A Field Relevance Model for Structured Document Retrieval

Jin Young Kim and W. Bruce Croft

Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts Amherst
{jykim,croft}@cs.umass.edu

Abstract. Many search applications involve documents with structure or fields. Since query terms often are related to specific structural components, mapping queries to fields and assigning weights to those fields is critical for retrieval effectiveness. Although several field-based retrieval models have been developed, there has not been a formal justification of field weighting.

In this work, we aim to improve the field weighting for structured document retrieval. We first introduce the notion of field relevance as the generalization of field weights, and discuss how it can be estimated using relevant documents, which effectively implements relevance feedback for field weighting. We then propose a framework for estimating field relevance based on the combination of several sources. Evaluation on several structured document collections show that field weighting based on the suggested framework improves retrieval effectiveness significantly.

1 Introduction

A recent trend toward web-based computing has led to increasingly more documents having structure with several components (or fields). Exploiting this structure is a significant challenge as part of improving retrieval effectiveness on such collections. From a ranking perspective, this typically involves modeling each document as a set of fields, and scoring documents by combining field-level evidences appropriately.

Several field-based retrieval models have been developed, including BM25F [14], a Mixture of Field Language Models (MFLM) [11], and a Probabilistic Retrieval Model for Semi-structured data (PRM-S) [5]. While these models are based on different retrieval paradigms and assumptions, one common component is the weights assigned to each field.

For the BM25F and MFLM models, field weights are fixed across different query terms and estimated based on held-out training queries. In the PRM-S model, field weights are assumed to be different for each query term, and are estimated using probabilistic classification based on collection statistics. While the estimation of field weights in PRM-S was shown to be relatively effective in previous evaluation [5, 4], the use of classification for field weighting is not theoretically grounded, and the estimation is based on a limited source.

In this work, we aim to provide an interpretation of field weighting from the relevance perspective, and improve the estimation of per-term field weights. We first introduce the notion of field relevance, which is defined as the distribution of per-term user intent over document fields. In our proposed retrieval model, the field relevance model, field relevance estimates for each query term are used as per-term and per-field weights. We then describe how field relevance can be estimated using relevant documents, which enables relevance feedback for structured document retrieval.

To improve the quality of field relevance estimation in practical scenarios, we introduce a framework for estimating per-term field weights based on the combination of several sources. We use several new sources including both the unigram and bigram language model of the top-k retrieved documents, as well as the field weight estimates from previous retrieval models. We performed experiments on several structured document collections with different characteristics. The results show that field relevance estimates based on the proposed framework can improve retrieval performance significantly, and that the proposed techniques produce a notable improvement in the estimation of field relevance.

The rest of this paper is organized as follows. In Section 2, we review field-based retrieval models and field weighting techniques suggested in previous work. In Section 3 and 4, we introduce the concept of field relevance and how we can incorporate relevance feedback for field weighting, followed by a framework for field relevance estimation using a combination of several sources. We then present retrieval results using several collections, including a TREC email collection, in Section 5. In Section 6, we provide an overview of related work, and we conclude the paper in Section 7.

2 Retrieval Models for Structured Documents

In this section, we review existing retrieval models for structured documents, and the techniques employed for field weighting in these models.

The following notations will be used throughout this paper. We assume that a query $Q = (q_1, ..., q_m)$ is composed of m words, and the collection C contains n field types $F = (F_1, ..., F_n)$, and that every document D in the collection is composed of fields $(F_1, ..., F_n)$. We also denote per-field weights as $(w_1, ..., w_n)$.

2.1 Existing Retrieval Models

BM25F Robertson et al. [14] introduced BM25F as a modification of the BM25 model where field-level evidence is combined at the raw frequency level. The BM25F score $BM25F(Q, D)$ is calculated as:

$$BM25F(Q, D) = \sum_{q_i \in Q} idf(q_i) \frac{Score(q_i, D)}{k_1 + Score(q_i, D)} \tag{1}$$

where term score $Score(q_i, D)$ is calculated as:

$$Score(q_i, D) = \sum_{F_j \in D} \frac{w_j tf(q_i, F_j, D)}{(1 - b_j) + b_j \frac{length(F_j, D)}{length(F_j, C)}} \tag{2}$$

Here, *idf* indicates global inverse document frequency, *tf* and *length* denotes per-field term frequency and length, respectively. Also, field weight parameter w_j is used to combine field-level frequency into document-level frequency, and another field-level parameter b_j controls the degree of length normalization. Robertson et al. [14] suggests training w_j and b_j based on held-out queries.

Mixture of Field Language Models Ogilvie et al. [11] suggested a mixture of field language models by linear interpolation (MFLM) for known-item search in structured document collections. A document score in the MFLM is calculated by taking the weighted average of field-level query-likelihood scores as follows:

$$P(Q|D) = \prod_{i=1}^{m} \sum_{j=1}^{n} w_j P(q_i|F_j, D) \tag{3}$$

MFLM also has per-field weights w_j, which are estimated based on maximizing retrieval performance in training queries, similarly to BM25F.

Probabilistic Retrieval Model for Semi-structured Data Kim and Croft [5] recently introduced the probabilistic retrieval model for semi-structured data (PRM-S). PRM-S employs a probabilistic mapping $P_M(F_j|q_i)$ between query terms and document fields, which is calculated by probabilistic classification of a given query term q_i into the field F_j:

$$P_M(F_j|q_i, C) = \frac{P(q_i|F_j, C)}{\sum_{F_k \in F} P(q_i|F_k, C)} \tag{4}$$

As a retrieval model, PRM-S is an extension of MFLM in that it also combines field-level query-likelihood scores into a document score. In contrast to BM25F and MFLM where field weights do not vary across query terms, however, PRM-S employs the mapping probability as per-term and per-field weights:

$$P(Q|D) = \prod_{i=1}^{m} \sum_{j=1}^{n} P_M(F_j|q_i) P(q_i|F_j, D) \tag{5}$$

3 Field Relevance Model

Previous retrieval models discussed so far used several sources to estimate field weights. However, there has not been a natural way to incorporate relevance feedback for the estimation. In other words, even when a set of relevant documents were known, it was not easy to exploit this information for retrieval.

In this section, we introduce the notion of field relevance and a corresponding retrieval model (the field relevance model). We then investigate how field relevance can be estimated either when relevant documents are known (relevance feedback) or not (pseudo-relevance feedback).

3.1 Field Weighting as Field Relevance

The notion of relevance is central to the field of information retrieval, yet the multi-faceted nature of relevance led to many definitions and controversies. Although there has been numerous efforts [6, 7, 15] to model and incorporate the relevance in a retrieval model, most have focused on modeling relevance in non-structured document collections.

In structured document retrieval, however, the fields within each document encode different aspects of information, and we can also find implicit structure within a user's keyword query. Given the structure found in both queries and documents, we can argue that the degree of topical relevance depends on the matching of the structure as well as terms.

As an illustration, consider a query 'james meeting 2011' issued for an email collection. Assume that a user formulated this query based on the memory of an email whose *sender* is 'james', whose *subject* and *body* fields include 'meeting', and that has the term '2011' in *date* field. A query term may have matches in the fields that user did not intend, (e.g., 'james' can be found in *body* field), but the term scores from such fields should be considered less important since those do not match with the user's structural intent.

Since traditional models of relevance feedback focused on adjusting query-term weights, they cannot capture this variation in relevance with respect to the matching between structural components of a document and a query. To overcome this limitation, it is necessary that structural components of a collection be considered in modeling relevance. We now formally define field relevance and the corresponding retrieval model in the context of a keyword query.

Field Relevance Given a query $Q = (q_1, ..., q_m)$, *field relevance* $P(F_j|q_i, R)$ is the *distribution* of *per-term* relevance over document fields.

Field Relevance Model Based on field relevance estimates $P(F_j|q_i, R)$, the *field relevance model* combines field-level scores $P(q_i|F_j, D)$ for each document using field relevance as weights.

$$Score(D, Q) = \prod_{i=1}^{m} \sum_{j=1}^{n} \hat{P}(F_j|q_i, R) P(q_i|F_j, D) \qquad (6)$$

From users' perspective, field relevance can be regarded as their per-term query intent over document fields. Alternatively, we can interpret field relevance as the generalization of field weighting components found in existing retrieval models. It is dependent on both word and document fields, unlike the per-field weights of BM25F and MFLM.

The field relevance as defined above looks similar to the mapping probability $P_M(F_j|q_i)$ in PRM-S. However, while the mapping probability is conceptually based on a classification framework, it is based on the notion of relevance, and we argue that this opens up new possibilities for estimation. In Section 4, we also show how the mapping probability can be incorporated to improve the estimation of field relevance.

3.2 Estimating Field Relevance by Relevance Feedback

Based on this notion of field relevance, we discuss how relevance feedback can be incorporated into the existing structured document retrieval framework. Ideally, if we assume the knowledge of relevant documents, we can directly use the language model of relevant documents to estimate field weights.

$$P(q_i|F_j, R) := P(q_i|F_j, D_R) \tag{7}$$

In other words, the term distribution of known relevant documents across different fields indicates the relevance of each field for given query-term. Going back to our earlier example on the query 'james meeting 2011', if we knew in which fields in the relevant email the query terms are located, we could easily identify relevant fields for each query term. As this is based on information not available in many cases, we call this the 'oracle' field weight estimate in what follows.

Since we use the field relevance as field weights in our retrieval model, this allows *true* relevance feedback in field weighting — the knowledge of relevant documents can be naturally incorporated into the estimation. This suggests the possibility to improve retrieval effectiveness if a user is willing to provide relevance judgments.

In more practical scenarios, where relevance judgments are not available, we need to find alternative sources by which we can approximate the field-level term distribution of relevant documents. One way is to use the top-k retrieved documents as an approximation of relevant documents, as was done in previous work [7]. In this work, to improve the quality of estimation further, we combine this with other sources of estimation.

4 Estimating Field Relevance by Combining Sources

In the previous section, we introduced the notion of field relevance as the generalization of field weights, and described how we can estimate field relevance when relevant documents are known. In practice, field relevance needs to be estimated based on the information available without the knowledge of relevant documents. Given the size of the parameter space, however, it is challenging to estimate the value per field and query term.

To address this concern, we introduce a learning framework where field relevance can be estimated based on the combination of several sources. Since each source gives the distribution of field relevance for each query term, we have only as many parameters as the number of sources.

Here we introduce our estimation framework more formally. We first define the field relevance estimate $\hat{P}(F_j|q_i, R)$ as a linear combination of several sources. Here, $\Lambda = (\lambda_1, ...\lambda_k)$ denotes weights used for the mixture.

$$\hat{P}(F_j|q_i, R) = \sum_{k=1}^{p} \lambda_k P_k(F_j|q_i) \tag{8}$$

In what follows, we present our framework for estimating field relevance based on the combination of several sources. We first introduce the sources we employed, and then how effective mixture weights Λ can be found to combine these sources.

4.1 Sources of Estimation

Collection Language Model As introduced in Section 2.1, PRM-S estimates per-field and per-term weights based on collection statistics. This is a reasonable choice assuming that the field-level term distribution of relevant documents will be similar to that of the collection. It also explains the empirical effectiveness of PRM-S [5, 4].

In our framework, we incorporated as a source the likelihood of observing a query term q_i in the unigram field language model of the collection.

$$P(F_j|q_i, C) = \frac{P(q_i|F_j, C)}{\sum_{F_k \in F} P(q_i|F_k, C)} \tag{9}$$

While this unigram language model was shown to be effective in previous evaluation with PRM-S [5, 4], it is limited in that it ignores the dependency between query terms. To address this problem, we use a field-level bigram language model whose probability is dependent on the previous query term as well as the current query term.

$$P(F_j|q_i, q_{i-1}, C) = \frac{P(q_i, q_{i-1}|F_j, C)}{\sum_{F_k \in F} P(q_i, q_{i-1}|F_k, C)} \tag{10}$$

Top-k Retrieved Documents As an effort to approximate the field-level term distribution of relevant documents, we described how a field-level collection language model can be used as a source of estimation. However, in many cases the field-level term distribution of relevant documents will diverge significantly from that of the collection. We somehow need ways to approximate the term distribution of relevant documents more closely.

To solve this problem, we propose using the top-k retrieved documents for a given query. Specifically, we combine the field-level language models of documents retrieved by some ranking methods to build a new language model for each field, and use it to approximate per-field and per-term weights:

$$P(F_j|q_i, D_{TopK}) = \frac{P(q_i|F_j, D_{TopK})}{\sum_{F_k \in F} P(q_i|F_k, D_{TopK})} \tag{11}$$

The idea of using top-k retrieved documents to approximate some aspect of relevance was introduced in Lavrenko and Croft [7], and our approach is similar in that we use top-k retrieved documents to approximate some dimension of relevance. The difference is that we use it to estimate field relevance, whereas their goal was to estimate term weights.

We use similar techniques to build the field language model of top-k documents as in previous work [7]. The probability is estimated based on the weighted average of the top-k retrieved documents, where the weights are the query-likelihood scores for those documents:

$$P(q_i|F_j, D_{TopK}) = \sum_{D \in TopK} P(w|F_j, D) \prod_{i=1}^{n} P(q_i|D) \tag{12}$$

Similarly to the case of a collection language model, we use bigram language models of the top-k documents to estimate field relevance.

$$P(F_j|q_i, q_{i-1}, D_{TopK}) = \frac{P(q_i, q_{i-1}|F_j, D_{TopK})}{\sum_{F_k \in F} P(q_i, q_{i-1}|F_k, D_{TopK})} \tag{13}$$

Per-Field Weights based on Training Queries Previous field-based retrieval models [14, 11] introduced ways of estimating per-field weights based on the retrieval effectiveness in training queries. Although field relevance in this work is defined to be dependent on each query term as well as field, we can incorporate these per-field weights as one of the sources to increase the reliability of the estimation.

4.2 Combining Sources

Given the sources as described above, we need to find a reasonable set of parameters to combine sources into a final estimate of field relevance. If we assume that we have training queries with relevance judgments, we can use coordinate ascent search [10, 1] to find a set of parameters that maximize the target metric in the training queries. Since we have only 5 parameters, this is computationally tractable. As for the choice of target metric, we followed previous work [10, 1, 14, 11], which used metrics of retrieval effectiveness, such as MAP or NDCG.

5 Experiments

In this section, we present an experimental evidence for proposed field relevance estimation technique and the field relevance model. We first describe the collections we used, and then present results for retrieval experiments. We then analyze the relationship between the quality of field relevance estimates and retrieval effectiveness. Since field relevance is the generalization of field weights in previous retrieval models, we use two terms interchangeably henceforth.

5.1 Experimental Setup

We first introduce the collections and other experimental settings. We used three collections with different document and query characteristics, and different numbers of relevant documents per query.

Firstly, we used a well-known TREC collection for structured documents (emails). The TREC 2005 Enterprise Track known-item task [3] used a crawl of the W3C mailing list, containing 198,394 documents with average length of 10kb. For each document, the indexed fields were *subject*, *body*, *to* (receiver), *date* and *from* (sender). Among the 150 queries provided, according to the TREC guideline, 25 were set aside for training of model parameters and the rest were used for testing. This collection has only one relevant document per query.

As another instance of a collection with rich structures, we then used the IMDB collection, which consists of 437,281 documents. Each document corresponds to a movie and was constructed from text data[1]. The fields were 'title', 'year', 'releasedata', 'language', 'genre', 'country', 'location', 'colorinfo', 'cast', 'team'. We used 50 queries (10 for training and 40 for evaluation) developed in a previous study [5]. This collection has two relevant documents per query on average.

Finally, we used the Monster[2] job description collection composed of 1,034,795 documents. Here, the documents were longer, with mostly full-text content. Each document is composed of fields like 'resumetitle', 'summary', 'jobtitle', 'school', 'experience', 'location' ,'skill' and 'additionalinfo'. The 60 queries we used (20 for training and 40 for evaluation) were requests for job descriptions created by real users of the Monster service. This collection has 15 relevant documents per query on average.

During indexing, each word was stemmed using the Krovetz stemmer and standard stopwords were eliminated. Indri[3] was used as a retrieval engine for all the retrieval experiments. Mean Average Precision (MAP) was used as the measure of retrieval performance for all experiments, since there were one or more relevant documents for each query with no grades in relevance judgments.

For baselines in our experiments, we used Document Query-Likelihood (DQL) [13], BM25F, Mixture of Language Models (MFLM) and the Probabilistic Retrieval Model for Semi-structured data (PRM-S). Since each retrieval model

[1] Available in http://www.imdb.com/interfaces#plain
[2] Licensed from Monster: http://www.monster.com
[3] http://www.lemurproject.org

required a different set of parameters to be tuned in advance, we used a training and test split for each query set.

For parameters that required training for each document field, such as per-field weights w_j and b_j, we performed a coordinate ascent search using training queries which find the best-performing parameter combination. As for the proposed method – field relevance model (FRM) – we similarly found the mixture weights Λ that maximize the retrieval performance in the training queries.

Similarity Metrics for Field Relevance An important assumption of our work is that the quality of field relevance estimate is correlated with retrieval effectiveness. Since we have an *oracle* estimate of field relevance based on relevant documents, we can directly evaluate given field relevance estimates against oracle estimates. For this purpose, we first calculate the per-term similarities between a given field relevance estimate and an oracle estimate as below, and then average these to create a query-level similarity.

As a first similarity measure, as we defined field relevance as a probability distribution, a natural choice is the **Kullback-Leibler divergence** between two per-term estimates of field relevance. Alternatively, we can compare the **cosine similarity** of an oracle estimate and the given estimate of field relevance. Finally, if we regard the problem of field relevance estimation as the relevance ranking of fields for a given query term, we can define a **precision** measure for each query term.

5.2 Retrieval Effectiveness

Table 1 shows retrieval results for all collections. Although the difference varies according to each collection, the field relevance model consistently improved baseline methods in all collections we tested. Among baseline methods, field-based retrieval models (BM25F and MFLM) showed better performance except for the case of BM25F model in the Monster collection, which might be due to the parameter estimation using a small number (20) of training queries.

The improvements over DQL, BM25F, MFLM and PRM-S method were statistically significant (using the paired t-test with p-value < 0.05) in all three collections we tested. Especially, the performance of FRM in TREC collection represents an improvement over already strong baseline (the best performance among TREC submission was 0.621 [2]). Finally, *oracle* estimates of field relevance in FRM_O shows the upper bound of performance one can get with ideal field relevance estimation. Here, oracle estimates were derived from the known relevant documents in each test collection, as described in Section 3.2.

Note that the only difference between MFLM, PRM-S and FRM is how field weights are estimated. Since FRM employs the per-field weight in MFLM and the unigram collection field-language model in PRM-S as the sources of combination, we can infer that additional sources used for FRM resulted in the improvement.

To gain further insights on the impact of different sources on retrieval effectiveness, we performed a feature ablation study where we omitted a set of sources

Table 1. Retrieval performance for three collections used. FRM_O is based on the oracle estimate of field relevance.

	DQL	BM25F	MFLM	PRM-S	FRM	FRM_O
TREC	0.542	0.597	0.601	0.624	0.668	0.794
IMDB	0.408	0.524	0.612	0.637	0.657	0.704
Monster	0.429	0.279	0.460	0.542	0.558	0.716

from the estimation of field relevance. We denote here five sources used for field relevance estimation as *cug* (collection unigram field-level language model (FLM)), *cbg* (collection bigram FLM), *tug* (top-k documents unigram FLM), *tbg* (top-k documents bigram FLM), and *prior* (per-field weight estimated using training queries).

Table 2. Feature ablation results in TREC collection. Each column denotes the performance where top-k document features (tug/tbg), bigram features (cbg/tbg), collection features (cbg/cug) and prior feature were omitted, respectively.

Features Omitted	None	tug/tbg	cbg/tbg	cbg/cug	prior
MAP	0.668	0.662	0.651	0.648	0.644

The results in Table 2 shows the impact from the omission of each source group on performance. You can see that all the source groups have a positive impact on the performance, and the omission of *prior* has the most impact on performance. This shows the importance of having a reliable back-up method in case the per-term field relevance estimation is difficult.

5.3 Field Relevance Estimation and Performance

Since we hypothesized that the performance advantage of FRM is based on improved estimation of field relevance, we then compared retrieval models in terms of the quality of field relevance estimates. For this experiment, we used three similarity metrics for field relevance introduced in Section 5.1, which measure the similarity with the oracle estimate found using relevant documents. The initial alphabet for each retrieval model (MFLM, PRM-S, FRM) denotes the estimates used in different retrieval methods.

Table 3. Quality of estimated field relevance compared to oracle estimation using the aggregated per-term KL-divergence (KL), cosine similarity (Cos) and precision ($P@1$). Higher value means higher quality, except for KL.

	KL_M	KL_P	KL_F	Cos_M	Cos_P	Cos_F	P@1_M	P@1_P	P@1_F
TREC	2.994	1.099	0.821	0.636	0.719	0.765	0.528	0.582	0.642
IMDB	2.764	0.723	0.529	0.405	0.814	0.876	0.478	0.802	0.820
Monster	4.121	1.481	1.381	0.358	0.650	0.675	0.015	0.467	0.481

The results above shows that field relevance model (FRM) improves estimation over MFLM and PRM-S, which use limited evidence for field relevance estimation. This result is consistent with our expectation that per-field and per-term estimation in PRM-S is better than per-field estimate in MFLM, and that the quality of estimation is improved by the combination of sources for FRM.

6 Related Work

Related work can be found in the area of structured document retrieval, and Section 2 provides a detailed review. Other recent work [12] showed that a keyword query can be refined into a structured query by mapping each query term into a set of structural fragments and transforming these fragments into the XPath query that represents the original information need most appropriately. Several other works [9] tried tagging a given query with labels that corresponds to different structural components of the document.

The modeling of field relevance can be considered as an extension of many efforts to model some aspect of relevance. The relevance-based language model [7] is a well-known model of topical relevance. Here, a relevance distribution is estimated from top-k retrieved documents, which is in turn used to enrich the initial representation of the information need given as a query.

As an extension of this work, Lavrenko et al. [8, 16] introduced the structural relevance model, which estimates a term-based relevance model per field. For retrieval, they combine field-level scores based on relevance models into a document score using fixed weights. Since our work focuses on estimating per-field and per-term weights, their model can be potentially improved based on the results here. However, they focus on modeling term-level relevance, whereas our work focuses on per-term field relevance.

The field relevance estimation technique in Section 4 is based on a linear combination of features, and we can relate this work to other term weighting models with linear combination of features. Metzler et al. [10] studies linear feature-based models in detail, focusing on optimization techniques such as a coordinate ascent. Bendersky et al. [1] employs similar techniques in estimating effective weights for terms and concepts.

7 Conclusions

In this work, we introduced the notion of field relevance as a generalization of the field weighting component of existing models for structured document retrieval. Field relevance can be estimated using relevant documents, thereby providing a natural way to incorporate relevance feedback for field weighting. We then showed how field relevance can be effectively estimated by combining many sources, which is shown to improve retrieval effectiveness significantly over strong baselines.

As a first effort in defining and estimating relevance in structured document retrieval as the distribution over fields, this work leaves a number of challenges. Our combination-based estimation framework can be improved by incorporating new sources and optimization techniques. The retrieval performance with true relevance feedback in Section 5.2 shows that there is still room for improvement. Lastly, we want to better understand the performance advantage of the proposed retrieval method as the function of document and query characteristics.

8 Acknowledgements

This work was supported in part by the Center for Intelligent Information Retrieval and in part by ARRA NSF IIS-9014442 . Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

References

1. M. Bendersky, D. Metzler, and W. B. Croft. Learning concept importance using a weighted dependence model. WSDM '10, pages 31–40, New York, NY, USA, 2010. ACM.
2. N. Craswell and S. R. Hugo Zaragoza. Microsoft cambridge at trec-14: Enterprise track. In *The Fourteenth Text REtrieval Conference*, 2005.
3. N. Craswell and A. P. D. Vries. Overview of the trec-2005 enterprise track. In *In The Fourteenth Text REtrieval Conf. Proc.*, 2005.
4. J. Kim and W. B. Croft. Retreival experiments using pseudo-desktop collections. In Proceedings of CIKM '09, Hong Kong, China, pages 1297–1306, 2009.
5. J. Kim, X. Xue, and W. B. Croft. *A Probabilistic Retrieval Model for Semistructured Data*. In Proceedings of ECIR '09. Springer, 2009.
6. V. Lavrenko. *A generative theory of relevance*. PhD thesis, 2004. AAI3152722.
7. V. Lavrenko and W. B. Croft. Relevance based language models. SIGIR '01, pages 120–127, New York, NY, USA, 2001. ACM.
8. V. Lavrenko, X. Yi, and J. Allan. Information retrieval on empty fields. In *HLT-NAACL*, pages 89–96, 2007.
9. X. Li, Y.-Y. Wang, and A. Acero. Extracting structured information from user queries with semi-supervised conditional random fields. In *SIGIR '09*, New York, NY, USA, 2009. ACM.
10. D. Metzler and W. Bruce Croft. Linear feature-based models for information retrieval. *Information Retrieval*, 10:257–274, June 2007.
11. P. Ogilvie and J. Callan. Combining document representations for known-item search. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference*, pages 143–150, New York, NY, USA, 2003. ACM.
12. D. Petkova, W. B. Croft, and Y. Diao. Refining keyword queries for xml retrieval by combining content and structure. ECIR '09, pages 662–669, Berlin, Heidelberg, 2009. Springer-Verlag.
13. J. Ponte and W. B. Croft. A language modeling approach to information retrieval. pages 275–281, New York, NY, 1998. ACM, ACM.
14. S. Robertson, H. Zaragoza, and M. Taylor. Simple BM25 extension to multiple weighted fields. In *In Proceedings of CIKM '04*, pages 42–49, New York, NY, USA, 2004. ACM.
15. S. E. Robertson and K. S. Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27(3):129–146, 1976.
16. X. Yi, J. Allan, and W. B. Croft. Matching resumes and jobs based on relevance models. In *SIGIR*, pages 809–810, 2007.