Improving Entity Ranking for Keyword Queries

John Foley*, Brendan O'Connor† & James Allan*
Center for Intelligent Information Retrieval*, Statistical Social Language Analysis Lab†
College of Information and Computer Sciences
University of Massachusetts
{jfoley, brenocon, allan}@cs.umass.edu

ABSTRACT

Knowledge bases about entities are an important part of modern information retrieval systems. A strong ranking of entities can be used to enhance query understanding and document retrieval, or can be presented as another vertical to the user.

Given a keyword query, our task is to provide a ranking the entities present in the collection of interest. We are particularly interested in approaches to this problem that generalize to different knowledge bases and different collections. In the past, this kind of problem has been explored in the enterprise domain through Expert Search. Recently, a dataset was introduced for entity ranking from news and web queries from more general TREC collections.

Approaches from prior work leverage a wide variety of lexical resources: e.g., natural language processing and relations in the knowledge base. We address the question of whether we can achieve competitive performance with minimal linguistic resources.

We propose a set of features that do not require index-time entity linking, and demonstrate competitive performance on the new dataset. As this paper is the first non-introductory work to leverage this new dataset, we also find and correct certain aspects of the benchmark. To support a fair evaluation, we collect 38% more judgments and contribute annotator agreement information.

1. INTRODUCTION

Entities are the people, places and concepts that exist in the world and in a growing variety of knowledge bases (e.g., YAGO [17], DBPedia [1]). These entities are often naturally related to users' information needs. Some queries that are more factual in nature (e.g., "how tall is Dilma Rousseff", or "what is Petrobras?") can even be answered directly from these knowledge bases. In contrast, queries that still focus on entities but that have a deeper information need, (e.g., "politicians implicated in operation lava jato") could possibly be answered by a very rich knowledge base, but could also be satisfied by understanding the relevant entities within a target collection, such as a news or web corpora.

In this work, we explore methods for ranking general-knowledge entities in a target collection in response to a short, keyword query. Because we are interested in alternate domains and languages, we focus on developing a strong, simple approach to this problem that limits the amount of linguistic resources needed for system development. We note that entity retrieval over the knowledge-base graphs is less relevant to our work, since we consider a well-linked knowledge base to be a lexical resource that may not always be available, but research in this area is ongoing [27].

There is a large amount of recent work that automates the search and ranking of collection and query-relevant entities, particularly to support search tasks [9, 16, 26, 28, 21]. Even earlier, in the TREC Enterprise track, a similar problem was explored: finding people within a corpus that would be best to answer a particular question [7, 24, 3, 2].

Until recently, there was no dataset to explore this problem intrinsically on general web entities. Schuhmacher et al. introduced a new dataset based on the TREC Robust04 and Web13/14 data [25], and evaluated a learning-to-rank approach based upon features of entities linked in pseudo-relevant documents. These entity links could be compared to the sort of document-person co-occurrence or authorship information used in expert search.

The approach presented in Schuhmacher et al. is based around document-entity links, and it is quite successful [25]. We found that running the recommended entity linker was computationally prohibitive for our needs. While other entity linkers might be more efficient, generalizing to new languages is an open research area [18, 19, 15]. In particular: the organizers of TAC-KBP observe that a 90% linking accuracy requires about 20,000 query mentions labeled for training [19].

Two questions motivate our work:

1. Can we build a competitive entity ranking approach with only dictionary-based entity recognition [5], and resolve ambiguity at query-time?

2. Can we minimize the linguistic, lexical, and knowledge-base resources needed to implement such an approach?

In this work, we describe an approach that does not rely upon entity linking, yet performs remarkably well. We believe this indicates that we can create entity ranking systems in less-studied domains: especially on private data (e.g., expert search, a corporation's email, e-Discovery) and on low-

CIKM 2016 Indianapolis, IN

resource languages (e.g., the Cherokee language Wikipedia has only 720 articles[1].

2. ENTITY-TAGGING

We describe a simple tagging procedure that allows us to build up document-entity relationships without actually performing entity linking. This is important because although there has been substantial research on entity linking [23, 13], constructing an entity linker for novel types of entities or novel languages still requires non-trivial engineering and research effort [19, 15].

To give a further sense of the inherent ambiguity of our approach, and to demonstrate that this is not an alternative to entity-linking in the general case, we present the results of our system's tagger on the headline from a news article within the Robust04 corpus in Figure 1.

Figure 1: Actual output created by our tagging system. The number on each brace represents the number of entries in DBPedia with that title (before disambiguation terms). "HEAT" links to 26 possible articles, including albums, songs, and temperature.

We hypothesize that this dictionary-based *tagging* is sufficient for use in entity ranking since other documents that are related to the query (e.g., "el nino") can help us determine in what sense these mentions were intended ("HEAT" is temperature, not music), and ultimately, what entities are actually relevant to the query. Our approach does not actually need to know which entity was in any single document.

Compared to the output of a traditional entity linking system, which generates a single link for each candidate span, the output of our tagging approach is a bipartite graph of "title mentions" and entities. Our scoring models operate over these bipartite graphs in top ranked documents. An advantage of our approach is that despite having a high false-positive rate, it is not limited to entities of types supported by existing entity linkers.

2.1 Performance Implications

As a simple demonstration, we observed that our method is capable of tagging the half-million news documents in the Robust04 corpus in six minutes on a single core desktop computer, whereas the processing of a popular NER [22] toolkit on that corpus took more than 20 hours. These numbers, while anecdotal, at least suggest that our method would likely be more efficient than any existing solutions.

We believe that efficiency is necessary to an approach that generalizes to an evolving knowledge base: if the knowledge base is changing, you want it to be efficient to re-tag documents to incorporate recent changes.

3. ENTITY SCORING MODELS

In this section, we develop features for a model that will help us relate a users' query q, the pseudo-relevant documents in a target collection, and the entities in a "knowledge-base". Because we are interested in generalizing to evolving or private domains, we model our knowledge base as a collection of textual records, rather than a large collection of relations, where each record refers to an entity e.

Having previously tagged our documents to reflect all the possibly mentioned entities e, we can now dervie scores that reflect their importance based on their occurrence in the knowledge base $K(e,q)$, and in the top-relevant documetns D_q. Our features each make an assumption that some sort of similarity can be created between an entity record e, and the user's query q. In the case that this is not possible, we do briefly explore our features under the assumption that our knowledge base similarity function is equal to a constant, the no-KB case: $K_{\text{NONE}}(e,q) = 1$. Obviously this is weaker than assuming we have a language model similarity between our query and its knowledge-base entry. To further explore $K_{\text{ABS}}(e,q)$ and $K_{\text{WIKI}}(e,q)$ which correspond to only abstracts and full wikipedia pages.

Now we focus on leveraging the information in D_q, or the set of top documents under the retrieval model score $S(d,q)$ where d is an individual document, and q is our query.

We additionally have some other sources of information: the bag of entities probability for a document $M(e,d) = freq(e,d)/freq(*,d)$, and the collection-discounted version of the above: $M_D(e,d) = M(e,d)/(freq(e,*)/freq(*,*))$. M captures if the entity is frequent within a document, normalized for length, and M_D captures whether an entity is frequenty in a document normalized to the collection. The latter naturally captures the idea of stop-entities, such as "The".

Note that we base our features loosely on the term-weighting function Relevance Modeling [20], with entities substituted for terms in the top relevant documents, with additional discounting, or alternate combinations of cross-document importances.

First we have a comparison between giving importance to high-frequency entity occurrences $M(e,d)$, and entity occurrences that occur proportionally more locally than globally: $M_D(e,d)$. Both formulations capture important information.

$$f_1(e,q) = \sum_{d \in D_q} K(e,q)S(d,q)M(e,d) \tag{1}$$

$$f_2(e,q) = \sum_{d \in D_q} K(e,q)S(d,q)M_D(e,d) \tag{2}$$

Then we have our "product" features, which discount entities that do not occur in many documents much more strongly; one that mixes in the document score, and one that assumes all top documents are equally helpful. We actually calculate these products in log space in order to avoid numerical underflow.

$$f_3(e,q) = \prod_{d \in D_q} K(e,q)S(d,q)M_D(e,d) \tag{3}$$

$$f_4(e,q) = \prod_{d \in D_q} K(e,q)M_D(e,d) \tag{4}$$

We also include a document-independent feature, which is simply the knowledge-base similarity itself.

$$f_5(e,q) = K(e,q) \tag{5}$$

[1] `http://wikipedia.org/wiki/List_of_Wikipedias`; May 2016

Given each of these five features, we can substitute our knowledge-base similarity function for our three separate cases $K_{\text{NONE}}, K_{\text{ABS}}, K_{\text{WIKI}}$. Surprisingly, we do not find that the K_{WIKI} features dominate: the different levels of query-knowledge-base similarity are all important ranking features and are used in our RankLib models.

3.1 Static Features

Unlike previous work, we find that static, popularity priors are helpful - perhaps because we forego such features in an entity-linking step. The first static prior we add is a PageRank($\lambda = 0.15, \tau = 0.001$) feature calculated from the Wikipedia graph as specified in the textbook by Croft et al. [8]. The second feature is a popularity score derived from the FACC entity links provided for ClueWeb09 [14] (the chance a random entity is that entity).

4. EXPERIMENTAL SETUP

In these experiments, we use titles and abstracts from DBPedia[2], as reproducible preprocessing of Wikipedia.

4.1 Corrections in Evaluation Measures

In Schuhmacher et al [25], the evaluation numbers are miscalculated because non-relevant documents were given a weight of 1, instead of 0. The measures presented in prior work are therefore inflated by 10-20%, due to this miscalculation. Numbers presented in this work are lower still due to the increased number of true relevant labels collected, as discussed in the next subsection. We re-evaluate all measures based upon ranking files.

Instead of NDCG, we choose to evaluate with mean average precision (MAP) because only one of the datasets has multilevel relevance judgments. Although our models are trained on MAP, we additionally show P@5, because we collected judgments to that depth, and the ability to use our entity ranking for an extrinsic task is likely to be dependent on high early precision.

4.2 Entity Judgment extension

We pooled our best non-learning-to-rank models (along with runs from prior work [25]) and evaluated all methods fully to a depth of five. We did this in two stages.

Initially, we created an evaluation set as a pilot evaluation: following the methodology of Schuhmacher et al [25], we had graduate students in our lab create additional judgments while initially testing our new techniques, labeling only those documents that were entirely new to our approach, as nearly all of our top-5 documents were unjudged.

In order to obtain diversity in annotators, we had workers on Amazon's Mechanical Turk judge the (query, document) pairs from pooled runs. We limited workers to those having achieved Master's qualification, and offered $0.08 per label.

Information about the collected data is available in Table 1. The final relevance set we used involved majority-voting, where ties were broken toward relevance. We make our enhanced entity judgments available to future work[3].

5. RESULTS

We derive train/test splits using five-fold cross validation as split by the RankLib learning-to-rank software package [10].

[2] `http://wiki.dbpedia.org/Downloads2015-04`

[3] Suppressed for review.

Table 1: Analysis of Judgments Collected locally and on Mechanical Turk

	Robust04	ClueWeb12
Queries	25	22
Original Judgments [25]	1250	3306
Pilot Judgments	322	609
MTurk Judgments	730	1085
MTurk-MTurk agreement	75% of 48	92% of 64
Original-MTurk agreement	76% of 213	80% of 334
Pilot-MTurk agreement	78% of 309	80% of 427
Final Judgments	1946	4353

Table 2: Presentation of Results: (*) represents a statistical improvement over the previous row $p < 0.05$ with a pairwise randomization test. All unsupervised approaches are presented above the dividing line.

	Robust04		ClueWeb12	
Model	MAP	P@5	MAP	P@5
Best K_{NONE} (f_3)	0.152	0.536	0.062	0.317
Best K_{ABS} (f_2)	0.201*	0.648*	0.142*	0.411*
Best K_{WIKI} (f_2)	0.229*	0.752*	0.246*	0.654*
RankLib	0.299*	0.712	0.249	0.764*
RankLib+Static	0.370*	0.816*	0.304*	0.782
REWQ [25]	0.406	0.792	0.300	0.845*
Fusion	0.570*	0.856*	0.391*	0.818

Surprisingly, we find acceptable performance on news even without a knowledge base using a single feature, which suggests that an approach can be somewhat successful even without rich resources. We even see quite good performance using only one of our features with a knowledge base similarity based on the article text (f_2), which suggests that even without any annotations or entity links, an entity ranking system of reasonable quality is possible. Going to a machine learned model and then incorporating static features both yield statistically significant improvements.

On both datasets, there is no statistical difference between our run with static features and the best run from prior work [25]. We retrained the proposed model with our expanded set of judgments, expecting some improvement, but we were unable to match the run files they provided, possibly due to the non-determinism of the algorithms in RankLib, so we compare to the most optimistic scores.

A promising result for our simpler approach is that not only are we able to tie results from prior work (RankLib+Static $\approx$ REWQ), we are able to achieve statistically significant improvements over prior work when we include both sets of features into a "Fusion" model.

6. RELATED WORK

In the past, entity search, linking, and ranking have been thoroughly studied in TREC and TAC challenges [11, 4, 12, 6]. However, Schuhmacher et al. are the first to expand this problem to modern ad-hoc retrieval queries and test collections [25]. We note that entity retrieval over the knowledge-base graphs is less relevant to our work, since we consider a well-linked knowledge base to be a lexical resource, but research in this area is ongoing [27].

Similar problems have been explored in the enterprise domain through Expert Search [7, 24, 3], however, the presence of a strong authorship relation between entities and docu-

ments makes expert finding different. Moreover, we note that our validation of non-linking approaches motivates further study of expert search approaches to our task, something that prior work [25] dismissed out of hand. We note that our models are most similar to Balog et al's Model 2 [2], but applied with less linking certainty (emails are unambiguous) and on a much larger knowledge base.

While there is much modern work on entity linking, constructing an entity linker for novel types of entities or novel languages still requires non-trivial engineering and research effort [19, 15]. In particular, although recent TAC entity linking work has focused on multilingual efforts, the organizers of this challenge observe that a 90% linking accuracy requires about 20,000 query mentions labeled for training. Although there is work on new techniques that require less supervision and are more robust, these techniques still require substantial resources: name translation dictionaries, similarity scores, and surface forms [18]. It is for this reason that we are interested in approaches to this entity ranking problem that don't require entity linking. We wish to highlight that dictionary-based entity recognition is not novel [5], but we are the first to use it on a modern test collection for entity ranking.

7. CONCLUSION

In this work, we focus on the task of entity ranking in the context of exploring a target corpus. We substantially improve a recent dataset for this problem by adding annotator agreement and more labels. For the task itself, we propose an approach that does not rely upon entity linking for performance, freeing our approach from requiring linguistic resources typically annotated by experts. The only labels used by our system were acquired inexpensively at high quality through crowd sourcing, suggesting that our approach is likely to be a strong, simple and useful approach in new domains, such as private data and low resource languages.

Although our approach is straightforward, we find that it is competitive to the work which introduced this modern dataset. We hope that our expansion of the truth data and revisiting this task will encourage others to explore more interesting approaches to this task in the future.

8. ACKNOWLEDGMENTS

This work was supported in part by the Center for Intelligent Information Retrieval and in part by NSF grant #IIS-0910884. 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.

9. REFERENCES

[1] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. *Dbpedia: A nucleus for a web of open data.* Springer, 2007.

[2] K. Balog, L. Azzopardi, and M. De Rijke. Formal models for expert finding in enterprise corpora. In *SIGIR'06*, pages 43–50.

[3] K. Balog, Y. Fang, M. de Rijke, P. Serdyukov, and L. Si. Expertise retrieval. *Foundations and Trends in Information Retrieval*, 6(2–3):127–256, 2012.

[4] K. Balog, P. Serdyukov, and A. P. d. Vries. Overview of the trec 2010 entity track. Technical report, DTIC Document, 2010.

[5] A. Boldyrev, G. Weikum, and M. Theobald. *Dictionary-Based Named Entity Recognition.* PhD thesis, Universität des Saarlandes Saarbrücken, 2013.

[6] D. Carmel, M.-W. Chang, E. Gabrilovich, B.-J. P. Hsu, and K. Wang. Erd'14: entity recognition and disambiguation challenge. In *ACM SIGIR Forum, 2014*, volume 48, pages 63–77.

[7] N. Craswell, A. P. de Vries, and I. Soboroff. Overview of the trec 2005 enterprise track. In *TREC*, pages 199–205, 2005.

[8] W. B. Croft, D. Metzler, and T. Strohman. *Search engines: Information retrieval in practice.* Addison-Wesley Reading, 2010.

[9] J. Dalton, L. Dietz, and J. Allan. Entity query feature expansion using knowledge base links. In *SIGIR'14*, pages 365–374.

[10] V. Dang. Ranklib. `https://sourceforge.net/p/lemur/wiki/RankLib`, 2015.

[11] G. Demartini, T. Iofciu, and A. P. De Vries. Overview of the INEX 2009 entity ranking track. In *Focused Retrieval and Evaluation*, pages 254–264. Springer, 2010.

[12] J. Dunietz and D. Gillick. A new entity salience task with millions of training examples. *EACL'14*, page 205, 2014.

[13] P. Ferragina and U. Scaiella. Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). In *CIKM'10*, pages 1625–1628.

[14] E. Gabrilovich, M. Ringgaard, and A. Subramanya. FACC1: Freebase annotation of clueweb corpora, version 1. `http://lemurproject.org/clueweb09/FACC1/`, June 2013.

[15] F. Hasibi, K. Balog, and S. E. Bratsberg. On the reproducibility of the tagme entity linking system. In *Advances in Information Retrieval*, pages 436–449. Springer, 2016.

[16] J. Hoffart, D. Milchevski, and G. Weikum. Stics: searching with strings, things, and cats. In *SIGIR'14*, pages 1247–1248.

[17] J. Hoffart, F. M. Suchanek, K. Berberich, and G. Weikum. Yago2: A spatially and temporally enhanced knowledge base from wikipedia. *Artificial Intelligence*, 194:28–61, 2013.

[18] Y. Hong, D. Lu, D. Yu, X. Pan, X. Wang, Y. Chen, L. Huang, and H. Ji. Rpi blender tac-kbp2015 system description. In *Proc. Text Analysis Conference*, 2015.

[19] H. Ji, J. Nothman, B. Hachey, and R. Florian. Overview of TAC-KBP2015 Tri-lingual Entity Discovery and Linking. 2015.

[20] V. Lavrenko and W. B. Croft. Relevance based language models. In *SIGIR 2001*, pages 120–127, New York, NY, USA, 2001. ACM.

[21] X. Liu and H. Fang. Latent entity space: a novel retrieval approach for entity-bearing queries. *Information Retrieval Journal*, 18(6):473–503, 2015.

[22] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky. The Stanford CoreNLP natural language processing toolkit. In *ACL'14 Demo*, pages 55–60, 2014.

[23] P. McNamee and H. T. Dang. Overview of the TAC 2009 knowledge base population track. In *TAC'09*, volume 17, pages 111–113, 2009.

[24] D. Petkova and W. B. Croft. Proximity-based document representation for named entity retrieval. In *CIKM 2007*, pages 731–740, 2007.

[25] M. Schuhmacher, L. Dietz, and S. Ponzetto. Ranking entities for web queries through text and knowledge. In *CIKM'15*.

[26] C. Xiong and J. Callan. EsdRank: Connecting Query and Documents through External Semi-Structured Data. In *CIKM'15*.

[27] N. Zhiltsov, A. Kotov, and F. Nikolaev. Fielded sequential dependence model for ad-hoc entity retrieval in the web of data. In *SIGIR'15*, pages 253–262.

[28] G. Zuccon, B. Koopman, and P. Bruza. Exploiting inference from semantic annotations for information retrieval: Reflections from medical IR. In *ESAIR'14*, pages 43–45, 2014.