Entity Centric Query Expansion for Enterprise Search

Xitong Liu, Hui Fang
University of Delaware
Newark, DE, USA
{xtliu, hfang}@udel.edu

Fei Chen[1], Min Wang[2]
[1]HP Labs, Palo Alto, CA, USA
[2]HP Labs, Beijing, China
{fei.chen4, min.wang6}@hp.com

ABSTRACT

Enterprise search is important, and the search quality has a direct impact on the productivity of an enterprise. Many information needs of enterprise search center around *entities*. Intuitively, information related to the entities mentioned in the query, such as related entities, would be useful to reformulate the query and improve the retrieval performance. However, most existing studies on query expansion are term-centric. In this paper, we propose a novel entity-centric query expansion framework for enterprise search. Specifically, given a query containing entities, we first utilize both unstructured and structured information to find entities that are related to the ones in the query. We then discuss how to adapt existing feedback methods to use the related entities to improve search quality. Experiment results show that the proposed entity-centric query expansion strategy is more effective to improve the search performance than the state-of-the-art pseudo feedback methods on longer, natural language-like queries with entities.

Categories and Subject Descriptors: H.3.3 [Information Search and Retrieval]: Retrieval Models

General Terms: Algorithm

Keywords: Entity Centric, Enterprise Search, Retrieval, Query Expansion, Combining Structured and Unstructured Data

1. INTRODUCTION

Today any enterprise has to deal with a sheer amount of information such as emails, Web pages, relational databases, etc. The quality of enterprise search is critical to reduce business costs and produce positive business outcomes.

Despite the great progress on Web search, there are still many unsolved challenges in enterprise search [11]. In particular, enterprise data contain not only unstructured information such as documents and Web pages, but also a rich set of structured information such as relational databases. These structured data usually center around entities since relational databases are designed based on Entity-Relation models. Furthermore, the unstructured data, which capture information complimentary to structured data, also contain rich information about entities and their relationships, embedded in text. Clearly, a large portion of enterprise information centers around entities. Therefore, it would be interesting to study how to fully utilize the unique characteristic of enterprise data, i.e., *entities*, as a bridge to seamlessly combine both *structured* and *unstructured* data to improve enterprise search quality.

One of the important search problems in every enterprise is to provide effective self-service IT support, where an enterprise user submits a query to describe a problem and expects to find relevant information for solving the problem from a collection of knowledge documents. For example, a user may submit a query "XYZ cannot access intranet" to find solutions to the network connection problem that he has with his computer "XYZ". However, as the knowledge documents seldom cover information about specific IT assets such as "XYZ", there might be many documents relevant to "cannot access intranet" but not to query entity, i.e., "XYZ". With existing search engines, the user may not be able to efficiently find the solution to his problem. It is clear that query entities, i.e., those mentioned in a query, should play an important role in the query expansion process, since they often represent one or multiple aspects of the information need. Intuitively, a document mentioning entities that are related to the query entities are more likely to be relevant to the query than those not mentioning any related entities. For example, if we could know that "XYZ" is a PC and"ActivKey" is required for the authentication of employees so that PCs can access the intranet, we would expand the original query with the related entities, i.e., "ActivKey", to improve search accuracy.

In this paper, we study the problem of entity-centric query expansion for enterprise search. Given a query involving entities, the goal is to utilize the entity relationships embedded in both *structured* and *unstructured* information to find entities that are related to the query and use them to improve the enterprise search performance.

We first study how to identify related entities. The structured data contain explicit information about relations among entities such as foreign-key relationships. However, the entity relationship information is often hidden in the unstructured data. We apply Condition Random Fields models to learn a domain-specific entity recognizer, and apply the entity recognizer to documents and queries to identify entities from the unstructured information. If two entities co-occur

CIKM'12, October 29–November 2, 2012, Maui, HI, USA.

in the same document, they are related. The relations can be discovered by the context terms surrounding their occurrences.

With the entities and relations identified in both structured and unstructured data, we propose a general ranking strategy that systematically integrates the entity relationships from both data types to rank the entities that have relationships with the query entities. Intuitively, related entities should be relevant not only to the entities mentioned in the query but also the query as a whole. Thus, the ranking strategy is determined by not only the relationships between entities, but also the relevance of the related entities for the given query.

We conduct experiments over a real world enterprise data collection to evaluate the proposed methods. We find that the performance of entity identification is satisfying, and the proposed entity ranking methods are effective to find related entities for a given query. Moreover, experiment results show that entity centric query expansion methods are more effective than the state-of-the-art pseudo feedback methods to improve the retrieval performance over longer, natural language-like queries with entities.

2. RELATED WORK

Enterprise search is an important topic as the failure of finding relevant information will significantly cause the loss of productivities and therefore profit [10]. However, compared with Web search, enterprise search has received less attention in the research community. Hawking [11] discussed several challenges in enterprise search, but it remains unclear what are the effective strategies to improve search accuracy. The enterprise track in TREC [7, 4] has attracted more research efforts on improving enterprise search quality including expert finding [1, 19] and document search [12, 16]. However, to our best knowledge, few work has been done on utilizing entities to improve enterprise search.

Our work is also related to entity retrieval. The entity track of TREC conference focused on the problem of finding related entities [2, 3]. The goal is to retrieve entities that are related to a structured query from a document collection. The entity ranking of INEX [8] also focused on retrieving related entities with the emphasis on the type of the target entities (i.e., categories) rather than the relation between the target and input entities. Liu et al. [15] studied the problem of finding relevant information with specified types. Unlike previous work, we use unstructured queries to find the related entities, and no entity type is specified in the query and no explicit relation is specified either. Moreover, the related entities are not returned directly but utilized to improve document search accuracy.

Query expansion is a well known strategy to improve retrieval performance [22, 14]. A common strategy in most existing query expansion methods are term-based. Specifically, they use different strategies to select expansion *terms* from feedback documents, user feedback or external sources, and update the existing query through some re-weighting strategies. On the contrary, we study the feasibility of using related entities for query expansion.

3. FINDING RELATED ENTITIES

Since structured information is designed based on entity-relationship models, it is straightforward to identify entities and their relationships there. However, the problem is more challenging for the unstructured information, where we do not have any information about the semantic meaning of a piece of text. In this section, we will first discuss how to identify entities in the unstructured information and then propose a general ranking strategy to rank the entities based on the relationships in both unstructured and structured information. Table 1 explains the notations used in the paper.

Table 1: Notations.

Q	An entity-centric query
E_Q	A set of entities mentioned in query Q
E_R	The related entities for query Q
Q_E	The expanded query of Q
$\mathcal{D}$	An enterprise data collection
$\mathcal{D}_{TEXT}$	The unstructured information in $\mathcal{D}$
$\mathcal{D}_{DB}$	The structured information in $\mathcal{D}$
e_i	An entity in the structured information $\mathcal{D}_{DB}$
$E(T)$	A set of entities in the text T

3.1 Entity Identification in Unstructured Data

Unlike structured information, unstructured information does not have semantic meaning associated with each piece of text. As a result, entities are not explicitly identified in the documents, and are often represented as sequences of terms. Moreover, the mentions of an entity could have more variants in unstructured data. For example, entity "Microsoft Outlook 2003" could be mentioned as "MS Outlook 2003" in one document but as "Outlook" in another.

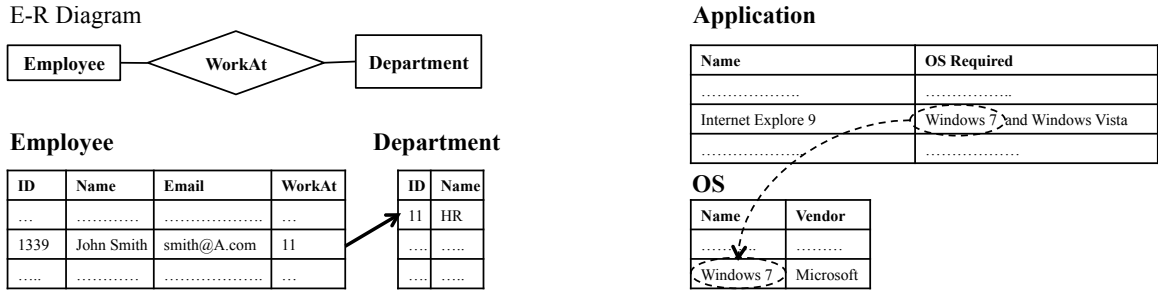
The majority of entities in enterprise data are *domain specific* entities such as IT assets. These domain specific entities have more variations than the common types of entities. To identify entity mentions from the unstructured information, following existing studies on named entity identification [18, 9, 5], we train a model based on Conditional Random Fields (CRFs) [13] with various features including dictionary, regular expression and part of speech tags. Specifically, the model makes binary decision for each term in a document, and the term will be labeled as either an entity term or not. We trained the model on a training document set with their entity mentions manually labeled. Note that the training set is different from the test collections we used in the experiments.

After identifying entity mentions in the unstructured data (denoted as em), we need to connect them with the entities in the structured data (denoted as e) to make both the unstructured and structured data integrated. Specifically, we first construct a list of candidate entities from the structured data. Given an entity mention in a document, we calculate its string similarity based on the SoftTFIDF string similarity function proposed by Cohen et al. [6] with every one on the candidate list and select the most similar one. Note that this step is done offline.

3.2 Entity Ranking

The next challenge is how to rank candidate entities for a given query. The underlying assumption is that the relevance of the candidate entity for the query is determined by the relationships between the candidate entity and the entities mentioned in the query. Formally, the relevance score of a candidate entity e for a query Q can be computed as:

$$R(Q,e) = \sum_{e^Q \in E_Q} R(e^Q, e), \quad (1)$$



(a) Foreign key links between entities

(b) Entity mentioned in the attribute field

Figure 1: Entity relations in structured data.

where $R(e^Q, e)$ is the relevance score between query entity e^Q and e based on their relationships in collection $\mathcal{D}$. We now discuss how to exploit the characteristics of both unstructured and structured information to compute the relevance score between two entities, i.e. $R(e^Q, e)$, based on their relationships.

3.2.1 Using Relationships from the Structured Data

In relational databases, every table corresponds to one type of entities, and every tuple in a table corresponds to an entity. The database schema describes the relations between different tables as well as the meanings of their attributes.

We consider two types of entity relationships. First, if two entities are connected through foreign key links between two tables, these entities will have the same relation as the one specified between the two tables. For example, as shown in Figure 1(a), entity "John Smith" is related to entity "HR", and their relationship is "WorkAt". Second, if one entity is mentioned in an attribute field of another entity, the two entities have the relation specified in the corresponding attribute name. As shown in Figure 1(b), entity "Windows 7" is related to entity "Internet Explorer 9" through relation "OS Required". We now discuss how to compute the relevance scores between entities based on these two relation types.

The relevance scores based on *foreign key relations* are computed as:

$$R^{LINK}(e^Q, e) = \begin{cases} 1 & \text{if there is a link between } e^Q \text{ and } e, \\ 0 & \text{otherwise.} \end{cases}$$

The relevance scores based on *field mention relations* are computed as:

$$R^{FIELD}(e^Q, e) = \sum_{e \in E(e^Q.text)} 1 + \sum_{e \in E(e.text)} 1,$$

where $e.text$ denotes the union of text in the attribute fields of e. We can get the final ranking score by integrating the two types of relevance score through linear interpolation:

$$R^{DB}(e^Q, e) = \alpha R^{LINK}(e^Q, e) + (1-\alpha) R^{FIELD}(e^Q, e), \quad (2)$$

where α is a coefficient to control the influence of two components.

3.2.2 Using Relationships from Unstructured Data

Unlike in the structured data where entity relationships are specified in the database schema, there is no explicit entity relationship in the unstructured data. Since the co-occurrences of entities may indicate certain semantic relations between these entities, we use the co-occurrence relationships in this paper. Our experiment results (c.f. Section 5) showed that such co-occurrence relationships can already result in good performance in entity ranking and query expansion. We may also apply advanced NLP techniques to automatically extract relations [20], and we leave it as our future work.

After identifying entities from unstructured data and connecting them with candidate entities as described in the previous subsections, we are able to get the information about co-occurrences of entities in the document sets. If an entity co-occurs with a query entity in more documents and the context of the co-occurrences is more relevant to the query, the entity should have higher relevance score.

Formally, the relevance score can be computed as follows:

$$R^{TEXT}(e^Q, e) = \sum_{d \in \mathcal{D}_{TEXT}} \sum_{c \in WINDOW(e^Q, e, d)} S(Q, c), \quad (3)$$

where d denotes a document in the enterprise collection, and $WINDOW(e^Q, e, d)$ is the set of all possible context windows in which the two entities co-occur in d . The basic assumption is that the relations between the two entities can be captured through their context. Thus, the relevance between the query and the context terms can be used to model the relevance of the relationships between two entities for the given query. The window size is set to 64 based on preliminary results. If the distance of two entities is longer than the window size, they will be considered non-related. Note that $S(Q, c)$ measures the relevance score between the query and content of context window of the two entities. Since both Q and c essentially are bag of words, the relevance score between them can be estimated by existing document retrieve models.

4. ENTITY CENTRIC QUERY EXPANSION

We now discuss how to utilize the related entities to improve the performance of document retrieval. As shown in Section 1, we observe that the related entities, which are relevant to the query but are not directly mentioned in the query can serve as complementary information to the original query terms. Therefore, integrating the related entities into the query can help the query to cover more information aspects, and thus improve the performance of document retrieval.

Language modeling [17] has been a popular framework for document retrieval in the recent decade. One of the popular retrieval models is KL-divergence [22], where the relevance

score of document D for query Q can be estimated based on the distance between the document and query models, i.e.

$$S(Q,D) = \sum_w p(w|\theta_Q) \log p(w|\theta_D).$$

To further improve the performance, Zhai and Lafferty [22] proposed to update the original query model using feedback documents as follows:

$$\theta_Q^{new} = (1-\lambda)\theta_Q + \lambda\theta_{\mathcal{F}}, \quad (4)$$

where θ_Q is the original query model, $\theta_{\mathcal{F}}$ is the estimated feedback query model based on feedback documents, and λ controls the influence of the feedback model.

Unlike previous work where the query model is updated with terms selected from feedback documents, we propose to update it using the related entities. Following the sprit of model-based feedback methods [22], we propose to update the query model as follows:

$$\theta_Q^{new} = (1-\lambda)\theta_Q + \lambda\theta_{ER}, \quad (5)$$

where θ_Q is the query model, θ_{ER} is the estimated expansion model based on related entities and λ controls the influence of θ_{ER}. Given a query Q, the relevance score of a document D can be computed as:

$$S(Q,D) = \sum_w ((1-\lambda)p(w|\theta_Q) + \lambda p(w|\theta_{ER})) \log p(w|\theta_D). \quad (6)$$

The main challenge here is how to estimate $p(w|\theta_{ER})$ based on related entities.

Given a query, we have discussed how to find related entities E_R in the Section 3. We think the top ranked related entities can provide useful information to better reformulate the original query. Here we use "bags-of-terms" representation for entity names, and a name list of related entities can be regarded as a collection of short documents. Thus, we propose to estimate the expansion model based on the related entities as follows:

$$p(w|\theta_{ER}) = \frac{\sum_{e_i \in E_R^L} count(w, N(e_i))}{\sum_{w'} \sum_{e_i \in E_R^L} count(w', N(e_i))}. \quad (7)$$

where E_R^L is the top L ranked entities from E_R, and $N(e)$ is the name of the entity e.

5. EXPERIMENTS

5.1 Experiment Design

We construct two enterprise data sets using real-world data from HP, denoted as **ENT**. Each data set consists of two parts: unstructured documents and structured databases. The unstructured documents consist of 59,706 knowledge base documents which are provided by the IT support department. Most of the documents are talking about how-to and troubleshooting for the software products used HP. The structured data include a relational database which contains the information about 2,628 software products. 60 Queries are collected from the internal IT support forum as the query set. Almost all the queries are described in natural languages, and the average query length is 8 terms, which is much longer than keyword queries used in Web search. The queries are selected so that every query contains at least one entity. Let us consider a query from the query set, i.e., "Office 2003 SP3 installation fails on Windows XP". It mentions two entities: "Office 2003" and "Windows XP". For each query, we ask human assessors to manually label the relevance of every entity (for evaluating finding related entities) and every document (for evaluating document retrieval).

We use MAP (Mean Average Precision) as the main measurement for the performance. P@3 (Precision at rank 3), P@10 (Precision at rank 10) and R-precision (R is the number of relevant results for a given query) are also reported.

Models	Equations	MAP	P@3	P@10	P@R
R^{TEXT}	Plugging (3) in (1)	0.530	0.593	0.348	**0.463**
R^{DB}	Plugging (2) in (1)	0.131	0.253	0.122	0.158
R^{BOTH}	(8)	**0.545**	**0.642**	**0.354**	0.462

Table 2: Optimal Results of Finding Related Entities on ENT.

5.2 Finding Related Entities

We evaluated the effectiveness of our entity ranking methods. By plugging Equation (3) and (2) into Equation (1), we can get different entity ranking models, which are denoted as R^{TEXT} and R^{DB}, respectively. Moreover, structured and unstructured data may contain different relationships between entities. Thus, it would be interesting to study whether combining these relationships could bring any benefits. We can combine them through a linear interpretation:

$$R^{BOTH}(Q,e) = \beta R^{TEXT}(Q,e) + (1-\beta)R^{DB}(Q,e), \quad (8)$$

where β balances the importance of the relationships from the two sources.

Table 2 shows the optimal results. We can find that the performance of R^{TEXT} is much better than R^{DB}, showing that the relationships in the unstructured documents are more effective than those in the structured data. The R^{BOTH} model can reach the best performance, but its improvement over R^{TEXT} is small.

We then studied the parameter values used to achieve the optimal performance on **ENT**. Specifically, α in Equation (2) is set to 0.7, indicating that the foreign link relations is more important than entity mention relations. And β in Equation (8) is set to 0.7, which suggests that the unstructured data contributes most to rank the related entities. By analyzing the data, we find that the main reason for the worse performance of structured data based entity ranking (i.e. R^{DB}) is that the number of relations between entities (either foreign key links or entity mention in the attribute field) is much smaller than that within the unstructured data. Only 37.5% of entities has relationships in the structured data. We expect the performance of R^{DB} could be improved when the structured data can provide more information about entity relations.

5.3 Effectiveness of Query Expansion

The entity-centric expansion function is shown in Equation (6). In the experiments, we estimate $p(w|\theta_Q)$ by maximum likelihood, i.e. $p(w|\theta_Q) = \frac{count(w,Q)}{|Q|}$, where $count(w,Q)$ is the number of occurrences of w in Q and $|Q|$ is the query length. And $p(w|\theta_D)$ can be estimated using smoothing methods such as Dirichlet Prior [21].

Thus, the basic retrieval model (i.e., when $\lambda = 0$) is the KL-divergence function with Dirichlet Prior smoothing [21], which is one of the state-of-the-art retrieval functions. We denote it as *Baseline*. To compare our models with other language model based query expansion models, we choose

Models	MAP	P@3	P@10	P@R
$Baseline$	0.216	**0.172**	0.105	0.177
$BaselineFB$	0.220	0.167	0.105	0.174
QE^{TEXT}	0.256⋆	0.161	**0.112**	0.196
QE^{DB}	0.229	0.167	0.103	0.183
QE^{BOTH}	**0.260**⋆†	0.161	**0.112**	**0.210**

Table 3: Optimal Results of Entity Centric Query Expansion on ENT. ⋆ and † denote the improvement over $Baseline$ and $BaselineFB$ is statistically significant at 0.05 level by Wilcoxon signed-rank test.

Models	MAP	P@3	P@10	P@R
$Baseline$	0.216	**0.172**	**0.105**	0.177
$BaselineFB$	0.220	0.167	**0.105**	0.174
QE^{TEXT}	0.245	0.128	0.102	0.180
QE^{DB}	0.229	0.161	0.098	0.177
QE^{BOTH}	**0.250**	0.139	**0.105**	**0.193**

Table 4: Results of Entity Centric Query Expansion on ENT using 5-Fold Cross Validation.

model-based feedback [22], which is a state-of-the-art pseudo relevance feedback model, to do query expansion and use it as a stronger baseline, which is denoted as $BaselineFB$.

As described in Section 4, we can expand queries with the names of entities which are related to the query. Specifically, the entity name based expansion model (i.e., Equation (7) using entity lists from R^{TEXT}, R^{DB} and R^{BOTH} are denoted as QE^{TEXT}, QE^{DB} and QE^{BOTH}, respectively. The optimal results are reported in Table 3. Comparing it with Table 2, we can find that the better the quality of related entities, the better the performance of query expansion. QE^{BOTH} performs best as R^{BOTH} generates the best result of related entities. Moreover, we can find that QE^{BOTH} can outperform both baselines significantly, showing that entity names have more beneficial effects than the terms chosen by language modeling approach to do query expansion.

Since the optimal parameter settings for the same query expansion model may vary on different data sets, we would like to see the robustness of our models by automatically learning the optimal parameter settings within one data set. We conduct 5-fold cross validation over the query set on **ENT**, and the results are shown in Table 4. It is clear that the proposed entity-based expansion methods are more effective than the two baseline methods.

6. CONCLUSION AND FUTURE WORK

In this paper we study the problem of improving enterprise search quality using related entities to do query expansion. In particular, we propose a domain specific entity identification method based on CRFs, a general ranking strategy that can find related entities based on different entity relationships from both unstructured and structured data, and an entity-centric query expansion method that can utilize related entities to estimate a new query model. We then conduct experiments over a real-world enterprise data set to exam the effectiveness of both finding related entities and entity based query expansion methods. Experiment results demonstrate that our proposed entity ranking methods can retrieve high quality related entities. Moreover, results also show that entity based query expansion method can outperform the state-of-the-art term based query expansion methods over long, natural language like queries with entities.

There are many interesting future research directions. First, it would be interesting to leverage relation among entities to improve the performance. Second, we plan to study how to utilize the related entities to aggregate search results.

7. ACKNOWLEDGEMENTS

This material is based upon work supported by the HP Labs Innovation Research Program. We thank the anonymous CIKM reviewers for their useful comments.

8. REFERENCES

[1] K. Balog, L. Azzopardi, and M. de Rijke. Formal Models for Expert Finding in Enterprise Corpora. In *SIGIR*, pages 43–50, 2006.

[2] K. Balog, A. P. de Vries, P. Serdyukov, P. Thomas, and T. Westerveld. Overview of the TREC 2009 Entity Track. In *Proceedings of TREC*, 2010.

[3] K. Balog, P. Serdyukov, and A. P. de Vries. Overview of the TREC 2010 Entity Track. In *Proceedings of TREC*, 2011.

[4] K. Balog, I. Soboroff, P. Thomas, P. Bailey, N. Craswell, and A. P. de Vries. Overview of the TREC 2008 Enterprise Track. In *Proceedings of TREC'08*, 2008.

[5] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. H. Jr., and T. M. Mitchell. Toward an Architecture for Never-Ending Language Learning. In *AAAI*, 2010.

[6] W. W. Cohen, P. Ravikumar, and S. E. Fienberg. A Comparison of String Distance Metrics for Name-Matching Tasks. In *IJCAI*, pages 73–78, 2003.

[7] N. Craswell, A. P. de Vries, and I. Soboroff. Overview of the TREC 2005 Enterprise Track. In *Proceedings of TREC'05*, 2005.

[8] G. Demartini, T. Iofciu, and A. de Vries. Overview of the INEX 2009 Entity Ranking Track. In *Focused Retrieval and Evaluation*, 2010.

[9] A. Doan, L. G. R. Ramakrishnan, and S. Vaithyanathan. Introduction to the Special Issue on Managing Information Extraction. *SIGMOD Record*, 37(4), 2009.

[10] S. Feldman and C. Sherman. The High Cost of Not Finding Information. In *Technical Report No. 29127, IDC*, 2003.

[11] D. Hawking. Challenges in Enterprise Search. In *Proceedings of ADC'04*, pages 15–24, 2004.

[12] M. Kolla and O. Vechtomova. Retrieval of Discussions from Enterprise Mailing Lists. In *SIGIR*, pages 881–882, 2007.

[13] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *ICML*, pages 282–289, 2001.

[14] V. Lavrenko and W. B. Croft. Relevance-Based Language Models. In *SIGIR*, pages 120–127, 2001.

[15] X. Liu, H. Fang, C.-L. Yao, and M. Wang. Finding Relevant Information of Certain Types from Enterprise Data. In *CIKM*, pages 47–56, 2011.

[16] C. Macdonald and I. Ounis. Combining Fields in Known-Item Email Search. In *SIGIR*, pages 675–676, 2006.

[17] J. M. Ponte and W. B. Croft. A Language Modeling Approach to Information Retrieval. In *SIGIR*, pages 275–281, 1998.

[18] S. Sarawagi. Information Extraction. *Foundations and Trends in Databases*, 1(3):261–377, 2008.

[19] P. Serdyukov, H. Rode, and D. Hiemstra. Modeling Multi-step Relevance Propagation for Expert Finding. In *CIKM*, pages 1133–1142, 2008.

[20] D. Zelenko, C. Aone, and A. Richardella. Kernel methods for relation extraction. *J. Mach. Learn. Res.*, 3:1083–1106, Mar. 2003.

[21] C. Zhai and J. Lafferty. A Study of Smoothing Methods for Language Models Applied to Ad Hoc Information Retrieval. In *SIGIR*, pages 334–342, 2001.

[22] C. Zhai and J. Lafferty. Model-Based Feedback in the Language Modeling Approach to Information Retrieval. In *CIKM*, 2001.