Building Simulated Queries for Known-Item Topics

An Analysis using Six European Languages

Leif Azzopardi
Dept. of Computing Science
University of Glasgow,
Glasgow G12 8QQ
leif@dcs.gla.ac.uk

Maarten de Rijke
ISLA, University of Amsterdam
Kruislaan 403, 1098 SJ
Amsterdam
mdr@science.uva.nl

Krisztian Balog
ISLA, University of Amsterdam
Kruislaan 403, 1098 SJ
Amsterdam
kbalog@science.uva.nl

ABSTRACT

There has been increased interest in the use of simulated queries for evaluation and estimation purposes in Information Retrieval. However, there are still many unaddressed issues regarding their usage and impact on evaluation because their quality, in terms of retrieval performance, is unlike real queries. In this paper, we focus on methods for building simulated known-item topics and explore their quality against real known-item topics. Using existing generation models as our starting point, we explore factors which may influence the generation of the known-item topic. Informed by this detailed analysis (on six European languages) we propose a model with improved document and term selection properties, showing that simulated known-item topics can be generated that are comparable to real known-item topics. This is a significant step towards validating the potential usefulness of simulated queries: for evaluation purposes, and because building models of querying behavior provides a deeper insight into the querying process so that better retrieval mechanisms can be developed to support the user.

Categories and Subject Descriptors

H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*performance evaluation*

General Terms

Experimentation

Keywords

Query simulation, query generations, evaluation, multilingual retrieval

1. INTRODUCTION

Evaluation plays a central role in Information Retrieval because it enables the benchmarking and comparison of different retrieval techniques [17]. However, manually building test collections for evaluation is a time consuming and expensive process. With more and more collections becoming available and only finite resources available, the range of tasks that can be evaluated is restricted. A cost-effective alternative to manually building test collections is to construct simulated (or artificial) test collections instead [1, 11, 15, 16]. While this involves a number of compromises regarding the realism of the generated test collection, the solution has many benefits.

Simulation provides an inexpensive avenue for testing, training, and evaluating retrieval algorithms along with the ability to precisely control the experimental conditions. For instance, the length (i.e., long vs. short), style (highly discriminative terms vs. popular terms), quality (noisy query terms, translations, etc) and number of queries that can be produced for a given topic can be greatly varied to define a specific scenario. This enables selective evaluation of particular query types. By considering different query types the relationship between query characteristics and algorithm performance can be better understood and help guide the development of retrieval algorithms and query performance prediction [12].

Recently, there has been number of studies which have constructed a simulated test collection for evaluation and training purposes [1, 11, 12]. However, little research has been performed on investigating their validity, utility and limitations for evaluation. Consequently, there are many open questions and issues concerning their use. What is the information need? How should topics be generated? Can the models of query generation produce reasonable or interpretable queries? Are these queries really like user queries? Do they give the same indication of performance, or the same ranking of systems? And what about relevance judgments? So while using simulated test collections is cost-effective and potentially useful in the context of evaluation, they are not well understood [10].

The focus of this study is on one particular type of simulated test collections built from the automatic generation of known-item topics [1, 11]. The approach generates known-item topics by selecting a document, the known-item, and producing a query for that known item. Since the task of known-item finding has a clear and precise semantics (i.e., find the known item), this removes issues relating to acquiring user judgements or defining an information need (as it is implicit in the task). The main concern for these test collections, is the production of the known-item topics and whether they are representative or reflective of actual known items. The challenge is to develop models that produce

SIGIR'07, July 23–27, 2007, Amsterdam, The Netherlands.
Copyright 2007 ACM 978-1-59593-597-7/07/0007 ...$5.00.

known-item topics that are comparable to manually created known-item topics.

Previous studies using simulated topics have resulted in varying levels of performance. A very early study found that simulated queries for ad hoc retrieval performed very poorly compared to real queries [15, 16]. However, more recent studies have shown mixed performances from simulated queries. Azzopardi and de Rijke [1] report that performance for known-item finding queries is reasonably similiar to real queries, but either somewhat lower or somewhat higher. During WebCLEF 2006 simulated queries were also used to generate queries for the known-item task in over 20 languages [2]. Simulated topics resulted in substantially poorer performance than manual topics for many of the languages. As a result, only a weak correlation between the ranking of systems using simulated and real queries was found [2]. In sum, current models for generating simulated topics do not appear to be providing comparable performance to manual topics. The cause of this problem, we believe, is that the models used to generate the queries and topics are not a good representation of the actual querying processes.

In this paper, we examine the problems of developing useful simulated queries for known-item finding, and attempt to identify generation methods that produce topics that are comparable to real topics. On six different European language collections, we test current models of known-item generation. Our main contribution consists of two parts: (1) a detailed analysis of a number of factors that impact the performance of the generated queries; and (2) extensions and refinements of current query models in terms of term selection and non-uniform document priors. This results in query models whose replicative validity can be established.

The remainder of this paper is structured as follows. In the next section, we briefly review the related work in this area before describing a generative probabilistic approach to query generation for known-item finding tasks. Then, in Section 4 we perform an empirical analysis on the WebCLEF multilingual web retrieval tracks using different query models. We analyse and revise the models in Section 5 and 6. Finally, we discuss our findings and conclude in Section 7.

2. RELATED WORK

Simulated queries have been used in the past for a variety of applications, including parameter estimation, sampling, and for simulated topics.

In [15] we find an early attempt to not only generate topics, but also the collection, based on the distributions of real collections and queries. A model for generation is proposed to construct documents and topics—queries and relevance judgments for ad hoc retrieval. To determine whether their simulated model was comparable to the real systems, they tried to validate the model. According to Zeigler [18], there are three kinds of validation that can be performed on a simulation; predictive, structural and replicative. A model has *predictive* validity if it can produce the same data output as the real system (i.e., comparing the query terms for a given known item from the simulated model and real system). A model has *structural* validity if the way it operates is a reflection of how the real system operates. And *replicative* validity is if the model produces output that is similar to the output of the real system. Tague et al. [15] focus on replicative validity by comparing the performance of the simulated queries to the performance of the real queries and seeing if they were drawn from the same distributions. However, their simulated collection and topics resulted in very poor performance and was not comparable to real topic. Consequently, they were unable to produce a model with replicative validity. In this paper, we adopt their methodology for testing whether the performance of the topics of the simulated is comparable to the performance of real queries; see Section 4.

Berger and Lafferty [4] proposed a sophisticated language model (the translation model) which required significant training to set the model parameters effectively. Query and document pairs were required as training data, where queries were created to obtain these pairs. Queries were formed by taking the title of the document to be one example of a query for that document. These were then used for the estimation process.

Callan and Connell [7] proposed Query Based Sampling which used random queries to probe collections. Queries consisted of a single term, that has been drawn from a sample population of documents according to different selection strategies, such as: term frequency, document frequency, inverse document frequency, or average term frequency. The documents returned in response to the query were used to form a representation of the collection. It was believed that random queries, while unrealistic, would lead to an unbiased representation of the resource. Thus, there was no requirement for the queries to be realistic.

In [11], a synthetic test collection for annotation retrieval is created by generating queries for a particular document by randomly sampling terms without replacement. Then, a corresponding annotation is generated by again randomly sampling terms without replacement from the document. The idea was to create query-annotation pairs where the queries did not match the annotation so that they could evaluate this context.

Jordan et al. [12] generated simulated queries which they refer to as “controlled” queries, which were used to estimate parameters in psuedo-relevance feedback. They generated queries from a set of pseudo-relevant documents using a number of different selection strategies: (1) single term highly discriminative queries, (2) two term query queries composed of the highest discriminative term and the other term was selected at random, and (3) varied length queries, where terms were selected which had the highest relative entropy (i.e., the highest term discrimination). The relative entropy was computed with respect to the previous terms (i.e., term dependence). This process is analogous to query expansion, where extra query terms are selected to add to the query; for instance, Cai et al. [5] used the most discriminative terms from top ranked documents.

Azzopardi and de Rijke [1] focus on generating queries for known-item finding. Here, the querying behavior of users for a known item is formalized within a generative probabilistic framework. They generated known-item topics on the TREC Enterprise collection and showed that generated queries which were more discriminative performed better than queries that were randomly selected or selected proportional to term frequency. However, in absolute numbers the performance of the generated queries was less than the performance of manual queries. A similar result was found at WebCLEF 2006 [2]. This disparity has prompted this study into the reasons why generated queries were not as successful, and how to improve the current models of query generation. We adopt the framework set out in [1] for studying query generation because it enables many different models to be devised, of varying sophistication, while still being intuitive and simple. We describe the generative process in more detail in the following section.

It has been acknowledged in the literature that the simulated queries produced are far from realistic. However, since the cost of building test collections is very high, this motivates addressing this problem, because one of the main advantages of simulated queries is that numerous queries can be produced at minimal cost along with the ability to generate a multitude of different querying behaviors [1, 12]. This has huge ramifications for evaluating systems because it provides finer grained control over the experimental conditions that would be otherwise impossible in a real environment.

Further, by focusing on building models of query generation we can develop a better understanding of how users query a system, where we can then develop mechanisms which help to address the users' information retrieval tasks. Such query models are also useful for identifying structure and context within the query [6], estimating the difficultly of the query for performance prediction [8], automatically completing queries by predicting subsequent terms [3], and automatic query expansion[5].

3. SIMULATED KNOWN-ITEM QUERIES

In [1], a probabilistic framework is presented for the generation of simulated queries for the task of known-item search. This approach models the following behavior of a known-item searcher. It is assumed that the user wants to retrieve a particular document that they have seen before in the collection, because some need has arisen calling for this document. The user then tries to re-construct or recall terms, phrases and features that would help identify this document, which they pose as a query.

The basic algorithm for generating known-item queries is based on an abstraction of the actual querying process, where the following steps are undertaken:

- Initialize an empty query $q = \{\}$
- Select the document d_k to be the known-item with probability $p(d_k)$
- Select the query length s with probability $p(s)$
- Repeat s times:
    - Select a term t_i from the document model of d_k with probability $p(t_i|\theta_d)$
    - Add t_i to the query q.
- Record d_k and q to define the known-item/query pair.

Since the known item is the relevant document there is no need for any explicit relevance judgments. The benefits are immediately obvious. By repeatedly performing this algorithm numerous queries can be generated quickly and inexpensively. But before this can be performed, the probability distributions $p(d_k)$, $p(s)$, and $p(t|\theta_{d_k})$ need to be defined. These distributions are an important part of the generative process as they characterize the behavior of the user which we are trying to simulate. By using different probability distributions the various types and styles of queries can be generated. The distribution with the biggest influence appears to be the definition of the user's language model of the document $p(t_i|\theta_{d_k})$, from which the query terms are sampled.

Formally, the user's querying model $p(\cdot|\theta_m^{d_k})$ can be expressed as shown in Eq. 1, where m is the model of the user querying behavior for the document d_k. The process is a mixture between sampling from the document and sampling from the collection (or noise):

$$p(t_i|\theta_m^{d_k}) = (1 - \lambda) \cdot p(t_i|d_k) + \lambda \cdot p(t_i) \quad (1)$$

Given this user querying model, the quality of the query generated can be directly influenced by varying the λ parameter. As λ tends to zero, the user's recollection of the original document improves. Conversely, as λ tends to one, the user's memory of the document degrades. If $\lambda = 1$, then the user knows the document exists but they have no idea as to which terms appear in the document (and randomly selects query terms).

To provide different types of user querying behaviors, then, it is important to define the probability distribution defining the likelihood of selecting the term t_i from the document d_k, i.e., $p(t_i|d_k)$. The three broad sampling strategies that have been used to characterize this selection process are: (1) popular, (2) random, and (3) discriminative [1, 7, 12].

Popular selection considers the use of information such as term frequency, location, etc. which makes the term stand out in some way to the user so that the user recalls this feature. We capture popularity by assuming that more frequent terms are more likely to be used as query terms and use the empirical probability of a term in the document to represent this selection strategy:

$$p(t_i|d_k) = \frac{n(t_i, d_k)}{\sum_{t_{i'}} n(t_{i'}, d_k)}, \quad (2)$$

where $n(t_i, d_k)$ is the number of occurrences of t_i in d_k.

Random selection makes the assumption that the user will indiscriminately recall terms in the document. Admittedly, this does not seem like a very likely strategy that a user may have when issuing a query, but it provides a baseline to compare other selection strategies, as this is the most naive. The probability distribution for the random selection strategy is set as:

$$p(t_i|d_k) = \frac{b(t_i, d_k)}{\sum_{t_j \in d_k} b(t_j, d_k)}, \quad (3)$$

where $b(t_i, d_k)$ denotes the presence of a term in a document, where $b(t_i, d_k) = 1$ if t_i occurs in d_k, and zero otherwise.

Discriminative selection assumes that the user will consider information outside the document, and that they will consider the document with respect to the collection. That is, the user may try to select query terms that will discriminate the document they want from the other documents in the collection. Selection of this kind is generally based around the informativeness of a term; the probability distribution we define to represent this strategy is:

$$p(t_i|d_k) = \frac{b(t_i, d_k)}{p(t_i) \cdot \sum_{t_j \in d_k} \frac{b(t_j, d_k)}{p(t_j)}}, \quad (4)$$

where $p(t_j)$ is the probability of a term occurring in the collection defined by the maximum likelihood estimate (i.e. $p(t_i|d_k)$ is proportional to the inverse collection frequency of a term).

Table 1: Basic statistics of manual queries.

	Number of			Query length		
Lang.	Docs.	Terms	Qrys.	Min.	Max.	Avg.
ES	33,772	1,399,309	143	1	13	6.01
UK	66,345	1,768,353	68	2	12	5.14
NL	148,040	887,819	83	1	9	3.49
PT	146,563	576,126	56	2	17	5.98
HU	324,961	537,220	44	1	9	3.50
DE	438,481	873,631	70	1	7	3.12

4. EXPERIMENTAL SET-UP

How well do the query generation models introduced in Section 3 work? More specifically, can we establish replicative validity for them? Our goal is to better understand the process of query generation through an extensive study of factors influencing the quality of the known-item topics generated. To this end we generate known-item topics using multiple document collections, in six European languages. After describing the collections, we describe the manual topics used for comparison, the test method used for the comparison, and the results. In Section 5 we follow with a detailed analysis of the results.

Data. The document collections we used are from the EuroGOV corpus [13], a multilingual web corpus built from a crawl of European government-related sites, which contains over 3.5 million pages from 27 primary domains, covering over twenty languages; there is no single language that dominates the corpus.

From this corpus, we selected only those languages which had a sufficiently large number (over 40) known-item topics for that domain and language. This was to ensure that we had enough examples to be able to observe statistically significant differences. This restriction resulted in using the following languages from the domains: German (DE), Spanish (ES), Hungarian (HU), Dutch (NL), Portuguese (PT), and English (UK). Each domain from the EuroGOV collection was individually indexed, using language-specific stopword lists, but we did not apply stemming. Table 1 shows the collection statistics, along with the number of manual known-item topics (Qrys.), and the average, minimum, and maximum query lengths with stopwords removed.

Known-Item Topics. We used the (manually created) known-item queries for WebCLEF 2005 and 2006 [2, 14] as a reference point. We produced a set of simulated queries for each of the six languages (DE, ES, HU, NL, PT, UK) from three user query models using: (Model 1) popular sampling (i.e., Eq. 2); (Model 2) random sampling (i.e., Eq. 3); and (Model 3) discriminative sampling (i.e., Eq. 4). The amount of noise on all models was $\lambda = 0.2$, which reflects the amount of noise on average within the manual queries. The length of the queries generated where drawn from a poisson distribution with the mean set according to the average length of a query (rounded to the nearest whole number) given the particular language. In all, 100 known-item query-document pairs were generated for each model and each language.

Validation of Query Models. In order to establish replicative validity of a query model we need to determine whether the generated queries from the model are representative of the corresponding manual queries. Tague and Nelson [16] validated whether the performance of their generated queries was similar to real queries across the points of the precision-recall graph using the Kolmogorov-Smirnov (KS) Test. Here, we compare the Mean Reciporal Rank (MRR) of each model against the MRR of the manual queries using the KS Test as a way to validate the query models. The KS Test is an independent two-sample test which is used to test the hypothesis that the two samples may reasonably be assumed to come from the same distribution. So if the two distributions of MRRs are not significantly different, we can say that the query model produces known-item topics which are comparable to manual queries (in terms of performance). Otherwise, if they are significantly different, the query models produce known-item topics which are not comparable, resulting in performance which is significantly lower or higher.

Results. Table 2 presents the performance of the generated queries and the manual queries on three popular, but different, retrieval models: TF.IDF, OKAPI BM25, and a Language Model using Bayes Smoothing with a Dirichlet Prior set to 2000. An asterisk denotes whether the performance is comparable to the manual queries (i.e., *not* significantly different), otherwise the performance is not comparable (i.e., significantly different) according to a two tailed KS-Test where the significance level was set to 5%.

Table 2: Mean Reciprocal Rank: On manual and simulated queries.

Lang.	**Query Type**	**TF.IDF**	**OKAPI**	**LM**
ES	**Manual**	0.2217	0.3102	0.2482
	Model 1	**0.1805***	**0.2383***	**0.2195***
	Model 2	**0.1742***	**0.2733***	**0.2331***
	Model 3	0.3109	0.3675	0.3636
UK	**Manual**	0.3548	0.4836	0.4232
	Model 1	**0.3047***	**0.4741***	**0.4475***
	Model 2	0.1941	0.4440	0.3381
	Model 3	0.6359	0.6778	0.6711
NL	**Manual**	0.4158	0.6133	0.4842
	Model 1	0.1136	0.1763	0.1729
	Model 2	0.0885	0.1910	0.1458
	Model 3	0.2703	0.2940	0.3119
PT	**Manual**	0.0866	0.2161	0.1362
	Model 1	0.0763	0.1165	0.0951
	Model 2	0.0380	0.0620	0.042
	Model 3	**0.1540***	**0.2011***	**0.1868***
HU	**Manual**	0.2812	0.3683	0.2754
	Model 1	0.0148	0.0086	0.0064
	Model 2	0.0173	0.0346	0.0405
	Model 3	0.0338	0.0315	0.0321
DE	**Manual**	0.3231	0.5038	0.3588
	Model 1	0.0258	0.0464	0.0473
	Model 2	0.0402	0.0714	0.0484
	Model 3	0.0286	0.0274	0.0274

On inspection of the results, we see that Model 3 generates queries which obtain higher MRR scores than the other models. This is to be expected, as model 3 should generate queries with more discriminative terms. And, with respect to retrieval models, OKAPI consistently performed the best (or close to the best) regardless of query model or language.

The absolute scores for the simulated queries for ES, UK, PT, and to some extent NL, are all in the same general range as the absolute scores for the manual queries. However, there is a quite clear difference in performance between manual and simulated queries for DE and HU.

When we consider whether the performances of the simulated queries are comparable to the manual queries, we find that Model 1 and 2 for ES, Model 2 for UK, and Model 3 for PT are not significantly different (regardless of retrieval model). However, for all the other languages and models the performance is significantly different. And so, for NL, HU and DE none of the query models generated known-item topics that are comparable to manual topics. Put differently, so far we have not established replicative validity for all query models on all languages. We have established that specific models can achieve replicative validity on particular languages using particular query models.

5. ANALYSIS

In this section we provide a detailed analysis of the results obtained in Section 4. The most striking finding in Section 4 is that the performance of the simulated queries is generally quite lower than the manual queries (i.e., for NL, HU and DE, and for Model 1 and 2 for PT) when the performance is not comparable (the exceptions are model 3 on ES and UK, which result in considerably better performance). In most cases the MRR scores were extremely low and resulted from many of the simulated known-item topics ending in complete failure (i.e., not retrieving the known item within the top 1000 documents). Consequently, the proposed models do not always produce queries which perform like manual queries. Human users must be generating topics in a different manner such that they are more likely to be successful.

We have examined several related factors which appear to affect the quality of the queries which may influence the way in which users construct successful queries: the size of the collection, the size of the vocabulary, the document frequency of terms in the collection, the document frequency of query terms, and the importance of a document (as estimated by its inlink count). Note, while this is not an exhaustive list of factors, it represents a reasonable set to investigate initially. Below, we consider each in turn.

Collection Size. First, as the document collections increase in size the quality of the generated queries appears to decrease. Intuitively, it is more difficult to pin point the document in the larger collections (i.e., HU and DE). To test whether collection size has a large impact on the performance, we performed further experiments using 1/3rd and 1/6th of the DE and HU collections in order to reduce their size. While small improvements are obtained (See Table 3), they are still not comparable (i.e., simulated queries are still significantly different from manual ones). So we believe some other factor is likely to have a greater influence.

Table 3: Mean Reciprocal Rank: On simulated queries with reduced collection size.

Lang.	Query Type	TF.IDF	OKAPI	LM
DE 1/3	**Model 1**	0.0380	0.0654	0.0562
	Model 2	0.0527	0.0550	0.0626
	Model 3	0.1022	0.1039	0.1129
DE 1/6	**Model 1**	0.0401	0.0418	0.0427
	Model 2	0.0361	0.0683	0.0800
	Model 3	0.0967	0.1147	0.1299
HU 1/3	**Model 1**	0.0473	0.0624	0.0538
	Model 2	0.0502	0.0702	0.0552
	Model 3	0.0904	0.0915	0.0933
HU 1/6	**Model 1**	0.0324	0.0572	0.0594
	Model 2	0.0505	0.0504	0.0565
	Model 3	0.0599	0.0746	0.0754

Vocabulary Size. If we consider the size of the vocabulary, it is much smaller in HU and DE than in UK or ES. For example, the UK collection has about 1.7 million terms, while the DE collection has 0.8 million terms. Consequently, about 0.9 million more terms can be issued in English queries, which can be used to narrow down the search (and most of these occurred only in a handful of documents). Obviously, more index terms available for selection when submitting a query will enable the selection of terms that will be more likely to retrieve the document, however, in manual queries in HU, DE, and NL all perform relatively well. The impact of the size of the vocabulary does not seem to affect their performance as much. So instead of vocabulary size, we examine the document frequency of the terms in the collection and those in the queries.

DF of Terms in Collection. When we consider the discriminative power of terms (according to document frequency (DF)) in each collection, we notice that the HU and DE collections contained terms which occur in many more documents than the other languages. Table 4 shows the DF of the n-th term, ordered by DF. Submitting a term from the top 10,000 would result in over 870 and 1,635 documents being returned for DE and HU, respectively, whereas for ES and UK, the number of documents is smaller by almost an order of magnitude. Obviously, this is related to the number of documents in the collections, but suggests it is more difficult to select query terms that have sufficient discriminative power in these collections compared to others. Terms in the HU and DE collections appear in so many more documents that using any such term will return many documents (which will most probably lower the MRR). And so, terms from these languages are less discriminative than UK or ES terms, and taking into account the size of the vocabulary this means they also have fewer of them. Hence, the selection of query terms in such cases needs to be highly discriminative and popular, in order to identify the known item with high MRR.

Table 4: DF of Terms in the Collection. Number of documents given the term at different ranks, ordered by DF.

Lang.	100	1K	10K	50K
ES	4,909	2,249	147	32
UK	13,408	3,336	186	25
NL	27,288	5,466	350	20
PT	91,350	3,512	135	12
HU	92,975	15,563	1,635	28
DE	94,330	19,780	870	42

Table 5: Mean average DF and mean minimum DF of query terms.

Lang.	Stat.	Man.	Mod.1	Mod.2	Mod.3
ES	Avg.	945	1,011	686	413
	Min.	587	573	573	257
UK	Avg.	2,876	2,269	2,095	1,059
	Min.	1,294	1,240	1,141	171
NL	Avg.	5,021	8,747	9,812	2,657
	Min.	3,246	7,053	6,574	1,115
PT	Avg.	12,388	20,440	21,500	10,541
	Min.	6,807	17,637	17,537	6,358
HU	Avg.	14,139	46,151	35,910	35,052
	Min.	8,570	43,898	28,996	25,347
DE	Avg.	18,707	29,655	50,582	18,057
	Min.	4,243	18,003	33,134	13,621

DF of Terms in Queries. Next, we examined the document frequency (DF) of the terms in the queries (See Table 5). The DF of query terms from the generated queries were far higher than the manual queries, especially on the HU and DE collections. For example, for HU manual queries the average DF was 14,139, while for the three models of simulated queries the average DF was 46,151, 35,910 and 35,052, respectively. Consequently, the user querying models did not appear to sufficiently favor terms that were as discriminative as those in the manual queries. However, for ES and UK queries the difference in average DF was small, which attributes to their comparable performance against the manual queries on models 1 and 2. Conversely, the DF was somewhat lower for these queries on model 3, which is reflected in the higher performance.

Document Importance. Here, we consider whether the importance of a document has an impact on the generation of a known item. We estimate the importance of a document in terms of its inlinks. Table 6 shows the average number of inlinks for a page in the collection, manual topics, and the simulated topics.[1] The average number of inlinks in the collection is quite low, but the targets of manual known-item queries had substantially more inlinks. Documents of simulated known items created using the uniform document prior, had very low inlink counts that did not resemble the manual known items in this respect. Since manual topics are generated by selecting documents that tend to be linked more, it seems sensible that this prior on selection should be incorporated into the query generation process. However, it is not clear why this would provide any improvements in performance, as the retrieval models considered in this paper do not use any link information when scoring documents!

Other Factors. There are any number of other factors that could be influencing the quality and realism of the queries produced for a known item. For instance, the selection of bigrams, the selection of terms from structured fields, variations in selection strategies (i.e., querying is a mixture of different query models depending on the state of the user), length of a known item, etc. Another factor, specific to DE, HU and NL is the possible influence of morphological normalization on the effectiveness of the query generation process. For these languages de-compounding could have been employed—before or after the query sampling process. In the former case it could increase the size of the vocabulary, and it could do so in a way that drastically changes the statistics of terms [9]. However, whether this, or the other factors, would affect the replicative validity of the generation models, are left for future analysis and consideration.

Table 6: Inlink statistics for the collection, manual known items and simulated known items.

	Average number of inlinks in the:			
		Manual	Simulated topics	
Lang.	**Coll.**	**topics**	**w/o prior**	**w. prior**
DE	7.3	142.9	2.1	2925.6
ES	5.4	21.1	3.9	168.8
HU	15.6	9763.8	2.1	40490.2
NL	20.3	388.8	9.2	4273.2
PT[2]	0.001	0.169	0.0	0.0
UK[2]	0.002	0.010	0.0	0.015

[2] These collections do not have many links within the collection (which results in the very low values).

6. IMPROVING SIMULATED TOPIC PERFORMANCE

The analysis in the previous section suggests that two factors are quite different between simulated and manual topics: (1) the discrimination of query terms, and (2) the number of inlinks of known items. To determine whether theses two factors do have an impact on the quality of simulated topics, we perform a series of follow-up experiments where we use a different selection strategy and a non-uniform document prior. By applying these changes to the generation sequence we hope to improve the performance of the topics so that they are more realistic (in the sense that they share more of the characteristics of manual known items). Hopefully, this will result in query models whose replicative validity can be established.

Improving the Term Discrimination. To improve the discrimination of query terms selected we propose a combined strategy of popular and discrimination, where terms that are highly discriminative within the collection, and also very popular within the document are more likely to be selected. To represent this strategy we make the probability of a term being selected proportional to the tf/idf of the term in the document. The *Popular + Discrimination selection* selection strategy is defined as:

$$p(t_i|d_k) = \frac{n(t_i, d_k) \cdot \log \frac{N}{df(t_i)}}{\sum_{t_j \in d_k} \left(n(t_j, d_k) \cdot \log \frac{N}{df(t_j)} \right)}, \qquad (5)$$

where N is the number of documents in the collection and $df(\cdot)$ is the document frequency of a term.

For each of our six collections, we generated another 100 queries using the popular+discrimination selection strategy (Model 4). All other parameter values are held the same as in the initial experiments (Section 3). Table 7 reports the performance of the topics using Model 4, without inlink priors. Using the popular+discriminative selection strat-

[1] Inlink counts for simulated known-item topics are averaged over all topics generated for each collection.

Table 7: Mean Reciprocal Rank: On topics generated using Model 4, without inlink prior.

Lang.	Query Type	TF.IDF	OKAPI	LM
ES	Model 4	0.3425	0.4622	0.4267
UK	Model 4	0.5843	0.7013	0.6726
NL	Model 4	0.2531	0.3576	0.3314
PT	Model 4	**0.1478***	**0.2017***	**0.1713***
HU	Model 4	0.0623	0.0739	0.0661
DE	Model 4	0.0462	0.0535	0.0482

egy (Model 4) produces results similar to the discriminative selection strategy (Model 3) and this only results in producing comparable queries for the PT collection. Unfortunately, then, trying to improve the term selection strategy does not succeed in increasing the performance for the HU, DE and NL collections. As a consequence, we still do not have replicative validity for all languages.

While it appeared that the more discriminative term selection strategies (Models 3 and 4) would be more suitable for the latter languages (DE, HU and NL), because terms in these languages had a high DF, this did not necessarily translate in significantly better retrieval performance. We posit that users combat the DF problem by issuing specific combinations of terms, which result in a lower DF when combined. Consequently, independently selecting terms during the process may not sufficiently replicate this phenomenon. This suggests that in order to build query models whose outputs are more comparable to manual queries, the term dependencies between query terms need to be captured, i.e., term selection based on the conditional probability given the previous query terms (i.e., $p(t_t|t_{i-1}, \ldots, t_1, \theta_{d_k})$).

Encoding the Document Importance. From our analysis of inlink counts in the previous section, it seems reasonable to set a non-uniform document prior for the selection of the known item. This is to encode the querying behavior of users where we have seen that they are more likely to retrieve a known item that is important (defined by the number of inlinks). To examine this proposition we set the document prior $p(d_k)$ to be:

$$p(d_k) = \frac{in(d_k) + 1}{\sum_{d_i}(in(d_i) + 1)}, \qquad (6)$$

where $in(d_k)$ is the number of inlinks to document d_k and the summation ranges over the entire document collection.

We now turn to an empirical examination of the four query models introduced so far, but now with a non-uniform document prior as defined in Eq. 6. For each of our six collections and four models we generated another 100 queries. Table 8 shows the performance of each of the query models for each collection and retrieval model. Clearly, the non-uniform prior has a significant impact on the performance. From the number of inlinks of known items with the non-uniform prior (shown in last column of Table 6), it would seem that the prior is too strong and results in documents with a larger than average number of inlinks than manual topics. Regardless, the performance on these queries is surprisingly high, despite the fact that none of the retrieval models use any link information. For all languages we obtain substantial increases to the MMR for the majority of the query models.

Specifically, for ES, model 2 queries are comparable to manual topics instead of models 1 and 2 (in the uniform prior case), while for the UK collection, model 1 and 2 queries are now comparable as opposed to only model 1 previously. In all other instances, the performance of the models was substantially higher than what manual queries obtain.

Table 8: Mean Reciprocal Rank: On simulated queries with inlink Prior

Lang.	Query Type	TF.IDF	OKAPI	LM
ES	**Model 1**	0.2533	0.4467	0.3577
	Model 2	**0.2786***	**0.4099***	**0.3430***
	Model 3	0.5620	0.6368	0.6080
	Model 4	0.3523	0.5418	0.4563
UK	**Model 1**	**0.3334***	**0.4994***	**0.4363***
	Model 2	**0.3602***	**0.5171***	**0.4883***
	Model 3	0.7741	0.7640	0.7719
	Model 4	0.5526	0.6812	0.6279
NL	**Model 1**	0.1662	0.2898	0.2453
	Model 2	0.1637	0.2394	0.2013
	Model 3	0.6464	**0.6189***	**0.6204***
	Model 4	0.3151	**0.4392***	**0.3751***
PT	**Model 1**	0.1383	0.1729	0.1712
	Model 2	0.1395	0.1370	0.1653
	Model 3	**0.1979***	**0.2249***	**0.2200***
	Model 4	**0.1468***	**0.1796***	**0.1654***
HU	**Model 1**	**0.2707***	**0.2753***	**0.3166***
	Model 2	**0.3317***	**0.3228***	**0.4149***
	Model 3	0.6908	0.6181	0.6033
	Model 4	0.5431	**0.4688***	0.4664
DE	**Model 1**	0.1445	0.1762	0.1405
	Model 2	0.1878	0.1615	0.1847
	Model 3	**0.4012***	0.3767	**0.4092***
	Model 4	0.2625	0.3179	0.3093

The application of the inlink prior in the process results in simulated topics for HU on models 1 and 2, and PT on model 3, that are now comparable with the manual topics. The generation process has been modified to reflect the desire of seeking important pages, and this has ameliorated the retrieval performance, to fall in line with manual queries.

For the DE and NL simulated queries some comparable performance is now found with Model 3, but only on two of the retrieval models. These results show that the known-item document selected is important and improves performance (substantially so in the case of DE topics), but the simulated topics are still under performing compared to the manual topics.

7. DISCUSSION

We analyzed the performance of simulated known-item queries in six European languages where the differences in languages considered raised many issues concerning the automatic generation of queries. Different languages require different models to cater for discrimination of terms, and models which do not arbitrarily select random known items but known items which tend to be more important. The latter change to models previously proposed in the literature is very surprising because despite the retrieval models not using any link information the performance of these queries is substantially better across all collections. This reveals an interesting pattern of behavior on behalf of the user: they prefer to retrieve known items which tend to be more important.

This difference in performance between generation models with and without the prior on importance suggests that important pages have certain characteristics which make them more likely to be retrieved highly in the ranking, regardless of retrieval model. Retrieving a random document in the collection is substantially more difficult than retrieving more important documents.

While querying models currently employed do not always adequately capture the querying process that a user adopts when formulating a known-item query, we have identified several areas where the process can be improved. For particular languages certain strategies have proved to be more appropriate. It is possible to generate queries for known-item topics that have comparable performance to the manual queries for UK and ES with Popular/Uniform strategies, while the other languages are better with more discriminative term selection models coupled with the importance prior. However, further research is required to develop more sophisticated models of known-item topic generation which extend the process further to consider other factors, such as term dependence, document length, de-compounding, etc.

8. CONCLUSIONS

We have analyzed a number of factors which affect the quality of the queries produced by previous query generation models. This has led to refinements of the query generation process, employing an improved term selection method and document priors based on inlink counts. Known-item topics can now be simulated in such a way that they produce similar performance to manually created know-item topics. Put differently, the generation models are replicatively validated: the performance of the output of both processes is sampled from the same underlying distribution. Thus, we have identified specific generation models for specific languages that produce simulated queries that are replicatively valid.

While this is an important step and contribution toward using simulated known-item topics, further work is required in order to build models of query generation that are not only replicatively, but also predictively valid. E.g., we have not examined whether the simulated queries, themselves, are similar to real queries for a given known item. I.e., are the query terms generated the same as the real terms? This is also left to further work, along with examining a number of other arising research questions, such as: why are particular documents favored by the user, but also easier to retrieve, and why are random documents harder to find? And, do system rankings resulting from replicatively valid query models correlate well with system rankings resulting from manual topics?

9. ACKNOWLEDGMENTS

Krisztian Balog was supported by the Netherlands Organisation for Scientific Research (NWO) by a grant under project number 220-80-001. Maarten de Rijke was also supported by NWO under project numbers 017.001.190, 220-80-001, 264-70-050, 354-20-005, 600.065.120, 612-13-001, 612.-000.106, 612.066.302, 612.069.006, 640.001.501, 640.002.501, and by the E.U. IST programme of the 6th FP for RTD under project MultiMATCH contract IST-033104.

10. REFERENCES

[1] L. Azzopardi and M. de Rijke. Automatic construction of known-item finding test beds. In *Proc. 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 603–604, 2006.

[2] K. Balog, L. Azzopardi, J. Kamps, and M. de Rijke. Overview of WebCLEF 2006. In A. Nardi, C. Peters, and J. Vicedo, editors, *Working Notes CLEF 2006*, Sept 2006.

[3] H. Bast and I. Weber. Type less, find more: fast autocompletion search with a succinct index. In *Proc. 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 364–371, 2006.

[4] A. Berger and J. Lafferty. Information retrieval as statistical translation. In *Proc. 22nd ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 222–229, Berkeley, CA., 1999.

[5] D. Cai, C. J. van Rijsbergen, and J. Jose. Automatic query expansion based on divergence. In *Proc. 10th International Conference on Information Knowledge Management*, pages 419–246, 2001.

[6] P. Calado, A. S. da Silva, R. C. Vieira, A. H. F. Laender, and B. A. Ribeiro-Neto. Searching web databases by structuring keyword-based queries. In *Proc. 11th international conference on Information and knowledge management*, pages 26–33, New York, NY, USA, 2002.

[7] J. P. Callan and M. E. Connell. Query-based sampling of text databases. *Information Systems*, 19(2):97–130, 2001.

[8] S. Cronen-Townsend, Y. Zhou, and W. B. Croft. Predicting query performance. In *Proc. 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 299–306, 2002.

[9] V. Hollink, J. Kamps, C. Monz, and M. de Rijke. Monolingual document retrieval for European languages. *Information Retrieval*, 7:33–52, 2004.

[10] M. Inoue. The remarkable search topic-finding task to share success stories of cross-language information retrieval. In *Proc. SIGIR 2006 Workshop on New Directions in Multilingual Information Access*, 2006.

[11] M. Inoue and N. Ueda. Retrieving lightly annotated images using image similarities. In *Proc. 2005 ACM symposium on Applied computing*, pages 1031–1037, 2005.

[12] C. Jordan, C. Watters, and Q. Gao. Using controlled query generation to evaluate blind relevance feedback algorithms. In *Proc. 6th ACM/IEEE-CS joint conference on Digital libraries*, pages 286–295, 2006.

[13] B. Sigurbjörnsson, J. Kamps, and M. de Rijke. EuroGOV: Engineering a multilingual Web corpus. In C. Peters, F. C. Gey, J. Gonzalo, G. Jones, M. Kluck, B. Magnini, H. Müller, and M. de Rijke, editors, *Accessing Multilingual Information Repositories*, LNCS 4022. Springer Verlag, 2006.

[14] B. Sigurbjörnsson, J. Kamps, and M. de Rijke. Overview of WebCLEF 2005. In C. Peters, F. Gey, J. Gonzalo, H. Müller, G. Jones, M. Kluck, B. Magnini, and M. de Rijke, editors, *Accessing Multilingual Information Repositories*, LNCS 4022, pages 810–824. Springer, Sept 2006.

[15] J. Tague, M. Nelson, and H. Wu. Problems in the simulation of bibliographic retrieval systems. In *Proc. 3rd annual ACM conference on Research and development in information retrieval*, pages 236–255, 1981.

[16] J. M. Tague and M. J. Nelson. Simulation of user judgments in bibliographic retrieval systems. In *Proc. 4th annual international ACM SIGIR conference on Information storage and retrieval*, pages 66–71, 1981.

[17] E. M. Voorhees. The philosophy of information retrieval evaluation. In *Revised Papers from the Second Workshop of the Cross-Language Evaluation Forum on Evaluation of Cross-Language Information Retrieval Systems*, pages 355–370, London, UK, 2002. Springer-Verlag.

[18] B. P. Zeigler. *Theory of Modelling and Simulation.* Wiley, 1976.