Relevance Model Revisited: With Multiple Document Representations

Ruey-Cheng Chen, Chiung-Min Tsai, and Jieh Hsiang

National Taiwan University,
1 Roosevelt Rd. Sec. 4, Taipei 106, Taiwan
cobain@turing.csie.ntu.edu.tw
cmtsai@mail.lis.ntu.edu.tw
jhsiang@ntu.edu.tw

Abstract. In this work, we extended Lavrenko's relevance model [6] and adapted it to the cases where an additional layer of document representation is appropriate. With this change, we are able to aggregate heterogeneous data sources and operate the model in different granularity levels. We demonstrated this idea with two applications. In the first task, we showed the feasibility of using a carefully-selected vocabulary as the query expansion source in a language model to enhance retrieval effectiveness. The proposed query refinement model outperformed the relevance model counterpart in terms of MAP by 17.6% under rigid relevance judgment. In the second task, we established a ranking scheme in a faceted search session to sort the facets based on their corresponding relevance to the query. The result showed that our approach improved the baseline performance by roughly 100% in terms of MAP.

Key words: Bayesian relevance model, multiple representations

1 Introduction

Over the past few years, the abundant information on the Web along with the rise of Web users has silently changed the usual shape of information retrieval in the way we model documents. A document used to be only a collection of terms, and retrieval models were acting merely based on this piece of information to direct user to the documents that were most likely what they needed. There was also a time when user-contributed tags or human annotations were not so popular or even made available to the researchers. Today, the table has turned. We have seen many successful IR applications that made use of these external resources to improve retrieval efficiency, cases where the use of metadata items and keywords, data that surrounds the documents but is usually left outside of a formal model, could benefit the research field.

Presence of augmented contents poses a new challenge to researchers in the way retrieval is operated on top of heterogeneous data sources. One intuitive way to get over this is to mix text data from various sources into one distribution, which is a commonly-used technique in the literature. Doing so, however, might

give rise to other issues since the new representation did not necessarily share the same context with the original one, making it difficult to assign weights or distribute probability masses. Recall that we have experienced the same issue when trying to mix the contents in the document title with that in the document body to create an single index; any attempt for adjusting the weight of title words might end up harming theoretical soundness.

This issue has brought our attention to the possibility of building additional layers of document representations into existing retrieval frameworks, in which we are able to aggregate information contents and to manipulate them in different granularity levels as necessary. Consider applications that could take advantage of this idea, such as facilitating cross-language information retrieval in a corpus where documents possess multiple representations over different languages [5]; receiving user queries in a general vocabulary while operating the retrieval process in another level where documents are indexed with more specific terms; or searching for named entities that are relevant to the user queries in a collection where manually-labeled, high-quality annotations were already made available at the document level.

The aforementioned thoughts have motivated our research on formal retrieval methods. In addition to the regular term set T, we look for ways to build a *secondary document representation* S into a model so as to enable simple interaction between the two layers, such as retrieving elements from one side with that of the other as the query. In mathematical terms, we want to estimate the probability $\Pr(s|t)$, given that a document d_i is indexed in two different representations T_i and S_i, where $T_i \subset T$ and $S_i \subset S$. A concrete application of this is to let the user form a query $\mathbf{q} \subset T$ and to estimate the probability $\Pr(s|\mathbf{q})$ accordingly.

To achieve our goal, in this paper we propose a generalization over the relevance models [6, 5]. The original relevance model was adapted to the cases where two individual term distributions were available in the text collection. We take a Bayesian generative approach, starting from a graphical model in which documents, terms, and hyperparameters are all explicitly specified and then break down the full-blown probabilistic network to a few equations that could easily implemented at the index level.

The rest of the work is structured as follows. First, we introduce a Bayesian generative process that involves two individual term distributions in Section 2 and show that the process leads to a generalized relevance model. Then, in Section 3, we present two applications of our model. In Section 3.1, the idea of query expansion in language modeling is revisited by considering user queries in a set of terms indexed in general vocabulary and further expanding the set by collecting relevant concepts in a more specific vocabulary; the other task we introduce in Section 3.2 is about named-entity ranking, in which we use $\Pr(s|\mathbf{q})$ to sort the named entities retrieved in a facet search session and present the entities according to their relevance to the query. In each task, the corresponding evaluation benchmark is also described. We briefly summarize the related work in Section 4. Finally, we discuss a few issues arise in the development of the work and give out concluding remarks in Section 5.

2 Model

In this model, we hypothesize the existence of the two independent multinomial distributions $\tau^{(j)}$ and $\phi^{(j)}$ associated with each document d_j; we further assume that, in each document, the terms observed in the *primary* representation T_j are drawn from the first multinomial, and those in the *secondary* representation S_j drawn from the second. The advantage of making this independent assumption can be stated in two respects. First, separation of the generation processes on both sides may lead to a cleaner framework, which makes model inference a bit easier; second, we want to highlight the fact that these two terms sets do not necessarily share the same context and further simplify the dependency relations in the resulting Bayesian network.

2.1 The Generative Framework

Figure 1 shows our proposed Bayesian generative framework. Consider the collection is composed of N documents, in which each document d_j is associated with two sets of terms T_j and S_j. We refer T_j as the primary representation and S_j the secondary representation. Each document d_j possesses two multinomial distributions, denoted by $\tau^{(j)}$ and $\phi^{(j)}$, from which elements in T_j and S_j are drawn independently, respectively. The two multinomials for a document could also be understood as two independent document models $p(w|\tau^{(j)})$ and $p(w|\phi^{(j)})$ that work on two different sets of vocabularies. Two Dirichlet distributions $\{\alpha_i; i \in [1, |T|]\}$ and $\{\beta_k; k \in [1, |S|]\}$ are assumed in the model to govern the generation of these multinomials.

As suggested in the upper-half of Figure 1, the input query $\mathbf{q}$ is modeled as a set of observed terms drawn from an unknown document model x in the collection. Specifically, we assume that the query terms and the primary representation of document x are encoded in the same vocabulary. By making this assumption, we are able to match the query against all the document models and to estimate the query likelihood $\Pr(\mathbf{q}|d_j)$. This leads us to probabilistic generative methods, such as language modeling. Recall that the unknown document x is connected to another unknown variable y. The variable y represents the most probable term generated from the secondary multinomial distribution $\phi^{(x)}$.

The entire generative process can be summarized as follows:

1. For each document d_j,
   (a) $\tau^{(j)} \sim \text{Dirichlet}(\alpha_i; i \in [1, |T|])$
   (b) $\phi^{(j)} \sim \text{Dirichlet}(\beta_k; i \in [1, |S|])$
   (c) For $i \in \{1, \ldots, |T_j|\}$, $t_i \sim \text{Mult}(\tau^{(j)})$
   (d) For $k \in \{1, \ldots, |S_j|\}$, $s_k \sim \text{Mult}(\phi^{(j)})$
2. For some unknown document d_x,
   (a) For $i \in \{1, \ldots, |\mathbf{q}|\}$, $q_i \sim \text{Mult}(\tau^{(x)})$

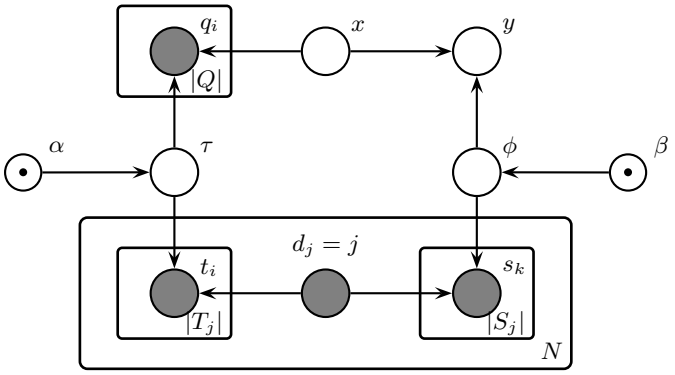

Fig. 1. The document-centric generative model is shown in the plate notation, in which shaded nodes denote observed variables. The document d_j generates two representations T_j and S_j, where each term in T_j is represented as t_i and that in S_j represented as s_j. The query terms are represented as q_i's, which are drawn from an unknown document model x; the document is associated with an unknown term $y \in S$.

2.2 Inference

The discussions have brought us to the focus of this work, which is to explicitly estimate the likelihood of one term s in the secondary term domain being triggered[1] by another set of terms in the primary term domain. This task can be framed as an optimization problem:

$$y^* = \arg\max_{y \in S} \Pr(y|\mathbf{q}, \mathbf{t}, \mathbf{d}, \mathbf{s}) \tag{1}$$

where $\mathbf{t}$ and $\mathbf{s}$ represent all the observed terms in the primary/secondary representations in the collection, respectively; $\mathbf{q}$ represents the input query and $\mathbf{d}$ denotes all the observed documents. The right-hand-side of Equation (1) can be broken down into the following form:

$$\begin{aligned}
\Pr(y|\mathbf{q}, \mathbf{t}, \mathbf{d}, \mathbf{s}) &= \sum_{x \in [1,N]} \Pr(y|x, \mathbf{q}, \mathbf{t}, \mathbf{d}, \mathbf{s}) \Pr(x|\mathbf{q}, \mathbf{t}, \mathbf{d}, \mathbf{s}) \\
&\propto \sum_{x \in [1,N]} \Big\{ \int \Pr(y|x, \phi^{(x)}) \Pr(\phi^{(x)}|d_x, \mathbf{s}_x) \mathrm{d}\phi^{(x)} \\
&\quad \int \Pr(\mathbf{q}|x, \tau^{(x)}) \Pr(\tau^{(x)}|d_x, \mathbf{t}_x) \mathrm{d}\tau^{(x)} \Pr(x) \Big\} \\
&= \sum_{x \in [1,N]} \frac{\beta_y + c_{y,x}}{\sum_k \beta_k + c_{k,x}} \frac{\mathcal{B}(\{\alpha_i + c_{i,x} + c_{i,q}\})}{\mathcal{B}(\{\alpha_i + c_{i,x}\})} \Pr(x)
\end{aligned} \tag{2}$$

[1] We coin the term due to the lack of obvious dependency between both sides.

The last line follows the conjugacy of Dirichlet distribution, as in:

$$\Pr(\phi^{(x)}|x, y, d_x, \mathbf{s}_x) \sim \text{Dirichlet}(\{\beta_k + c_{k,x}; \forall k\})$$
$$\Pr(\tau^{(x)}|x, \mathbf{q}, d_x, \mathbf{t}_x) \sim \text{Dirichlet}(\{\alpha_i + c_{i,x}; \forall i\}).$$

Recall that $\mathcal{B}(\cdot)$ is the multinomial beta function defined by:

$$\mathcal{B}(a_1, \ldots, a_n) = \frac{\prod_i \Gamma(a_i)}{\Gamma(\sum_i a_i)}.$$

2.3 Hyperparameters

There are two particular prior distributions that we want to study in this work. Consider a prior as a vector of size $|S|$: $(\beta_1, \beta_2, \ldots, \beta_{|S|})$. The first one is the *uniform* prior:

$$(c, c, \ldots, c),$$

in which we let $\beta_k = c$ for each $k \in [1, |S|]$ and c denote some constant. The second one is called *smoothed-Dirichlet*, which is widely-used as a smoothing method in the language modeling applications for information retrieval [11]:

$$(\mu \Pr(s_1|C), \mu \Pr(s_2|C), \ldots, \mu \Pr(s_{|S|}|C)),$$

where $\Pr(s_k|C)$ denotes the probability of generating term s_k (in the secondary term domain) from the collection by viewing the entire collection as one language model; μ represents some constant.

2.4 Computational Efficiency

One interesting feature of our model is that it offers seamless integration with the language-model-based retrieval method. It comes from the fact that we could further simplify Equation (2) by assigning α as a smoothed-Dirichlet prior and $\Pr(x)$ as a uniform prior. When $\alpha_i = \mu \Pr(t_i|C)$, it can be shown that:

$$\frac{\mathcal{B}(\{\alpha_i + c_{i,x} + c_{i,q}\})}{\mathcal{B}(\{\alpha_i + c_{i,x}\})} \propto \mathrm{L}_{\mathrm{LM}}(\mathbf{q}|d),$$

where $\mathrm{L}_{\mathrm{LM}}(\mathbf{q}|d)$ denotes the query likelihood score in the language model with Dirichlet smoothing scheme. In this case, the model scores becomes:

$$\Pr(y|\mathbf{q}, \mathbf{t}, \mathbf{d}, \mathbf{s}) \propto \sum_{x \in [1,N]} \frac{\beta_y + c_{y,x}}{\sum_k \beta_k + c_{k,x}} \mathrm{L}_{\mathrm{LM}}(\mathbf{q}|x) \tag{3}$$

It can be shown that the expected complexity for computing the summation is linear to the size of top-ranked documents, namely n, and the average size of secondary domain terms associated with each document, i.e., $(\sum_j |S_j|)/N$.

3 Experimental Results

In this section, we introduce two applications of our proposed framework. In the first experiment, we use a different set of terms to refine the query model so as to leverage retrieval effectiveness; in the second experiment, we showed that we were able to retrieve the named entities relevant to the user query from a database by supplying the entities as the secondary document representation. Both experiments were conducted in a general way so as to be reproduced by any follow-up research.

3.1 Query Refinement Using the Secondary Representation

The probability estimate $\Pr(y|\mathbf{q}, \mathbf{t}, \mathbf{d}, \mathbf{s})$ that we obtained from the proposed model can be seen as a *refined query model*. In other words, we employ a two-stage retrieval strategy by launching an initial retrieval run to populate a refined model, and then re-submit the model as a new query to the retrieval system. Our idea here differs from the previous efforts [6] in the use of a more specific vocabulary as the secondary representation. The major obstacle, however, for realizing such type of model-based query refinement lies in the restriction of the language modeling framework, according to which the input query should be specified as a sequence of terms. We used a technique that enabled a second-round retrieval by *realizing* the probability distribution $\Pr(y|\mathbf{q}, \mathbf{t}, \mathbf{d}, \mathbf{s})$ as a sequence of *expected terms*, in which each term s_k appears exactly $\mathrm{E}(s_k|\mathbf{q}, \mathbf{t}, \mathbf{d}, \mathbf{s})$ times. It can be shown that the technique reduces to the KL-divergence retrieval framework [10].

The benchmark corpora that we used here was the NTCIR-4 Test Collection. We considered a subtask of the NTCIR-4 CLIR Task [4]: Chinese-to-Chinese monolingual document retrieval. The dataset comprised of totally 381,681 newswire articles and 59 test topics. Only title queries were used in this experiment. Limited preprocessing steps were applied to the corpus in advance. All the punctuation marks, whitesapces, and separators were discarded and the remaining texts were send to a simple tokenizer. As a result, the output was a mixture of CJK-character bigrams and English word unigrams, which formed the primary document representations in our model and in other baseline methods. The secondary representation we chose was the top-500,000 CJK-character bigrams in the entire collection ranked by the corresponding *tf-idf* score; the candidate set was formed by excluding those bigrams occurred less than 5 times. The parameters described here were determined through experimentation.

Our model was developed from scratch in C++, and the rest of the experimental runs were established on top of the Lemur toolkit[2]. From all the retrieval methods supported by the tool, we selected `tfidf`, `okapi`, and `indri` (a language-model-based method) as the baseline methods; our own language model implementation is denoted as `lm`. Standard query expansion techniques for each method were also tested: Pseudo-relevance feedback (`PRF`) was implemented for both `tfidf` and `okapi` models, and an adaptation of Lavrenko's

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

relevance model (RM) was also built into the core of Indri index; our refinement model is denoted as BRM in this experiment.

We stuck with the default parameter values for all the participant models in the regular runs; for standard query expansion, we set the number of feedback documents as 20 and the number of feedback terms as 100. The refined query was computed under the same restriction in BRM, where terms other than the top-ranked 100's were completely ignored. Recall that our model reviewed all the retrieved top-1000 documents to achieve the probability estimates. The performance is measured in terms of mean average precision and precision-at-5.

Table 1. The overall performance result is summarized in this table. Regular retrieval runs are listed in the upper-half and expansion runs in the lower-half. Boldfaced values indicate the best-performers. Evaluation results for using the rigid and the relax judgment sets are both listed. Note that improvement (+%) for each expansion run is calculated against the performance of the corresponding regular run.

	Rigid Judgment				Relax Judgment			
Regular	MAP	(+%)	P@5	(+%)	MAP	(+%)	P@5	(+%)
tfidf	0.181		0.264		0.213		0.335	
okapi	0.185		0.278		0.223		0.356	
indri	0.174		0.258		0.216		0.346	
lm	0.170		0.251		0.209		0.322	
Expansion								
tfidf+PRF	0.217	(+19.9%)	0.295	(+11.7%)	0.264	(+23.9%)	0.383	(+14.3%)
okapi+PRF	**0.224**	(+21.1%)	**0.315**	(+13.3%)	**0.270**	(+21.1%)	**0.400**	(+12.4%)
indri+RM	0.180	(+3.4%)	0.271	(+5.3%)	0.222	(+2.8%)	0.342	(-1.2%)
lm+BRM	0.207	(+21.8%)	0.302	(+20.3%)	0.261	(+24.9%)	0.369	(+14.6%)

Table 1 shows the experimental result. Among all the regular runs, our language modeling implementation (lm) achieved the lowest MAP under both judgment sets; the retrieval performance was greatly improved when the refinement model (lm+BRM) was applied, achieving 0.207 for rigid relevance judgment and 0.261 for relax in terms of MAP. It turned out that the proposed query refinement model greatly enhanced the performance of relevance model counterpart in Indri (indri+RM) by 17.6% in terms of MAP. Since our proposed model can be viewed as a relaxed (or adapted) version of the original relevance model, this encouraging result partly confirmed our hypothesis that bad expansion terms could be avoided by biasing the expansion source toward a secondary document representation that possesses a cleaner, highly-discriminative vocabulary.

The overall best performance, however, still fell on the tfidf-side. The expansion run okapi+PRF achieved 0.224 under rigid and 0.270 under relax judgment set in terms of MAP. Generally, the performance for all the expansion runs can be summarized as: $\texttt{okapi} + \texttt{PRF} > \texttt{tfidf} + \texttt{PRF} > \texttt{lm} + \texttt{BRM} \gg \texttt{indri} + \texttt{RM}$. As we can see in Table 1, even though the proposed method achieved roughly comparable performance as tfidf+PRF, the difference between the performance of

tfidf-based methods and that of language-model-based methods remains rather significant.

Table 2. The performance results of all the methods (except the proposed one) using only the top-500,000 bigram representation. Boldfaced values indicate the best performers.

	Rigid Judgment		Relax Judgment	
Method	MAP	P@5	MAP	P@5
`tfidf+PRF.500k`	0.190	0.288	0.241	0.373
`okapi+PRF.500k`	0.196	**0.302**	0.247	**0.383**
`indri+RM.500k`	0.155	0.248	0.197	0.309
(*The above runs were indexed against top-50k bigrams only*)				
`lm+BRM`	**0.207**	**0.302**	**0.261**	0.369
(*This run was operated on both representations*)				

One thing to note is that readers might argue that the performance improvement of our model was largely contributed by using the terms in the secondary representation as expansion source. Table 2 summarizes the performance results for another round of experiments conducted on the same corpus where only the secondary representation (i.e., top-500,000 bigrams) was available. The result shows that using the secondary representation alone was not enough for achieving high retrieval effectiveness. The performance for all the participating runs except our proposed method decreased due to the choice of the document representation. Note that we do not intend to claim superiority of our method over the other runs, since our model possessed better knowledge about the complete document distribution and making comparison against other approaches in this case would not be appropriate.

3.2 Retrieving Query-Relevant Facets

Faceted search [3, 8, 7] can be seen as a two-phase extension of the ordinary retrieval task: In the first phase, one populates a set of documents (the *result set*) by querying the underlying retrieval model; then, in the second phase, one exploits the associations between the facet entries and the documents, and returns the facet items back to the users. The collected facets are usually presented in descending order of the corresponding number of occurrences in the result set, which we called the *count-based scheme*. The scheme is mostly preferable because it is fast and easy to implement. However, it treats all the collected facet counts equally important and disregards the differences on the degree of relevance between the retrieved documents. In other words, a facet found in a low-rank document might appear as good as that found in a top-rank one when this simple ranking scheme is employed.

This can also be a problem when the user wants to consult the retrieved facets to guide the subsequent exploratory search. Since the facets are not sorted in

terms of relevance, the user might be mislead by top-ranked facets that are not actually relevant to the input query but merely possess more counts in the lower-ranked documents. We argue that a reasonable facet ranking algorithm should always assign higher weights to the facets discovered from the top-ranked documents; therefore, the algorithm itself may need to co-operate with the retrieval model for obtaining better facet ranking. This observation has brought us to a simple solution: We can build in the facets associated with each document as a secondary representation and utilize Equation 2 to rank the facets.

The very nature of this task makes our proposed model feasible. One major difficulty that we faced, however, was the lack of standard benchmark on the facet ranking task. As a result, we went for a custom benchmark based on one text collection in our own applications. The test corpus that we chose was the Tan-Hsin Archive[3], which is currently the most sizable and complete collection of historical documents in Taiwan dating back from 1776 to 1895. We took a subset of 15,314 documents from the collection to form the test set. Each document in the collection is associated with a set of human-annotated facets. By considering topic coverage and ease of evaluation, we tested model performance only on person names as the designated facet set. The number of unique person names in the dataset is 10,918, and the total number of occurrences of these facets is 37,489. Thus, the average number of person names associated with a document is 2.45. All these information were already made available in the metadata of the documents.

To prepare the query topics, we first fetched all the query log entries from the database server. We put together a set of 105 query topics, each of which had obtained more than 50 hits in our database. We randomly chose 28 topics out from the set and used them in the following evaluation. The relevance judgment was made manually. For each query topics, we prepared a list of the top 100 facets returned by the baseline retriever that operated merely based on occurrence counts[4]; the domain expert would then go over the entire list, labeling each facet as relevant or irrelevant accordingly. We considered 3 competing runs in the experiment: (i) `baseline`, with which facets were ranked by the corresponding number of counts in the result set; (ii) `uniform-prior`, with which facets were ranked by the proposed model using uniform prior for β; and (iii) `smoothed-prior`, with which facets were ranked by the proposed model using Dirichlet-smoothed prior for β. Note that the language-model-based approach proposed in [2] was partly involved in this experiment. The method is roughly comparable[5] to a special case of the `uniform-prior` method for $\beta = 0$.

We took the top-1000 documents retrieved by using Dirichlet-smoothed language model with $\mu = 2,500$ as the result set. The hyperparameter was pre-

[3] `http://www.lib.ntu.edu.tw/project/en/index.htm`

[4] Relevance judgment created this way might greatly favor the baseline method, as noted by the friendly anonymous reviewer. A better way to do this is through pooling. As of writing, we are not able to revise the benchmark; however, the biases result is still able to demonstrate the effectiveness of the proposed method.

[5] The equivalence takes place only when the mixture component λ_{CA} is set to 0.

determined through experimentation. All the algorithms took the same set of document IDs as input. For our proposed methods, we reused the retrieved scores returned by the document-level retrieval model, as stated in the preceding sections. The output was a sorted list of all the associated facets in descending order of the ranking score. For efficiency, we used only the top-100 facets returned by each run as the final results. The performance was assessed both in terms of mean-average precision and precision-at-10.

Table 3. The performance results for all the test runs. The top performers are shown in boldface.

Method		MAP	P@10
`baseline`		0.2996	0.3857
`uniform-prior`	$\beta = 0$	0.5775	0.6179
	$\beta = 1$	**0.5899**	**0.6464**
	$\beta = 10$	0.5877	**0.6464**
`smoothed-prior`	$\mu = 0.01$	0.5781	0.6179
	$\mu = 1$	0.5782	0.6214
	$\mu = 100$	0.4956	0.5429

The performance of the baseline was quite moderate, with MAP a bit less than 0.3 and P@10 around 0.39. All the other models showed significant improvement over the baseline, ranging from 30% to 100% in terms of MAP. From the result, we find that the performance of our best model (`uniform-prior` with $\beta = 1$) was slightly better than that of the language-model-based counterpart (`uniform-prior` with $\beta = 0$), though only to a limited extent, by 2.15% in terms of MAP. Out of curiosity, we also ran a line search for each model by varying the hyperparameter β. We found that model performances were quite stable across different experimental runs. Generally speaking, `uniform-prior` and `smoothed-prior` achieved comparable performance. The best one was found in the `uniform-prior` run with $\beta = 0.2$, achieving 0.591 in MAP. The positive observation in this experiment suggested that a relevance-based model is probably more effective than a count-based model in the facet ranking task.

4 Related Work

Our proposed framework was greatly inspired by the recent advances on language modeling applications in information retrieval, including relevance models [6, 5], model-based feedback [10], and Bayesian language model [9]. In this respect, our solution can be seen as a two-stage Bayesian extension of the regular language model over the facet data. Our approach differs from the previous efforts in the formal definition of a secondary document representation and a corresponding generative Bayesian model. Specifically, our work departs from the original relevance model [6] and its cross-lingual counterpart [5] in the way Dirichlet priors are associated with the both representations.

To the best of our knowledge, this work is among the earliest attempt for retrieving relevant facets in the digital library community. Interestingly, we learned that similar attempts were made in the area of expert finding. In [1], Amitay et al. formulated the problem of expert finding as a two-layered retrieval task and proposed a solution based on the *inverse entity-frequency* that achieved moderate performance on the task. Balog et al. [2] followed up in the same direction by proposing a language modeling framework with the similar rationale to extend the use of language model to the underlying co-occurrence statistics by exploiting person-term and person-document links. Our contribution departs from these two previous efforts, not only in the application domain, but in the way document relevance is incorporated into the model and the support for prior belief.

5 Discussion and Concluding Remarks

In this work, we propose a Bayesian framework that enables the use of a secondary document representation in language modeling. The framework extends Lavrenko's relevance model to offer interoperability between two different term domains. Moreover, we show that our method is capable of working with a language-model-based retrieval engine to achieve high efficiency in computation.

Two applications are introduced in this paper to evaluate the performance of the proposed solution. In the first task, we show that the presence of a secondary document layer that is made of a carefully-selected vocabulary could greatly enhance retrieval effectiveness. Even though the proposed query refinement model did not achieve the best performance, the model still beaten the relevance model baseline by 17.6% (rigid) and 22.7% (relax) in term of MAP and gained comparable performance as `tfidf` with pseudo-relevance feedback. The refinement model alone improved the MAP of the regular language modeling run by 21.8% (rigid) and 24.9% (relax). In the second application, where the model was used to rank named entities returned in the query session by the underlying retrieval system, the proposed solution achieved encouraging result on the custom benchmark. Our approach outperformed the baseline and the language-model-based approach by roughly 100% and 2.15%, respectively, in terms of MAP.

Despite the early success in the evaluation results, there is still room for improvements. We see our contributions here as a starting point toward further exploration on several issues that have not yet been covered in this study: the use of different prior families, the formal inference model for the hyperparameters, potential applications on the other datasets, etc. These challenges should be the focus of our future work.

Acknowledgments

We thank Po-Yu Chen and the staff in Special Collection Department, National Taiwan University Library for their support on preparation of the test data. The research efforts described in this paper are supported under the National

Taiwan University Digital Archives Project (Project No. NSC-98-2631-H-002-005), which is sponsored by National Science Council, Taiwan.

References

1. Amitay, E., Carmel, D., Golbandi, N., HarEl, N., Ofek-Koifman, S., Yogev, S.: Finding people and documents, using web 2.0 data. In: Proceedings of the SIGIR 2008 Workshop on Future Challenges in Expertise Retrieval (fCHER). pp. 1–6 (2009)
2. Balog, K., Azzopardi, L., de Rijke, M.: A language modeling framework for expert finding. Inf. Process. Manage. 45(1), 1–19 (January 2009)
3. Hearst, M., Elliott, A., English, J., Sinha, R., Swearingen, K., Yee, K.P.: Finding the flow in web site search. Commun. ACM 45(9), 42–49 (September 2002)
4. Kishida, K., Chen, K., Lee, S., Kuriyama, K., Kando, N., Chen, H., Myaeng, S., Eguchi, K.: Overview of CLIR task at the fourth NTCIR workshop. In: Proceedings of NTCIR. vol. 4, pp. 1–38 (2004)
5. Lavrenko, V., Choquette, M., Croft, W.B.: Cross-lingual relevance models. In: SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 175–182. ACM, New York, NY, USA (2002)
6. Lavrenko, V., Croft, W.B.: Relevance based language models. In: SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 120–127. ACM, New York, NY, USA (2001)
7. Roy, S.B., Wang, H., Das, G., Nambiar, U., Mohania, M.: Minimum-effort driven dynamic faceted search in structured databases. In: CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management. pp. 13–22. ACM, New York, NY, USA (2008)
8. Yee, K.P., Swearingen, K., Li, K., Hearst, M.: Faceted metadata for image search and browsing. In: CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems. pp. 401–408. ACM, New York, NY, USA (2003)
9. Zaragoza, H., Hiemstra, D., Tipping, M.: Bayesian extension to the language model for ad hoc information retrieval. In: SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval. pp. 4–9. ACM, New York, NY, USA (2003)
10. Zhai, C., Lafferty, J.: 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. pp. 403–410. ACM, New York, NY, USA (2001)
11. Zhai, C., Lafferty, J.: A study of smoothing methods for language models applied to information retrieval. ACM Trans. Inf. Syst. 22(2), 179–214 (April 2004)