Document knowledge base linking for information retrieval

[Blind]

ABSTRACT

Entity linking and wikification are the tasks of matching mentions of an entity, such as a person, place, or organization, or a concept with its representation in a knowledge base such as Wikipedia. While there has been some investigation into use of entity linking in information retrieval, its usage may be hampered by the computational expense of constructing accurate entity annotations on large corpora, the frequent need for training data to construct entity links, and the ambiguity involved in real-world entity linking. We present a method by which a "bag of links" from a document to a knowledge base may be generated using standard information retrieval techniques with Wikipedia. We also suggest a document expansion model that employs these links, which is effective in improving retrieval results.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]

1. INTRODUCTION

Entity linking and wikification have been the subject of much research, spurred recently by the Text Analysis Conference (TAC) Knowledge Base Population (KBP) entity linking track. Systems are designed to link mentions of entities or concepts found in the text to their representations in a knowledge base such as Wikipedia[1] or Freebase[2].

These links enrich documents by connecting knowledge base information such as hyperlink graphs, entity membership in categories, and additional text to the annotated documents. It seems reasonable that information retrieval (IR) systems might take advantage of this added source of information to improve the quality of search results. Unfortunately, producing high quality entity annotations is often a computationally expensive process. In addition, these systems generally require training data, which increases the barrier to use. Much entity linking and wikification research has also assumed more amenable circumstances that complicate their application to real-world text. For example, until 2014, the KBP entity linking track supplied explicit mention boundaries to participants.

For the purposes of ad hoc document retrieval, however, the granularity of linking entities to specific mention spans in the text is not necessarily required. Since retrieval is concerned with scoring documents in their entirety, we may simplify the task from mapping *mentions* to entities, as is more common, to mapping *documents* to entities. We argue that such a "bag of links" provides many of the same benefits as more fine-grained entity linking or wikification processes for the purposes of ad hoc document retrieval while improving computational efficiency and robustness against overfitting. This is because explicitly mentioned entities are often not directly related to the subject matter of documents: for example, a news article may explicitly mention the publisher of a book, which is less useful for document expansion purposes than unmentioned entities relating to the book's subject matter.

In this paper, we present a document expansion retrieval model to incorporate a "bag of links" into the document retrieval task. This model is in contrast to prior work applying entity links to IR, which has focused on *query* expansion. Our document expansion model yields improvement over IR baselines.

2. RELATED WORK

2.1 Entity Linking, Wikification, and Entity Retrieval

Entity linking and wikification have been studied extensively, particularly in the context of TAC KBP [10]. Entity linking systems often exploit knowledge base structure to help match and disambiguate entities. For example, Cucerzan [5] employs Wikipedia disambiguation pages and redirection pages to help identify various entity surface forms. Most systems employ some form of context, which often refers to the co-occurrence of entities as evidence for disambiguation [15, 6]. Alternatively, context may refer to the text surrounding an entity mention, which can be used to disambiguate knowledge base entries [14, 6].

The related area of entity retrieval is also relevant to our work, e.g. [1, 4]. Entity retrieval refers to the task of retrieving entities, rather than documents, in response to a specified information need. This was studied at the TREC entity retrieval track [2]. Though entity retrieval tasks differ

[1]http://wikipedia.org
[2]http://freebase.org

CIKM 2016 Indianapolis, Indiana, USA

ACM ISBN 978-1-4503-2138-9.

DOI: 10.1145/1235

from ours in that entities are explicitly requested by the user, the querying of a corpus to retrieve entities is conceptually similar to much of our knowledge base linking approach.

2.2 Document Expansion in IR

Prior applications of knowledge base links in information retrieval use entity links in a feedback process for query expansion. Xiong and Croft, for example, employ FACC1[3] annotations on the document collection with supervised learning to expand queries [19]. Other researchers use entity links identified in query texts, rather than documents, to aid in expansion [20, 3]. Still others use entity links in *both* the query and documents [7, 12] to expand queries. Apart from entity linking, knowledge bases have also been used for query expansion in several related tasks, such as document filtering [18] and blog search [8, 16].

In contrast, our approach is concerned with *document* expansion. This expansion entails linearly interpolating the document language model with a second model estimated from the knowledge base links, discussed further in Section 4. Our idea is closely related to Wei and Croft's LDA-based document model, which smooths the document language model with latent Dirichlet allocation probabilities [17]. Liu and Croft's *CBDM* model performs a similar type of document expansion by interpolating the probability of a query term in a document cluster with its probability in the document [13].

3. CONSTRUCTING KNOWLEDGE BASE LINKS

3.1 Underlying Retrieval Model

Throughout this paper we rely on the language modeling retrieval framework [11], though this is not strictly necessary and imposes no particular mathematical constraints on our approach.

More specifically, our framework for all of the retrievals carried out in this work is the query likelihood (QL) ranking method. Given a query Q and a document D, we rank documents on $P(Q|\theta_D)$, where θ_D is the language model (typically a multinomial over the vocabulary V) that generated the text of document D. Assuming independence among terms and a uniform distribution over documents, each document is scored by

$$\log P(Q|D) = \prod_{w \in Q} P(w|Q) \cdot \log P(w|\theta_D). \quad (1)$$

We follow standard procedures for estimating the probabilities in Eq. 1. We simply use the maximum likelihood estimate of $\hat{P}(w|Q) = \frac{c(w,Q)}{|Q|}$ where $c(w,Q)$ is the frequency of word w in Q. For $P(w|\theta_D)$ we estimate a smoothed language model by assuming that document language models in a given collection have a Dirichlet prior distribution:

$$\hat{P}(w|\theta_D) = \frac{c(w,D) + \mu \hat{P}(w|C)}{|D| + \mu} \quad (2)$$

where $\hat{P}(w|C)$ is the maximum likelihood estimate of the probability of seeing word w in a "background" collection C (typically C is the corpus from which D is drawn), and $\mu \geq 0$ is the smoothing hyper-parameter.

[3]http://lemurproject.org/clueweb09/FACC1/

3.2 Linking with Document Pseudo-Queries

To find candidate entities to "link" to a given document D, we begin by treating the text of D as a pseudo-query which we pose against a collection of entities C_E. To transform a document into a pseudo-query we apply two transformations. First we remove all terms from D that appear in the standard Indri stoplist[4]. Next, we prune our pseudo-query by retaining only the $0 < k \leq |D|$ most frequent words in the stopped text of D. The integer variable k is a parameter that we choose empirically. Let Q_D be the pseudo-query for D, consisting of the text of D after our two transformations.

We obtain a list of candidate entities by running Q_D over an index of our knowledge base, C_E, where each entry in this index is the text of an entity E's knowledge base node. More formally, we rank the entities in our knowledge base against D using Eq. 1, substituting Q_D for the query and E_i—the text of the i^{th} entity—for the document. Let π_i be the log-probability for entity E_i with respect to D given by Eq. 1.

We now have a ranked list of tuples $\{(E_1, \pi_1), (E_2, \pi_2), ..., (E_N, \pi_N)\}$ relating knowledge base entry E_i to D with log-probability π_i. We take the top n entries where $0 \leq n \leq N$. We call these top entries $\mathcal{E}_D$ and designate them as our knowledge base links for D. Finally, we exponentiate each π_i and normalize our entity scores so they sum to 1 over the n retained entities. Assuming a uniform prior over entities, we now have a probability distribution over our n retained entities: $P(E|D)$.

Since this procedure does not depend on the query, we may compute $\mathcal{E}_D$ once at indexing time and reuse our knowledge base links across queries.

4. KB-LINKED RETRIEVAL MODEL

We would now like to incorporate our knowledge base links into a retrieval model over documents. Though many knowledge bases provide structured information such as hyperlink graphs and entity category information, in this work we focus only on the textual content supplied for each entry.

We assume that a query is generated by a mixture of the document language model θ_D and a language model θ_K representing the concepts linked from the knowledge base. We assume that θ_K can be estimated using the description texts of the linked knowledge base concepts $\mathcal{E}_D$. This mixture model may be expressed as:

$$\hat{P}^{\lambda}(Q|D) = \prod_{i=1}^{|Q|} (1-\lambda) P(q_i|D) + \lambda P(q_i|\mathcal{E}_D) \quad (3)$$

The larger λ is, the more we believe that the knowledge base concepts are responsible for generating Q, and the less we believe that the document is responsible for generating Q. We estimate $P(q_i|\mathcal{E}_D)$ in expectation:

$$P(q_i|\mathcal{E}_D) = \sum_{E \in \mathcal{E}_D} P(q_i|E) P(E|D). \quad (4)$$

Like $P(q_i|D)$, we estimate $P(q_i|E)$ as a Dirichlet-smoothed query likelihood, but using the description text for entry E

[4]http://www.lemurpro ject.org/stopwords/stoplist.dft

to estimate of the underlying model. By virtue of our entity-document scoring and normalization, we also have $P(E|D)$.

5. EVALUATION

5.1 Data

To perform knowledge base linking, we make use of the September 1, 2015 dump of English Wikipedia. We build an Indri[5] index over the Wikipedia page text. The text of each Wikipedia page also serves as the "description text" used in Eq. 4.

We test our approach using the TREC 2004 robust topics. These 250 topics are used with data from TREC disks 4 and 5. In addition, we use the AP newswire collection from TREC disks 1 and 2 with topics 101-200. Using slightly older test collections helps to ensure that the knowledge base's coverage includes information about the topics; as noted in [9], updates to knowledge bases may lag considerably behind the occurrence of events.

For comparison, we also report the results of our model using entity links produced by Apache Stanbol[6]. We process the test collections with Stanbol's default "enhancer," which supplies entity links and corresponding confidence scores. Though Stanbol links to DBpedia[7] resources, these generally correspond to Wikipedia pages and may be converted easily. We select the link with the highest confidence for each mention in a given document and limit the set of links to n, the number of entities used in our bag-of-links approach. These links form the set of document links, and their confidence scores may be normalized to provide an estimate of $P(E|D)$.

5.2 Runs

We produce three runs per collection:

- *baseline-ql*, a baseline query likelihood run
- *kb-ql*, incorporating knowledge base links using Eq. 3
- *stanbol-ql*, which uses Stanbol entity annotations in place of our document-level links.

We remove stop words in documents and entity descriptions for all runs. For the *kb* and *stanbol* runs, we retrieve the top 1000 documents per query based on the default Indri query likelihood implementation. We then re-rank these documents by incorporating their knowledge base links as described in Section 4.

5.3 Parameters

The various parameters required for our approach, along with their meanings and the values used in our experiments, are shown in Table 1.

We sweep across values of λ and n at intervals of 0.1 and 10 respectively to investigate the sensitivity of our model to these parameters. The results of these sweeps are shown in Figure 1 and discussed further in Section 6.

For this work, we set k heuristically. In principle, this parameter need not be limited beyond the length of the document; however, this would increase computation time significantly, so we have opted to set it to the specified value.

[5]http://www.lemurproject.org/indri/
[6]http://stanbol.apache.org/
[7]http://dbpedia.org

Param	Meaning	Value
k	The maximum number of document terms to use in constructing Q_D.	20
n	The maximum number of knowledge base entries in $\mathcal{E}_D$.	10-100
λ	Mixing parameter controlling the weights of $P(q\|D)$ and $P(q\|E)$	0.0-1.0
μ	Used for Dirichlet smoothing of both $P(q\|D)$ and $P(q\|E)$.	2500

Table 1: Parameter settings for the entity linking procedure and retrieval model

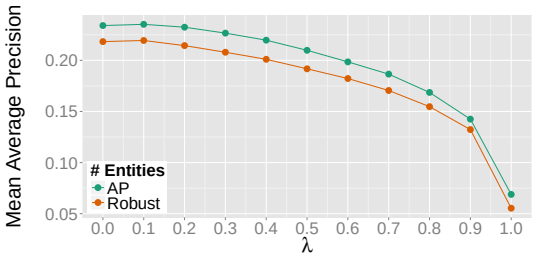

Figure 2: Sweeps over values of λ for AP and robust *stanbol* runs.

6. RESULTS

	Run	λ	n	MAP
Robust	*baseline-ql*	–	–	0.2183
	kb-ql	0.2	10	$0.2354^{\uparrow}$
	stanbol-ql	0.1	–	0.2194
AP	*baseline-ql*	–	–	0.2346
	kb-ql	0.2	20	$0.2680^{\uparrow}$
	stanbol-ql	0.1	–	0.2355

Table 2: The top-scoring runs and baselines by MAP. Values marked with ↑ are statistically significant improvements over baselines.

Retrieval performance of the baselines and top-scoring runs are shown in Table 2. Mean average precision (MAP) scores marked with ↑ are greater than the baseline run with statistical significance at $p < 0.05$ using a paired t-test. Note that baselines correspond to $\lambda = 0.0$.

As Figure 1 shows, performance of *kb* runs is not very sensitive to n, the number of linked entities. Though performance does vary somewhat based on n, a run's improvement or decline over the baseline depended only on the mixing parameter λ, with AP improving at $0.1 \leq \lambda \leq 0.9$ and robust at $0.1 \leq \lambda \leq 0.7$ for all values of n. Optimal performance for both collections occured at relatively small numbers of entities as shown in Table 2; this is a convenient result since it allows for more efficient document expansion.

Notably, our model did not perform well when Stanbol annotations were used as a source of knowledge base links, as seen in Figure 2. Both collections show very slight improvement at $\lambda = 0.1$, but this is not statistically significant and performance declines relative to the baseline at all other values of λ. We attribute this shortcoming to the overfitting of the Stanbol annotations to specific entity mentions in the

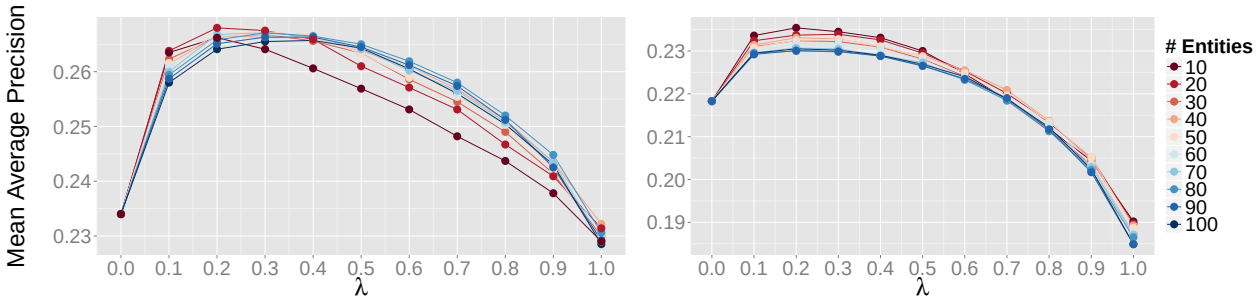

Figure 1: Sweeps over values of λ for AP (a) and robust (b) runs. $\lambda = 0.0$ is the baseline run scored only on the document text, while *lambda* $= 1.0$ is based entirely on the entity text.

text, rather than to concepts relevant to the document. Figure 2 does not present sweeps over n for *stanbol* runs because variations in performance at different values of n are negligible using Stanbol entity links. This is due to the low number of linked entities produced by Stanbol: an average of 5.96 and 4.70 per document for AP and robust respectively.

7. CONCLUSIONS

The results indicate that our approach for constructing knowledge base links between documents and Wikipedia produces useful data for document retrieval purposes. Our simple document expansion model that incorporates these links performs well compared to a query likelihood baseline. These outcomes support our argument that a "bag of links" to a knowledge base can provide helpful information for a document retrieval task. Further, the poor performance of our model using more traditional mention-to-entity links indicates that not only are document-to-entity links more efficient to produce, they also connect more useful knowledge base entries for the purposes of document retrieval.

In this paper, we have limited ourselves to using only knowledge base description text. However, knowledge base links provide a great deal more information. Future work may benefit from harnessing this knowledge base data, including hyperlink graphs and entity categories. Since our retrieval model performs *document* expansion, we also plan to investigate its utility when paired with *query* expansion techniques that employ knowledge base links.

8. ACKNOWLEDGMENTS

This work was supported in part by the US National Science Foundation under Grant No. [blind]. Any opinions, findings, conclusions, or recommendations expressed are those of the authors and do not necessarily reflect the views of the National Science Foundation.

9. REFERENCES

[1] S. F. Adafre, M. de Rijke, and E. T. K. Sang. Entity retrieval. *Recent Advances in Natural Language Processing*, 2007.

[2] K. Balog and P. Serdyukov. Overview of the TREC 2011 entity track. Technical report, NIST, 2011.

[3] W. C. Brandão, R. L. Santos, N. Ziviani, E. S. Moura, and A. S. Silva. Learning to expand queries using entities. *Journal of the Association for Information Science and Technology*, 65(9):1870–1883, 2014.

[4] M. Bron, K. Balog, and M. De Rijke. Ranking related entities: components and analyses. In *Proc. of CIKM*, pages 1079–1088. ACM, 2010.

[5] S. Cucerzan. Large-scale named entity disambiguation based on Wikipedia data. In *EMNLP-CoNLL*, volume 7, pages 708–716, 2007.

[6] J. Dalton and L. Dietz. A neighborhood relevance model for entity linking. In *Proc. of OAIR*, pages 149–156, 2013.

[7] J. Dalton, L. Dietz, and J. Allan. Entity query feature expansion using knowledge base links. In *Proc. of SIGIR*, pages 365–374. ACM, 2014.

[8] J. L. Elsas, J. Arguello, J. Callan, and J. G. Carbonell. Retrieval and feedback models for blog feed search. In *Proc. of SIGIR*, pages 347–354. ACM, 2008.

[9] J. R. Frank, M. Kleiman-Weiner, D. A. Roberts, F. Niu, C. Zhang, C. Re, and I. Soboroff. Building an entity-centric stream filtering test collection for trec 2012. In *TREC '12*. NIST, 2013.

[10] H. Ji, J. Nothman, and B. Hachey. Overview of TAC-KBP 2014 entity discovery and linking tasks. In *Proc. of TAC'14*, 2014.

[11] J. Lafferty and C. Zhai. Document language models, query models, and risk minimization for information retrieval. In *Proc. of SIGIR*, pages 111–119. ACM, 2001.

[12] R. Li, L. Hao, X. Zhao, P. Zhang, D. Song, and Y. Hou. A query expansion approach using entity distribution based on Markov random fields. In *Information Retrieval Technology*, pages 387–393. Springer, 2015.

[13] X. Liu and W. B. Croft. Cluster-based retrieval using language models. In *Proc. of SIGIR*, pages 186–193. ACM, 2004.

[14] R. Mihalcea and A. Csomai. Wikify!: linking documents to encyclopedic knowledge. In *Proc. of CIKM*, pages 233–242. ACM, 2007.

[15] V. Stoyanov, J. Mayfield, T. Xu, D. W. Oard, D. Lawrie, T. Oates, and T. Finin. A context-aware approach to entity linking. In *Proc. of AKBC-WEKEX*, pages 62–67. ACL, 2012.

[16] W. Weerkamp, K. Balog, and M. de Rijke. A generative blog post retrieval model that uses query expansion based on external collections. In *Proc. of ACL-IJCNLP*, pages 1057–1065. ACL, 2009.

[17] X. Wei and W. B. Croft. LDA-based document models

for ad-hoc retrieval. In *Proc. of SIGIR*, pages 178–185. ACM, 2006.

[18] C. Xie. An entity-centric query expansion approach to cumulative citation recommendation in knowledge base acceleration. In *International Conference on Fuzzy Systems and Knowledge Discovery*, pages 1355–1359. IEEE, 2015.

[19] C. Xiong and J. Callan. Query expansion with Freebase. In *Proc. of ICTIR*, pages 111–120. ACM, 2015.

[20] Y. Xu, G. J. Jones, and B. Wang. Query dependent pseudo-relevance feedback based on Wikipedia. In *Proc. of SIGIR*, pages 59–66. ACM, 2009.