Validating Query Simulators: An Experiment Using Commercial Searches and Purchases

Bouke Huurnink, Katja Hofmann, Maarten de Rijke, and Marc Bron

ISLA, University of Amsterdam, The Netherlands
{bhuurnink,k.hofmann,derijke,m.m.bron}@uva.nl

Abstract. We design and validate simulators for generating queries and relevance judgments for retrieval system evaluation. We develop a simulation framework that incorporates existing and new simulation strategies. To validate a simulator, we assess whether evaluation using its output data ranks retrieval systems in the same way as evaluation using real-world data. The real-world data is obtained using logged commercial searches and associated purchase decisions. While no simulator reproduces an ideal ranking, there is a large variation in simulator performance that allows us to distinguish those that are better suited to creating artificial testbeds for retrieval experiments. Incorporating knowledge about document structure in the query generation process helps create more realistic simulators.

1 Introduction

Search engine transaction logs offer an abundant source of information about search "in the wild." An increasing number of studies have been performed into how searches and result clicks recorded in transaction logs may be used to evaluate the performance of retrieval systems [6, 7, 9] . However, transaction logs often contain information that can be used to breach the privacy of search engine users. In addition, their content may contain commercially sensitive information. Therefore companies are reluctant to release such data for open benchmarking activities.

A solution may lie in using simulation to generate artificial user queries and judgments. Simulation is a method by which large numbers of queries and judgments may be obtained without user interaction, as a substitute for hand-crafted individual retrieval queries and explicitly judged sets of relevant documents. Simulated queries have been compared to manually created queries for information retrieval [2, 13]. Reproducing absolute evaluation scores through simulation has been found to be challenging, as absolute scores will change with, e.g., recall base. However, reproducing exact retrieval scores is not essential to developing a useful simulator when we wish to rank retrieval systems by their performance. Following this argument, the aim is to make a simulator that allows us to identify the *best performing* retrieval system.

Consequently, we assess simulation approaches based on how well they predict the relative performance of different retrieval systems. More precisely, we examine whether simulated queries and relevance judgments can be used to create an artificial evaluation testbed that reliably ranks retrieval systems according to their performance. To the best of our knowledge this question has not been addressed so far.

M. Agosti et al. (Eds.): CLEF 2010, LNCS 6360, pp. 40–51, 2010.

In the novel simulation approaches that we introduce, we integrate insights into users' search goals and strategies to improve the simulation, e.g., patterns regarding the items typically sought and/or purchased. We also enrich the simulation process by exploiting characteristics of the document collection in the commercial environment in which we work, in particular the fielded nature of the documents.

We assess the validity of the output of the simulation approaches that we consider by correlate system rankings produced by our simulator output data with a gold standard system ranking produced by real judgments. Ideally, a large number of judgments should be used to produce this gold standard ranking, so as to obtain stable average evaluation scores. As sets of manually judged queries are usually small and difficult to obtain, we use implicit judgments derived from the transaction log of a commercial environment, where users search for and purchase documents.

Purchase-query pairs (i.e., user queries and their subsequent purchases) can be identified from the transaction logs and we use these purchases as implicit positive relevance judgments for the associated queries [7]. In addition, we use a set of training purchase-query pairs to help inform our simulators. We use the simulators to create sets of artificial purchase-query pairs to use as evaluation testbeds. Each simulator is then assessed by determining whether its testbed produces similar retrieval system rankings to the gold standard testbed created using real-world (logged) purchase-query pairs.

The main contributions of this paper are:

1. A large-scale study of the correlation between system rankings derived using simulated purchase-query pairs and a system ranking obtained using real queries and (implicit) assessments.
2. Novel query simulation methods that exploit characteristics from real queries as well as document structure and collection structure.
3. A detailed analysis of factors that impact the performance of a simulator.

Our results can be used by creators of evaluation collections who have access to transaction log data but are unable to release it and by experimenters who want to create realistic queries for a document collection without having access to the transaction logs.

2 Related Work

We describe developments in creating simulated queries and relevance assessments for retrieval evaluation.

Research into simulation for information retrieval systems goes back to at least the 1980s. Early work focused on simulating not only queries and relevance assessments, but also the documents in a collection. Tague et al. [12] developed simulation models that produced output data similar to real-world data, as measured for example by term frequency distributions. However, retrieval experiments showed that evaluations using the simulated data did not score retrieval systems in the same way as evaluations using real-world data [13]. Gordon [4] suggested that simulators should focus on creating queries and relevance judgments for pre-existing (non-simulated) document collections: the remainder of the work that we describe has been performed under this condition.

A problem when simulating queries is to identify multiple relevant documents for a query. One solution has been to use document labels or pre-existing relevance judgments to identify sets of multiple related documents in a collection [1, 10, 11]. Queries are back-generated from the related documents, which are then considered relevant for that query. This allows queries to be defined so that they conform to predefined criteria, e.g., long vs. short. While queries generated in this manner can be used to evaluate retrieval systems, for validation purposes there is typically no user interaction data available. Therefore it is unclear how they compare to "real-world" queries.

An alternative to identifying multiple relevant documents per query is to focus on identifying a single relevant document per query. In the context of online search, Dang and Croft [3] work under this condition. They exploit hyperlinked anchor texts, using the anchor text as a simulated query and the hyperlink target as a relevant document. Azzopardi et al. [2] study the building of simulators for known-item queries—queries where the user is interested in finding a single specific document. Their approach is to first select a target known-item document from the collection, and to subsequently back-generate a query from the terms in the selected document. In both of these studies, simulator output was compared to real-world data with respect to validity for evaluating retrieval systems. Dang and Croft [3] modified the anchor text queries and compared system evaluations on the simulated modified queries to system evaluations on similarly modified queries and clicks taken from an actual search log. They found that, in terms of systems evaluation, the simulated queries reacted similarly to various modification strategies as real queries. Azzopardi et al. [2] compared their simulated known-item queries to sets of 100 manually created known-item queries, examining absolute evaluation scores attained by evaluating systems on the simulator output data. The simulator output sometimes produced retrieval scores statistically indistinguishable to those produced using real known-item queries. Factors found to affect the simulator assessments include the strategy used to select known items, and the strategy used to select query terms from a target document. The optimal simulator settings varied per document collection and per retrieval system.

Our work is similar to [2] as our goal is to assess simulators for retrieval evaluation. However, we focus on relative performance instead of absolute scores as we argue that this is a more feasible and useful goal. Instead of comparing simulators to explicit judgments for known-item queries, we compare our approaches to a large number of purchase-query pairs that are derived from implicit judgments obtained from a transaction log. We apply and extend simulation strategies developed in [2], and compare these to strategies that take characteristics of logged queries into account.

3 Simulation Framework

First, we describe our setup for assessing query/document pair simulators. Then we detail the simulators, including how they select target documents and query terms.

3.1 Validating Simulators

In validating simulators we are particularly interested in how closely the system rankings using simulated purchase-query pairs resemble the system rankings using real

purchase-query pairs. To assess a simulator, we first run it to create an evaluation testbed consisting of a set of queries with associated relevant documents (one relevant document per query, thus resembling a known-item task). The relevance judgments thus generated are then used to score retrieval systems. We evaluate the performance of each retrieval system on the automatically generated testbed in terms of Mean Reciprocal Rank (MRR), a standard evaluation measure for known-item tasks. Next, systems are ranked by retrieval performance. The ranking is compared to the one obtained by ranking the systems using real purchase-query pairs from a commercial transaction log.

Comparisons between system rankings are couched in terms of the rank correlation coefficient, Kendall's τ. A "better" (read: more realistic) simulator would achieve a higher rank correlation score with the gold standard ranking. Kendall's τ has been used previously for determining the similarity of evaluation testbeds generated by different sets of human assessors. Voorhees [14] considered evaluation testbeds with $\tau \geq 0.9$ to be equivalent. An ideal simulator would produce exactly the same rankings as the real queries, i.e., $\tau = 1.0$.

3.2 Simulating Purchase Queries

We base our framework for simulating purchase-query pairs on the known-item search simulation framework presented in [2]. Purchase queries are similar to known-item searches in that, for both types of search, a single query is (usually) associated with a single relevant document. For purchase queries the user may not necessarily know beforehand exactly which item they wish to obtain, but usually purchases a single item. In our transaction log, 92% of the purchase queries led to exactly one purchased item.

We extend the framework in [2] by incorporating information about the document fields from which query terms are typically drawn (detailed below). We observe that users of search systems tend to select query terms from specific parts of a document, e.g., users in our setting have been found to frequently issue queries containing terms from document titles [8]. To take this into account, we allow the simulator to use information about document fields to select query terms.

The resulting simulator framework is summarized in Algorithm 1. First, we select a document d_p from the collection C that is considered to be the target document to be purchased, according to a distribution D_d. We determine the length of the query by drawing from a distribution D_l that is estimated using a random subset of the training data (a sample of real purchase queries). For each term to be added to the query, we then determine the field from which the term should be drawn according to distribution D_f, and finally sample a term from the associated term distribution D_t.

In this setting, the crucial problems become: (1) determining D_d, i.e., which document should to be the target of a query; (2) determining D_f, i.e., which field should be selected as the source for a query term; and (3) determining D_t, i.e., which terms should be used to search for the target document. We implement and compare multiple strategies for addressing each problem, which we will discuss in turn.

D_d: Selecting target documents. We investigate the effect of varying the selection of documents to use as simulated purchases. In previous work selecting target documents according to document importance as captured by inlink counts was found to have a

Algorithm 1. Generalized overview of our purchase query simulator

Initialize an empty query $q = \{\}$
Select the document d_p to be the purchased document with probability $p(d_p)$
Select the query length l with probability $p(l)$
for i in 1 .. l **do**
  if Using field priors **then**
    Select the document field f from which a term should be sampled with probability $p(f)$
    Select a term t_i from the field model (θ_f) of f with probability $p(t_i|\theta_f)$
  else
    Select a term t_i from the document model (θ_d) with probability $p(t_i|\theta_d)$
  Add t_i to the query q
Record d_p and q to define the purchase-query pair

positive effect in obtaining scores closer to retrieval scores using "real" queries [2]. We operate in an environment where inlink information is not available. Therefore, we formulate two target selection strategies that are expected to be representative of a lower and upper bound in simulation accuracy: (1) a uniform selection strategy, and (2) an oracle selection strategy that selects documents that are known to have been purchased.

Uniform. All documents in the collection are equally likely to be selected (samples are drawn with replacement). This strategy only requires the presence of a document collection and does not assume additional information. The probability of selecting a document is $p_{\text{Uni}}(d_p) = |C|^{-1}$, where $|C|$ is the collection size.

Oracle. For each logged purchased document, a query is generated. This strategy exactly mirrors the distribution of purchased documents that is observed in the test collection. The probability of selecting a document is determined by: $p_{\text{Ora}}(d_p) = \#(d_p) \cdot (\sum_{d \in D_p} \#(d))^{-1}$, where $\#(\cdot)$ is the number of times a document is purchased and D_p is the set of purchased documents.

We expect the oracle selection strategy to result in a simulator for which the resulting system rankings more closely resemble a ranking resulting from real queries. If the two document selection strategies lead to large differences in correlations with a system ranking produced by real queries, this would mean that more complex strategies for generating purchase distributions should be investigated further.

***D_t*: Selecting query terms.** The second step in developing a purchase query simulator is to generate query terms that a user might use to (re)find a given target document. Many strategies are possible — we focus on the effect of existing term selection methods and the effect of selecting terms from different document fields. The following selection methods, previously defined for known-item search in [2], are investigated:

Popular. Query terms are sampled from the purchased document using the maximum likelihood estimator. Frequently occurring terms in the document are most likely to be sampled. The probability of sampling a term is determined by: $p(t_i|\theta_d) = n(t_i, d) \cdot (\sum_{t_j \in d} n(t_j, d))^{-1}$, where $n(t, d)$ is the number of times t occurs in d.

Uniform. Query terms are sampled from the document using a uniform distribution (each term has an equally likely chance of being sampled): $p(t_i|\theta_d) = |d|^{-1}$, where $|d|$ is the number of unique terms in a document.

Discriminative. Query terms are sampled from the document using their inverse collection frequency. Terms that rarely occur in the collection are most likely to be sampled. The probability of sampling these terms is determined by:

$$p(t_i|\theta_d) = \frac{b(t_i, d)}{p(t_i) \cdot \sum_{t_j \in d} \frac{b(t_j,d)}{p(t_j)}}, \tag{1}$$

where $b(t, d) = 1$ if t is present in d and 0 otherwise and $p(t)$ is given by:

$$p(t_i) = \frac{\sum_{d \in C} n(t_i, d)}{\sum_{d \in C} \sum_{t_j \in d} n(t_j, d)}. \tag{2}$$

TF.IDF. Query terms are sampled from the document according to their TF.IDF value. Terms that occur rarely in the collection, but frequently in the document, are most likely to be sampled. Writing $df(t)$ for the document frequency of a term, we put:

$$p(t_i|\theta_d) = \frac{n(t_i, d) \cdot \log \frac{|C|}{df(t_i)}}{\sum_{t_j \in d} n(t_j, d) \log \frac{|C|}{df(t_j)}}. \tag{3}$$

Other term sampling strategies are possible, e.g., ones that take term distances or co-occurrences into account, but these go beyond the scope of this paper.

Sampling strategies are expected to affect how realistic simulated queries are, as they constitute different models of how users create query terms when thinking about a target document. Query formulation is more complex in real life, but a model that explains a large part of real query generation well will result in a better simulator.

A simulator that uses a term selection method that is close to that of the term scoring method underlying a retrieval system used to evaluate that simulator will score high on the queries thus generated. This need not result in a good simulator, as we are comparing system rankings to those resulting from evaluation using real purchase queries.

D_f: Incorporating document structure. Beyond the influence of the term selection strategy, we observe that users of online search systems tend to select query terms from specific parts of a document [8]. In the collection that we used for development (see Section 4), we observed the program description was the most frequent source of query terms, followed by the summary, title, and recording number fields. Table 1 gives a description of the fields that are used in our document collection, and specifies the likelihood that query terms are matched in the respective fields. We obtained the probabilities by matching terms in queries from the development set to their field locations in the associated purchased documents. If a term occurred in multiple fields, each field was counted once. Note that this analysis is not representative of the entire group of queries issued in the archive, as only certain types of queries have been included in this paper (see Section 4.)

In our simulation experiments, we systematically explore the use of document fields for generating simulated queries, using the four most frequent sources of query terms on the one hand, and the distribution of field priors on the other:

Table 1. Names and descriptions of fields available in our experimental document collection; $p(f)$ indicates the probability that the field contains a purchase query term

Name	Description	$p(f)$	Name	Description	$p(f)$
beschrijving	description of the program	0.4482	immix_docid	document id	0.0037
dragernummer	number of recording	0.1303	zendgemachtigde	broadcast rights	0.0020
hoofdtitel	program title	0.1691	rechten	copyright owner	0.0012
samenvatting	summary of the program	0.1406	genre	the type of a program	0.0008
selectietitel	titles of program sections	0.0348	dragertype	format of recording	0.0005
makers	creators of the program	0.0241	deelcatalogus	catalog sub-collection	0.0000
namen	names mentioned in the program	0.0190	dragerid	id of recording	0.0000
persoonsnamen	people in program	0.0153	dragersoort	type of recording	0.0000
geografische_namen	locations mentioned in program	0.0051	herkomst	origin of the program	0.0000
trefwoorden	keywords	0.0051	publid	publication id	0.0000

Whole Document. Query terms are sampled from the entire document without incorporating any information about fields.

Description. Query terms are sampled from the description field only.

Summary. Query terms are sampled from the summary field only.

Title. Query terms are sampled from the title field only.

Recording number. Query terms are sampled from the recording number field only.

Field priors. Query terms are drawn from any part of the document, but terms from fields that are more frequent sources of real query terms are more likely to be used as a source for query terms. This model corresponds to the full setup outlined in Algorithm 1, including the optional field selection step. Field prior probabilities are estimated using the development collection, and correspond to $p(f)$ in Table 1.

4 Experimental Setup

We describe the setup of our experiments in terms of the data used, the settings used when applying our simulators, and the retrieval systems used to assess the simulators.

4.1 Experimental Data

Our experimental data consists of a collection of documents and a large set of purchase-query pairs. We obtain this data from an audiovisual archive, the *Netherlands Institute for Sound and Vision*. The archive maintains a large collection of audiovisual broadcasts. Customers (primarily media professionals) can search for and purchase broadcasts (or parts of broadcasts). Their actions are recorded in transaction logs [8].

Each audiovisual broadcast in the archive is associated with a textual catalog entry that describes its content and technical properties. Search in the archive is based on text search of these entries. As our document collection we use a dump of the catalog entries made on February 1, 2010.

Our purchase-query pairs are derived from transaction log data gathered between November 18, 2008 and February 1, 2010. In some cases purchases are made for fragments of a broadcast; following the practice at TREC and other evaluation forums [5], we consider the entire broadcast relevant if it contains relevant information, i.e., if a

fragment has been purchased. The transaction log includes queries that contain date filters and advanced search terms. We exclude these types of queries from our experiments, leaving their simulation for future work. In some cases a query resulted in purchases for multiple broadcasts, here we consider each purchase-query pair separately. In total we derived $31,237$ keyword-only purchase-query pairs from the collection.

Our documents and purchase-query pairs are divided into a training and a test set. The training set is used to derive probability estimates for simulation purposes. The test set is used to produce the gold standard ranking of retrieval systems. In order to preserve the same distribution of documents and purchase-query pairs in the training and test set, documents were randomly assigned to either set. Purchase-query pairs were then assigned to a set depending on the assignments of the purchased document.

4.2 Simulator Settings

We create a simulator for each combination of the target, term, and field selection models described in Section 3.2. Query lengths for the simulators are drawn from the distribution of query lengths in the training queries. Query terms are taken directly from documents without modification. We generate 1,000 purchase-query pairs per simulator. These are then used to evaluate the retrieval systems described below.

4.3 Retrieval Systems

To produce system rankings, we need multiple retrieval systems that generate diverse retrieval runs. To this end we use a variety of indexing and preprocessing methods and scoring functions. We use two open source toolkits for indexing and retrieval: Indri[1] (based on language modeling) and Lucene[2] (based on the vector-space model) to create a total of 36 retrieval systems. The main difference between the toolkits is the document scoring method, but they also differ in terms of pre-processing and indexing strategy; both are frequently used in real-world search applications and retrieval experiments. Some of the 36 retrieval systems are designed to give very similar performance, to capture subtle differences in ranking of similar systems; others are designed to give very different retrieval performance, to capture large fluctuations in performance.

We build three indexes, one using Indri, and two using Lucene. For the Indri index we use the standard pre-processing settings, without stemming or stop word removal. For the first Lucene index, we apply a standard tokenizer, for the second we additionally remove stop words and apply stemming using the Snowball Stemmer for Dutch.

The Indri retrieval toolkit is based on language modeling and allows for experimentation with different smoothing methods, which we use to generate runs based on a number of models. Documents receive a score that is based on the probability that the language model that generated the query also generated the document. This probability estimate is typically smoothed with term distributions obtained from the collection. The setting of these smoothing parameters can have a large impact on retrieval performance. Here we generate retrieval systems using Dirichlet smoothing with the parameter range $\mu = 50, 250, 500, 1250, 2500, 5000$. In this manner, we generate a total of

[1] `http://www.lemurproject.org/indri/`
[2] `http://lucene.apache.org/`

6 smoothing-based retrieval runs. Some of these 6 systems can be expected to produce similar retrieval results, allowing us to capture small differences in system rankings.

Both Indri and Lucene provide methods for indexing per field, allowing us to create alternative retrieval systems by forming different combinations of fields; Table 1 shows the names and descriptions of the indexed fields. We generate 10 fielded retrieval runs for each index (for a total of 30 runs), based on one or more of the following fields: *content* (all text associated with a document), *free(text)* (summary and description), *meta* (title and technical metadata), and *tags* (named entities, genre). The 10 field combinations can be expected to give very different performance, while applying a specific field combination to three index types will result in smaller variations in performance.

5 Results and Analysis

Our experiments are designed to validate simulation approaches by assessing how well their simulated purchase-query pairs rank retrieval systems in terms of their performance. A second goal is to identify the best performing simulator, i.e., the simulator that results in rankings that are closest to the gold standard ranking produced using real queries. In this section we provide an overview and analysis of our experimental results.

The correlation coefficients for the simulators produced in our experiments are given in Table 2. The simulator with the lowest coefficient of 0.286 uses *Discriminative* term selection in combination with *Summary* field selection and *Oracle* target selection, indicating that this simulator setting is particularly unsuited for generating realistic purchase-query pairs. The simulator with the highest correlation coefficient of 0.758 uses *Field Prior* field selection in combination with *Uniform* target selection, and *TF.IDF* term selection. None of the simulators achieves the value of $\tau \geq 0.9$ that indicates the equivalence of two testbeds created by human assessors, indicating that there is still plenty of room for improvement in creating simulators that realistically reproduce

Table 2. Correlation coefficients of system rankings using simulated queries and a system ranking using real-world data. The simulator with the highest coefficient overall is highlighted in bold. Shading has been used to differentiate between ranges of correlation coefficients: darkest shading for $\tau \geq 0.7$, medium shading for $0.5 \leq \tau < 0.7$, light shading for $0.3 \leq \tau < 0.5$, and no shading for $\tau < 0.3$.

	***Uniform* Target Model**				***Oracle* Target Model**			
	Term Model				*Term Model*			
Field Model	*Popular*	*Uniform*	*Discrim.*	*TF.IDF*	*Popular*	*Uniform*	*Discrim.*	*TF.IDF*
Whole Document	0.714	0.741	0.666	0.690	0.650	0.697	0.667	0.677
Description	0.393	0.348	0.347	0.360	0.382	0.373	0.371	0.375
Summary	0.352	0.340	0.365	0.355	0.435	0.476	0.286	0.384
Title	0.444	0.461	0.418	0.432	0.385	0.373	0.405	0.392
Recording Number	0.654	0.682	0.645	0.674	0.684	0.704	0.673	0.686
Field Priors	0.738	0.745	0.714	**0.758**	0.738	0.721	0.624	0.687

human querying and purchasing behavior. However, the large variance in simulator assessments does give some insight into which simulator strategies are preferable in the framework that we employ.

Incorporating field priors. Overall, we can see that simulators incorporating *Field Prior* field selection produce the most reliable system rankings, as measured by correlation to a system ranking using real purchase-query pairs. Except for one case, using field priors consistently and substantially improves over sampling from the whole document without taking field information into account.

The least reliable system rankings are produced by restricting field selection to a single field, as is the case for the *Title*, *Summary*, and *Description* field selection models. An exception is the *Recording Number* field. Simulators using this field selection model achieve correlations that are, depending on the term and target selection models, in many cases similar to the correlations achieved when using the whole document. This points to an important aspect of the real queries, namely, that many of them are high precision in nature. Professional users often know very well what they are looking for and how to find it, and searching on the highly distinctive recording number allows a document to be quickly targeted. Simulators missing this high-precision field lose out when trying to predict which retrieval systems will perform well in this setting.

Selecting known purchased documents. Simulators selecting documents that are known to be purchased (i.e., using *Oracle* target selection) generally do not produce better evaluation testbeds than simulators that select documents uniformly from the entire collection. This is somewhat surprising as Azzopardi et al. [2] found that a non-uniform sampling strategy based on document importance produced better simulators. However, the cited work validated simulators according to their ability to reproduce the absolute

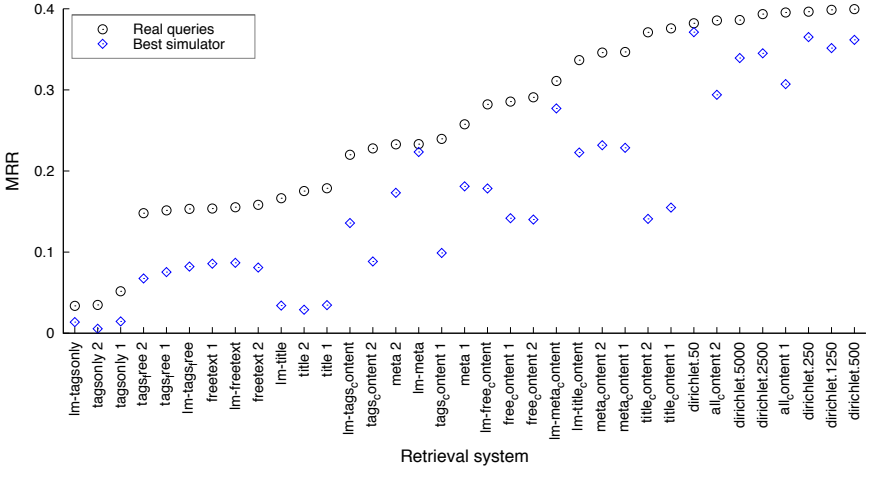

Fig. 1. Evaluation scores of retrieval systems in terms of MRR, on real purchase-query pairs derived from transaction logs, and on simulated purchase-query pairs generated by the best simulator ($D_d = Uniform, D_t = Uniform, D_f = FieldPriors$). Retrieval systems are sorted by their performance on real purchase-query pairs.

retrieval evaluation scores, while in this work we validate simulators according to their ability to rank retrieval systems. This may account for the difference in our findings, although further work is necessary to confirm this.

The impact of term selection. Comparing simulator assessment scores across the term selection models, we find subtle variations in correlation coefficients. No single model scores high consistently. In our experiments, then, the optimal term selection model is dependent on the target selection and field selection models that are used.

Absolute retrieval evaluation scores. For illustration purposes we briefly discuss absolute retrieval scores. Figure 1 plots the retrieval scores of retrieval systems evaluated with real purchase-query pairs from transaction logs, against the retrieval scores of the same systems when applied to the artificial purchase-query pairs generated by the best simulator. Some clusters of system rankings are correctly reproduced by the simulator (systems scoring better on real queries also tend to perform better on the simulated queries, and vice versa), even though absolute retrieval scores are generally lower than those obtained using the real queries.

6 Conclusion

We have explored the design and validation of simulators for generating purchase-query pairs, consisting of a query and an associated relevant purchased document, in a commercial setting. We developed a purchase query simulation framework incorporating new and previously existing simulation approaches. The framework incorporates models for (1) selecting target purchased documents, (2) selecting query terms, and (3) incorporating document structure in the query term selection process. By varying these models we created 48 simulators. Each simulator was used to produce an artificial evaluation testbed of simulated purchase-query pairs. The simulators were validated in terms of how well their testbeds ranked retrieval systems, as compared to the gold standard ranking obtained using a testbed of real logged purchase-query pairs.

No simulator produced an evaluation testbed that, according to the commonly accepted threshold for significance, ranked retrieval systems equivalently to the gold standard. This indicates that there is still plenty of room for improvement in the intrinsically difficult query simulation task. However, the large variation in simulator assessments did highlight broad categories of more successful and less successful approaches. Simulators were helped by explicitly including information about the parts of documents that query terms are drawn from in the term selection model. Further, in our setting, uniform selection of target purchased documents worked at least as well as a non-uniform selection of known purchased documents. This contrasts with previous findings in a different setting, where selecting "important" documents as targets for known-item searches resulted in improved simulator performance.

With large collections of queries being logged, one promising direction for improving simulators is further leveraging of the distributions of real-world queries and implicit judgments. For example, this study did not take the use of phrases as queries into account—their distribution could be estimated in sets of training queries and incorporated in a simulator. Other directions for future work include assessing the simulation

approach in multilingual archival environments such as the European Library, and examining in further detail the multimedia aspect of this particular study, for example by exploring in detail the simulation of queries that target specific visual content.

Acknowledgements

This research was supported by: the European Union's ICT Policy Support Programme as part of the Competitiveness and Innovation Framework Programme, CIP ICT-PSP under grant agreement nr 250430; the DuOMAn project carried out within the STEVIN programme which is funded by the Dutch and Flemish Governments under project nr STE-09-12; the Center for Creation, Content and Technology (CCCT); the Dutch Ministry of Economic Affairs and Amsterdam Topstad under the Krant van Morgen project; and the Netherlands Organisation for Scientific Research (NWO) under project nrs 640.001.501, 612.066.512, 612.061.814, 612.061.815, 640.004.802.

References

[1] Azzopardi, L.: Query side evaluation: an empirical analysis of effectiveness and effort. In: SIGIR 2009, pp. 556–563. ACM, New York (2009)
[2] Azzopardi, L., de Rijke, M., Balog, K.: Building simulated queries for known-item topics: an analysis using six European languages. In: SIGIR 2007, pp. 455–462. ACM Press, New York (2007)
[3] Dang, V., Croft, B.W.: Query reformulation using anchor text. In: WSDM 2010, pp. 41–50. ACM Press, New York (2010)
[4] Gordon, M.D.: Evaluating the effectiveness of information retrieval systems using simulated queries. J. American Society for Information Science and Technology 41(5), 313–323 (1990)
[5] Harman, D.K.: The TREC test collection, chapter 2, pp. 21–52. TREC: Experiment and Evaluation in Information Retrieval (2005)
[6] He, J., Zhai, C., Li, X.: Evaluation of methods for relative comparison of retrieval systems based on clickthroughs. In: CIKM 2009, pp. 2029–2032. ACM, New York (2009)
[7] Hofmann, K., Huurnink, B., Bron, M., de Rijke, M.: Comparing click-through data to purchase decisions for retrieval evaluation. In: SIGIR 2010, Geneva, ACM, New York (July 2010)
[8] Huurnink, B., Hollink, L., van den Heuvel, W., de Rijke, M.: The search behavior of media professionals at an audiovisual archive: A transaction log analysis. J. American Society for Information Science and Technology 61(6), 1180–1197 (2010)
[9] Joachims, T.: Optimizing search engines using clickthrough data. In: KDD 2002, pp. 133–142. ACM, New York (2002)
[10] Jordan, C., Watters, C., Gao, Q.: Using controlled query generation to evaluate blind relevance feedback algorithms. In: JCDL 2006, New York, NY, USA, pp. 286–295. ACM, New York (2006)
[11] Keskustalo, H., Järvelin, K., Pirkola, A., Sharma, T., Lykke, M.: Test collection-based IR evaluation needs extension toward sessions–a case of extremely short queries. Inf. Retr. Technology, 63–74 (2009)
[12] Tague, J., Nelson, M., Wu, H.: Problems in the simulation of bibliographic retrieval systems. In: SIGIR 1980, Kent, UK, pp. 236–255. Butterworth & Co., Butterworths (1981)
[13] Tague, J.M., Nelson, M.J.: Simulation of user judgments in bibliographic retrieval systems. SIGIR Forum 16(1), 66–71 (1981)
[14] Voorhees, E.M.: Variations in relevance judgments and the measurement of retrieval effectiveness. In: SIGIR 1998, pp. 315–323. ACM Press, New York (1998)