The Sindice-2011 Dataset for Entity-Oriented Search in the Web of Data

Stéphane Campinas* Diego Ceccarelli‡ Thomas E. Perry*
Renaud Delbru* Krisztian Balog† Giovanni Tummarello*

*Digital Enterprise Research Institute
National University of Ireland, Galway
Galway, Ireland
firstname.lastname@deri.org

†Norwegian University of
Science and Technology
Trondheim, Norway
krisztian.balog@idi.ntnu.no

‡ ISTI-CNR
Dipartimento di Informatica
Università di Pisa
Pisa, Italy
diego.ceccarelli@isti.cnr.it

ABSTRACT

The task of entity retrieval becomes increasingly prevalent as more and more (semi-) structured information about objects is available on the Web in the form of documents embedding metadata (RDF, RDFa, Microformats, and others). However, research and development in that direction is dependent on (1) the availability of a representative corpus of entities that are found on the Web, and (2) the availability of an entity-oriented search infrastructure for experimenting with new retrieval models. In this paper, we introduce the Sindice-2011 data collection which is derived from data collected by the Sindice semantic search engine. The data collection (available at `http://data.sindice.com/trec2011/`) is especially designed for supporting research in the domain of web entity retrieval. We describe how the corpus is organised, discuss statistics of the data collection, and introduce a search infrastructure to foster research and development.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.1 Content Analysis and Indexing; H.3.3 Information Search and Retrieval; H.3.4 Systems and Software

General Terms

Measurement, Performance, Experimentation

Keywords

Entity search, Web of Data, Entity corpus

1. INTRODUCTION

Entity retrieval, that is, returning "objects" (such as people, organisations, locations, products, etc.) in response to users' information needs, has received considerable attention recently from various research communities, as well as from the commercial sector. According to a recent study, more than half of web queries target a particular entity or instances of a given entity type [9]. Supporting the search and discovery of entities, therefore, is essential for ensuring a satisfying user experience.

EOS, SIGIR 2011 workshop, July 28, Beijing, China

The field of Information Retrieval (IR) is characterized by rigorous attention to evaluation and measurement. International benchmarking campaigns, such as the Text REtrieval Conference (TREC) and the Initiative for the Evaluation of XML Retrieval (INEX) play a key role in fostering IR research by providing a common platform, evaluation methodology, and relevance judgements to assess the quality of information access systems. The introduction of the expert finding task at the TREC Enterprise track in 2005 was a significant milestone on the path to entity-oriented retrieval. The goal of the expert finding task is to create a ranking of people who are experts in a given topical area [4]. Later, in 2007, INEX launched an Entity Ranking track [10]; here, entities are represented by their Wikipedia page and two tasks are considered: (i) *entity ranking*, where a query and target categories are given, and (ii) *list completion*, where a textual query, example entities, and, optionally, target categories are provided as input. In 2009, the Entity track at TREC started with the goal to perform entity-oriented search tasks on the Web, and defined the *related entity finding* (REF) task [2]. REF requests a ranked list of entities (of a specified type) that engage in a given relationship with a given source entity. The collection used there is a general Web crawl and entities are identified by their homepages. The 2010 edition of the track introduced an *entity list completion* task [3], similar to that of INEX, but the collection is a Semantic Web crawl, specifically, the Billion Triple Challenge 2009 (BTC-2009) dataset[1]. Looking at these developments over time, a shift of emphasis can be observed from the document-oriented web to the data-oriented web, or "Web of Data."

From the Web of Documents to the Web of Data.

Compared to the Web of Documents, the Web of Data is much more structured. However, since each Web of Data source might have its own defined schema, ranging from loosely to strictly defined, the data structure does not follow strict rules as in a database. Even within a given a data source, the schema might not be fixed and may change as the information grows. The information structure evolves over time, and new records can require new attributes. We therefore consider the Web of Data as being *semi-structured* [1].

The most important property of the Web of Data is that it is naturally organised around entities[2], and that each of these entities is uniquely identified by a Uniform Resource Identifier (URI). The

[1] `http://vmlion25.deri.ie/`

[2] Note that under standard Semantic Web terminology, this concept is referred to as *resource description*. We use "entity" instead of "resource" in order to emphasize that a resource describes an entity.

Web of Data, therefore, provides a natural setting for entity retrieval, also recognized by the Semantic Web Workshop series that organised the Semantic Search Challenge in 2010[3] and in 2011[4]. Two tasks are addressed: (i) *entity search* (2010 and 2011), where queries refer to one particular entity [7], and (ii) *list search* (2011), where queries target a group of entities that match certain criteria (similar in spirit to the list completion tasks at INEX and TREC, but here only a keyword query is provided, without type information or example entities).

The data collection used in both editions of the Semantic Search Challenge and also at the 2010 TREC Entity track, as a representative of Web of Data, is the BTC-2009 dataset. This collection, however, is not representative anymore of what can be found on the Web. It is mainly composed of RDF documents crawled in early 2009, and do not contain (or only in a very small proportion) data from RDFa or Microformats; as we show in Section 4, RDFa and Microformats make up more than half of the data in our collection. There exists a newer version of the BTC collection, BTC-2010[5] that contains 3 billion statements crawled in the early 2010. However, the aforementioned problem still persists.

Contribution.

The first contribution of this work is a new data collection, referred to as the *Sindice-2011 Dataset*, that is an accurate reflection of the current Web of Data. This dataset is made available to be used by the broad research community. One particular type of usage is to evaluate semantic search systems using this dataset. For instance, the 2011 edition of the TREC Entity track uses this collection to evaluate entity-related search tasks. Other communities could use this dataset for evaluating entity coreference resolution techniques or for benchmarking RDF data management systems.

Specifically, the Sindice-2011 Dataset contains 11 billion statements from 231 million documents that correspond to 1.738 billion entities. The exact values are reported in the Table 2. The data has been collected by the Sindice semantic search engine [8] from 2009 to 13/05/2011. The Sindice-2011 Dataset has been designed with the central aim of supporting research in web entity search. To conform to this particular need, the original Sindice crawl, which is composed of and organised around semi-structured documents, is processed and transformed into a corpus of entities.

Our second contribution is a collection of tools to help researchers working with this data. Specifically, these tools, written in Java, provide methods to process, index, and search for entities in the Sindice-2011 Dataset.

The remainder of the paper is organised as follows. Section 2 presents the Web of Data model as well as our entity-centric model. In Section 3, we discuss the creation and the organisation of the data collection. Section 4 provides statistics about the dataset. Section 5 introduces the tools and infrastructure for facilitating IR research and development using this collection. Finally, we conclude in Section 6.

2. WEB OF DATA

We use the term *Web of Data* to refer to the collection of machine processable content, exposed on the Web through metadata standards, e.g., Microformats, RDF, or RDFa. Without entering into a discussion about terminology, we use Web of Data as a casual synonym for the Semantic Web (or Web 3.0), and view it as a superset of Linked (Open) Data. Next in this section, a brief explanation of the RDF data model is given before presenting the Web of Data model. Readers familiar with these concepts may wish to proceed to the next section.

[3] `http://km.aifb.kit.edu/ws/semsearch10/`
[4] `http://km.aifb.kit.edu/ws/semsearch11/`
[5] `http://km.aifb.kit.edu/projects/btc-2010/`

2.1 Resource Description Framework

The Resource Description Framework (RDF) is a generic data model that can be used for interoperable data exchange. In a nutshell, RDF facilitates the machine understanding of resource (entity) descriptions, hence allowing an automated processing of them. In RDF, a resource description is composed of statements about a given resource. A statement is a triple consisting of a subject, a predicate and an object, and asserts that a subject has a property with some value. A set of RDF statements forms a directed labelled graph. In an RDF graph, a node can be of the three types: URI, literal and blank node. An URI serves as a globally-unique identifier for a resource. A literal is a character string with an optional associated language and datatype. A blank node represents a resource for which a URI is not given. A blank node is considered as an identifier which is always scoped to the containing graph, e.g., a web document.

2.2 Web of Data Model

We define the Web of Data as part of the Hypertext Transfer Protocol (HTTP) accessible Web that returns semi-structured information using standard interchange formats and practices. The standard data interchange formats include HTML pages which embed RDFa or Microformats as well as RDF models using different syntaxes such as RDF/XML. It is possible to abstract the following core concepts in semi-structured data web publishing:

A Dataset is a collection of entity descriptions. One dataset is usually the content of some database which powers a web application exposing metadata, be this a dynamic web site with just partial semantic markups or a Semantic Web database which exposes its content, such as a Linked Open Data dataset. Datasets, however, can also come in the form of a single RDF document, e.g., an individual FOAF[6] file posted on a person's homepage.

An Entity description is a set of assertions about an entity and belongs to a dataset. Typical examples of entities include documents, people, events, products, etc. The assertions provide information regarding the entity such as attributes, for instance, the firstname and surname for a person, or relationships with other entities, for instance, the family members for a person. Each entity description can be uniquely identified, either using a URI or a blank node jointly with the graph identifier it originates from, i.e., the document's URL.

A View represents a single accessible piece of information, i.e., a web document, that provides a full or partial view over the content of the dataset. In the case of Linked Open Data, a typical view is the RDF model returned when one dereferences the URI of a resource. The Linked Open Data views are one-to-one mappings from the URI to the complete entity description. This, however, is more of an exception than a rule; other kinds of web data publishing mostly provide only partial entity descriptions in the views. For example, in the case of Microformats or RDFa, views are pages that talk about different aspects of the entity, e.g., one listing social contacts for a person, another listing personal data and a third containing all the posts by a user.

[6] FOAF: `http://www.foaf-project.org/`

3. CREATION AND ORGANISATION OF THE DATA COLLECTION

The Sindice-2011 Dataset is based on documents containing semi-structured data gathered from the Web since 2009. More information about the provenance and content of the data is provided in §3.1. Given the initial format of the data, i.e., a set of semi-structured documents, an extraction step is necessary in order to transform the crawled data into a collection of entities (§3.2). Finally, because of the inherent structure of the Web of Data, we organise the collection of entities into two reusable hierarchical structures, each one providing a particular view of the information about an entity (§3.3).

3.1 General Characteristics of the Data Collection

Sindice started crawling Semantic Web data in 2009. Since then, the data has been continuously refreshed by the Sindice crawler. This ensures a relatively fresh snapshots of the Web of Data. The crawl covers various domains: e-commerce, social networks, social communications, events, scientific publications, and more. Part of the data is coming from Linked Data[7] datasets, such as DBpedia or Geonames. However, the collection also covers data published on the Web by individuals or large companies, such as LinkedIn, BestBuy, IMDB, and so forth. In total, the data is coming from more than 200 thousand second-level domains, as reported in Table 2. The collection covers various standardised data formats for web data publishing, such as RDF, RDFa, and Microformats. All the data is converted to RDF using Any23[8]. The dataset is highly heterogeneous. As shown in Table 2, the entities are described using more than 600 thousand ontologies and more than 5 million predicates.

3.2 Entity Extraction

A pre-processing step is required in order to extract entities from documents. The entity extraction algorithm is implemented using the MapReduce [5] programming model. In the map function in Alg. 1, the input key $< c >$ is the URL of a document, and the value $< s, p, o >$ is a triple in the document. The map phase simply outputs the same triple for two different join keys, one triple for the entity appearing as a subject and one triple for the entity appearing as an object. The outputs are then partitioned, sorted, and merged by the MapReduce framework. In the reduce phase, defined in Alg. 2, all the triples for each join key $< c, e >$ are aggregated together to build a graph containing the incoming and outgoing relations of an entity e. The resulting entity graph is similar to the sub-graphs pictured using dashed lines in Figure 1a.

Furthermore, we filter all entity graphs that are composed of only one or two triples. In general, such entity graphs do not bear any valuable information, and can be removed in order to reduce the noise as well as the size of the dataset.

The Table 1 reports two examples of extracted entities. For each one, the URL of the document, the identifier of the entity (URI or blank node identifier), and the content of the entity description is provided. Keeping track of the URL of the document is important when an entity is identified by a blank node, as the only way to uniquely identify such an entity is to combine the URL with the blank node identifier.

3.3 Data Organisation

The data collection is organised using two different structures, each one providing a particular view over the entities. The *document-entity centric format* (Sindice-DE) provides a representation of the entity within its context, the context, here, being a document. The *entity-document centric format* (Sindice-ED), on the other hand, provides a complete view of an entity across multiple contexts. These two structures are described in the next subsections.

Algorithm 1: Sindice-DE map function

input : Key $< c >$, Tuple $< s, p, o >$
output: Key $< c, e >$, Tuple $< s, p, o >$

`output` $(< c, s >, < s, p, o >)$
`output` $(< c, o >, < s, p, o >)$

Algorithm 2: Sindice-DE reduce function

input : Key $< c, e >$, List<Tuple> T
output: Key $< c, e >$, Graph g

g $\longleftarrow \emptyset$
for $< s, p, o > \in T$ **do**
  g.add$(< s, p, o >)$
end
`output` $(< c, e >$, g$)$

3.3.1 Document-Entity Centric Format

In this collection, data is organised hierarchically in a document-entity centric format: there is one node in the first level of the hierarchy (directory in the file system) for each document, and each second-level node (sub-directory) below represents an entity within a document. Such a collection can be directly generated from the output of the reduce function of Alg. 2. This collection is useful to keep a global view of the context (i.e., document) where the entity is found. In general, a document provides contextual information about an entity, such as a list of people known by the entity in a social network or a list of publications authored by the entity in scientific publisher databases. Such a data structure is useful in certain retrieval tasks, where contextual information can help in disambiguating an entity.

3.3.2 Entity-Document Centric Format

In this collection, data is organised hierarchically in an entity-document centric format: there is one node in the first level of the hierarchy (directory in the file system) for each entity, and each second-level node (sub-directory) below represents a document that contains information about the entity. This collection is useful in providing a global view of an entity across multiple contexts, i.e., documents where pieces of information about the entity have been found. Such a data structure is useful in certain retrieval tasks, where the completeness of information prevails over quality. Aggregating information across contexts, however, is a double-edged sword. It leads to a more complete picture about an entity, but can also lead to entity descriptions of lower quality, as there is no control on what people publish about an entity on the web.

To create such a collection, a second data pre-processing step over the output of Alg. 2 is required in order to group entity information across contexts. The entity aggregation algorithm is also implemented using the MapReduce programming model. In the map function in Alg. 3, the input key $< c, e >$ is composed of the URL of a document and the identifier of an entity within this document. The value is the entity graph computed in Alg. 2. The map phase simply annotates the graph with the URL of the document, and outputs the entity identifier as the join key and the annotated

[7]Linked Data: `http://linkeddata.org/`
[8]Any23: `http://any23.org/`

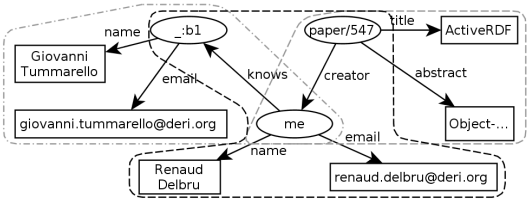
URL: http://dbpedia.org/resource/Modern_Times_(film)
ID: http://dbpedia.org/resource/Modern_Times_(film)

<http://dbpedia.org/resource/Modern_Times_(film)>	<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>	<http://dbpedia.org/ontology/Film> .
<http://dbpedia.org/resource/Modern_Times_(film)>	<http://dbpedia.org/property/director>	<http://dbpedia.org/resource/Charlie_Chaplin> .
<http://dbpedia.org/resource/Modern_Times_(film)>	<http://dbpedia.org/property/name>	"Modern Times" .
<http://dbpedia.org/resource/Modern_Times_(film)>	<http://dbpedia.org/property/writer>	"Paulette Goddard" .

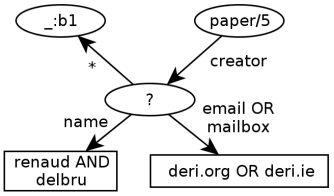
URL: http://belfast.ratemyarea.com/events/bumps-babies-and-pare-201663
ID: _:node997bf4dc6291719673b348d1a331f71

_:node997bf4dc6291719673b348d1a331f71	<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>	<http://www.w3.org/2006/vcard/ns#VCard> .
_:node997bf4dc6291719673b348d1a331f71	<http://www.w3.org/2006/vcard/ns#fn>	"The Windmill Restaurant" .
_:node997bf4dc6291719673b348d1a331f71	<http://www.w3.org/2006/vcard/ns#org>	_:node15qbkbg04x42625 .
_:node997bf4dc6291719673b348d1a331f71	<http://www.w3.org/2006/vcard/ns#url>	<http://belfast.ratemyarea.com/places/the-windmill-restaura-105433> .

Table 1: Examples of extracted entities



(a) Visual representation of an RDF graph. The RDF graph is divided (dashed lines) into three entities identified by the nodes *_:b1*, *me* and *paper/547*.



(b) Star-shaped query matching the entity me where ? is the bound variable and ⋆ a wildcard

Figure 1: In these graphs, oval nodes represent resources and rectangular ones represent literals. For space consideration, URIs have been replaced by their local names.

graph as the value. The outputs are then partitioned, sorted, and merged by the MapReduce framework. In the reduce phase, the identify function (i.e., a function that copies the supplied intermediate data to the output) is applied and the output is a list of annotated graphs for each entity.

Furthermore, as a blank node identifier is only valid within its context, entities with blank-node identifiers are filtered from this collection. We also filter all incoming relations of type `rdf:type`. An entity representing a common concept reused on the web, such as `foaf:Person`, can have hundreds of millions of incoming links. These incoming links may consist of a huge amount of data, adding heavy load to the processing, while providing only very limited information about the entity; therefore, it is preferable to filter them. For the same reasons, we also filter incoming relations of type `dbpedia:wikilinks`.

Algorithm 3: Sindice-ED map function

input : Key $< c, e >$, Graph g
output: Key $< e >$, Graph g

g.setContext(c)
`output` ($< e >$, g)

4. STATISTICS

In this section, we report and discuss statistics about the data collection. We organise these statistics into three groups: data heterogeneity (§4.1), entity size (§4.2), and term distribution (§4.3). Additionally, we check the coverage of the dataset for results from previous evaluation campaigns (§4.4).

4.1 Data Heterogeneity

Figure 3a shows the distribution of data format across documents, i.e., how many documents are using a particular data format.

Description	Statistic
Documents	230,643,642
Domains	4,316,438
Second-level domains	270,404
Entities	1,738,089,611
Statements	11,163,347,487
Bytes	1,274,254,991,725
Literals	3,968,223,334 (35.55%)
URIs (as objects)	5,461,354,581 (48.92%)
Blank nodes (as objects)	1,733,769,572 (15.53%)
Unique ontologies	663,062
Unique predicate URIs	5,396,273
Unique literal words	114,138,096
Unique URIs (as objects)	409,589,991

Table 2: Statistics about the Sindice-2011 dataset

It is worth noting that one document can use more than one format. More than half (142 million) of the documents are containing some RDFa markup, a third (87 million) are containing plain RDF. If we aggregate the different Microformats together, more than half (130 million) of the documents are containing some Microformats markup. This shows one aspect of the diversity of the data collection, but it also illustrates that all standardised data formats well covered.

With respect to schema heterogeneity, the collection of entities is described using more than 600 thousand ontologies and with more than 5 million predicates, as reported in Table 2. Figure 3b depicts the distribution of the frequency of ontologies used across documents, i.e., the probability for an ontology to be used in exactly n documents. The distribution shows a power-law distribution following a Zipf function with a slope of $\alpha = 2.27$. Notice that most ontologies (99%) are used only once. However, the distribution tail is sparse, suggesting that a few ontologies are used in a large proportion of documents. For example, one ontology (`http://purl.org/dc/terms/`) is used in more than 150 million documents and another one (`http://www.w3.org/2006/vcard/ns\#`) is used in more than 64 millions documents. In Figure 3c, we report the distribution of frequency of predicates used across documents, i.e., the probability for a predicate to be used in exactly n documents. Similarly to the distribution of frequency of use of ontologies, the frequency of use of predicate URIs shows a power-law distribution following a Zipf function with a slope of $\alpha = 1.45$. A large number of predicates (98%) are used only once. However, the distribution tail is sparse, suggesting that a few predicates are used in a large proportion of documents. For example, two predicates are used in more than half of the documents (`http://www.w3.org/1999/02/22-rdf-syntax-ns\#type` in 155 million documents and `http://purl.org/dc/terms/title` in 140 million documents), and six other predicates are used in more than a quarter of the documents.

Figure 2 depicts the number of statements for each triple pattern, where U stands for URI, L for Literal, and B for Blank Node. For example, the pattern UUU represents triples with URIs at the subject, predicate, and object positions. It is worth noticing that the dataset consists of a large proportion of blank nodes, most of them coming from Microformats documents, which are removed from Sindice-ED.

With respect to entity types covered by the dataset, it is difficult to give a precise number for a particular concept, as one concept can be described by many different classes and ontologies. Instead, we report some observations from the list of the top-100 classes based on their document frequency and their probability distribution[9]. The most frequently used types describe persons and organizations. Products and business entities defined using the *GoodRelations* e-commerce ontology[10] are also very frequent. Further popular types concern reviews, bibliographic resources, locations and geographical data, bio-medical data, events, movies, and music.

4.2 Entity Size

We measure the size of an entity in terms of the number of triples in its RDF graph. Recall that we removed all entities having less than 3 triples from the Sindice-2011 Dataset. Figure 4a and Figure 4b show the distribution of entity sizes, i.e., the probability for an entity to have exactly n triples, for Sindice-DE and for Sindice-ED, respectively. The distributions show a power-law distribution following a Zipf function with a slope of $\alpha = 3.6$ for Sindice-DE and $\alpha = 2.6$ for Sindice-ED. In Sindice-DE, most entities are very small: 25% of them have three of four triples and 60% have ten or fewer triples. The average entity size is 5.32 triples, with a maximum entity size of 19,617 triples. In Sindice-ED, the entities are generally larger, with 20% of them having three or four triples and 75% having ten or fewer triples. The average entity size is 14.19 triples, with a maximum entity size of 5,454,094 triples. Such difference in size is explained by the fact that entities in Sindice-ED are aggregates of information coming from multiple documents.

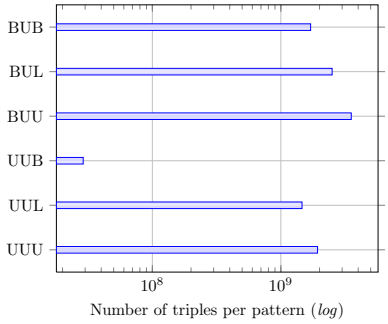


Figure 2: Distribution of triple patterns.

We also compute the length of a literal as the number of words in a literal. We exclude stop words and single character terms. Figure 4c shows the distribution of literal lengths, i.e., the probability for a literal to have exactly n words. The distribution shows a power-law distribution following a Zipf function with a slope of $\alpha = 2.58$. Most of the literals are very small, with 57% of them having exactly one word and 97% having ten or fewer words. The average literal length is 2.57 words, with a maximum literal length of 166,127 words.

4.3 Term Distribution

Fig 5a plots the distribution of words in literals and Figure 5b shows the distribution of URIs. The graphs depict the probability for a word or a URI to appear in exactly n documents. These distributions exhibit a similar slope ($\alpha \simeq 2$), indicating that a URI or a word follows the same probability of occurrence throughout the documents. However, the number of unique URIs (409,589,991) is orders of magnitude greater than the number of unique terms (114,138,096), which is due to the inherent function of URIs being used as unique identifiers (hence their cardinality is theoretically infinite).

In Figure 5c and in Figure 5d we plot, respectively, the distribution of literal terms over the predicates and the distribution of URIs over the predicates. The graphs show the probability, for each literal term or URI, to appear in exactly n distinct predicates. As in the previous graphs, these distributions exhibit a power-law distribution with slopes 2.36 (for literal terms) and 3.22 (for URIs). We note that the Zipf's α parameter in the URIs distribution is distinctly higher than in the literal terms distribution. This means that URIs often tend to appear jointly with a small number of predicates, while in literals this behavior is less pronounced. This is probably due to the fact that a term can represent different things and thus can appear in a larger set of predicates with different meanings.

4.4 Coverage

We checked all the URIs in the relevance assessment files from the 2010 and 2011 editions of the Semantic Search Challenge and from the ELC task of the 2010 TREC Entity track. The Sindice-

[9] `http://data.sindice.com/trec2011/resources/top100_class_distribution`

[10] `http://www.heppnetz.de/projects/goodrelations/`

2011 Dataset covers 99% of these URIs; the missing ones are either (1) not exist anymore on the Web or (2) synthetic URIs that were generated to replace blank nodes with URIs in the BTC-2009 dataset.

5. SEARCH TOOLS AND INFRASTRUCTURE

To support IR research and development with the Sindice-2011 Dataset, we provide a search infrastructure based on the Semantic Information Retrieval Engine, SIREn [6]. SIREn is an IR system designed for searching entities, specifically, according to the requirements of the Web of Data. Given a description of an entity, i.e., a star-shaped query such as the one in Figure 1b: locate the most relevant entities. Since RDF is semi-structured, SIREn is aimed to support three types of queries: (1) full-text search (keyword based) when the data structure is unknown, (2) semi-structured queries (complex queries specified in a star-shaped structure) when the data schema is known, or (3) a combination of the two (where full-text search can be used on any part of the star-shaped query) when the data structure is partially known. SIREn also supports top-k query processing using an adaptation of the well-known TF-IDF scoring function, providing a standard baseline for experimentation.

The developed tools provide functionality to process, index, and retrieve entities. Using this infrastructure, new entity retrieval models and algorithms can be rapidly deployed and tested on top of the Sindice-2011 Dataset. For example, one can implement a two-step retrieval process where the infrastructure is used to retrieve a subset of the data collection that might be relevant to a query, and then, in a subsequent step, more advanced processing and ranking techniques are performed on this result set. One can also (1) implement on top of the infrastructure a query expansion technique that combines full-text and structured queries, (2) modify the way entities are ranked by adding weights to certain attributes or values, or (3) (an advanced user) can change how entities are indexed and implement her own query ranking function directly within the query processing framework. The search infrastructure can be downloaded from `https://github.com/rdelbru/trec-entity-tool`.

6. SUMMARY

In this paper, we have presented the *Sindice-2011 Dataset*, a collection providing an accurate reflection of the current *Web of Data*, i.e., a collection of semi-structured data (e.g., RDF, RDFa, Microformats, etc.). Statistics about the dataset are reported, which can be useful for developing appropriate search systems. With a sum of 11 billion statements and with information organized around 1.7 billion entities, the Sindice-2011 Dataset represents a real-world data collection that allows the research community to make a significant step forward in web entity search. In order to further support the research in that direction, we also provide a search infrastructure based on *SIREn*, the Semantic Information Retrieval Engine. The Sindice-2011 Dataset is available at `http://data.sindice.com/trec2011/`.

Bibliography

[1] S. Abiteboul. Querying Semi-Structured Data. In *Proceedings of the 6th International Conference on Database Theory*, pages 1–18, 1997.

[2] K. Balog, A. P. de Vries, P. Serdyukov, P. Thomas, and T. Westerveld. Overview of the TREC 2009 entity track. In *Proceedings of the Eighteenth Text REtrieval Conference (TREC 2009)*. NIST, 2010.

[3] K. Balog, P. Serdyukov, and A. P. de Vries. Overview of the TREC 2010 entity track. In *Proceedings of the Nineteenth Text REtrieval Conference (TREC 2010)*. NIST, 2011.

[4] N. Craswell, A. D. Vries, and I. Soboroff. Overview of the TREC-2005 enterprise track. In *The Fourteenth Text REtrieval Conference Proceedings*, TREC 2005, 2006.

[5] J. Dean and S. Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51:107–113, January 2008.

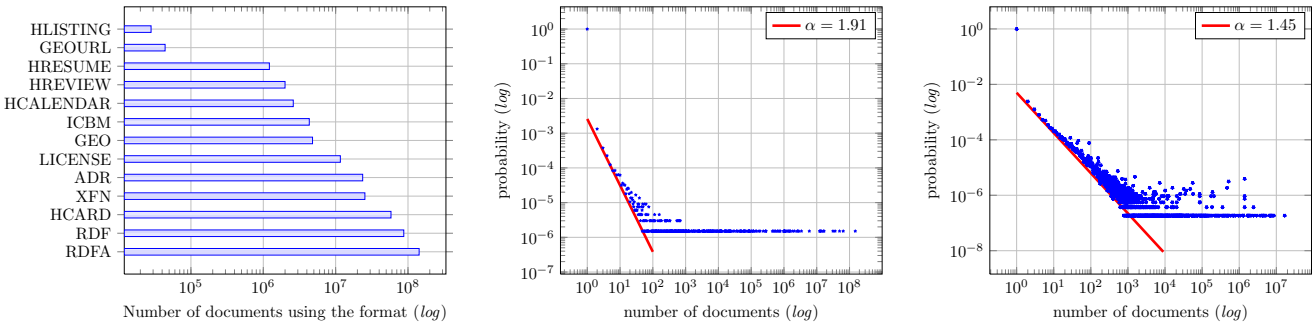
[6] R. Delbru, S. Campinas, and G. Tummarello. Searching web data: An entity retrieval and high-performance indexing model. *Web Semantics: Science, Services and Agents on the World Wide Web*, 2011.

[7] H. Halpin, D. M. Herzig, P. Mika, R. Blanco, J. Pound, H. S. Thompson, and D. T. Tran. Evaluating ad-hoc object retrieval. In *Proceedings of the International Workshop on Evaluation of Semantic Technologies*, IWEST 2010, 2010.

[8] E. Oren, R. Delbru, M. Catasta, R. Cyganiak, H. Stenzhorn, and G. Tummarello. Sindice.com: a document-oriented lookup index for open linked data. *Int. J. Metadata Semant. Ontologies*, 3:37–52, November 2008.

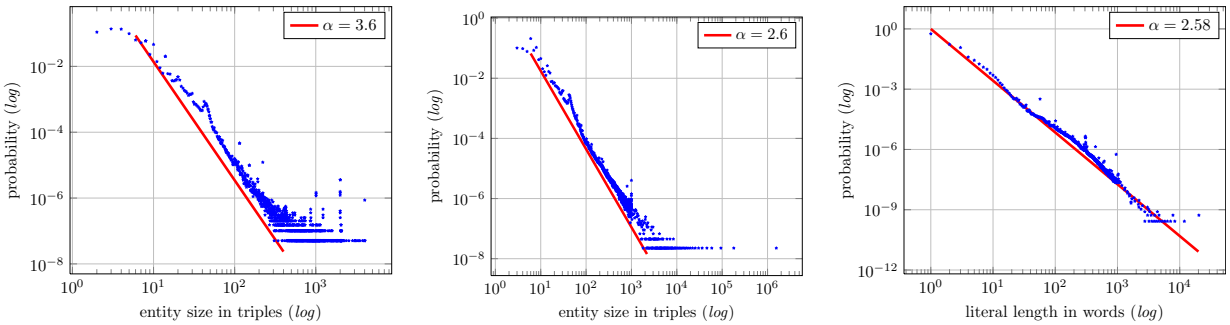
[9] J. Pound, P. Mika, and H. Zaragoza. Ad-hoc object retrieval in the web of data. In *Proceedings of the 19th international conference on World Wide Web*, pages 771–780, New York, New York, USA, 2010. ACM Press.

[10] A. P. Vries, A.-M. Vercoustre, J. A. Thom, N. Craswell, and M. Lalmas. Overview of the INEX 2007 entity ranking track. In N. Fuhr, J. Kamps, M. Lalmas, and A. Trotman, editors, *Focused Access to XML Documents*, volume 4862 of *Lecture Notes in Computer Science*, pages 245–251. Springer Verlag, Heidelberg, 2008.



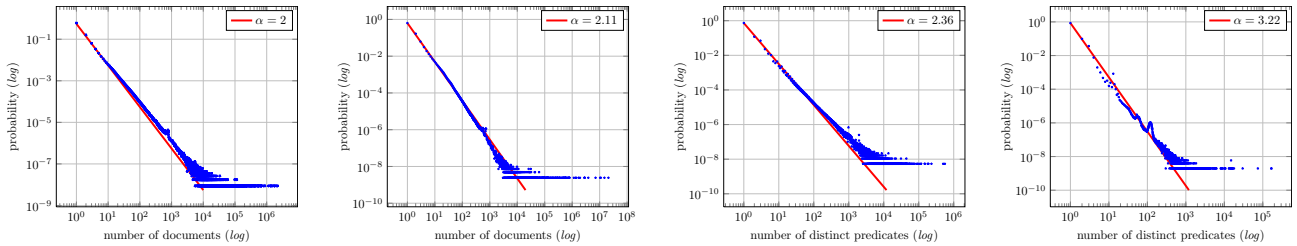
(a) Data formats (b) Frequency of ontologies used (c) Frequency of use of predicate URIs

Figure 3: Distributions of semantically structured data.



(a) Entity sizes in Sindice DE (b) Entity sizes in Sindice ED (c) Literal lengths

Figure 4: Entity size distributions.



(a) Terms in literals (b) URIs (c) Literal terms over predicates (d) URIs over predicates

Figure 5: Term distributions.