Exploring Topic-based Language Models for Effective Web Information Retrieval

Rongmei LI[1] Rianne Kaptein[2] Djoerd Hiemstra[1] Jaap Kamps[2]

[1] University of Twente, Enschede, the Netherlands
[2] University of Amsterdam, Amsterdam, the Netherlands

ABSTRACT

The main obstacle for providing focused search is the relative opaqueness of search request—searchers tend to express their complex information needs in only a couple of keywords. Our overall aim is to find out if, and how, topic-based language models can leads to more effective web information retrieval. In this paper we explore retrieval performance of a topic-based model that combines topical models with other language models based on cross-entropy. We first define our topical categories and train our topical models on the .GOV2 corpus by building parsimonious language models. We then test the topic-based model on TREC8 small Web data collection for ad-hoc search. Our experimental results show that the topic-based model outperforms the standard language model and parsimonious model.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Information theory

Keywords

Information Retrieval, Language Models, Parsimonious Language Models

1. INTRODUCTION

The Internet is the source of an enormous amount of data where specific information is difficult to find by simple browsing. Nowadays, Web users predominantly use Web search engines to retrieve information. As the Internet continues to grow exponentially, there will be increasing numbers of "relevant" pages to rank. These search results tend to have varying degrees of relevance to the interests of the searcher. The main obstacle to satisfying the needs of searchers is the relative opaqueness of search request—searchers tend to express their complex information needs in only a couple of keywords. But searchers like to have very focused results that are closer to their topic of interest at maximum extent.

DIR-2008 April 14-15, 2008, Maastricht, the Netherlands.

Efforts have been made in improving the search results for particular type of queries. There are specialized search engines, such as Google Scholar [9] that focuses on scientific documents; Ask [2] that provides an interactive way to either expand or narrow a user's search by a list of query topic related links. Alternatively, search users can use other techniques to retrieve topic-based information, such as hierarchical directories [8, 30, 32], currently emerging semantic web applications [24, 25, 28], and many other software agents and collaborative filtering systems [1, 10, 15].

However, specialised search engines only provide focused search if the user is first able to find the specialised search engine of his/her choice. Whereas specialized search engines are part of the answer to the increasing size of the web—they may identify more structured information and provide more focused search—they also introduce new information overload problems. Moreover, techniques like collaborative filtering provide complementary retrieval for topic-based information but not a search solution. Each alternative has its own representation of Web data. Nevertheless, none of these presentations alone are sufficient for focused search.

In this paper, we hypothesize that a restricted search within a topic may help focus retrieval. Our overall aim is to find out if, and how, topic-based search leads to more effective retrieval. In this topic-based search, user queries are first classified into their topical categories so that a user's intent is clearly defined; then a search is carried out within that topic. We may make use of the underlying structure of topical categories and present the difference of language usage between topics by probabilistic language modeling. On the bag-of-words assumption [6], a topical model can be constructed as the probability distribution of topical terms of a group of documents that are relevant to that topic. These topical models have the potential to let a generic search engine provide the quality of a dedicated vertical search engine. That is, using a range of topics models, the same search engine could provide focused search on a range of topics, whilst at the same time avoiding the need to locate a specialized search engine beforehand.

One of the main obstacles of focused search is the ambiguity of a user's queries. They are often short and dynamic and thus it is difficult for search engines to understand a user's intent correctly. In our envisaged framework, the choice of topic-based models depends on the correct interpretation of the user's intention on his/her own topic of interest. To disambiguate queries and restrict search, contextual information can be helpful such as (implicit/explicit) user feedback, query log, user geographical information (IP address),

previously visited Web pages, previously read and created documents and email [4, 16, 22, 26, 33]. The explicit or interactive user feedback on the topical category is one of the effective ways to reveal latent user intent. To evaluate the utility of such feedback, without having to resort to interactive experiments, we will simulate explicit user feedback with a survey asking test persons to classify retrieval queries into the predefined topical category.

In the rest of the paper, we first briefly introduce some topic-based language models in literature in Section 2. In Section 3, we present various language models including the topic-based language models. Section 4 explains our survey on topical categories in detail. We document and discuss our extensive experiments in Section 5. Finally, we draw our conclusion in Section 6.

2. RELATED WORK

In literature, there is a range of studies [3, 4, 18, 21, 29, 31] on building topic-based language models (LMs) to advance retrieval performance. A closely related approach is cluster-based retrieval where a cluster is a group of documents that are semantically close to each other. At query time, a list of documents will be retrieved based on the clusters that they belong to. Liu and Croft [18] re-examined the cluster-based retrieval in their recent work by using the language modeling approach. They first classified documents into clusters and then applied their cluster-based models to rank clusters and retrieve documents in the ranked clusters. Both query dependent and independent clustering algorithms were used. The first model was derived from the normal query likelihood retrieval model. It computed the likelihood of query generation from a cluster. The second model is a mixture of the document model, cluster model, and the collection/background model. Both models can be smoothed by common smoothing techniques.

Azzopardi et al. [3] adopted the same premise as Liu and Croft that similar documents will match the same information needs [27]. They proposed a topic-based LM that utilized an underlying topical structure within documents for better representation of document models. Different from the work of Liu and Croft, they improved the Bayes smoothing method by using a document dependant term prior to smooth the document model. The document-dependent term prior was estimated based on a term distribution over topics and the distribution of topics over a document. In case that a document was about only one topic, their method resulted in cluster-based retrieval.

Wei and Croft [29] tried to construct topical models from the Open Directory for ad-hoc search recently. In their work, each query was assigned to the "deepest" categories and only the first-level Web pages in a category were used as a topical collection for computing the topical model. Their topical model was estimated by the maximum likelihood estimator and was used for smoothing the query model. The KL-divergence between the modified query model and the document model was then calculated to rank documents. They also investigated two ways to combine topical models with the relevance model at model and query level respectively.

Bai et al. [4] proposed a framework based on language modeling approach. It integrated multiple contextual factors with the original query model by linear interpolation. Contextual factors included context around query, context within query, and blind feedback documents. Documents were then ranked by the KL-divergence score between the document model and the integrated query model. They constructed their domain model from context around query. For each domain, the specific part of the domain was extracted from a set of documents by applying the EM-algorithm. Queries were classified into corresponding domains based on their divergence score with domain model. They studied further on using feedback documents in context models.

3. LANGUAGE MODELS

The term LM originates from probabilistic models of language generation developed for automatic speech recognition systems in the early 1980s. Since 1998, LMs have had quite an impact on information retrieval (IR), especially text retrieval. It was first introduced by Ponte and Croft [20] and later explored in [5, 11, 19, 23]. It has been shown to perform well empirically [33]. Besides, the relative simplicity and effectiveness of the language modeling approach, together with the fact that it leverages statistical methods that have been developed in speech recognition and other areas, make it an attractive framework in which to develop new text retrieval methodology [33].

An important problem in the language modeling approach is the sparse data problem. It can eliminate a relevant document from a user's consideration due to one missing query term. The fundamental solution to this problem is so-called smoothing technique. Smoothing is the task to assign some non-zero probability to query terms that do not occur in a document. In IR setting, the simplest smoothing is to linearly interpolate the document model with a general collection model [5, 23]. There are other approaches to smoothing language models [13], some of which have been suggested for IR as well. For instance, smoothing using the geometric mean and backing-off by Ponte and Croft [20]; and Dirichlet smoothing and absolute discounting suggested in a study by Zhai and Lafferty [33].

3.1 Standard Models

A statistical LM is a probabilistic mechanism for explaining the generation of text. It basically defines a distribution over all possible word sequences. For IR a LM is defined for each document as the probability $P(t_1, t_2, ..., t_n|D)$ of generating a sequence of n query terms $t_1, ..., t_n$ from a given document. The documents are then ranked by that probability. The document having largest probability is considered most likely relevant to the given query. The standard language modeling approach uses a mixture of the document model $P(t_i|D)$ with a general collection/background model $P(t_i|C)$. The approach needs a parameter lambda λ which is set empirically on some test collection, or alternatively estimated by the EM-algorithm on a test collection. The simplest LM is the unigram LM, which is a word distribution over a natural language. In this paper, we employ unigram LMs, whose effectiveness for IR tasks has been demonstrated in the literature [12].

$$P(t_1, ..., t_n|D) = \prod_{i=1}^{n} (\lambda P(t_i|D) + (1 - \lambda)P(t_i|C)) \qquad (1)$$

The document model $P(t_i|D)$ can be estimated in many ways. The simplest method is called maximum likelihood

estimation that can be computed as follows:

$$P(t_i|D) = \frac{tf(t_i, D)}{\sum_t tf(t, D)} \quad (2)$$

where $tf(t_i, D)$ is the number of occurrences of term t_i in document D. Similarly we can calculate the background model $P(t_i|C)$ using maximum likelihood estimator:

$$P(t_i|C) = \frac{cf(t_i)}{\sum_t cf(t)} \quad (3)$$

where $cf(t_i)$ is the term frequency in the background collection.

3.2 Parsimonious Models

The standard LM tends to estimate the distribution for every term in a document. For a long document, the model will contain a long list of term probabilities. However, there are many general terms that often appear in every document. They are less discriminative and thus contributes less to distinguish a relevant document from others in regard to a query. To eliminate these terms from models, the so-called parsimonious LM is introduced [12]. Similar to the standard LM, the parsimonious model ranks documents based the smoothed document model. The difference is this model uses the EM-algorithm to estimate the term distribution in a document. At the expectation step, the expectation score is computed for all terms. The general terms should have smaller expectation score as they have relatively higher probabilities $P(t|C)$ in the background model. The algorithm uses a fixed value for μ. A low value of μ adds more weight on the background probability. In this way, we can further reduce the expectation score for the general terms. At the maximization step, the expectation score is normalized and compared to a given threshold. Terms having higher score will be preserved in the pruning process while terms having lower score will be eliminated from the next iteration. Some general terms will not pass the threshold test as their normalized expectation score will be low. This selection process will continue till the maximized term distribution will not change significantly anymore. Getting rid of terms or token that are common in general English, the resulting model thus has fewer terms than the standard model with full text indexing. In another words, the parsimonious model preserves specific terms that appear in a document frequently but relatively less often in the whole collection. This compact representation is very important for modeling data with terabyte scale such as the collection of Web pages on Internet.

$$\text{E-step}: \quad e_t = tf(t, D) \cdot \frac{\mu P(t|D)}{\mu P(t|D) + (1-\mu)P(t|C)}$$

$$\text{M-step}: \quad P(t|D) = \frac{e_t}{\sum_t e_t}, \text{i.e. normalize the model}$$

This learning process is unsupervised. It requires no information from a user query or relevance judgement.

3.3 Topical Models

Each Web page or document may consist of several topics. In this paper, we assume a single Web page or document has its central topic. This topic is described by the terms with high co-relation. These terms occur more often in the relevant Web pages than the irrelevant Web pages. For instance, Web pages about *education* will have many more terms like school, department, courses, and exams than Web pages about other topics do. Web pages about *health* will contain more terms like disease, treatment, dose and etc. Taking advantage of the similar language usage on a topic, we can easily distinguish the topic of a Web page from that of other Web pages. Consequently we may focus our retrieval result to a user's need better than the standard LMs.

To present a topic precisely, a topical model $P(t|T)$ can be built on the characteristic topical terms instead of general terms. The topical terms should be highly specific to that topic. In regard to this concern, parsimonious LM can facilitate us to compute topical models using a set of known relevant documents for a topical category. During retrieval, a query topic is classified into the defined topical categories and only the corresponding topical model will be used.

3.4 Combined Topic-based Model

There are many LMs that compute the ranking score for a Web page, such as the standard LM and the parsimonious LM. They only use information of the term distribution in a user query, a document, and a background document collection. The topical model emphasizes more on the typical language usage on a topic. We hope to integrate the additional information to the known models for improving topic-based search. Inspired by the relevance model [12, 14, 16], we can use cross-entropy to measure the information gain between models. The information gain can be calculated by the cross-entropy of document model and query model. The cross-entropy score is high when two models differ from each other. Otherwise, it is low. This divergence computation is not symmetric.

In Web IR, we like to know if the cross-entropy score is low between the model of a Web page and the model of a given query. Naturally the first way is to compute the divergence between an estimated query model $P'(t_i|Q)$ and an estimated document model $P'(t_i|D)$. The document model $P(t_i|D)$ can be smoothed with the background model $P(t_i|C)$ to avoid zero-probabilities for terms not occurring in a document. The query model $P(t_i|Q)$ is expanded with the corresponding topical model $P(t_i|T)$ to represent the category of the topic of request. The weights (λ and α) of interpolation can be estimated empirically. The divergence score is summed up over the whole language vocabulary with length l. A document will have low divergence score when its probability distribution is similar to that of the query. It is thus considered most likely relevant to the given query. This topic-based model is flexible to be extended to other models. When the weight of α is 1, the model becomes a standard LM.

$$P'(t_i|D) = \lambda P(t_i|D) + (1-\lambda)P(t_i|C)$$

$$P'(t_i|Q) = \alpha P(t_i|Q) + (1-\alpha)P(t_i|T)$$

$$score(D) = \sum_{i=1}^{l} \left[P'(t_i|Q) \cdot \log(P'(t_i|D)\right] \quad (4)$$

4. USER FEEDBACK

It is very important for a search engine to understand a user's interest on topics for focusing search results on topic-based search. In our envisaged framework, retrieval is most

Table 1: Training queries and Relevant Web-pages

Topical Category	# Queries	# Rel. Pages
Health	13	2,261
Animals	9	1,844
Education	4	821
Transport	4	658
Environment	3	922
Career	3	808
Human rights	3	248
Art and culture	2	476
Nuclear energy	1	312
Space	1	285
Terrorism	1	203
Total	44	8,838

accurate when the topic-based model matches a user's topical interest. Using feedback is the natural way to confirm the understanding of a user's intent. Blind/pseudo feedback that uses the observed documents from the initial search result to build up a topical context has lead to effective IR [14, 16, 33]. The direct way to know a user's intention is the user feedback on the defined topical categories. Using this information, the user's interest on a topical category can be decided and hence the topic-based model can be selected for ranking Web pages. The most well-known topical categories are DMOZ [8], Yahoo! Directory [32], and Wikipedia [30]. Among them, ODP categories have been used for topic-based modeling [7, 17, 29]. In this paper, we decide to use our own topical categories that are derived from available data. There are explicit and implicit user feedback. The implicit user feedback can be clicking through a Web link by a Internet user. The explicit user feedback can be simulated by a user survey. In the survey, a group of test persons has to classify the given queries into the predefined topical categories. One query is allowed to be assigned to one or two categories. Test persons must come from diverse knowledge background that has connection with difference of gender, profession, geographical location, and nationality. They are not required to be computer literate or familiar with Web search. We hope the diversity can present the view of the average Internet users on topic categorization.

5. EXPERIMENTS

In this section we present detailed information on the user survey of query categorization. It will be used for testing our LMs. We describe our experiments on training topical LMs and testing the standard LM, the parsimonious LM, and the topic-based LM. Comparison among the LMs will be discussed at the end of this section based on the experimental results.

5.1 Training Topical LMs

Our topical LMs are computed on TREC Terabyte Track data, using the .GOV2 collection and the corresponding 700–850 queries. We first defined 12 topical categories including *others* rather arbitrarily upon informal inspection of the query set, and then assigned the 150 queries into one of them ourselves manually. Our query categorization is presented in Table 1. Among 150 queries, there are 46 queries being classified into their categories. The rest 104 queries do not belong to any topical categories and thus are considered as the *others* category.

Table 2: Examples of Topical Models

Education Topical Model			**Health Topical Model**		
school	58845	0.057078	cancer	59641	0.028327
students	46860	0.047065	have	40261	0.004632
that	42252	0.003344	pharmacy	37979	0.018717
with	33955	0.003836	may	37305	0.004732
education	33197	0.029556	board	36771	0.014006
program	25379	0.017923	health	36079	0.009305
not	23746	0.001185	shall	32879	0.008640
studenx	22752	0.022771	it	29912	0.000759
state	21961	0.003650	care	29347	0.010954
district	17810	0.014852	patients	27794	0.013110
schools	16987	0.016637	which	27505	0.001874
programs	15807	0.012223	has	26296	0.001942
have	14831	0.000308	research	25682	0.001676
services	13803	0.004738	these	25005	0.003302
their	12809	0.006870	any	24898	0.002416
gifted	11905	0.012500	drug	24828	0.011081

After queries are categorized, we can train our topical LMs based on their relevant Web pages. For each topical category, we can find the relevant Web pages from .GOV2 data collection using the TREC relevance judgements. For instance, there is one query belonging to the *space* category. According to the TREC relevance judgement, we know there are 285 relevant Web pages to this *space* topical category in .Gov2 collection. Applying the similar method, we can summarize the overall relevant Web pages in Table 1. For each topical category, we treat all relevant Web pages on that topic as a single document and calculate the term probability using EM-algorithm as the parsimonious model. The weight μ is set at 0.1 and the pruning threshold is 0.0001. In total, we have computed 11 topical models for our defined topical categories.

Table 2 shows the examples of the constructed *education* and *health* topical models. For each topical model, the first column lists terms appearing in the relevant Web pages, the second column lists the number of term occurrences, and the last column is the term probability in the relevant collection. The *education* topical model contains education related terms such as school, students, education, program, and etc. These terms have higher occurrences and probabilities than other terms. From the content of the topical model (see Table 2), we can easily distinguish the *education* topic from the *health* topic.

5.2 Survey

To test our topic-based model, we need testing queries that have clear and well defined topic. In reality, this is usually obtained by explicit user feedback. In our lab setting, we simplify this process by manually classifying queries into our own topical categories. To accomplish this, a user survey is conducted on the queries 401–450 of the TREC-8 Web data collection. The queries contain three fields, namely title, description, and narrative. The *title* has 2.5 terms on average. The *description* contains a complete sentence stating the topic of a query. The *narrative* gives a paragraph information about relevant and/or irrelevant context for the topic. We have 12 categories that are defined at the training

Table 3: Testing Queries and Categories

Topical Category	# Queries	% Average Votes
Health	8	0.88
Animals	3	0.67
Art and culture	5	0.52
Environmen	5	0.75
Transport	8	0.58
Human rights	2	0.56
Space	1	0.47
Education	1	0.64
Terrorism	1	0.59
Others	16	0.60
Total	50	-

stage of topical LMs, namely: health, animals, education, transport, environment, career, human rights, art and culture, nuclear energy, space, terrorism, and others. A group of test persons has to assign each of the 50 queries into one or two categories manually. The test persons are people with and/or without knowledge of information retrieval. The survey is distributed on Internet and survey results are collected by email. We summarize the received categorizations in Table 3 based on the following rules for the query classification:

- For each respondent, if a query is assigned to both *others* and another category, the query is considered to be in the latter category only. If a query is only assigned to *others* category, the query has the category of *others*.
- Overall, a query is considered to be of the most frequently assigned category. If there is a tie between at least two categories, which are not *others*, the query is randomly assigned to one of them.

We can view an assigned topical category, as a "vote" by the test person that this query belongs to this category. For each query, the percentage vote is computed as the fraction of votes for a category and the total number of votes for that topic. The largest percentage vote is the category for that query. When the percentage vote is tied between *others* and another category, we choose the latter category for the concerned query. For other kinds of ties, we break them by random choice. The average percentage votes are the average value of the percentage votes of all queries in that category. The categorization summary (see Table 3) shows that only 10 out of 12 topical categories have assigned queries and only 34 out of 50 queries can be classified into a category other than *others*. Excluding the *others* category, there are 9 categories that we can use for testing our topic-based LMs in later subsections. The larger the average percentage vote, more test persons agree on that categorization on average. It therefore indicates the confidence on that categorization. The *health* category wins the most agreement while the *space* is the least confident categorization.

5.3 Testing LMs

We test our LMs on TREC-8 ad-hoc small Web data collection. This collection contains 247,491 Web pages extracted from 969 different Web domains. For the queries 401-450, there are 2,279 relevant Web pages in total. Only

Table 4: Summary of LM Performance on Average

# queries	metric	run 1	run 2	run 3	run 4
34	p@10	0.3941	0.4882	0.4500	0.4824
	map	0.2456	0.2978	0.2578	0.2789
	bpref	0.2602	0.2956	0.2707	0.2805
50	p@10	0.3640	0.4280	0.4040	0.4260
	map	0.2331	0.2686	0.2428	0.2571
	bpref	0.2481	0.2722	0.2571	0.2637

title terms are used in our query model. Their average length is 2.5 terms. TREC evaluation program (`trec_eval`) is used to compute performance metrics for each query and the overall average performance. Three measures are taken into account, namely Mean Average Precision (MAP), Binary PREFerence (BPREF), and Precision at rank 10 (P@10). To compare the retrieval performance, we implement the following models and present the best results as follows:

- Run 1: standard LM with $\lambda = 0.9$
- Run 2: standard LM with $\lambda = 0.9$ + topic-based LM with $\lambda = 0.9$ and $\alpha = 0.4$ (equation 4)
- Run 3: parsimonious LM with $\lambda = 0.9$
- Run 4: parsimonious LM with $\lambda = 0.9$ + topic-based LM with $\lambda = 0.9$ and $\alpha = 0.4$ (equation 4)

In the experiments of standard LM (run 1) and parsimonious LM (run 3), we compute the ranking score for all Web pages. In the experiments of the topic-based LMs, we take the top 1,000 Web pages in the ranking list of the standard LM and parsimonious LM respectively and re-rank them by computing the divergence score based on two different topic-based model respectively. We assume that these top 1,000 pages retrieved by the basic models contain all, or at least a large fraction of, the relevant pages. This choice will save computation cost without losing the generality of performance for the topic-based LM that does improve the retrieval effectiveness. The topic-based models use 9 out of 13 trained topical models as only 34 test queries are able to be classified into 9 topical categories (excluding *others*). For each of the 34 categorized queries, a corresponding topical model is chosen manually for computing the ranking score.

We summarize the main performance of the 4 runs in Table 4 and Table 5. The breakdown over topic categories in Table 5 shows that the standard LM + topic-based LM (run 2) outperforms the standard LM (run 1) in terms of P@10, MAP, and BPREF except *education* and *terrorism*. On average of 34 classified queries, the standard LM + topic-based LM gains 23.9%, 21.3%, and 13.6% improvement in terms of P@10, MAP, and BPREF over the standard LM respectively. The parsimonious LM + topic-based LM (run 4) is more accurate than the original parsimonious model (run 3) for 6 topical categories. The exceptions are *human rights*, *education*, and *terrorism*.

The overall performance is shown in Table 4. On average of 34 classified queries, the parsimonious LM + topic-based LM gains 7.2%, 8.2%, and 3.6% improvement in P@10, MAP, and BPREF over the parsimonious LM respectively. Using the standard LM and the parsimonious LM to compute the ranking score for the rest of 16 queries in *others*

Table 5: Summary of LM Performance per Topical Category

run	metric	health	animal	art and culture	environment	transport	human rights	space	education	terrorism
	#queries	8	3	5	5	8	2	1	1	1
run 1	p@10	0.3875	0.3667	0.4200	0.5000	0.3875	0.4500	0.0000	0.1000	0.5000
	map	0.3215	0.2558	0.1548	0.2978	0.2679	0.1399	0.0069	0.1550	0.3013
	bpref	0.3233	0.2277	0.1876	0.3339	0.2862	0.1367	0.0150	0.0282	0.3653
run 2	p@10	0.6000	0.4333	0.5000	0.6200	0.4000	0.4500	0.2000	0.0000	0.6000
	map	0.3965	0.2714	0.2169	0.3887	0.2911	0.2247	0.0414	0.0156	0.2756
	bpref	0.3784	0.2400	0.2239	0.3789	0.3057	0.1965	0.0673	0.0368	0.3480
run 3	p@10	0.5125	0.3667	0.5200	0.5000	0.4000	0.5500	0.0000	0.1000	0.6000
	map	0.3543	0.2547	0.1850	0.3079	0.2619	0.1387	0.0066	0.0142	0.3093
	bpref	0.3546	0.2368	0.1971	0.3542	0.2709	0.1560	0.0102	0.0394	0.3714
run 4	p@10	0.6000	0.3667	0.5600	0.6200	0.4000	0.4000	0.0000	0.0000	0.6000
	map	0.3853	0.2564	0.2063	0.3727	0.2654	0.1390	0.0205	0.0113	0.3022
	bpref	0.3716	0.2374	0.2071	0.3858	0.2706	0.1442	0.0367	0.0342	0.3631

category, we can have an overview of the average performance for all 50 queries in TREC-8 Web data collection. On average for all, the topic-based models are more effective than either the standard or the parsimonious model alone. For the same runs, the additional *others* queries are mitigating the effect of the topics models. In run 2, the overall performance still increases with 17.6%, 15.2%, and 9.7% in terms of P@10, MAP, and BPREF. In run 4, the increase in performance is 5.4%, 5.9%, and 2.6% in terms of P@10, MAP, and BPREF. The performance can plausibly be further improved by using additional topical models for these *others* queries based on the IR performance by individual category (see Table 5).

Topical category wise, the topic-based models are not superior to other models in case of the *education* topic. This category has only one query that has the *title* field of *inventions* and *scientific discoveries*. Our definition to this topic is related to school, education, programs, classes, exams, etc. It is obvious that search users have different understanding on this concept or notion. Therefore, the *education* topical model differs a lot from the test *education* topic. As a result, it is not surprising that there is no improvement on IR performance for this topical category. In addition to the mismatching between topic and topical models, our topical models have inherent problem that they are trained on a small number of queries (at most 13 in case of *health* topic). The quality of a topical model depends heavily on the distribution of the training queries on its topical space. The larger coverage of the training queries in the topical space, higher quality the topical model can be trained. Our promising results indicate our topical models have fine coverage over their topical space.

6. CONCLUSIONS

This paper explores the possibility of using topical models for better Web IR. We present a topic-based model that combines topical models and other language models using cross-entropy. The empirical results show that the performance of the topic-based model can be superior to the standard LMs and the parsimonious LMs on average.

We are currently extending our research in various directions. First, we are including the use of well-known external topical categories (for instance, ODP categories). Second, we are experimenting with the automatic query classification, instead of explicit user feedback, and compare its effectiveness with standard pseudo-relevance feedback. Third, we are incorporating the classification probability into the model, and thereby allowing multiple-topic classifications of the same Web page.

Acknowledgments

This work is sponsored by the Netherlands Organization for Scientific Research (NWO), under project number 612-066-513.

7. REFERENCES

[1] Amazon, 2008. `http://amazon.com/`.

[2] Ask. The other search engine, 2008. `http://ask.com/`.

[3] L. Azzopardi, M. Girolami, and C. van Rijsbergen. Topic based language models for ad hoc information retrieval. In *IJCNN '04: Proceedings of the international joint conference on neural networks*, pages 3281–3286. IEEE, 2004.

[4] J. Bai, J.-Y. Nie, G. Cao, and H. Bouchard. Using query contexts in information retrieval. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 15–22, New York, NY, USA, 2007. ACM.

[5] A. Berger and J. Lafferty. Information retrieval as statistical translation. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 222–229, New York, NY, USA, 1999. ACM.

[6] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003. ISSN 1533-7928.

[7] P. A. Chirita, W. Nejdl, R. Paiu, and C. Kohlschütter. Using odp metadata to personalize search. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 178–185, New York, NY, USA, 2005. ACM.

[8] DMOZ. Open directory project, 2008. `http://dmoz.org/`.

[9] Google Scholar. Standing on the schoulders of giants, 2008. `http://scholar.google.com/`.

[10] Half, 2008. `http://half.ebay.com/`.

[11] D. Hiemstra and W. Kraaij. Twenty-one at trec7: Ad-hoc and cross-language track. In *Text REtrieval Conference*, pages 174–185, 1998.

[12] D. Hiemstra, S. Robertson, and H. Zaragoza. Parsimonious language models for information retrieval. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 178–185, Sheffield, United Kingdom, 2004. ACM.

[13] F. Jelinek. *Statistical methods for speech recognition*. MIT Press, Cambridge, MA, USA, 1997.

[14] J. Lafferty and C. Zhai. Document language models, query models, and risk minimization for information retrieval. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 111–119. ACM, 2001.

[15] Last FM. The social music revolution, 2008. `http://last.fm/`.

[16] V. Lavrenko and W. B. Croft. Relevance models in information retrieval. In *Language Modeling for Information Retrieval*, pages 11–56. Kluwer Academic Publishers, 2003.

[17] F. Liu, C. Yu, and W. Meng. Personalized web search by mapping user queries to categories. In *CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management*, pages 558–565, New York, NY, USA, 2002. ACM.

[18] X. Liu and W. B. Croft. Cluster-based retrieval using language models. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 186–193, New York, NY, USA, 2004. ACM.

[19] D. R. H. Miller, T. Leek, and R. M. Schwartz. A hidden markov model information retrieval system. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 214–221, New York, NY, USA, 1999. ACM.

[20] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 275–281. ACM, 1998.

[21] H. Rode and D. Hiemstra. Conceptual language models for context-aware text retrieval. In *Proceedings of the 13th Text Retrieval Conference (TREC)*. NIST Special Publications, 2005.

[22] X. Shen, B. Tan, and C. Zhai. Implicit user modeling for personalized search. In *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 824–831, New York, NY, USA, 2005. ACM.

[23] F. Song and W. B. Croft. A general language model for information retrieval. In *CIKM '99: Proceedings of the eighth international conference on Information and knowledge management*, pages 316–321. ACM, 1999.

[24] Swoogle. Semantic web search, 2008. `http://swoogle.umbc.edu/`.

[25] SWSE. Answers before links, 2008. `http://swse.deri.org/`.

[26] J. Teevan, S. T. Dumais, and E. Horvitz. Personalizing search via automated analysis of interests and activities. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 449–456, New York, NY, USA, 2005. ACM.

[27] C. J. van Rijsbergen. *Information Retrieval, 2nd edition*. Butterworths, London, 1979.

[28] Watson. Exploring the semantic web, 2008. `http://watson.kmi.open.ac.uk/WatsonWUI/`.

[29] X. Wei and W. B. Croft. Investigating retrieval performance with manually-built topic models. In *Proceedings of RIAO 2007 - 8th Conference - Large-Scale Semantic Access to Content (Text, Image, Video and Sound)*, 2006.

[30] Wikipedia. The free encyclopedia, 2008. `http://wikipedia.org/`.

[31] J. Xu and W. B. Croft. Cluster-based language models for distributed retrieval. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and developme nt in information retrieval*, pages 254–261. ACM, 1999.

[32] Yahoo! Directory, 2008. `http://search.yahoo.com/dir`.

[33] C. Zhai and J. Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *CIKM '01: Proceedings of the tenth international conference on Information and knowledge management*, pages 403–410, New York, NY, USA, 2001. ACM.