UMass at TREC WEB 2014: Entity Query Feature Expansion using Knowledge Base Links

Laura Dietz and Patrick Verga
University of Massachusetts
Amherst, MA, U.S.A.
{dietz,pat}@cs.umass.edu

Abstract

Entity linking tools predict links between entity mentions in text and knowledge base entries. In this work we leverage the rich semantic knowledge available through these links to understand relevance of documents for a query. We focus on the ad hoc task on the category A subset and demonstrate the benefit of entity-centric approaches even for non-entity queries like "dark chocolate health benefits".

1 Introduction

Recent advances in automatic entity linking and knowledge base construction have resulted in entity annotations for document and query collections. For example, Google's FACC1 data set [3] contains entity annotations for all documents in the ClueWeb collection. Understanding how to leverage these entity annotations embedded in text to improve ad hoc document retrieval is an open research area.

Query expansion is a commonly used technique to improve retrieval effectiveness. Most previous query expansion approaches focus on text, mainly using unigram concepts. In this TREC submission, we follow up on our SIGIR paper [2], where we propose a new technique, called entity query feature expansion (EQFE). Our approach is to enrich the query with features from relevant entities and their links to knowledge bases, including structured attributes and text. We use a graphical model that performs joint inference on the relevance of latent entities and relevance of documents from target collection.

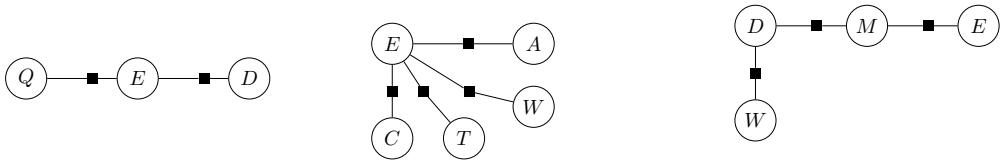
2 Approach

We assume availability of a general purpose knowledge base and the capability of establishing entity links from mentions in documents to the knowledge base. For this submission we use a Wikipedia dump from January 2012 (Wiki WEX dump), which we augment with extracted name variants from Wiki-internal anchor text and an anchor text resource from the open web [6], and merged with Freebase names and types. We index all knowledge base articles with the retrieval engine Galago.[1] We use entity links provided in the FACC1 dataset [3] for the ClueWeb12 corpus Category A and B.

We index all ClueWeb 12 Category A documents with Indri [2] and merge them with entity link annotations from the FACC1 dataset.

We devise a retrieval model is not just based on keywords in the query and keyword expansion, but that further reasons about which entities are relevant and then uses entity-information to rank documents. Figure 1 summarizes our retrieval model in factor graph notation, where each factor (black box) assigns a compatibility score to settings of indicent variable. We are using log-linear

[1] lemurproject.org/galago
[2] lemurproject.org/indri



(a) Joint retrieval problem, given Q.

(b) Representation of entity E.

(c) Representation of documents D.

Figure 1: Graphical model for joint document and entity retrieval model.

factors that are formed through an inner product of a feature vector ϕ with a parameter vector that is to be determined. In this section we explain three parts of the model: 1) how to retrieve entities that are relevant for the query; 2) given entities and a query, how to retrieve documents that are relevant; and 3) how to identify the relevant aspects of each entity.

2.1 Joint Entity Retrieval

We found different ways to derive indicators for relevant entities. One indicator is to perform probabilistic document retrieval with the query Q against the Galago index of knowledge base documents. We refer to this distribution as $E \sim \phi_{\text{kb}}(Q, E)$.

Alternatively, we can derive indicators for relevant entities through a pseudo-relevance feedback approach on entity links in ClueWeb documents. Using conventional keyword retrieval models, we retrieve an initial distribution over documents $D \sim \phi_{\text{ir}}(Q, D)$. In this work, we consider the sequential dependence model [5] with relevance model query expansion (SDM-RM3) for the initial document distribution [4]. Extending the idea of the relevance model to bags-of-entities, we derive a distribution over entities as a document-weighted mixture model over document-wise entity language models.

$$E \sim \phi_{\text{doc}}(Q, E) = \sum_d p(d|Q) \frac{\#\{e \in d\}}{\#\{\cdot \in d\}}$$

In order to prefer entities that are close to query keywords across many documents, we propose an alternative look onto the documents. Inspecting high ranked documents from the initial ranking, we consider the context surrounding each entity link using varying windows of 8 and 50 terms. Contexts are grouped by knowledge base entry, and all contexts surrounding the same entity are merged into one pseudo document which we call the entity contexts. We can score these entity contexts with the initial retrieval model for how relevant the entity is for the query. We refer to this entity distribution as $E \sim \phi_{\text{ecm}}(D, Q, E)$.

A last indicator can be derived by applying an entity linking tool to the query text and thereby identifying entity mentions in the query. For instance in the example query "obama family tree", the mention "obama" can be linked to the Wikipedia entry "Barack_Obama". In previous work we noticed that most entity linking tools do not work well on query text, due to lack of grammatical structure. As TREC web track queries are unlikely to mention entities directly, we omit this kind of source for this submission.

Given a parameter vector (which is to be determined), we can aggregate the different entity indicators into one distribution over entities $p(E|Q)$ as in Figure 1a.

2.2 Joint Document Retrieval

The joint document retrieval model combines keyword-based retrieval models with entity-based retrieval models. We use different state-of-the-art keyword-based probabilistic retrieval models such as the sequential dependence model, a query likelihood model, and relevance model query expansion. With weight parameters, these can be integrated into one distribution over documents, e.g. $D \sim \phi_{\text{ir}}(Q, D)$.

We combine these scores with additional indicators that take the distribution of query-relevant entities $p(E|Q)$ into account. We exploit that each entity has distributions over name aliases, words,

types, and an entity id associated. When mixed according to $p(E|Q)$, we can use these different distributions to derive a new retrieval model.

For instance, we derive a distribution over categories C from the knowledge base as

$$C \sim \int \phi(E, Q)\phi(E, C)\, dE,$$

where $\phi(E, C)$ denotes a distribution over Wikipedia category labels for the entity E which is smoothed with the collection-level category distribution. We use this distribution over categories as query expansions as well as for features for supervised re-ranking—a parameter is the cut-off for number of entities E considered.

Likewise, distributions over name aliases A, entity identifiers E, ontological Freebase types T, and words W (from the Wikipedia article) can be derived. Retrieval models over words W and aliases A match against the full text of the web documents. Since entity linking annotations already exist for all documents are already entity linked, entity IDs E can be matched against entity link targets, as well as types T and categories C can be matched against types of link targets. For name aliases we use the sequential dependence model with collection level smoothing; for entities E, words W, categories C, and types T we use a query likelihood model with collection level smoothing.

The score of a document D under each respective retrieval model can be turned into an entity-inspired feature $\phi(Q, D)$ over each vocabulary type or, given a weight vector, interpreted as a combined retrieval model.

2.3 Learning Query-specific Entity-information

So far, we derived entity-typical information directly from the knowledge base article. This follows the assumption that if an entity is relevant, then all of its aspects are equally relevant. This is not necessarily true. For example, the entity "Agriculture" is clearly relevant for a query about farming in a developing country, but its aspect on large-scale corn farming in the United States is not relevant.

So far all entity-characteristic words W are taken form the Wikipedia article, which is the basis of the WikiRM model. An entity-independent source is a relevance model estimated from retrieved documents [4]. Here, we suggest a third option; using the entity context derived through entity links. We build a collection-smoothed language model over context surrounding an entity's link to derive an alternative distribution over words W.

We also consider that depending on the context, an entity might be referred to via different names, e.g. referring to its function or nickname. We also consider the case of entities in documents that do not have an entry in the knowledge base. Both cases are addressed by deriving a distribution over named entity mentions M from documents through pseudo-relevance feedback.

2.4 Learning Procedure

In Sections 2.1 through 2.3, we discussed several relevance indicators for entities given the query and documents given entities.

It is not realistic to expect availability of relevance data for entities, as typical IR benchmarks like the TREC Web training queries from 2013 only include relevance judgments for documents. We suggest a learning procedure that integrates over latent entity variables E by computing the cross product of entity-query features and document-entity features.

We denote document-entity features through the vocabulary that is matched in the document, i.e. entity link with identifier E, name aliases A, and unlinked entity mentions M, as well as Wikipedia category C and Freebase type T as a surrogate through the enity identifier.

For each of these vocabularies a query-indicative distribution can be derived through different entity-relevance distributions. In particular through issuing the query agains the knowledge base ("kb"); through documents of a pseudo-relevance feedback pass ("doc"), and the entity context ("ecm").

The cross-product of these features is further merged with different traditional retrieval models, such as the baseline retrievals, query expansion and spam scores provided by the organizers.

Given relevance judgments on the document level, we can train a supervised re-ranking model. In this submission we use RankLib[3] with coordinate ascent.

3 Experimental Evaluation

We use an Indri index of the ClueWeb12 Category A collection created using default parameters. We do not apply spam filtering on the ClueWeb12 documents, because we noticed many relevant documents with spam score 0. For all queries from the 2013 training set and the 2014 test set we derive a pooled corpus using the top 10,000 documents retrieved by the following models:

- Query Likelihood; provided by organizers
- Query Likelihood with RM3; provided by organizers
- Terrier; provided by organizers
- Sequential Dependence Model (SDM) [5]; contributed as manual run
- WikiRM1 baseline (expansion for SDM); contributed as manual run

WikiRM is an external feedback model which uses the Wikipedia knowledge base as a text collection. WikiRM1 extracts terms from the highest ranked Wikipedia articled returned by querying the knowledge base and to be used as expansion terms for a sequential dependence model on the original query terms (SDM-RM3). Models similar to WikiRM1 were shown to be effective for these collections in previous work [1, 7]. While WikiRM1 uses Wikipedia as an external corpus, it does not leverage entity links, entity names, categories, or ontological types from the knowledge base.

We pool the top 10,000 results of each retrieval model and merge the pooled documents with entity link annotations from the FACC1 data set.

3.1 Submitted Runs

We submitted three automatic runs, and two baselines as manual runs. All runs use a knowledge base index built from a January 2012 Wikipedia dump and entity links provided in the FACC1 annotations. The automatic runs were created with supervised reranking using RankLib's coordinate ascent optimized for ERR@20 with no normalization and 1 start.

Our five runs are described below.

CiirAll1 Combination of all 40 features, all entity context features and all baseline features as listed in Table 1.

CiirSub1 and CiirSub2 Combination of a subset of 13 entity context features as marked with 'X' in Table 1.

CiirSdm (Manual Run) Indri sequential dependence model with standard parameters 0.8, 0.15, 0.05

CiirWikiRm (Manual Run) SDM with Wikipedia expansion model (generated with Indri). Parameters: SDM default parameters 0.8, 0.15, 0.05; RM weight 0.8/0.2

3.2 Results on Train/Validation data

We used a restricted training procedure due to time constraints before the submission. We trained the supervised models on a very limited training collection consisting the pooled top 100 documents retrieved by each method. Further, we used one re-start with coordinate ascent.

We measure the performance of each feature individually and the training set performance of the combined runs in terms of ERR-IA@20, ERR-IA@10, and MAP-IA.

Results on methods and invidual EQFE features on the training set are presented in Table 1. We see that all contributed methods outperform the best baseline contributed by the organizers by 20% in ERR-IA@10. Also, our automatic run All1 is only marginally better than Sub2. All1 includes features from the baselines contributed by the organizers while Sub2 is trained only on a subset of the features. The subset of features are denoted by an 'X' in the last column of Table 1.

[3] sourceforge.net/p/lemur/wiki/RankLib/

Run / Feature	$\phi(E, D)$	$\phi(Q, E)$	Sub1/2	ERR-IA@10	ERR-IA@20	MAP-IA
CiirAll1				0.640817	0.651061	0.145293
CiirSub2				0.646	0.65	0.192
CiirSub1				0.585188	0.593584	0.182323
`terrier-baseline`				0.488953	0.499958	0.151134
`CiirWikiRM (manual run)`			X	0.441862	0.449224	0.146358
`CiirSdm (manual run)`			X	0.393408	0.402837	0.159584
`rm-baseline`				0.370645	0.375283	0.126118
feature-contextFeatsentity-8	W	ECM	X	0.357623	0.366816	0.106399
`ql-baseline`				0.355609	0.365256	0.135077
`ql-spam-filtered`				0.348758	0.360416	0.10684
`rm-spam-filtered`				0.343325	0.353163	0.104951
feature-contextFeats-idQL-entity-50-20	E	ECM	X	0.333337	0.342083	0.057906
feature-contextFeats-idQL-entity-8-20	E	ECM		0.321489	0.332868	0.05712
feature-contextFeatsentity-50	W	ECM		0.321362	0.33268	0.092048
feature-names-mention-numEnts20	M	doc	X	0.317012	0.325854	0.075703
feature-wikipedia-20	W*	kb	X	0.31368	0.32321	0.082976
feature-contextFeats-names-descentity-50	A	ECM	X	0.309812	0.317491	0.066934
feature-wikipedia-5	W*	kb		0.307967	0.315215	0.082003
feature-contextFeats-names-descentity-8	A	ECM		0.302711	0.315073	0.076
feature-linkedEnts-top1-idQl-20	E	doc	X	0.291321	0.298747	0.051835
feature-top1names-numEnts20	A	doc	X	0.286737	0.294396	0.065631
feature-names-mention-numEnts10	M	doc		0.283586	0.293901	0.063472
feature-wikipedia-1	W*	kb		0.278271	0.286035	0.076898
feature-collection-20 (RM1)				0.273644	0.282796	0.064824
feature-wikipedia-names-numEnts10	A	kb		0.244849	0.25628	0.067349
feature-wiki-idQL-50	E	kb	X	0.234009	0.243401	0.038373
feature-wikipedia-names-numEnts20	A	kb		0.227854	0.238997	0.070966
feature-wikipedia-names-numEnts5	A	kb		0.206007	0.219887	0.062106
feature-wiki-idQL-20	E	kb		0.203334	0.213179	0.037272
feature-top1-numEnts20	W*	doc		0.202596	0.207457	0.037763
feature-wiki-idQL-10	E	kb		0.195772	0.20508	0.03657
feature-wiki-idQL-1	E	kb		0.190529	0.199752	0.03292
feature-wikipedia-names-numEnts1	A	kb		0.171216	0.182375	0.055716
feature-top1-numEnts10	W*	doc		0.16566	0.17228	0.031786
feature-top1-numEnts1	W*	doc		0.159872	0.164304	0.040393
feature-wiki-categoryQl-1	C	kb		0.141777	0.152451	0.031231
feature-categoryQl-20	C	doc		0.139191	0.143512	0.026689
feature-wiki-typeQl-5	T	kb		0.085307	0.090245	0.009785
feature-wiki-typeQl-1	T	kb		0.073605	0.087839	0.015047
feature-wiki-categoryQl-5	C	kb		0.069046	0.074361	0.01673
feature-fbTypeQl-20	T	doc		0.060378	0.068659	0.015846
feature-cluespam				0.024334	0.030719	0.008195

Table 1: Performance of individual features, baselines (typewriter) and combined methods (bold), ordered by ERR-IA@20. The letters in the $\phi(E, D)$ column refer to the type of the information. W denotes words, E entity IDs, T types, C categories, M mentions, A name aliases, and $W*$ words from KB article through entity links. The $\phi(Q, E)$ column refers to the indicator for relevant entities used where doc refers to corpus documents, kb to knowledge base documents, and ECM to entity contexts.

3.3 Results on Test data

We applied the learned re-ranking model to the pool of the top 10,000 retrieved documents from each retrieval method. The difference in characteristics between the training (top 100) and test set (top 10,000) led to suboptimal results.

Table 2: Best/Worst queries for Sub1/2 in comparison to SDM.

(a) Best

Query	Title
271	halloween activities for middle school
255	teddy bears
270	sun tzu
274	golf instruction
291	sangre de cristo mountains
263	evidence for evolution
300	how to find the mean
262	balding cure
280	view my internet history
294	flowering plants

(b) Worst

Query	Title
264	tribe formerly living in alabama
295	how to tie a windsor knot
283	hayrides in pa
252	history of orcas island
287	carotid cavernous fistula treatment
259	carpenter bee
267	feliz navidad lyrics
299	pink slime in ground beef
278	mister rogers
289	benefits of yoga

Model	MAP	ERR@20	NDCG@20
SDM	4.18	9.15	12.61
WikiRM1	4.00	9.31	12.80
SDM-RM3	3.53	7.61	11.00
EQFE	4.67	10.00	14.61

(a) Results on 2013 Cat B.

Model	ERR@20	NDCG@20	α-nDCG@20
CiirAll1	0.25	0.15	0.64
CiirSub1	0.11	0.06	0.36
CiirSub2	0.12	0.07	0.36
CiirSdm	0.23	0.13	0.53
CiirWikiRm	0.21	0.12	0.55

(b) Results on 2014 Cat A.

We present the official results in Table 3b. The reranking method with all features outperforms the SDM and WikiRM baselines. In contrast to the results on the training set, reranking based on feature subsets performed substantially worse achieving only about half the ERR@20 of the other methods.

Analyzing correlations in query-by-query performance, we notice that the performance of Sub1 is highly correlated to performance of Sub2, i.e., Sub1 is doing well when Sub2 is also doing well. This indicates that the few restarts are unlikely to be the issue. We notice that for ten queries, Sub1/2 are at least 1.5 times as good as the SDM baseline (cf. Table 2a where we also display the best queries).

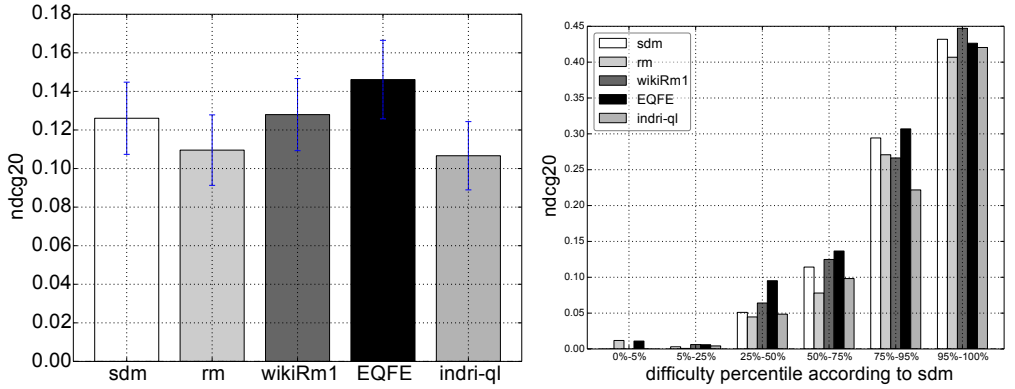
3.4 Crossvalidation experiments in Cat B

We also recap previous experiments on the ClueWeb12 category B subset with training queries from the 2013 dataset with five-fold-crossvalidation using the search engine Galago. The effectiveness of our query feature expansion is compared with sequential dependence model, the WikiRM1 method, and SDM expanded with Relevance Model (SDM-RM3), and Indri's query likelihood model (Indri-QL), as provided by the track organizers.

The overall retrieval effectiveness across different methods and collections is presented in Table 3a and Figure 2a. Our proposed EQFE model is the best performer on MAP for the ClueWeb12B collection. A paired t-test with α-level 5% indicates that the improvement of EFQE over SDM is statistically significant.

We further analyze whether the EQFE method improves particularly difficult or easy queries. To do that, we order queries by performance achieved by the SDM baseline. In Figure 2b we display the different difficulty percentiles, organizing the queries from most difficult to easiest. The 5% of the hardest queries are represented by the left-most cluster of columns, the 5% of the easiest queries in the right-most cluster of columns, the middle half is represented in two middle clusters (labeled "25%-50%" and "50%-75%").

This analysis shows that EQFE especially improves hard queries. EQFE outperforms all methods, except for the top 5% of the easiest queries. We achieve this result despite having on average 7 unjudged documents in the top 20 and 2.5 unjudged documents in the top 10 (in both the "5%-25%" and "25%-50%" cluster), which are counted as negatives in the analysis.



(a) Mean retrieval effectiveness with standard error bars on ClueWeb12B.

(b) Mean retrieval effectiveness across different query-difficulties, measured according to the percentile of the SDM method.

The WikiRM1 method, which is the most similar expansion method to EQFE, demonstrates the opposite characteristic, outperforming EQFE only on "easiest" percentiles.

4 Conclusions

We presented results from our Entity Query Feature Expansion approach [2] applied to data from the TREC web track 2013 Cat A, the test set from 2014 Cat A, and cross-validation experiments coducted on 2013 Cat B.

Acknowledgements

This work was supported in part by the Center for Intelligent Information Retrieval and in part by IBM subcontract #4913003298 under DARPA prime contract #HR001-12-C-0015. 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] Michael Bendersky, Donald Metzler, and W. Bruce Croft. Effective query formulation with multiple information sources. In *Proceedings of the fifth ACM international conference on Web search and data mining*, WSDM '12, pages 443–452, New York, NY, USA, 2012. ACM.

[2] Jeffrey Dalton, Laura Dietz, and James Allan. Entity query feature expansion using knowledge base links. In *SIGIR*, 2014.

[3] Evgeniy Gabrilovich, Michael Ringgaard, and Amarnag Subramanya. FACC1: Freebase annotation of ClueWeb corpora, version 1 (release date 2013-06-26, format version 1, correction level 0), June 2013.

[4] Victor Lavrenko and W. Bruce Croft. Relevance-Based Language Models. In *Proceedings of the ACM SIGIR 01 conference*, pages 120–127, 2001.

[5] Donald Metzler and W. Bruce Croft. A Markov random field model for term dependencies. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '05, pages 472–479, New York, NY, USA, 2005. ACM.

[6] Valentin I. Spitkovsky and Angel X. Chang. A Cross-Lingual dictionary for english wikipedia concepts. In *Conference on Language Resources and Evaluation*, 2012.

[7] Yang Xu, Gareth J. F. Jones, and Bin Wang. Query Dependent Pseudo-relevance Feedback Based on Wikipedia. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 59–66, New York, NY, USA, 2009. ACM.