The University of Amsterdam at TREC 2010

Session, Entity, and Relevance Feedback

Marc Bron Jiyin He Katja Hofmann Edgar Meij
Maarten de Rijke Manos Tsagkias Wouter Weerkamp

ISLA, University of Amsterdam
http://ilps.science.uva.nl/

Abstract: We describe the participation of the University of Amsterdam's ILPS group in the session, entity, and relevance feedback track at TREC 2010. In the Session track we investigate the use of blind relevance feedback for taking information about a previous query into account when retrieving documents for a follow-up query. In the Entity Track REF task we experiment with a window size parameter to limit the contexts that are considered by our co-occurrence model and explore the use of Freebase for type filtering, entity normalization and homepage finding. To address the ELC task we locate candidate entities based on objects shared with the example entities and rank candidates based on the predicates and objects they share with the example entities. In the relevance feedback track we evaluate a novel model that uses wikipedia as a pivot language for estimating query models.

1 Introduction

This year the Information and Language Processing Systems (ILPS) group of the University of Amsterdam participated in the session, entity and relevance feedback tracks. In this paper, we describe our participation for each of these tracks, in three largely independent sections: Section 3 on our session track participation, Section 4 on our participation in the entity track, and Section 5 on our work in the relevance feedback track. We detail the runs we submitted, present the results of the submitted runs, and, where possible, provide an initial analysis of these results. Before doing so, we describe the shared retrieval approach in Section 2. We conclude in Section 6.

2 Retrieval Framework

In this section we describe our general approach for each of the tracks in which we participated this year. We employ a language modeling approach to IR and rank documents by their log-likelihood of being relevant given a query. Without presenting details here, we only provide our final formula for ranking documents, and refer the reader to (Balog et al., 2008) for the steps of deriving this equation:

$$\log P(D|Q) \propto \log P(D) + \sum_{t \in Q} P(t|\theta_Q) \cdot \log P(t|\theta_D). \quad (1)$$

Here, both documents and queries are represented as multinomial distributions over terms in the vocabulary, and are referred to as *document model* (θ_D) and *query model* (θ_Q), respectively. The third component of our ranking model is the *document prior* ($P(D)$), which is assumed to be uniform, unless stated otherwise. Note that by using uniform priors, Eq. 1 gives the same ranking as scoring documents by measuring the KL-divergence between the query model θ_Q and each document model θ_D, in which the divergence is negated for ranking purposes (Lafferty and Zhai, 2001).

2.1 Modeling

Unless indicated otherwise, we smooth each document model using a Dirichlet prior:

$$P(t|\theta_D) = \frac{n(t,D) + \mu P(t)}{\sum_t n(t,D) + \mu}, \quad (2)$$

where $n(t,D)$ indicates the count of term t in D and $P(t)$ indicates the probability of observing t in a large background model such as the collection:

$$P(t) = P(t|C) = \frac{\sum_D n(t,D)}{|C|}. \quad (3)$$

μ is a hyperparameter that controls the influence of the background corpus which we set to the average document length.

As to the query model θ_Q, we adopt the common approach to linearly interpolate the initial query with an expanded part (Balog et al., 2008; Zhai and Lafferty, 2001):

$$P(t|\theta_Q) = \lambda_Q P(t|\hat{\theta}_Q) + (1 - \lambda_Q) P(t|Q), \quad (4)$$

where $P(t|Q)$ indicates the MLE on the initial query, $P(t|\hat{\theta}_Q)$ indicates the MLE of the expanded part, and the parameter λ_Q controls the amount of interpolation.

2.2 Significance testing

Throughout the paper we use the Wilcoxon signed-rank test to test for significant differences between runs. We report on significant increases (or drops) for $p < .01$ using $^{\blacktriangle}$ (and $^{\blacktriangledown}$) and for $p < .05$ using $^{\triangle}$ (and $^{\triangledown}$).

2.3 Clueweb

All the tracks we participated in this year make use of the Clueweb document collection. We do not use any form of stemming and remove a conservative list of 588 stopwords. We index the headings, titles, and contents as searchable fields and do not remove any HTML tags.

3 Session Track

The goal of the TREC Session track is to find out how retrieval systems perform over the course of a session and whether taking a previous query into account can help improve retrieval performance for a follow-up query. The current setting considers sessions consisting of an original query and one follow-up query that constitutes either a generalization, specialization, or a parallel move.

Our submission for the TREC Session track explores the use of blind relevance feedback to bias a follow-up query towards or against the topics covered in documents that were returned to the user in response to the original query. Blind relevance feedback takes the most discriminative terms from a set of documents retrieved for a query, and uses these to build a query model that incorporates information about the topic underlying the documents. We apply this method to an initial, diverse result list. Below we explain our approach in detail, and give an overview of our results.

3.1 Approach

Currently, little is known about users' expectations about retrieval systems' behavior throughout a session. Therefore, we based our submission on the following intuitions. Without contextual information about the users' preferred interpretation of a query, a retrieval system can return a standard retrieval run (cf. *Retrieval approach*, below). If the query is ambiguous, it may be better to provide a diverse result list to increase the likelihood of providing at least some relevant documents for different possible interpretations of the query(cf. *Diversification*). When additional information from a previous query can be taken into account, search results can be more focused. In our submission we use pseudo relevance feedback to combine information about the topics covered by the two queries (cf. *Pseudo Relevance Feedback*).

Retrieval approach Our retrieval system uses the framework explained above (cf. §2). We use a Dirichlet prior with $\mu = 1600$. Queries are constructed to emphasize phrases, as these are often found in web queries. Phrases and individual terms are combined with equal weights. An example query is shown below.

```
<query>
  <number>1</number>
  <text>#weight( 0.5 #1(legal advice) 0.5
        #combine(legal advice) )</text>
</query>
```

Figure 1: Example query, combining individual terms and phrases.

Retrieval runs are post-processed to filter out category and redirection pages from wikipedia, and to place the top result from wikipedia at the top of the result list.

Diversification We diversify runs following a *topic model-based approach*. It models documents as a mixture of topics and constructs a final result list by re-ranking an initial list so that as many topics as possible are represented in the top ranked documents.

Our approach is inspired by previous work on diversifying a ranked list with Maximal Marginal Relevance (MMR) by Carbonell and Goldstein (1998) and based on a topic modeling approach, i.e., LDA (Blei et al., 2003). It treats the re-ranking problem as a procedure of selecting a sequence of documents, where a document is selected depending on both its relevance with respect to the query and the documents that have already been selected before it, so as to have a set of documents that (i) are most relevant to the query and (ii) represent most if not all topical aspects.

We proceed as follows. First, we use LDA to extract 10 topics from the top 500 documents of the baseline retrieval run described above, so that each document is represented as a mixture of these 10 topics. We then start the re-ranking procedure by selecting the top relevant document in the initial list as the first document in the new ranked list. Then, we select a next document that can maximize the expected joint probability of presence of all topics in the selected result set. Since the sum of topic proportions within a document equals 1, the maximum joint probability (i.e., product of the probabilities of presence of each topic) occurs when the topics have equal proportion in the selected set. On the other hand, we use the retrieval score from the initial run as a prior probability that a document is selected as the next one, so as to take into account the relevance relation between the document and the original query.

Formally, given a query Q, a set of candidate documents $Ca = \{D_j\}_{j=1}^{n}$ and a set of latent topics $T = \{t_i\}_{i=1}^{m}$, a document is selected from Ca for inclusion in the ranked list S such that

$$\arg\max_{D \in Ca} \quad P(Q|D) \prod_{i=1}^{m} P(t_i \in S \cup \{D\}), \tag{5}$$

where $P(Q|D)$ is the query likelihood between the query Q and document D calculated as in a standard language modeling framework. The term $P(t_i \in S \cup \{D\})$ denotes the probability of a topic being present in the set $S' = S \cup \{D\}$, which is estimated by

$$P(t_i \in S') = \sum_{D_j \in S'} P(t_i \in D_j) P(D_j). \quad (6)$$

Pseudo Relevance Feedback RL3 runs are generated using blind relevance feedback as follows. First, retrieval runs for the original and the follow-up query individually are generated, using the *baseline* method and *diversification* as described above. From the top-10 documents of these runs, the 10 most discriminative relevance feedback terms are extracted to form the sets of expansion terms E_O (expansion terms extracted from results for the original query) and E_F (expansion terms extracted from results for the follow-up query). These are combined to form the query expansion E as follows:

RF1 $E = E_O$ - only use feedback terms extracted from the top-ranked results of the original query.

RF2 $E = E_F \setminus E_F$ - take feedback terms generated from results for the follow-up query and remove terms that were also extracted from results for the original query.

RF3 $E = E_O \cup E_F$ - combine relevance feedback terms of both queries.

These three approaches implement the following intuitions. First, we assume that results returned for the original query were helpful and can be used to focus or disambiguate results for the follow-up query. Second, we cover the assumption that results for the original query were not helpful. Finally, we consider the possibility that the underlying topic may best be represented by both queries.

As a final step in generating results when taking an original query into account, we remove documents that have been displayed in the top-10 of the response to the original query from the result list shown for the follow-up query.

3.2 Runs

All submitted runs were automatic category A runs.

uvaExt*.RL1 standard retrieval run using the original query + diversification using LDA

uvaExt1.RL2 standard retrieval run using the follow-up query

uvaExt1.RL3 combines uvaExt1.RL1 and uvaExt1.RL2 using RF1.

uvaExt2.RL2 standard retrieval run using the follow-up query + blind relevance feedback using the follow-up query

uvaExt2.RL3 combines uvaExt1.RL1 and uvaExt1.RL2 using RF2.

uvaExt3.RL2 standard retrieval run using the follow-up query + diversification using LDA

uvaExt3.RL3 combines uvaExt1.RL1 and uvaExt1.RL2 using RF3.

3.3 Results

Results for our submissions are listed in Table 1. Listed is $nsDCG@10.RL_{12,13}$ with and without taking duplicate documents into account. $nsDCG@10.RL_{12}$ measures session performance when the original query was not taken into account. $nsDCG@10.RL_{13}$ measures session performance when the original query was taken into consideration.

Table 1: Results. nsDCG@10 for RL_{12} and RL_{13}.

	with duplicates		*w/o duplicates*	
runID	n..RL_{12}	*n*..RL_{13}	*n*..RL_{12}	*n*..RL_{13}
uvaExt1	0.1356	0.1320	0.1416	0.1398
uvaExt2	0.1260	0.1297	0.1317	0.1373$^{\Delta}$
uvaExt3	0.1262	0.1279	0.1311	0.1356

We find overall best performance is achieved by run *uvaExt1* when duplicates are removed and the original query was *not* taken into account. This run retrieves document lists for each query. In all other cases, the follow-up run was diversified and in these cases, performance improves when taking the original query into account. In one case, this improvement is statistically significant (*uvaExt2*, no duplicates).

Performance when measured after removing duplicate documents improves in all cases. This is expected, as we remove duplicate documents that were previously displayed to the user at high ranks.

Table 2: Results, split by type of query reformulation.

	Generalization		*Specialization*		*Drift*	
runID	n..RL_{12}	*n*..RL_{13}	*n*..RL_{12}	*n*..RL_{13}	*n*..RL_{12}	*n*..RL_{13}
uvaExt1	0.1682	0.1694	0.1153	0.1032	0.1267	0.1273
uvaExt2	0.1558	0.1569	0.1040	0.1105	0.1213	0.1248
uvaExt3	0.1492	0.1659	0.1044	0.1029	0.1273	0.1190

Table 2 shows results when split by the type of query reformulation. Overall, we can see that scores for *Generalization* queries are highest, followed by *Drift*, and *Specialization*. This is expected as more general queries are expected to have more relevant documents, making it easier to retrieve these relevant documents. For *Generalizations* performance when taking the original query into account is highest for *uvaExt1*, which indicates that adding blind relevance feedback based

on the original query is helpful for this type of reformulation. The opposite is the case for *Specializations*. Here, the best performance when taking a previous query into account is achieved by *uvaExt2*, where feedback terms based on the original query are excluded. Finally, for *Drift* reformulations, best performance is again achieved by *uvaExt1*.

Our preliminary results indicate that blind relevance feedback can be helpful in taking an original query into account when retrieving documents for a follow-up query. The success of this approach appears to depend on the type of query reformulation.

4 Entity Track

The Entity Track consists of two tasks this year. The main task is the Related Entity Finding (REF) task introduced last year, where the goal is to find homepages of entities given a source entity, relation and target type. New this year is the second task: Entity List Completion (ELC). In the ELC task the goal is to find entities in structured data given a source entity, relation, target type and example entities.

4.1 Related Entity Finding Approach

In the REF task, we continue our experiments with co-occurrence models Bron et al. (2010). This year we use a generative model to rank candidate entities that combines the co-occurrence between the source entity and candidate entities with evidence for relevance to the relation from the snippets in which they co-occur. To the ranking provided by this model we apply type filtering based on Freebase and homepage finding using candidate entity names as queries to a web search engine. We briefly recall the derivation of our co-occurrence model below, followed by a description of each of the components.

We formulate the entity ranking problem as follows: rank candidate entities (e) according to $P(e|E,T,R)$, where E is the source entity, T is the target type, and R is the relation described in the narrative.

Instead of estimating this probability directly, we use Bayes' rule and reformulate it into:

$$P(e|E,T,R) \quad = \quad \frac{P(E,T,R|e)\cdot P(e)}{P(E,T,R)}. \tag{7}$$

Next, we drop the denominator as it does not influence the ranking of entities, and derive our final ranking formula as follows:

$$\begin{aligned} &P(E,T,R|e)\cdot P(e) \\ &\quad = P(E,R|e)\cdot P(T|e)\cdot P(e) && (8)\\ &\quad = P(E,R,e)\cdot P(T|e) \\ &\quad = P(R|E,e)\cdot P(E,e)\cdot P(T|e) \\ &\quad = P(R|E,e)\cdot P(e|E)\cdot P(E)\cdot P(T|e) && (9)\\ &\quad \overset{\text{rank}}{=} P(R|E,e)\cdot P(e|E)\cdot P(T|e) && (10) \end{aligned}$$

In (8) we assume that the type is independent of the source entity E and the relation R. Next, we rewrite $P(E,R|e)$ to $P(R|E,e)$ so that it expresses the probability that relation R is generated by the two (co-occurring) entities (e and E). Finally, we rewrite $P(E,e)$ to $P(e|E)\cdot P(E)$ in (9) as the latter is a more convenient form for estimation, and we drop $P(E)$ in (10) as it does not influence the ranking (for a fixed source entity E). Given equation (10) we are left with the following components:

- $P(e|E)$: pure co-occurrence model,
- $P(R|E,e)$: context dependent model, and
- $P(T|e)$: type filtering.

Pure co-occurrence model We use χ^2 to express the strength of associations between the source entity and candidates, without considering the nature of their relation:

$$\text{cooc}_{\chi^2}(e,E) = \frac{N\cdot(\,c(e,E)\cdot c(\overline{e},\overline{E}) - c(e,\overline{E})\cdot c(\overline{e},E)\,)^2}{c(e)\cdot c(E)\cdot(N-c(e)\,)\cdot(N-c(E)\,)},$$

where N is the total number of documents, and $\overline{e}$, $\overline{E}$ indicate that e, E do not occur, respectively (i.e., $c(\overline{e},\overline{E})$ is the number of documents in which neither e or E occurs).

Context-dependent model We take the context surrounding a source entity and candidate entity into account by constructing a co-occurrence language model (θ_{Ee}) from the contexts in which a source entity and candidate co-occur. By assuming independence between the terms in the relation R we arrive at the following estimate:

$$P(R|E,e) = P(R|\theta_{Ee}) = \prod_{t\in R} P(t|\theta_{Ee})^{n(t,R)}, \tag{11}$$

where $n(t,R)$ is the number of times t occurs in R. To estimate the co-occurrence language model θ_{Ee}, we collect the snippets in which the two entities co-occur into a pseudo document, which we name a co-occurrence document d_{Ee}, and obtain term probabilities as follows:

$$P(t|\theta_{Ee}) = \frac{n(t,d_{Ee}) + \mu\cdot P(t)}{\sum_{t'} n(t',d_{Ee}) + \mu}, \tag{12}$$

where $n(t,d_{Ee})$ is the number of times t appears in the co-occurrence document d_{Ee} of source entity E and candidate e, $P(t)$ is the collection language model, and μ is the Dirichlet smoothing parameter, set to the average length of the co-occurrence documents for a given source entity.

Type filtering To perform filtering based on target type we use Freebase[1] as our knowledge source. Freebase is a collection of data sources, i.e., DBPedia and WordNet, structured using a single format

[1] http://wiki.freebase.com

(schema). Each entity in Freebase has a unique ID and the ID is linked to objects that are either literals or other entities. For example the entity with ID "/guid/9202a8c04000641f8000000000028f64" is linked to the literal "Michael Schumacher" via a name predicate, but also to other entities such as "Ferrari". To maintain the link to the knowledge source from which an entity originates the entity ID in the original data source is kept as a literal, i.e., its DBPedia URI.

We first map a candidate entity to an entity in Freebase by exact sting matching, if no match is found we do not consider the entity as candidate. For each entity found in Freebase we find the set of category labels it is linked to. For the target types, i.e., person, organization, product or location, we create a manual mapping of the most frequent labels to each type. Given this mapping we estimate $P(T|e)$ as follows:

$$P(T|e) = \begin{cases} 1 & \text{if } lab(e) \cap lab(T) \neq \emptyset \\ 0 & \text{otherwise,} \end{cases}$$

where $lab(e)$ is the set of Freebase labels associated with entity e and $lab(T)$ is the set of labels mapped to target type T.

Hompage finding To perform homepage finding we again use Freebase as source of URLs of entity homepages. In addition we submit entity strings to a web search engine and collect the top 10 hits. If we find a match for the Freebase URL in ClueWeb then we use it as the entity homepage, otherwise we take the highest ranked URL returned by the search engine that is found in ClueWeb. If no homepage is found we remove the entity from the candidates.

4.1.1 Pre-processing

The main challenge this year was to construct a related entity finding system that runs on the complete ClueWeb Cat A collection. For Named Entity Recognition (NER) we used the Stanford tagger Finkel et al. (2005) which resulted in almost 2 billion unique entities. Removing entities with frequency lower than 5, replacing diacritics and removing entities longer than 128 characters, left us with 148 million entities. We then replaced entities by a unique identifier and indexed the documents using the Indri toolkit [2] resulting in 10 indexes one for each part of ClueWeb Cat A. We were able to perform these computationally expensive operations because we had access to the Big Grid cluster [3].

4.1.2 REF Runs

In our runs we experimented with the size of the window in which an entity is considered to co-occur with a source entity. To control the amount of context considered by the model we use a window size parameter indicating the number of tokens (white space delimited strings) considered to the left and right of the source entity. For example: a window size of 500 considers 500 tokens to the left and 500 to the right of the source entity making up a total context size of maximal 1000 tokens.

We also experiment with entity normalization and again turn to Freebase to provide a mapping of variants to a canonical entity. We consider all strings that are linked to the same Freebase ID with a name predicate as variants of the same entity.

In our manual runs we manually removed stop words from the queries and added the base forms of verbs and singular forms of plural terms to the narrative. For example we reformulated the query: "Carriers that BlackBerry makes phones for" to "carriers carrier blackberry make makes phone phones".

ilpsA500 An automatic recall oriented run with a window size of 500. Given the large context the source entity is less likely to miss any of the relevant target entities.

ilpsM50 A manual precision oriented run with a window size of 50. The intuition is that entities that occur close to the source entity are more strongly associated with it.

ilpsM50var A manual precision oriented run where entity variants are collapsed into a single variant. By collapsing variants we obtain more reliable co-occurrence statistics and a more complete context for an entity.

ilpsM50agfil A manual precision oriented run where we apply a more strict filtering strategy: entities with labels that belong to the target type but also to other types are ranked lower based on the number of non-target type category labels they have.

4.2 Entity List Completion Approach

In our participation of the ELC task we investigate two intuitions. First, candidate entities that are more similar to the example entities should be ranked higher than entities that are less similar. Second, entities that link to objects that share words with the narrative, source entity and target type should be ranked higher than entities that do not.

Our first challenge was understanding the structure of the Billion Triple Corpus (BTC) which consists of data structured as RDF triples, where an RDF triple consists of a "subject", "predicate" and an "object". An entity in the Billion triple corpus has a unique ID and can serve as subject. The object is either an entity ID or a literal such as the entity name, birth date or even its DBPedia description. Predicates denote a relation between the subject and the object. The terms in the predicate can explicitly mention the relation such as in the case of the predicate "birth place of" but

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

[3] http://www.biggrid.nl/

may also be a more abstract representation without any terms relevant to the relation.

In order to collect candidate entities we first obtain all objects that have one of the example entities as subject to form the set of example objects O_{ex}. Candidates are then all entities that have an object in common with the example entities:

$$C = \{s : triple(s,p,o), o \in O_{ex}\},$$

where $triple(s,p,o)$ is a triple in the BTC with subject s, predicate p and object o.

ilpsSetOL: baseline run To rank a candidate entity ($c \in C$) we take the predicates and objects from the set of triples which have the candidate as subject and store them as predicate-object tuples:

$$T(c) = \{(p,o) : triple(s,p,o), s = c\}.$$

We do this similarly for each example entity and calculate the candidate ranking score as the average Jaccard coefficient between the candidate entity and the example entities:

$$avgJ(c,Ex) = \frac{1}{|Ex|} \sum_{ex \in Ex} \frac{|T(c) \cap T(ex)|}{|T(c) \cup T(ex)|},$$

where ex is an example entity from the set of examples Ex and $|Ex|$ is the size of the example set.

ilpsSetOLnar: baseline combined with word overlap In our second run we combine the set overlap with the word overlap between the set of terms contained in an entity's predicate object tuples ($W(c)$) and the set of terms from the source entity, narrative and target type ($W(E,R,T)$). We calculate the word overlap as follows:

$$wo(W(c),W(E,R,T)) = \frac{|W(c) \cap W(E,R,T)|}{|W(E,R,T)| \cdot 2}.$$

We then combine the word overlap score with the average Jaccard coefficient to obtain our ranking score:

$$score(c) = avgJ(c,Ex) \cdot wo(W(c),W(E,R,T))$$

4.3 Results

By the time of writing of these working notes no results are available yet for the REF or ELC runs.

5 Relevance Feedback Track

Typical relevance feedback algorithms consider feedback documents as generative models from which to sample terms. We find that simply applying out-of-the-box relevance feedback algorithms to the single example document is not effective; such feedback algorithms degrade retrieval performance. To address this issue, we have implemented a novel model and our focus in our TREC participation this year is to evaluate its performance.

No results have been provided to the participants at the time of writing. As such, we limit our discussion to describing our algorithm.

Our algorithm makes use of the moderated contents of Wikipedia as a pivot language. Wikipedia articles can be created by anyone, but they are typically moderated by a relatively small group of volunteers. Moreover, Wikipedia has extensive guidelines in place, pertaining to the correct use of grammar and style. As a consequence (and unlike common web pages), the language used in each article tends to be "clean" and to the point. It is this particular feature of Wikipedia that we use to influence the estimation of the language model of web pages. The expanded query language model is interpolated with the initial query to obtain a final representation of the user's information need.

6 Conclusion

We have described the participation of the University of Amsterdam's ILPS group in the session, entity, and relevance feedback track at TREC 2010. No results are available yet for the Entity and Relevance Feedback tracks and consequently we can provide no analyses or conclusions about our approaches. In the Session track we focused on the use of blind relevance feedback for taking information about an original query into account when retrieving documents for a follow-up query. Our preliminary results indicate that the success of this approach depends on the type of query reformulation.

7 Acknowledgments

This research was supported by the DuOMAn project carried out within the STEVIN programme which is funded by the Dutch and Flemish Governments (http://www.stevin-tst.org) under project number STE-09-12, and by the Netherlands Organisation for Scientific Research (NWO) under project numbers 640.001.501, 640.002.501, 612.066.512, 612.061.814, 612.061.815, 640.004.802, and by the Virtual Laboratory for e-Science project (http://www.vl-e.nl), which is supported by a BSIK grant from the Dutch Ministry of Education, Culture and Science and is part of the ICT innovation program of the Ministry of Economic Affairs.

8 References

Balog, K., Weerkamp, W., and de Rijke, M. (2008). A few examples go a long way: constructing query models from

elaborate query formulations. In *SIGIR '08*.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022.

Bron, M., Balog, K., and de Rijke, M. (2010). Ranking related entities: Components and analyses. In *CIKM '10*.

Carbonell, J. and Goldstein, J. (1998). The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *SIGIR '98*, pages 335–336, New York, NY, USA. ACM.

Finkel, J. R., Grenager, T., and Manning, C. (2005). Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL '05*, pages 363–370.

Lafferty, J. and Zhai, C. (2001). Document language models, query models, and risk minimization for information retrieval. In *SIGIR '01*.

Zhai, C. and Lafferty, J. (2001). Model-based feedback in the language modeling approach to information retrieval. In *CIKM '01*.