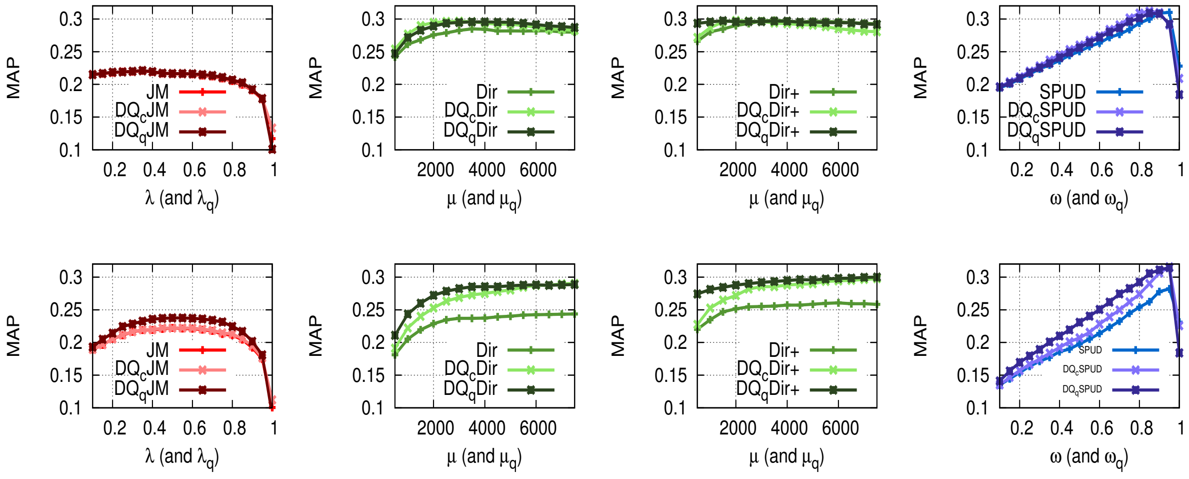


Figure 2: Performance trend (MAP) as the single smoothing hyper-parameter (λ, μ, and ω) changes for each language model on the WT2g tuning collection for *description only* queries (top) and for *description and narrative* queries (bottom).

Table 7: Optimal hyper-parameter on all retrieval methods over both types of verbose queries tuned for MAP on WT2g.

	WT2g (Web) Tuning Collection		
Retrieval	verbose queries		
Methods	MAP	NDCG	Optimal value
MATF	0.259	0.458	No hyper-parameters
BM25	0.248	0.461	$k_1 = 1.5, b = 0.6$
BM25+	0.258	0.466	$k1 = 3.0, b = 0.7$
TLM	0.277	0.460	$\mu = 2500, \lambda = 0.6$
JM	0.219	0.391	$\lambda = 0.5$
DQM$_c$JM	0.219	0.392	$\lambda = 0.5$
DQM$_q$JM	0.227	0.405	$\lambda = 0.5$
Dir	0.260	0.427	$\mu = 4000$
DQM$_c$Dir	0.284	0.465	$\mu = 4000$
DQM$_q$Dir	0.290	0.470	$\mu = 4000$
Dir+	0.277	0.442	$\mu = 4000$
DQM$_c$Dir+	0.290	0.471	$\mu = 4000$
DQM$_q$Dir+	0.295	0.478	$\mu = 4000$
SPUD	0.293	0.487	$\omega = 0.9$
DQM$_c$SPUD	0.300	0.490	$\omega = 0.85$
DQM$_q$SPUD	0.307	0.497	$\omega = 0.85$

7.3 Results

Table 8 shows the results of all of the single-pass retrieval methods on three collections. The *description and narrative* queries consistently outperform the *description only* queries. The baseline approaches (MATF, BM25, BM25+ and TLM) are all comparable with no method outperforming the others over all collections. Consistent with other studies, the language model with JM smoothing is the worst performing language model. Dir+ outperforms Dir in terms of MAP but both are significantly outperformed by the SPUD language model on each collection.

For each of the language models (JM, Dir, Dir+, and SPUD), the discriminative query model (DQM) improves performance. The best model is DQM$_q$ which uses the 500 topics as a background query model. The use of this background query model leads to a significant improvement on most of the collections across all of the language models. The increased improvement is quite large for the *description and narrative* queries. The two-stage language model (TLM) which uses two types of smoothing only slightly outperforms the Dir method, but is significantly outperformed by SPUD, DQM$_c$SPUD, and DQM$_q$SPUD.

Table 9 shows the performance of a tuned Dir language model (tuned per collection) and the Dir method incorporated with the TA feedback re-weighting method (Dir-TA) [17]. Also shown is the performance of the SPUD language model ($\omega = 0.85$) with the standard RM3[11] method (with interpolation of 0.5, 30 expansion terms, and 10 feedback documents) and the DQM$_q$SPUD with the same RM3 method. We can see that the DQM$_q$SPUD-RM3 method significantly outperforms the Dir-TA method.

8. DISCUSSION AND CONCLUSION

This work has focused on long documents and queries, and has shown that only one recently developed language model (SPUD) adheres to both the verbosity and scope hypotheses. Interestingly we have shown that different types of smoothing (that in JM and Dir) lead to different types of length normalisation (i.e. verbosity and scope respectively). The formal constraints outlined in this paper have potential implications regarding both *adversarial* and *session* search respectively. The ability to maliciously promote the ranking of a document by simple document concatenation raises issues regarding the trust and effectiveness of certain popular retrieval methods (e.g. Dir and BM25 among others). The SPUD method overcomes some of these issues by means of more principled normalisation. We argue that retrieval

[11] During the RM3 feedback-step, we set the document smoothing parameter to zero, essentially ranking terms as a query-likelihood weighted summation of term maximum-likelihood estimates. Tuning parameters chosen are those suggested in previous work [11].

Table 8: MAP and NDCG@10 of retrieval methods on Robust-04, WT10g, and Gov2 collections where the superscript $^{\blacktriangle}$ means significantly higher using a paired t-test ($p < 0.05$) compared to the standard maximum likelihood query model (e.g. DQM$_c$JM vs JM, and DQM$_q$JM vs JM). † means significantly higher than TLM, and ‡ means significant higher compared to BM25 and TLM. Best result in bold.

	Robust-04 (News)				WT10g (Web)				Gov2 (Web)			
Retrieval	desc		desc+narr		desc		desc+narr		desc		desc+narr	
Methods	MAP	NDCG	MAP	NDCG	MAP	NDCG	MAP	NDCG	MAP	NDCG	MAP	NDCG
MATF	0.251	0.472	0.242	0.461	0.181	0.370	0.190	0.373	0.265	0.562	0.248	0.569
BM25	0.241	0.469	0.239	0.470	0.179	0.396	0.196	0.409	0.258	0.545	0.268	0.584
BM25+	0.245	0.464	0.235	0.458	0.182	0.393	0.197	0.399	0.260	0.537	0.268	0.579
TLM	0.245	0.442	0.245	0.454	0.199	0.381	0.201	0.372	0.252	0.513	0.264	0.549
JM	0.234	0.424	0.239	0.448	0.143	0.302	0.165	0.322	0.192	0.385	0.253	0.531
DQM$_c$JM	0.234	0.424	$0.240^{\blacktriangle}$	0.449	0.144	0.302	$0.166^{\blacktriangle}$	0.322	0.192	0.385	$0.254^{\blacktriangle}$	0.532
DQM$_q$JM	$0.237^{\blacktriangle}$	$0.429^{\blacktriangle}$	$0.261^{\blacktriangle}_{\ddagger}$	$0.482^{\blacktriangle}_{\dagger}$	$0.152^{\blacktriangle}$	$0.313^{\blacktriangle}$	$0.174^{\blacktriangle}$	$0.340^{\blacktriangle}$	0.193	0.386	$0.269^{\blacktriangle}$	$0.563^{\blacktriangle}$
Dir	0.245	0.444	0.248	0.454	0.185	0.368	0.194	0.356	0.238	0.512	0.252	0.534
DQM$_c$Dir	0.240	0.427	$0.254^{\blacktriangle}_{\ddagger}$	0.461	$0.211^{\blacktriangle}$	0.377	$0.216^{\blacktriangle}_{\ddagger}$	$0.380^{\blacktriangle}$	$0.250^{\blacktriangle}$	$0.526^{\blacktriangle}$	$0.275^{\blacktriangle}_{\dagger}$	$0.565^{\blacktriangle}$
DQM$_q$Dir	$0.249^{\blacktriangle}$	0.446	$0.266^{\blacktriangle}_{\ddagger}$	$0.486^{\blacktriangle}_{\dagger}$	$0.202^{\blacktriangle}$	$0.379^{\blacktriangle}$	$0.220^{\blacktriangle}_{\ddagger}$	$0.394^{\blacktriangle}_{\dagger}$	$0.251^{\blacktriangle}$	$0.528^{\blacktriangle}$	$0.289^{\blacktriangle}_{\ddagger}$	$0.606^{\blacktriangle}_{\dagger}$
Dir+	0.248	0.439	$0.257_{\ddagger}$	0.456	0.192	0.365	0.200	0.350	0.240	0.507	0.254	0.532
DQM$_c$Dir+	0.238	0.420	$0.261_{\ddagger}$	0.459	$0.212^{\blacktriangle}$	0.368	$0.220^{\blacktriangle}_{\ddagger}$	$0.370^{\blacktriangle}$	$0.247^{\blacktriangle}$	$0.522^{\blacktriangle}$	$0.273^{\blacktriangle}_{\dagger}$	$0.562^{\blacktriangle}$
DQM$_q$Dir+	0.249	0.442	$0.273^{\blacktriangle}_{\ddagger}$	$0.487^{\blacktriangle}_{\dagger}$	$0.204^{\blacktriangle}$	0.372	$0.223^{\blacktriangle}_{\ddagger}$	$0.382^{\blacktriangle}$	0.249	0.524	$0.288^{\blacktriangle}_{\ddagger}$	$0.603^{\blacktriangle}_{\dagger}$
SPUD	$0.262_{\ddagger}$	$0.479_{\dagger}$	$0.266_{\ddagger}$	$0.486_{\dagger}$	0.202	0.385	$0.208_{\dagger}$	$0.385_{\dagger}$	$0.275_{\ddagger}$	$0.554_{\dagger}$	$0.287_{\ddagger}$	$0.589_{\dagger}$
DQM$_c$SPUD	$0.263_{\ddagger}$	$0.473_{\dagger}$	$0.270^{\blacktriangle}_{\ddagger}$	$0.491_{\dagger}$	$0.218^{\blacktriangle}_{\ddagger}$	0.390	$0.224^{\blacktriangle}_{\ddagger}$	$0.403^{\blacktriangle}_{\dagger}$	$0.287^{\blacktriangle}_{\ddagger}$	$0.559_{\dagger}$	$0.304^{\blacktriangle}_{\ddagger}$	$0.613^{\blacktriangle}_{\ddagger}$
DQM$_q$SPUD	$\mathbf{0.270}^{\blacktriangle}_{\ddagger}$	$\mathbf{0.483}_{\dagger}$	$\mathbf{0.288}^{\blacktriangle}_{\ddagger}$	$\mathbf{0.532}^{\blacktriangle}_{\ddagger}$	$\mathbf{0.218}^{\blacktriangle}_{\ddagger}$	$\mathbf{0.394}^{\blacktriangle}$	$\mathbf{0.233}^{\blacktriangle}_{\ddagger}$	$\mathbf{0.423}^{\blacktriangle}_{\dagger}$	$\mathbf{0.289}^{\blacktriangle}_{\ddagger}$	$\mathbf{0.560}^{\blacktriangle}_{\dagger}$	$\mathbf{0.327}^{\blacktriangle}_{\ddagger}$	$\mathbf{0.647}^{\blacktriangle}_{\ddagger}$

Table 9: MAP and NDCG@10 of feedback methods on Robust-04, WT10g, and Gov2 collections where the superscript $^{\blacktriangle}$ means significantly higher using a paired t-test ($p < 0.05$) compared to the Dir-TA method.

		Robust-04 (News)				WT10g (Web)				Gov2 (Web)			
		desc		desc+narr		desc		desc+narr		desc		desc+narr	
Retrieval Method	Feedback Method	MAP	NDCG	MAP	NDCG	MAP	NDCG	MAP	NDCG	MAP	NDCG	MAP	NDCG
Dir	None	0.251	0.456	0.255	0.467	0.185	0.368	0.194	0.364	0.253	0.517	0.262	0.566
Dir	TA	0.262	0.478	0.266	0.488	0.216	0.387	0.208	0.365	0.273	0.513	0.286	0.573
Dir	RM3	0.268	0.445	0.257	0.432	0.203	0.362	0.195	0.342	0.250	0.495	0.246	0.519
SPUD	RM3	$0.295^{\blacktriangle}$	$0.494^{\blacktriangle}$	$0.296^{\blacktriangle}$	0.487	$0.223^{\blacktriangle}$	0.383	0.222	0.384	$0.303^{\blacktriangle}$	$0.540^{\blacktriangle}$	0.300	0.591
DQM$_q$SPUD	RM3	$0.299^{\blacktriangle}$	$0.499^{\blacktriangle}$	$0.314^{\blacktriangle}$	$0.530^{\blacktriangle}$	$0.232^{\blacktriangle}$	0.390	$0.240^{\blacktriangle}$	$0.415^{\blacktriangle}$	$0.312^{\blacktriangle}$	$0.552^{\blacktriangle}$	$0.337^{\blacktriangle}$	$0.641^{\blacktriangle}$

methods that do not adhere to the verbosity hypothesis are open to certain types of manipulation.

Furthermore, when multiple queries (often of various length) are issued for the same information need, as is the case for session search [10], the interaction between document and query scope becomes increasingly important. While SPUD and Dir contain scope normalisation, the hyper-parameters of other retrieval methods (e.g. BM25 and JM) need to be tuned for queries of different length, making them less robust and theoretically deficient. Future work will look at evaluating the best retrieval methods presented in this work specifically on the tasked developed for the session track in TREC [10].

While other supervised retrieval methods have been developed specifically for verbose queries [1, 2] our methods are unsupervised, and achieve an effectiveness that outperform those reported in previous studies on the same collections. All of the methods and software used for this paper were developed with Lucene and are available for download[12]. Some approaches have aimed to estimate the performance of reformulate queries using query performance predictors [7, 4] and future work could look at incorporating some of these techniques into the expansion stage.

Finally, although the SPUD model uses the number of word types in a document as a measure of scope, there are other possible measures that also adhere to the constraints SC3 and SC4 introduced here. The *perplexity* [14] of the document model is one such measure that has been proposed. Perplexity is related to the information content (entropy) of the maximum likelihood model document model. Other measures of scope previously outlined [14] do not adhere to these SC3 and SC4 and so were ignored for those reasons. Therefore, it would be interesting future work to investigate perplexity.

9. REFERENCES

[1] Michael Bendersky and W. Bruce Croft. Discovering key concepts in verbose queries. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pages 491–498, New York, NY, USA, 2008. ACM.

[2] Michael Bendersky, Donald Metzler, and W. Bruce Croft. Parameterized concept weighting in verbose

[12] https://github.com/ronancummins/www2016

queries. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 605–614, New York, NY, USA, 2011. ACM.

[3] David Bodoff. Fuhr's challenge: conceptual research, or bust. In *ACM SIGIR Forum*, volume 47, pages 3–16. ACM, 2013.

[4] Ronan Cummins, Mounia Lalmas, Colm O'Riordan, and Joemon M Jose. Navigating the user query space. In *String Processing and Information Retrieval*, pages 380–385. Springer, 2011.

[5] Ronan Cummins and Colm O'Riordan. A constraint to automatically regulate document-length normalisation. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 2443–2446. ACM, 2012.

[6] Ronan Cummins, Jiaul H. Paik, and Yuanhua Lv. A Pólya urn document language model for improved information retrieval. *ACM Transactions of Informations Systems*, 33(4):21, 2015.

[7] Van Dang, Michael Bendersky, and W. Bruce Croft. Learning to rank query reformulations. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, pages 807–808, Geneva, Switzerland, 2010. ACM.

[8] Stephen P. Harter. A probabilistic approach to automatic keyword indexing. Part I. On the Distribution of Specialty Words in a Technical Literature. *Journal of the American Society for Information Science*, 26(4):197–206, 1975.

[9] Ben He and Iadh Ounis. On setting the hyper-parameters of term frequency normalization for information retrieval. *ACM Transactions on Information Systems (TOIS)*, 25(3), 2007.

[10] Evangelos Kanoulas, Ben Carterette, Mark Hall, Paul Clough, and Mark Sanderson. Overview of the trec 2011 session track. In *Proceedings of Text Retrieval Conference TREC 2011*, 2011.

[11] Yuanhua Lv and ChengXiang Zhai. A comparative study of methods for estimating query language models with pseudo feedback. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 1895–1898, New York, NY, USA, 2009. ACM.

[12] Yuanhua Lv and ChengXiang Zhai. Lower-bounding term frequency normalization. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, CIKM '11, pages 7–16. ACM, 2011.

[13] Yuanhua Lv and ChengXiang Zhai. When documents are very long, bm25 fails! In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 1103–1104. ACM, 2011.

[14] Seung-Hoon Na. Two-stage document length normalization for information retrieval. *ACM Transactions of Information Systems*, 33(2):8:1–8:40, 2015.

[15] Seung-Hoon Na, In-Su Kang, and Jong-Hyeok Lee. Improving term frequency normalization for multi-topical documents and application to language modeling approaches. In *Advances in Information Retrieval*, pages 382–393. Springer, 2008.

[16] Jiaul H. Paik. A novel tf-idf weighting scheme for effective ranking. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 343–352, New York, NY, USA, 2013. ACM.

[17] Jiaul H. Paik and Douglas W. Oard. A fixed-point method for weighting terms in verbose informational queries. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014, Shanghai, China, November 3-7, 2014*, pages 131–140, 2014.

[18] S. E. Robertson and S. Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '94, pages 232–241, New York, NY, USA, 1994. Springer-Verlag New York, Inc.

[19] S.E. Robertson, S. Walker, S. Jones, M.M. Hancock-Beaulieu, and M. Gatford. Okapi at trec-3. In *Proceedings of the Third Text REtrieval Conference (TREC 1994)*, pages 109–126, 1994.

[20] Stephen Robertson and Hugo Zaragoza. *The probabilistic relevance framework: BM25 and beyond.* Now Publishers Inc, 2009.

[21] Amit Singhal, Chris Buckley, and Mandar Mitra. Pivoted document length normalization. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '96, pages 21–29, New York, NY, USA, 1996. ACM.

[22] Peter D Turney, Patrick Pantel, et al. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37(1):141–188, 2010.

[23] Chengxiang Zhai and John Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *Proceedings of the Tenth International Conference on Information and Knowledge Management*, CIKM '01, pages 403–410, New York, NY, USA, 2001. ACM.

[24] ChengXiang Zhai and John Lafferty. Two-stage language models for information retrieval. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '02, pages 49–56, New York, NY, USA, 2002. ACM.

[25] Chengxiang Zhai and John Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Transactions of Information Systems*, 22:179–214, April 2004.

[26] Xiaofeng Zhou, Jimmy Xiangji Huang, and Ben He. Enhancing ad-hoc relevance weighting using probability density estimation. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 175–184, New York, NY, USA, 2011. ACM.