Semantic Driven Fielded Entity Retrieval

Shahrzad Naseri* Sheikh Muhammad Sarwar* James Allan

Center for Intelligent Information Retrieval
College of Information and Computer Sciences
University of Massachusetts Amherst
{shnaseri,smsarwar,allan}@cs.umass.edu

Abstract

A common approach for knowledge-base entity search is to consider an entity as a document with multiple fields. Models that focus on matching query terms in different fields are popular choices for searching such entity representations. An instance of such a model is FSDM (Fielded Sequential Dependence Model). We propose to integrate field-level semantic features into FSDM. We use FSDM to retrieve a pool of documents, and then to use semantic field-level features to re-rank those documents. We propose to represent queries as bags of terms as well as bags of entities, and eventually, use their dense vector representation to compute semantic features based on query document similarity. Our proposed re-ranking approach achieves significant improvement in entity retrieval on the DBpedia-Entity (v2) dataset over existing FSDM model. Specifically, for all queries we achieve 2.5% and 1.2% significant improvement in NDCG@10 and NDCG@100, respectively.

1 Introduction

In recent years, web search engines are moving toward answering users' query with a more focused response. Examples include entity cards as well as lists of named entities such as people, organizations, and locations as the answers or query suggestions. Studies over Bing [10] and Yahoo [23] web search queries has shown that over 70% and 50% of query logs are related to entities, respectively.

The core underlying most methods that provide such focused responses is collections of Knowledge Bases (KB). Knowledge bases provide a unified view of entities and the relationships between them. Knowledge bases such as DBpedia[1], YAGO[2], and Freebase[3] store entities information in a subject-predicate-object format, which is called Resource Description Framework (RDF) triple. Structured representation of entities available in KB made them attractive collections for entity search against natural language queries. In order to answer a users' query from the knowledge bases, the task of entity retrieval is defined as returning a ranked list of relevant entity articles to respond users' query.

Previous works represented a knowledge-base entity as a structured document by grouping RDFs into fields [2, 32] or tree structure [16]. For example, Zhiltsov et al. [32] define five fields such as names, attributes, categories, similar entity names, and related entities to represent an entity. They proposed Fielded Sequential Dependence Model (FSDM) and showed that term dependence is an important aspect for entity search. However, their work did not consider semantic matching of terms and documents which has become a popular choice for ad-hoc retrieval.

Capturing the semantic similarity between vocabulary terms and pieces of text is a long-standing problem in Information Retrieval (IR). Different methods have been proposed in this regard and one prevalent as well as recent choice among them is *word embedding.* Word embedding encodes the semantic information associated with a word by exploiting word co-occurrence information. Word2Vec [18] and Glove [21] are such methods which learns a low-dimensional vector using nueral networks and matrix factorization, respectively. We propose a method for entity retrieval that computes semantic match of query and each field of a document using embeddings for words and entities present in

*equal contribution

[1] `http://dbpedia.org`

[2] `http://www.mpi-inf.mpg.de/yago-naga/yago/`

[3] `http://freebase.org`

them.

We do our experiments on the DBpedia-Entity (v2) benchmark dataset [12] and use the train/test split provided by them to train a model that combines FSDM score and semantic features. DBpedia is often referred as the "database version of Wikipedia" and it is a community effort to extract structured information from Wikipedia. We demonstrate that significant gain can be achieved for similar entity search and natural language queries as well as all queries by incorporating semantic features. All resources, including a sample of the corpus we used to learn entity embeddings, source files for our model, runs and their evaluation results are made publicly available at https://tinyurl.com/sem-fielded-entity-retrieval.

The rest of this work is organized in the following manner: We provide some background on entity retrieval in section 2. In section 3 we discuss the formulation of our approach. Finally, we empirically validate our approach in section 5 and conclude in section 6.

2 Related Work

Guo et al. [10] and Pound et al. [23] show that over 70% and 50% of query logs of Bing and Yahoo, respectively, address entities. Motivated by that situation, an entity retrieval system returns ranked list of entities from a knowledge base to answer a user query. Various benchmarking campaigns focused on this task including INEX Entity Ranking [8], INEX Linked Data Track [28], the TREC Entity track [3, 1, 27], the Semantic Search Challenge [4, 11], and the Question Answering over Linked Data (QALD) challenge series [15]. DBpedia-Entity (v2), the dataset that is used by this shared task, gathers the queries from all of these previous challenges.

Existing methods take advantage of the fact that entities have rich fielded information and propose a variety of fielded retrieval methods such as BM25F [22, 13, 26] and FSDM [32]. In FSDM, different fields of an entity are categorized into five final fields: names, attributes, categories, related entity names, and similar entity names. FSDM incorporates term dependency based on ordered and unordered n-grams. Chen et al. [6] investigate learning to rank model on entity search which incorporates different features such as the FSDM score, BM25 score, etc.

There is substantial work in ad-hoc document retrieval that tries to take advantage of embeddings to improve retrieval effectiveness. Recently, Xiong et al. [29] described a method which presents documents and queries in both text and entity space, thus leveraging entity embeddings. However, such deep models need significant amounts of data to be effective. For this task, since the provided dataset is small, our model is more readily applicable.

Entity embeddings are also used in other tasks such as question answering [5], academic search [30], entity disambiguation [33], and for knowledge graph completion [31, 14]. The TREC-CAR (Complex Answer retrieval) task provides a large dataset on a large collection of knowledge articles from Wikipedia which present an opportunity for incorporating deep models in the task of entity retrieval. TREC-CAR shows that the RDF2Vec [25] is not as effective as the BM25 model in the paragraph ranking task [19].

3 Retrieval and re-ranking Approach

Our retrieval approach consists of two stages: we first create a pool of n documents using FSDM [32], and then we re-rank them using term and entity semantic features. Zhiltsov et al. proposed considering entities as documents with five different fields and used FSDM to retrieve entities [32]. In addition to the original five fields, another field *text* containing natural language description of an entity is incorporated in our setting.

Apart from using the top-n documents retrieved using FSDM, we use their scores in linear combination with our semantic similarity scores. We normalize the FSDM score using min-max normalization and use the result as a single feature or score in our approach. We compute two different types of similarity scores or semantic features based on two different query representations. This gives us two groups of semantic features that we linearly combine with the normalized FSDM score. We refer to the first group as "term semantics" and the second group as "entity semantics". For computing the entity semantics similarity score, we learned our own entity embedding vectors as described in Section 3.1, however, we used the pre-trained Glove word embeddings for term semantics.

Term Semantics

We compute the query embedding $\vec{q_t}$ using the average of the embedding of the query terms. For each field f_i of a document we also use the average of the word embedding of its terms to compute the representation $\vec{d_{f_i}}$ for that specific field. Then the score of that field f of a document is computed using $\cos \vec{q_t}, \vec{d_{f_i}}$. Finally, all the field scores are aggregated using a linear combination of the scores from each field using the following equation:

$$Score_t = \sum_{i}^{k} \lambda_i^t \times \cos \vec{q_t}, \vec{d_{f_i}}$$

where $Score_t$ represents term semantics.

Entity Semantics

In this approach, we represent the query as bag-of-entities. The semantic representation $\vec{q_e}$ of a query is computed as the average of the embedding of the entities present in the query. We compute the document representation in the same way as mentioned in the previous paragraph but using entities rather than terms. The query and document representations are used to compute entity semantics using the following equation:

$$Score_e = \sum_{i}^{k} \lambda_i^e \times \cos \vec{q_e}, \vec{d_{f_i}}$$

where $Score_e$ represents entity semantics.

Document Scoring

The score of a document is computed using Equation 1. It combines the term semantics, entity semantics, and normalized FSDM score. As we have six fields, we need to learn six parameters for term semantics, six for entity semantics, and one for FSDM. We learn these parameters using the Coordinate Ascent method for combining linear features proposed by Metzler et al. [17].

$$Score_D = Score_t + Score_e + \lambda \times Score_{FSDM} \quad (1)$$

3.1 Learning Entity Embeddings

Following the approach of Ni et al. [20], we learned embedding vectors for entities based on the Skip-gram [18] model. To this end, we replace the hyperlinks in the Wikipedia pages (that are links to other Wikipedia pages, i.e., entities) by a placeholder representing the entity. In this case, the hyperlink mentions (i.e. phrases) will be presented as a single "term" and the embedding of the entity (term) can be learned using Skip-gram model.

The following is an excerpt from Wikipedia in which entities are marked as italics:

> Albert Einstein was a German-born theoretical physicist who developed the *theory of relativity*, one of the two pillars of modern physics (alongside *quantum mechanics*). He is best known to the general public for his *mass–energy equivalence* formula $E = mc^2$ which has been dubbed "the world's most famous equation".

The excerpt will be changed to the following text in which hyperlinks (entities) are replaced by underscored of the title of the linked pages.

> Albert Einstein was a German-born theoretical physicist who developed the `Theory_of_relativity`, one of the two pillars of modern physics (alongside `Quantum_mechanics`). He is best known to the general public for his `Mass{energy_equivalence` formula $E = mc^2$ which has been dubbed "the world's most famous equation".

4 Experimental Setup

In this section, we introduce datasets (beyond DBpedia-Entity (v2)) that we used in our model. We also present our data processing approaches and hyperparameter settings.

4.1 Data Set

Our experiments are done using the dataset provided by the task, DBpedia-Entity (v2) [12]. We have used the same train/test split provided by them. We used the first 10 queries from training data to form our validation set. For embedding terms in queries and documents we used GloVe [21] pre-trained word embeddings. The word embeddings were originally learned from a 6 billion token collection (the Wikipedia dump 2014 plus the Gigawords 5). The entity embeddings are learned from the DBpedia 2016-10 full article Wikipedia pages dump.

4.2 Data Processing

We used the FSDM run in the DBpedia-Entity (v2) collection as the baseline method. We also consider the documents retrieved in that run as our initial document pool and re-ranked them using semantic features and FSDM score. For annotating entities in the query, we used the TagMe [9] mention detection tool. To learn the entity embeddings, we used the Word2Vec implementation in gensim [24]. Using the approach illustrated in Section 3.1, we learned embeddings of 3.0M entities out of 4.8M entities available in Wikipedia.

4.3 Hyperparameter settings

For learning the entity embedding vectors with 200 dimensions using Skip-gram model, we used the following hyperparameters: window-size=10, sub-sampling=$1\epsilon - 3$, cutoff min-count=0.

To learn the weights in our model, we used the coordinate ascent (CA) algorithm [17] to directly optimize NDCG@10. We start with random weights for all the features and use maximum 25 iterations with 2 restarts. We used the implementation of CA available at [7].

5 Experimental Results Discussion

Table 2 shows the result of incorporating semantic information with scores of our baseline FSDM model.

Term semantics refers to the similarity scores obtained for different fields of a document by considering the query as a term vector, while entity semantics consider the query as a bag of entities. We report the results based on different query groups in DBpedia-Entity v2 dataset. For the convenience of discussing our results, we provide one example for each type of query in Table 1.

Our results show that we achieve improvement by incorporating semantics (term semantics, entity semantics, or a combination of both) over all the query types. Incorporating entity semantic achieves the highest improvement in all query types except ListSearch queries. Note that for including entity information we only consider the query as a bag of entities. As a result of that choice, the converted list query "Professional sports teams in Philadelphia" would have two entities: *Professional sports teams* and *Philadelphia*. However, a *ListSearch* query is comprises three components: the target entity which is the entity to be retrieved, the source entity (*Philadelphia*), and the terms (*sports*, *teams*, *Professional*) that specify the relation between the target entity and the source entity. Our bag-of-entities query merges the terms that specify the relations between entities and is thus not helpful for that class of queries. As a consequence, we can see that incorporation of term semantics results in better performance compared to entity semantics in list search. Query term merging in this case might have been helpful if we have considered category or type embedding. Our work is more focused towards entity embedding, and we leave incorporating type embedding as future work.

Our approach yields the maximum (and significant) improvement for QALD query type. Including entity semantics resulted in 6.7% and 3.7% improvement over the FSDM baseline in term of NDCG@10 and NDCG@100, respectively. All these indicate that both term and entity semantics gives valuable gains in reranking.

Finally, we see significant improvement when we consider all the queries together. In this case, including both term and entity semantics resulted in the best NDCG@10. This is a statistically significant improvement over the baseline FSDM model. We also achieve significant improvement in NDCG@100 by incorporating entity semantics. However, in some cases incorporation of both term and entity semantics do not result in better performance compared to individually including them because of the increase in the number of features and lack of training data.

Table 1: Query types in DBpedia-Entity (v2) and their examples [12]

Query Type	Example
INEX-LD	*Electronic music geners*
ListSearch	*Professional sports teams in Philadelphia*
QALD-2	*Who is the mayor of Berlin?*
SemSearchES	*Brooklyn Bridge*

6 Conclusion and Future Works

In this study, we improve the accuracy of entity ranking by incorporating the similarity gained by comparing the query with each field of an entity document both in term and entity space. We demonstrate the efficiency of this model on a comprehensive benchmark dataset in comparison with the original FSDM model. In our experiments, we achieve statistically significant improvements over all of the queries. In order to increase the capacity of our model, we intend to learn separate vector embeddings for each field based on the content. For example, type embedding for the category field . Furthermore, we plan to adopt pairwise Learning to Rank (LTR) to determine feature weights. Moreover, by getting inspiration from the original FSDM paper which incorporates term dependencies, we hope to explore deep neural models such as RNN and LSTM in order to capture term sequence.

Acknowledgement

This work was supported in part by the Center for Intelligent Information Retrieval and in part by NSF grant #IIS-1617408. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsors.

References

[1] K. Balog, D. Carmel, and P. Arjen. de vries, daniel m. herzig, peter mika, haggai roitman, ralf schenkel, pavel serdyukov, thanh tran duc. In *The first joint international workshop on entity-oriented and semantic search (JIWES), ACM SIGIR Forum*, 2012.

[2] K. Balog and R. Neumayer. A test collection for entity search in dbpedia. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 737–740. ACM, 2013.

[3] K. Balog, P. Serdyukov, and A. P. d. Vries. Overview of the trec 2010 entity track. Technical report, NORWEGIAN UNIV OF SCIENCE AND TECHNOLOGY TRONDHEIM, 2010.

Table 2: Overall accuracy on each query group as well as all queries. The semantic and FSDM score are linearly combined using Coordinate Ascent algorithm. † indicates significant (p ¡ 0.05) improvement over the FSDM baseline measured by the Student's paired t-test.

	INEX_LD	
Methods	NDCG@10	NDCG@100
FSDM	0.4214	0.5043
FSDM + Entity Semantics	**0.4335**	**0.5119**
FSDM + Term Semantics	0.4224	0.5015
FSDM + Entity & Term Semantics	0.4291	0.5047
	ListSearch	
Methods	NDCG@10	NDCG@100
FSDM	0.4196	0.4952
FSDM + Entity Semantics	0.4247	0.4899
FSDM + Term Semantics	**0.4272**	**0.5004**
FSDM + Entity & Term Semantics	0.4242	0.4841
	QALD2	
Methods	NDCG@10	NDCG@100
FSDM	0.3401	0.4358
FSDM + Entity Semantics	**0.3628†**	**0.4521†**
FSDM + Term Semantics	0.3390	0.4291
FSDM + Entity & Term Semantics	0.3448	0.4330
	SemSearchES	
Methods	NDCG@10	NDCG@100
FSDM	0.6521	0.7220
FSDM + Entity Semantics	**0.6586**	**0.7281**
FSDM + Term Semantics	0.6500	0.7173
FSDM + Entity & Term Semantics	0.6583	0.7273
	all_queries	
Methods	NDCG@10	NDCG@100
FSDM	0.4524	0.5342
FSDM + Entity Semantics	0.4619†	**0.5408†**
FSDM + Term Semantics	0.4617†	0.5387†
FSDM + Entity & Term Semantics	**0.4639†**	0.5369

[4] R. Blanco, H. Halpin, D. M. Herzig, P. Mika, J. Pound, H. S. Thompson, and T. T. Duc. Entity search evaluation over structured web data. In *Proceedings of the 1st international workshop on entity-oriented search workshop (SIGIR 2011), ACM, New York*, 2011.

[5] A. Bordes, J. Weston, and N. Usunier. Open question answering with weakly supervised embedding models. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 165–180. Springer, 2014.

[6] J. Chen, C. Xiong, and J. Callan. An empirical study of learning to rank for entity search. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 737–740. ACM, 2016.

[7] R.-C. Chen. Coordinateascent. `https://github.com/rueycheng/CoordinateAscent`, 2018.

[8] G. Demartini, T. Iofciu, and A. P. De Vries. Overview of the inex 2009 entity ranking track. In *International Workshop of the Initiative for the Evaluation of XML Retrieval*, pages 254–264. Springer, 2009.

[9] P. Ferragina and U. Scaiella. Fast and accurate annotation of short texts with wikipedia pages. *IEEE software*, 29(1):70–75, 2012.

[10] J. Guo, G. Xu, X. Cheng, and H. Li. Named entity recognition in query. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 267–274. ACM, 2009.

[11] H. Halpin, D. M. Herzig, P. Mika, R. Blanco, J. Pound, H. Thompon, and D. T. Tran. Evaluating ad-hoc object retrieval. In *IWEST@ ISWC*, 2010.

[12] F. Hasibi, F. Nikolaev, C. Xiong, K. Balog, S. E. Bratsberg, A. Kotov, and J. Callan. Dbpedia-entity v2: A test collection for entity search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1265–1268. ACM, 2017.

[13] K. Y. Itakura and C. L. Clarke. A framework for bm25f-based xml retrieval. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 843–844. ACM, 2010.

[14] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu. Learning entity and relation embeddings for knowledge

graph completion. In *AAAI*, volume 15, pages 2181–2187, 2015.

[15] V. Lopez, C. Unger, P. Cimiano, and E. Motta. Evaluating question answering over linked data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 21:3–13, 2013.

[16] C. Lu, W. Lam, and Y. Liao. Entity retrieval via entity factoid hierarchy. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 514–523, 2015.

[17] D. Metzler and W. B. Croft. Linear feature-based models for information retrieval. *Information Retrieval*, 10(3):257–274, 2007.

[18] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[19] F. Nanni, B. Mitra, M. Magnusson, and L. Dietz. Benchmark for complex answer retrieval. In *Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval*, pages 293–296. ACM, 2017.

[20] Y. Ni, Q. K. Xu, F. Cao, Y. Mass, D. Sheinwald, H. J. Zhu, and S. S. Cao. Semantic documents relatedness using concept graph representation. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pages 635–644. ACM, 2016.

[21] J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[22] J. R. Pérez-Agüera, J. Arroyo, J. Greenberg, J. P. Iglesias, and V. Fresno. Using bm25f for semantic search. In *Proceedings of the 3rd international semantic search workshop*, page 2. ACM, 2010.

[23] J. Pound, P. Mika, and H. Zaragoza. Ad-hoc object retrieval in the web of data. In *Proceedings of the 19th international conference on World wide web*, pages 771–780. ACM, 2010.

[24] R. Řehůřek and P. Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. `http://is.muni.cz/publication/884893/en`.

[25] P. Ristoski and H. Paulheim. Rdf2vec: Rdf graph embeddings for data mining. In *International Semantic Web Conference*, pages 498–514. Springer, 2016.

[26] S. Robertson, H. Zaragoza, and M. Taylor. Simple bm25 extension to multiple weighted fields. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 42–49. ACM, 2004.

[27] P. Serdyukov and A. De Vries. Delft university at the trec 2009 entity track: Ranking wikipedia entities. Technical report, DELFT UNIV OF TECHNOLOGY (NETHERLANDS), 2009.

[28] Q. Wang, J. Kamps, G. R. Camps, M. Marx, A. Schuth, M. Theobald, S. Gurajada, and A. Mishra. Overview of the inex 2012 linked data track. In *CLEF (Online Working Notes/Labs/Workshop)*, 2012.

[29] C. Xiong, J. Callan, and T.-Y. Liu. Word-entity duet representations for document ranking. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 763–772. ACM, 2017.

[30] C. Xiong, R. Power, and J. Callan. Explicit semantic ranking for academic search via knowledge graph embedding. In *Proceedings of the 26th international conference on world wide web*, pages 1271–1279. International World Wide Web Conferences Steering Committee, 2017.

[31] B. Yang, W.-t. Yih, X. He, J. Gao, and L. Deng. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*, 2014.

[32] N. Zhiltsov, A. Kotov, and F. Nikolaev. Fielded sequential dependence model for ad-hoc entity retrieval in the web of data. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 253–262. ACM, 2015.

[33] S. Zwicklbauer, C. Seifert, and M. Granitzer. Robust and collective entity disambiguation through semantic embeddings. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 425–434. ACM, 2016.