

# DBpedia-Entity v2: A Test Collection for Entity Search

Faegheh Hasibi  
Norwegian University of Science and  
Technology  
faegheh.hasibi@ntnu.no

Fedor Nikolaev  
Wayne State University  
fedor@wayne.edu

Chenyan Xiong  
Carnegie Mellon University  
cx@cs.cmu.edu

Krisztian Balog  
University of Stavanger  
krisztian.balog@uis.no

Svein Erik Bratsberg  
Norwegian University of Science and  
Technology  
sveinbra@ntnu.no

Alexander Kotov  
Wayne State University  
kotov@wayne.edu

Jamie Callan  
Carnegie Mellon University  
callan@cs.cmu.edu

## ABSTRACT

The DBpedia-entity collection [2] has been used as a standard test collection for entity search in recent years. We develop and release a new version of this test collection, *DBpedia-Entity v2*, which uses a more recent DBpedia dump and a unified candidate result pool from the same set of retrieval models. Relevance judgments are also collected in a uniform way, using the same group of crowdsourcing workers, following the same assessment guidelines. The result is an up-to-date and consistent test collection. To facilitate further research, we also provide details about the pre-processing and indexing steps, and include baseline results from both classical and recently developed entity search methods.

## KEYWORDS

Entity retrieval; Test collection; Semantic search; DBpedia

### ACM Reference format:

Faegheh Hasibi, Fedor Nikolaev, Chenyan Xiong, Krisztian Balog, Svein Erik Bratsberg, Alexander Kotov, and Jamie Callan. 2017. DBpedia-Entity v2: A Test Collection for Entity Search. In *Proceedings of SIGIR '17, August 07-11, 2017, Shinjuku, Tokyo, Japan*, , 4 pages.  
DOI: 10.1145/3077136.3080751

## 1 INTRODUCTION

Entities are meaningful units of retrieval, as many information needs are centered around them [17]. For example, it has been found that more than 70% of Bing’s query volume is related to entities [9]; in the Allen Institute’s scholar search engine, more than half of the traffic is about research concepts (i.e., abstract entities) and another one third is about person names [26]. Over the course of the past decade, entity search has drawn a lot of attention from both academia and industry. Entities have also grown to be

first-class citizens in web search, often featured as entity cards. Much of this success can be attributed to the availability of large-scale knowledge repositories, which can provide rich semantic information organized around entities. Recently, there has been a shift of focus from semi-structured data sources (in particular, Wikipedia) to structured ones (DBpedia, YAGO, Freebase, etc.). To further research and development in this area, there is a need for a standard test collection—this is exactly what the resource we introduce in this paper, the *DBpedia-Entity v2 collection*, aims to be.

Balog and Neumayer [2] introduced the DBpedia-Entity test collection, by assembling search queries from a number of entity-oriented benchmarking campaigns and mapping relevant results to DBpedia. Over the past years, this has become a standard test collection for evaluating entity search research, see [5, 11, 16, 20, 27]. The main objective of this work is to create a new, updated version of this test collection. We shall refer to the original collection in [2] as *DBpedia-Entity v1* and to our updated version as *DBpedia-Entity v2*. The new version’s improvements are manifold. (1) The original collection contains only relevant results and relevance is binary for most of the queries; we use graded relevance judgments for all queries and also include all judged items, relevant or not. (2) The DBpedia knowledge base has grown significantly over the past years. Many new relevant entities were not judged in the old version; we use a recent DBpedia version and judge the relevance of new entities. (3) Judgments in the original collection have been assembled from multiple campaigns, each with its own setup; we obtain relevance labels under the same conditions for all queries in the collection.

We also present details about how the DBpedia dump is processed and indexed, reducing the inconsistency in preprocessing. We provide rankings using both traditional and recently-developed entity search methods, making future comparison with prior work much easier. All resources, including queries, relevance assessments (qrels), base runs, their evaluation results, and further details on indexing and preprocessing are made publicly available at <http://tiny.cc/dbpedia-entity>. We note that annotations for the *target type identification* [8] and *entity summarization* [12] tasks, using the same queries, are also available in this repository.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).  
SIGIR '17, August 07-11, 2017, Shinjuku, Tokyo, Japan  
© 2017 ACM. 978-1-4503-5022-8/17/08...\$15.00  
DOI: 10.1145/3077136.3080751

## 2 THE TEST COLLECTION

This section describes the test collection, including the knowledge base, queries, and process of collecting of relevance assessments.

### 2.1 Knowledge Base

We use DBpedia as our knowledge base; it is often referred to as “the database version of Wikipedia.” DBpedia is a community effort, where a set of rules (“mappings”) are collaboratively created to extract structured information from Wikipedia. Since its inception in 2007, there have been regular data releases; it also has a live extraction component that processes Wikipedia updates real-time. DBpedia is a central hub in the Linking Open Data cloud, and has been widely used in various semantic search tasks [1, 12, 13, 18].<sup>1</sup>

We use the English part of the 2015-10 version of DBpedia. It contains 6.2 million entities, 1.1 billion facts, and an ontology of 739 types. In comparison, version 3.7, that has been used in *DBpedia-Entity v1*, contains 3.64 million entities, over 400 million facts, and an ontology of 358 types. DBpedia 2015-10 is also believed to be much cleaner due to better extraction techniques developed by the DBpedia community.

*Preprocessing.* We require entities to have both a title and abstract (i.e., `rdfs:label` and `rdfs:comment` predicates)—this effectively filters out category, redirect, and disambiguation pages. Note that list pages, on the other hand, are retained. In the end, we are left with a total of 4.6 million entities. Each entity is uniquely identified by its URI.

### 2.2 Test Queries

The queries in *DBpedia-Entity v2* are the same as in *v1*. We distinguish between four categories of queries:

- **SemSearch ES** queries are from the ad-hoc entity search task of the Semantic Search Challenge series [4, 10]. These are short and ambiguous queries, searching for one particular entity, like “brooklyn bridge” or “08 toyota tundra.”
- **INEX-LD** queries are from the ad-hoc search task at the INEX 2012 Linked Data track [25]. They are IR-style keyword queries, e.g., “electronic music genres.”
- **List Search** comprises queries from the list search task of the 2011 Semantic Search Challenge (SemSearch LS) [4], from the INEX 2009 Entity Ranking track (INEX-XER) [6], and from the Related Entity Finding task at the TREC 2009 Entity track [3]. These queries seek a particular list of entities, e.g., “Professional sports teams in Philadelphia.”
- **QALD-2** queries are from the Question Answering over Linked Data challenge [15]. These are natural language questions that can be answered by DBpedia entities, for example, “Who is the mayor of Berlin?”

Originally, the SemSearch queries were evaluated using crowdsourcing on a 3-point relevance scale. All other benchmarks employed expert evaluators (trained assessors or benchmark organizers/participants) and have binary judgments.

<sup>1</sup>DBpedia is not the only general-purpose knowledge base available, but arguably the most suitable one. Alternatives include YAGO [24] (not updated regularly), Freebase (discontinued), and WikiData (still in its infancy).

**Query:**  
Who founded Intel?

**Entity:**  
Name: Gordon Moore  
Wikipedia article: [https://en.wikipedia.org/wiki/Gordon\\_Moore](https://en.wikipedia.org/wiki/Gordon_Moore)

Description: Gordon Earle Moore (born January 3, 1929) is an American businessman, co-founder and Chairman Emeritus of Intel Corporation, and the author of Moore's law. As of January 2015, his net worth is \$6.7 billion.

How relevant is this entity to the intent of the query?

Highly relevant  
 Relevant  
 Irrelevant

Figure 1: Crowdsourcing task design.

### 2.3 Relevance Assessments

In *DBpedia-Entity v1*, the relevance judgments (“qrels”) are assembled from several different benchmarks. These assessments were created using different annotation guidelines, judges (trained assessors vs. crowdsourcing), pooling methods, and even different corpora (various versions of DBpedia or Wikipedia). For *DBpedia-Entity v2*, we generate new relevance judgments for all queries using the same setup. We pool candidate results from the same set of systems, and use the same annotation procedure and guidelines.

*2.3.1 Pooling.* Following standard practice of IR test collection building, we employ a *pooling* approach, and combine retrieval results from four main sources:

- **Original qrels.** All relevant entities from *DBpedia-Entity v1* are included, to ensure that results that have previously been identified as relevant get re-assessed.
- **Previous runs.** We consider 37 different retrieval methods (“runs”) that have been evaluated on *DBpedia-Entity v1* in prior work [11, 20, 26, 27]. All entity URIs returned by these runs are mapped to DBpedia version 2015-10; entities not present in DBpedia 2015-10 are discarded. The pool depth is 20, i.e., we take the top 20 ranked entities from each run.
- **New runs.** We obtained retrieval results for DBpedia 2015-10 from 13 different systems, by three independent research groups; see Sect. 3 for the description of these methods. Results are pooled from these runs up to depth 20.
- **SPARQL results.** For QALD-2 queries, the ground truth is obtained by executing a SPARQL query (manually constructed by the campaign organizers [15]) over the knowledge base. We re-ran these queries against the DBpedia API endpoint to obtain up-to-date results, as the answers to some questions might have changed since (e.g., “Who is the mayor of Berlin?”).

The final assessment pool contains 50,516 query-entity pairs (104 entities per query on average).

*2.3.2 Collecting Relevance Judgments.* We collected the relevance judgments using the CrowdFlower crowdsourcing platform. For each record (i.e., query-entity pair) in our pool, we provided the workers with the query, the name and short description (DBpedia abstract) of the entity, as well as the link to the entity’s Wikipedia page; see Figure 1. Since narratives are only available for a small number of queries in our query set (those from TREC and INEX), we decided to keep the setup uniform across all queries, and present assessors only with the query text. To avoid positional bias, records were presented in a random order. Workers were then asked to

**Table 1: Query categories in DBpedia-Entity v2.**  $R_1$  and  $R_2$  refer to the average number of relevant and highly relevant entities per query, respectively.

Category	#queries	Type	$R_1$	$R_2$
SemSearch ES	113	named entities	12.5	3.0
INEX-LD	99	keyword queries	23.5	9.2
ListSearch	115	list of entities	18.1	12.7
QALD-2	140	NL questions	28.4	29.8
<b>Total</b>	467		21.0	14.7

judge relevance on a 3-point Likert scale: highly relevant, relevant, or irrelevant. We educated workers about the notion of entities and provided them with the following working definitions for each scale (each further illustrated with examples):

- **Highly relevant (2):** The entity is a direct answer to the query (i.e., the entity should be among the top answers).
- **Relevant (1):** The entity helps one to find the answer to the query (i.e., the entity can be shown as an answer to the query, but not among the top results).
- **Irrelevant (0):** The entity has no relation to the intent of the query (i.e., the entity should not be shown as an answer).

We have taken quality control very seriously, which was a non-trivial task for a pool size of over 50K. During the course of the assessment, the accuracy of workers was regularly examined with hidden test questions. 400 query-entity pairs were randomly selected as test cases and judged by three authors of the paper; 373 of these were then used as test questions (where at least two of the experts agreed on the relevance label). Only workers with qualification level 2 (medium) or 3 (high) on CrowdFlower were allowed to participate. They were then required to maintain at least 70% accuracy throughout the job; those falling below this threshold were not allowed to continue the job and their previous assessments were excluded. We collected 5 judgments for each record and paid workers a reasonable price of €1 per judgment. The final cost was over 3,500 USD, which makes this a very valuable test collection, also in the literal sense of the word. The Fleiss’ Kappa inter-annotator agreement was 0.32, which is considered fair agreement. To determine the relevance level for a query-entity pair, we took the majority vote among the assessors. In case of a tie, the rounded average of relevance scores is taken as the final judgment.

Further inspection of the obtained results revealed that crowd workers are less likely to find answers to complex information needs. They are less patient and make judgments primarily based on the provided snippets and Wikipedia pages. When it would be required to read the Wikipedia article more carefully, or to consult additional sources, users are less likely to label them as attentively as expert annotators would. To further the quality of the test collections, we collected expert annotations for cases with “extreme disagreements,” i.e., cases without majority vote, or cases that are found irrelevant by crowd workers, but are highly relevant according to the original qrels.<sup>2</sup> This resulted in the annotation of 8K query-entity pairs, each by two experts, with a Fleiss’ Kappa agreement of 0.48, which is considered moderate. The final label for

<sup>2</sup>This includes SPARQL query results for QALD queries, highly relevant judgments for SemSearch queries, and all TREC and INEX judgments.

**Table 2: Comparison of methods for DBpedia-entity v1 vs. v2 qrels.** The supervised methods (bottom block) are trained on v1; for methods trained on v2, we refer the reader to the online repository.

Method	Index	v1		v2	
		MAP	P@10	MAP	nDCG@10
BM25	A	0.0884	0.0971	0.1893	0.2558
PRMS	B	0.1571	0.1682	0.2895	0.3905
MLM-all	B	0.1618	0.1705	0.3031	0.4021
LM	B	0.1709	0.1837	0.3144	0.4182
SDM	A	<b>0.1860</b>	<b>0.1880</b>	<b>0.3259</b>	<b>0.4185</b>
LTR	A	0.1723	0.1831	0.2446	0.3464
LM+ELR	B+	0.1772	0.1895	0.3103	0.4123
SDM+ELR	A+	0.1901	0.1986	0.3284	0.4200
MLM-CA	A	0.1905	0.2008	0.3061	0.4117
BM25-CA	A	0.2067	0.2056	0.3265	0.4231
FSDM	A	0.2069	0.2039	0.3279	0.4267
BM25F-CA	A	0.2088	<b>0.2126</b>	<b>0.3361</b>	<b>0.4378</b>
FSDM+ELR	A+	<b>0.2210</b>	0.2089	0.3295	0.4335

the extreme disagreement cases was taken to be the expert-agreed label. If such a label did not exist, we took the rounded average between the two expert labels and the crowdsourcing decision (as a third label). Finally, queries that no longer have relevant results were removed (18 in total). Table 1 shows the statistics for the final v2 collection.

### 3 RETRIEVAL METHODS

We employ a range of entity retrieval approaches to obtain results for pooling. Below, we provide a brief overview of these methods, along with the details of indexing and pre-processing techniques.

#### 3.1 Indexing and Query Processing

Retrieval results were obtained using two indices (Index A and Index B), built from the DBpedia 2015-10 dump, following the general approach outlined in [27]. In particular, we used the same entity representation scheme with five fields (*names*, *categories*, *similar entity names*, *attributes*, and *related entity names*) as in [27]. Index A was constructed using Galago, while Index B was created using Elasticsearch. They use slightly different methods for converting entity URIs to texts. Index B also contains an extra *catchall* field, concatenating the contents of all other fields. An extra URI-only index was built according to [11], which is used for the ELR-based methods; we write ‘+’ to denote when this index is used. All the runs were generated using preprocessed queries; i.e., removing the stop patterns provided in [11] and punctuation marks. Further details are provided in the collection’s GitHub repository.

#### 3.2 Retrieval Methods

We consider various entity retrieval methods that have been published over the recent years [2, 5, 11, 20, 27]. Unless stated otherwise, the parameters of methods are trained for each of the four query subsets, using cross-validation (with the same folds across all methods). Table 2 shows the particular index version that was used for each method.

*Unstructured retrieval models.* This group of methods uses a flattened entity representation. Specifically, we report on **LM** (Language Modeling) [22], **SDM** (Sequential Dependence Model) [19], and **BM25** [23]. All LM-based methods use Dirichlet prior smoothing with  $\mu = 1500$  for index A, and  $\mu = 2000$  for index B. The BM25 parameters are  $k_1 = 1.2$  and  $b = 0.8$ . We also report on **BM25-CA** with parameters trained using Coordinate Ascent.

*Fielded retrieval models.* This category of methods employs a fielded entity representation (cf. Sect. 3.1). We report on **MLM-CA** (Mixture of Language Models) [21], **FSDM** (Fielded Sequential Dependence Model) [27], and **BM25F-CA** [23] (the -CA suffixes refer to training using Coordinate Ascent). We also report on **MLM-all**, with equal field weights, and on **PRMS** (Probabilistic Model for Semistructured Data) [14], which has no free parameters.

*Other models.* The **LTR** (Learning-to-Rank) approach [5] employs 25 features from various retrieval models and is trained using the RankSVM algorithm. The **ELR** methods [11] employ TAGME [7] for annotating queries with entities, and use the URI-only index (with a single catchall field) for computing the ELR component.

## 4 RESULTS AND ANALYSIS

In Table 2 we report on the performance of the different retrieval methods using both the original ( $v1$ ) and new ( $v2$ ) relevance judgments. (In case of the  $v1$  qrels, we removed entities that are not present in DBpedia 2015-10.) Methods in the top block of the table do not involve any training and use default parameter settings, while systems in the bottom block are trained for each query category using cross-validation. Training is done using the  $v1$  qrels. Since we have graded relevance judgments for  $v2$ , the “official” evaluation metric for the new collection is NDCG@10. However, to facilitate comparison with the  $v1$  results, we also report on MAP (at rank 100, accepting both levels 1 and 2 as relevant). At first glance, we observe that the absolute MAP values for  $v2$  are higher than for  $v1$ ; this is expected, as there are more relevant entities according to the new judgments. We also find that the relative ranking of methods in the top block remains the same when moving from  $v1$  to  $v2$ . On the other hand, methods that involve training (bottom block) show much smaller relative improvements over the models without training (top block) in  $v2$  as for  $v1$ . This is explained by the fact that training was done on  $v1$ . We note that we are not elaborating on the performance of individual methods as that is not the focus of this paper. One issue we wish to point out, nevertheless, is that default parameter settings may be unfitting for entity retrieval; in particular, observe the large difference between BM25 with default parameters vs. BM25-CA with trained parameters (which are  $b \approx 0.05$  and  $k_1$  in the range 2..6, depending on the query subtype). In the online repository, we further report on the supervised models trained on the new ( $v2$ ) qrels, and break down evaluation results into different query subsets.

## 5 CONCLUSION

This paper has introduced an updated version of a standard entity search test collection, by using a more recent DBpedia dump, a more consistent candidate document pool, and a unified relevance

assessment procedure. We have also provided details about processing and indexing, together with retrieval results for both traditional and more recent entity retrieval models. It is our hope that this new test collection will serve as the de facto testbed for entity search over structured data, and will foster future research.

*Acknowledgements.* This research was partially supported by the National Science Foundation (NSF) grant IIS-1422676. Any opinions, findings, and conclusions in this paper do not necessarily reflect those of the sponsors.

## REFERENCES

- [1] Krisztian Balog and Robert Neumayer. 2012. Hierarchical target type identification for entity-oriented queries. In *Proc. of CIKM '12*. 2391–2394.
- [2] Krisztian Balog and Robert Neumayer. 2013. A Test Collection for Entity Search in DBpedia. In *Proc. of SIGIR '13*. 737–740.
- [3] Krisztian Balog, Pavel Serdyukov, Arjen De Vries, Paul Thomas, and Thijs Westerveld. 2010. Overview of the TREC 2009 Entity Track. In *Proc. of TREC '09*.
- [4] Roi Blanco, Harry Halpin, Daniel M Herzig, Peter Mika, Jeffrey Pound, and Henry S Thompson. 2011. Entity Search Evaluation over Structured Web Data. In *Proc. of the 1st International Workshop on Entity-Oriented Search*. 65–71.
- [5] Jing Chen, Chenyan Xiong, and Jamie Callan. 2016. An Empirical Study of Learning to Rank for Entity Search. In *Proc. of SIGIR '16*. 737–740.
- [6] Gianluca Demartini, Tereza Iofciu, and Arjen P De Vries. 2009. Overview of the INEX 2009 Entity Ranking Track. In *INEX*. 254–264.
- [7] Paolo Ferragina and Ugo Scaiella. 2010. TAGME: On-the-fly Annotation of Short Text Fragments (by Wikipedia Entities). In *Proc. of CIKM '10*. 1625–1628.
- [8] Dario Garigliotti, Faegheh Hasibi, and Krisztian Balog. 2017. Target Type Identification for Entity-Bearing Queries. In *Proc. of SIGIR '17*.
- [9] Jiafeng Guo, Gu Xu, Xueqi Cheng, and Hang Li. 2009. Named Entity Recognition in Query. In *Proc. of SIGIR '09*. 267–274.
- [10] Harry Halpin, Daniel M Herzig, Peter Mika, Roi Blanco, Jeffrey Pound, Henry S Thompson, and Duc Thanh Tran. 2010. Evaluating Ad-hoc Object Retrieval. In *Proc. of the International Workshop on Evaluation of Semantic Technologies*.
- [11] Faegheh Hasibi, Krisztian Balog, and Svein Erik Bratsberg. 2016. Exploiting Entity Linking in Queries for Entity Retrieval. In *Proc. of ICTIR '16*. 171–180.
- [12] Faegheh Hasibi, Krisztian Balog, and Svein Erik Bratsberg. 2017. Dynamic Factual Summaries for Entity Cards. In *Proc. of SIGIR '17*.
- [13] Faegheh Hasibi, Krisztian Balog, and Svein Erik Bratsberg. 2017. Entity Linking in Queries: Efficiency vs. Effectiveness. In *Proc. of ECIIR '17*. 40–53.
- [14] Jinyoung Kim, Xiaobing Xue, and W Bruce Croft. 2009. A Probabilistic Retrieval Model for Semistructured Data. In *Proc. of ECIIR '09*. 228–239.
- [15] Vanessa Lopez, Christina Unger, Philipp Cimiano, and Enrico Motta. 2013. Evaluating Question Answering over Linked Data. *Web Semantics: Science, Services and Agents on the World Wide Web* 21, 0 (2013), 3–13.
- [16] Chunliang Lu, Wai Lam, and Yi Liao. 2015. Entity Retrieval via Entity Factoid Hierarchy. In *Proc. of ACL '15*. 514–523.
- [17] Edgar Meij, Krisztian Balog, and Daan Odijk. 2014. Entity Linking and Retrieval for Semantic Search. In *Proc. of WSDM '14*. 683–684.
- [18] Edgar Meij, Marc Bron, Laura Hollink, Bouke Huurnink, and Maarten de Rijke. 2011. Mapping Queries to the Linking Open Data Cloud: A Case Study Using DBpedia. *Web Semant.* 9, 4 (Dec. 2011), 418–433.
- [19] Donald Metzler and W Bruce Croft. 2005. A Markov Random Field Model for Term Dependencies. In *Proc. of SIGIR '05*. 472–479.
- [20] Fedor Nikolaev, Alexander Kotov, and Nikita Zhiltsov. 2016. Parameterized Fielded Term Dependence Models for Ad-hoc Entity Retrieval from Knowledge Graph. In *Proc. of SIGIR '16*. 435–444.
- [21] Paul Ogilvie and Jamie Callan. 2003. Combining Document Representations for Known-item Search. In *Proc. of SIGIR '03*. 143–150.
- [22] Jay M Ponte and W Bruce Croft. 1998. A Language Modeling Approach to Information Retrieval. In *Proc. of SIGIR '98*. 275–281.
- [23] Stephen E. Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Found. and Trends in IR* 3, 4 (2009), 333–389.
- [24] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: A Core of Semantic Knowledge. In *Proc. of WWW '07*. 697–706.
- [25] Qiuyue Wang, Jaap Kamps, Georgina Ramirez Camps, Maarten Marx, Anne Schuth, Martin Theobald, Sairam Gurajada, and Arunav Mishra. 2012. Overview of the INEX 2012 Linked Data Track. In *CLEF Online Working Notes*.
- [26] Chenyan Xiong, Russell Power, and Jamie Callan. 2017. Explicit Semantic Ranking for Academic Search via Knowledge Graph Embedding. In *Proc. of WWW '17*. 1271–1279.
- [27] Nikita Zhiltsov, Alexander Kotov, and Fedor Nikolaev. 2015. Fielded Sequential Dependence Model for Ad-Hoc Entity Retrieval in the Web of Data. In *Proc. of SIGIR '15*. 253–262.