Fouilla: Navigating DBpedia by Topic

Tanguy Raynaud, Julien Subercaze, Delphine Boucard, Vincent Battu, Frederique Laforest

Fouilla: Navigating DBpedia by Topic

Tanguy Raynaud, Julien Subercaze, Delphine Boucard, Vincent Battu, Frédérique Laforest
Univ Lyon, UJM Saint-Etienne, CNRS, Laboratoire Hubert Curien UMR 5516
Saint-Etienne, France
firstname.lastname@univ-st-etienne.fr

ABSTRACT

Navigating large knowledge bases made of billions of triples is very challenging. In this demonstration, we showcase Fouilla, a topical Knowledge Base browser that offers a seamless navigational experience of DBpedia. We propose an original approach that leverages both structural and semantic contents of Wikipedia to enable a topic-oriented filter on DBpedia entities. We devise an approach to drastically reduce the query time and to ensure a seamless browsing experience to the end user. We demonstrate how our system offers a novel and meaningful experience of DBpedia browsing by challenging the user to search for relevant information within the Knowledge Base in different use cases.

1 INTRODUCTION

The construction of Knowledge Bases is a very active domain in both industry and academia. Within the Semantic Web community, a large movement emerged to automatically or manually build Knowledge Bases, especially using Wikipedia as input. Projects such as DBpedia [4], Yago [6], Wikidata [7] are the most famous representatives of this movement. These Knowledge Bases are stored using some triple representation such as the Resource Description Format (RDF) standard. The DBpedia 2014 instance contains for example more than 3 billion triples and serves as one of the central interlinking hubs in the Linked Open Data on the Web [4]. These data were intended for machine-to-machine exploitation; but a suitable presentation makes them also relevant directly for end users. For instance a list of facts concerning a famous person can be more constructive than reading a long biography, particularly if this list is restricted to a particular topic that interests the most the reader. For example, one can be interested in Yannick Noah's music, his navigation should be restricted to the topic of Music, even if he is also well known as a tennis player. However, finding relevant information among the several billions of triples contained in those datasets is a tedious task for the end-user. For instance, the DBpedia Italy page alone contains several thousands triples, displayed without order. Such an amount of triples generally leads to user resignation, considering the time needed to find the required piece of information.

In this paper, we introduce the concept of *Topical browsing*. Our aim is to enable browsing among large Knowledge Bases, by displaying to the end user, for each topic, only pertinent elements. The end user can therefore find adequate information without SPARQL knowledge. Faceted browsing [5] has been extensively covered by the semantic web community [1–3], it consists of filtering data by several criteria (i.e. date, language, etc.) restricted to the ontology classes and properties. In Topical Browsing, the user selects a topic (e.g. History), navigates among entities related to this topic and sees only the triples that concern this topic. For example, a user is interested in Italy through the prism of Sports while another through the prism of Word War II. For each of these topics, the relevant triples of the Italy entity differ. In such circumstances, faceted browsing offers no solution to retrieve the entities relative to a defined topic if the knowledge graph does not explicitly contain an adequate relation (such as ' *Romans* ***partOf*** *History*'). On a huge dataset like DBpedia, topical browsing presents its very own challenges:

Topics. Identifying topics, delimiting their boundaries in terms of relevant items are the main encountered issues.

Efficient browsing. Despite the size of the dataset, the user shall not wait endlessly for queries to complete.

We demonstrate a fully functional system called Fouilla [fuja], aiming to provide to end users a meaningful browsing experience among the billions of triples of DBpedia. To build the topical dataset, our approach uses both the structural and semantic information present in Wikipedia. For each navigation action, Fouilla returns only entities and triples relevant to the chosen topic and offers a seamless browsing experience. Moreover, it allows users with no SPARQL experience to efficiently browse Knowledge Bases made of several billions of triples. Readers can experiment Fouilla at the following url: http://demo-satin.telecom-st-etienne.fr/fouilla/.

The paper is organized as follows. Section 2 presents our approach to identify topics, extract and rank relevant facts. Section 3 deals with scalability issues with ranking queries.Section4 discusses our demonstration settings.

2 TOPICS BUILDING AND PROVISIONING

Topical browsing experience relies on two major criteria. First, the topics titles and their content must be understandable as they are by the users, they must reflect common sense expectations. Second, the elements must be displayed in a meaningful order, that is, the elements must appear ranked by their relevance to the topic. Figure 1 presents an overview of the three steps involved in Fouilla for the creation of the topical database.

2.1 Topic Identification

To identify the set of topics, we leverage the bijective link between DBpedia and Wikipedia: each DBpedia resource URI corresponds to a Wikipedia article name. Wikipedia contains structural and navigational information that are not present in DBpedia. Such information is the raw material for topic identification. Structural and navigational information encompass all Wikipedia resources that do not offer encyclopedic content but have been created to structure the online encyclopedia and to ease the navigation of the end user. Fouilla uses different kinds of Wikipedia structural information to compute the *Topic Identification* step:

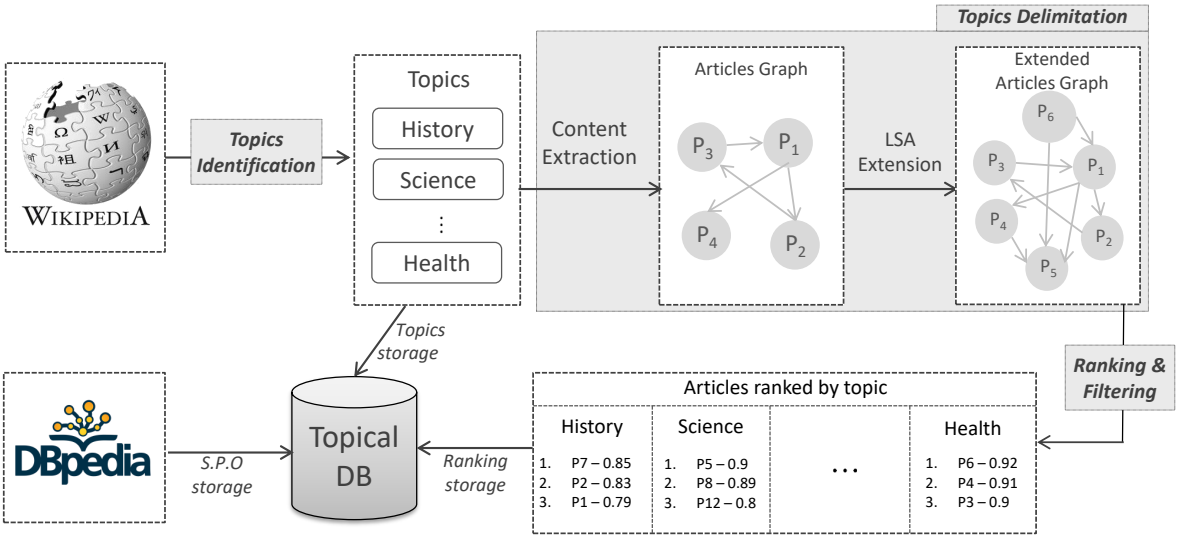


Figure 1: Overview of the three steps involved in the creation of the topical database.

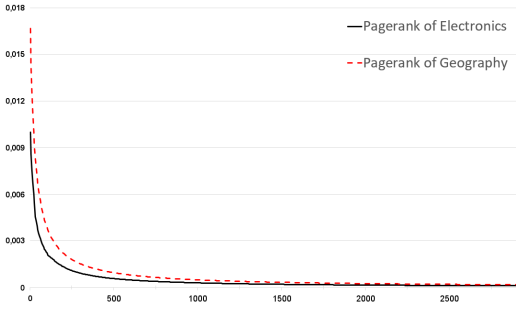


Figure 2: Examples of Topics Pagerank Score Distribution by Rank

Categories[1] are intended to group together pages on similar subjects. Contrary to the pages, they are organized as a hierarchical graph ranging from very general subjects (like *History*) to very precise ones (such as *Medieval legal texts*).

Portals carry semantics about the pages they link to. A *portal* serves as en entry point to navigate Wikipedia in a specific area e.g. *Geography*, or more precisely, *China*. As we write this paper, Wikipedia offers more than 1490 portals and among them, 173 are *featured portals*. *Featured portals*[2] have been cherry picked by Wikipedia editors as high-quality portals about general subjects and are therefore good candidates for our set of topics.

Outlines are special pages that provide a hierarchical list which presents the more relevant articles and categories associated with a given subject. The resources linked by *Outlines* can per consequent be safely considered as highly relevant to the corresponding subject. *Outlines* are categorized in a main page[3].

Lists are meta pages that provide a list of articles which are, in one way or another, relevant to a common subject.

In order to define the set of topics, Fouilla takes advantage of the aforementioned information contained in Wikipedia. In order to identify the topics automatically, the structure of Wikipedia is entirely analyzed, and data from featured portals, top level categories, and outlines are extracted and combined to finally obtain a list of 50+ topics. This list is available at the following URL: http://datasets-satin.telecom-st-etienne.fr/traynaud/Topics.html.

2.2 Topic Delimitation

Our next step is to determine which DBpedia triples are of interest for each topic. The huge number of triples, namely several billions, makes this task clearly the main challenge in the creation of our topical browsing system. In order to associate topics with the corresponding DBpedia triples, our approach consists of two steps.

The first step aims to bind topics with highly relevant pages by taking advantage of the structural and navigational information provided by Wikipedia. Our topic delimitation method uses a combination of data extracted from the *Outlines* and a supervised crawling of the *Categories*. Even if *Outlines* in Wikipedia do not follow a fixed format, they offer a great entry point for each corresponding topic: an empirical analysis has shown that all articles, categories and *Lists* of articles linked from *Outlines* can be considered as of the utmost interest regarding a topic. A supervised crawling on the *Categories* is then used to complement the set of relevant articles. In order to avoid the insertion of unrelated articles, a recursive crawling filters any branch that leads to a cycle, and

[1] https://en.wikipedia.org/wiki/Category:Wikipedia_categories
[2] https://en.wikipedia.org/wiki/Wikipedia:Featured_portals
[3] https://en.wikipedia.org/wiki/Portal:Contents/Outlines

	$PR < 5.10^{-6}$		$PR < 10^{-5}$		$PR < 5.10^{-5}$	
	NDCG	Speedup	NDCG	Speedup	NDCG	Speedup
Arts (1.6)	.99	26.89	.96	52.8	.58	122
Politics (1.2)	1	19.87	1	36.08	.76	84.35
⋮	⋮	⋮	⋮	⋮	⋮	⋮
WW1 (0.1)	1	.97	1	1.71	.99	3.24
Average	.99	6.19	.96	10.02	.68	22.75

Table 1: Experimental result for various pagerank (PR) cuts. NDCG@100 and query speedup are presented for sample topics and in average.

never explores a branch with a depth greater than four starting from the seed category.

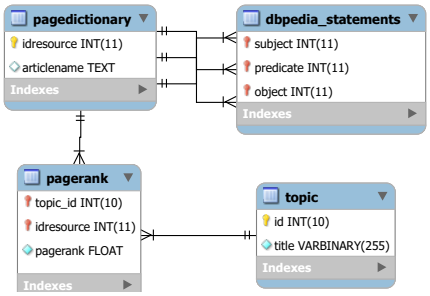


Figure 3: Database Layout

The second step enlarges the set of relevant articles obtained in step 1. While structural and navigational information provides us with highly relevant pages, due to their very nature, relevant articles may not be enlisted in these resources. The initial set was constructed using structural information from Wikipedia. To offer a complementary approach, this second step relies on the content of articles: we include additional articles that are semantically close to the articles extracted at step 1. For this purpose, we use Latent Semantic Analysis (LSA) [8]. LSA bases the similarity measure on a matrix of word count for each article, processed with the method of cosine distances. To ensure high quality results, we leverage the pagerank distribution of articles within a topic. We experimentally noticed that pagerank follows a similar distribution across the topics; Figure 2 shows this distribution. We then use the most relevant pages as input to the LSA, that is, we leverage derivative information to induce a cut before the pagerank score stagnation. We observe that half the articles obtained by LSA are not redundant with those obtained in the first step. This semantic expansion allows us to obtain the desired result: increasing recall while not having to trade for precision.

2.3 Triples Ranking & Filtering

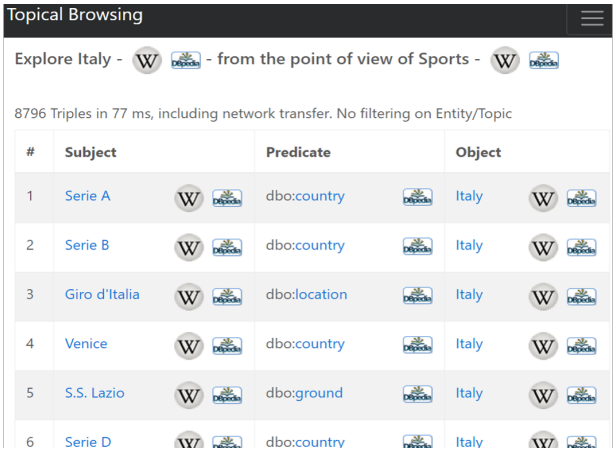
After the expansion step, we operate a final cut to ensure that irrelevant entities and triples are discarded. During the topic identification step, we computed a per-topic pagerank on the subgraph of articles obtained from structural information. We here recompute such a pagerank, including elements obtained with LSA. This pagerank provides us with a relevance score for each article within the topic. We consider as significant within a topic, the pages that are strongly related within the graph, using a similar technique as the one to restrict the LSA input, i.e. based on the derivative of the distribution. This cut approach based on the derivative is generic and enables us to compute a per-topic threshold as opposed to a global threshold, that would lead to poor results. Once again, regarding the pagerank distribution, we consider as semantically unrelated any article located beyond curve stagnation.

3 SCALABILITY AND USER EXPERIENCE

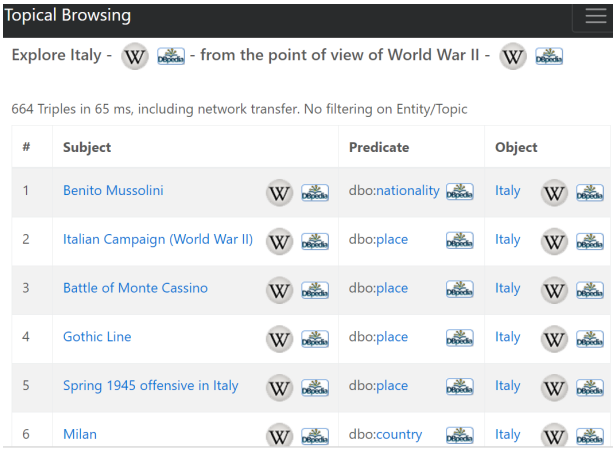
Data storage relies on an open-source RDBMS, following the schema depicted in Figure 3. Each DBpedia concept is assigned a unique integer ID through the table *pagedictionary*, their pagerank per topic are to be found in the table *pagerank*. The triples are represented in the table *dbpedia_statements*. Querying the database is very efficient for listing top-k entities for a given topic, thanks to indexes on *pagerank*. However, for listing top-k triples for a specific topic, the database must compute a slow query: an aggregated score is computed for each triple, then the results are sorted and displayed. The table *pagerank* contains more than 20M rows. Even with a carefully tuned server and NVM-E disks, these queries run in over 15 seconds, which is not acceptable for the end user experience. To address this issue, we rely on the pagerank distribution presented in Figure 2. Our intuition is that removing the tail of the pages in each topic will not affect the ordering of the top-k, for reasonable values of k. For this purpose, we experimented different *cuts* in the data, based on the pagerank value. We evaluated their influence on the rank, as well as the query speedup. To take fully advantage of the cut, we added an index on *(topic_id,pagerank)* on table *pagerank*.

To evaluate the influence of cuts on ranking, we use the normalized discounted cumulative gain (NDCG), which is a standard measure in information retrieval. The NDCG ($\in [0, 1]$) measures the quality of a ranking against an ideal ranking. In our case the ideal ranking is the one obtained without limit on the pagerank values. The second valuable measure is the query speedup obtained via the cut. Table 1 summarizes our experiments, for NCDG up to the 100th item, which is equivalent to 10 pages of browsing. The experiments validate our intuition insofar as for the first value of the cut (5.10^{-5}), we are able to obtain a significant average speedup of 6 times against a minimal impact on the NDCG (0.99). For greater values, the trade-off becomes less acceptable. One of the most important insights from Table 1 is that, the larger the topic, the greater the speedup. For instance for the topic *Politics* containing 1.2M items, the speedup is much greater than the average over all topics. From a practical point of view, this means that the slowest queries are the ones that benefit the most from the cut. Since the number of items is greatly reduced, the $n \cdot log(n)$ sorting time that dominates the query follows.

To summarize, the browsing of topics uses a pagerank cut at 5.10^{-5} to ensure a seamless user experience. For entity level browsing, the selectivity on the entity being very high, the cut is not required and would be a disadvantage as it would reduce the number of displayed items.



(a) Sport



(b) WWII

Figure 4: Views of the entity *Italy* from two (very) different topics

4 SYSTEM DEMONSTRATION

We have developed a Web Interface to the topical database. It is available at the following url: http://demo-satin.telecom-st-etienne.fr/fouilla/. Users can experience the various navigation paths, and evaluate the added value of our system when dealing with DBpedia contents on different scenarios.

Browsing Efficiency The visitor at our booth chooses a topic of his/her interest and searches the relevant pages using the standard DBpedia Web Interface. Since it is not tailored for this type of search, the user will have difficulties in finding relevant information on the topic.In order to demonstrate the efficiency of our approach, the user is asked to restart the experience, but through our Fouilla interface : he/she gets accurate results in a few clicks.

Topic Exploration What are the most important facts and entities related to a topic? We here showcase the visitor how our system can be used to explore the facts regarding a given topic. Once the topic is selected, its corresponding top-ranked entities are retrieved and displayed on the web interface. Then, the user can navigate through the entities, predicates and triples using the provided controls. For instance, Wikipedia and DBpedia icons located close to the entities names allow fast navigation to the corresponding pages.

Entity View DBpedia entities (such as cities, countries, years, etc...) consist of a large and various number of triples of any kind. Such entities commonly contain many thousands of triples. Thus, they are difficult to visualize. For instance, finding relevant information on the topic *Word War II* in *Italy* is practically impossible with the DBpedia Web interface, because of the random display of thousands of triples. We here present the user the filtering and domain delimitation capability of our system. Figure 4 depicts such an example of the entity *Italy* viewed from two topics: *Sports* and *Word War II*. As one can see, our system retrieves only triples that are relevant for both topics. The hyperlinks on entities are entry points for further topical navigation.

Finally, several sample queries are available and runnable with and without pagerank filtering. The visitor can judge the gain on the user experience as well as the negligible impact on ranking.

5 CONCLUSION

In this paper, we introduce the concept of *Topical Browsing*, and propose the demonstration of Fouilla, the first topical browser on DBpedia. We automatically extracted a set of 50+ topics from Wikipedia structural and semantic content and we delimited subgraphs for each topic within DBpedia using both ranking and machine learning techniques. We experimentally validated our pagerank cut that enables to drastically reduce the application response time, without loss of ranking quality. Our system offers a novel browsing experience in the billions of triples of the Knowledge Base. The proposed topical delimitation approach allows the user to find a needle in a very large haystack without any SPARQL knowledge.

REFERENCES

[1] L. Dali, B. Fortuna, et al. Query-independent learning to rank for RDF entity search. *The semantic web: Research and applications*, pages 484–498, 2012.
[2] L. Ding, R. Pan, et al. Finding and ranking knowledge on the Semantic Web. In *ISWC*, pages 156–170, 2005.
[3] A. Hogan, S. Decker, and A. Harth. Reconrank: A scalable ranking method for semantic web data with context. In *Workshop on Scalable Semantic Web Knowledge Base Systems*, 2006.
[4] J. Lehmann, R. Isele, M. Jakob, et al. DBpedia–a large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web*, 6(2):167–195, 2015.
[5] K. A. Ross et al. A faceted query engine applied to archaeology. In *VLDB*, pages 1334–1337, 2005.
[6] F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: a core of semantic knowledge. In *WWW*, pages 697–706. ACM, 2007.
[7] D. Vrandečić and M. Krötzsch. Wikidata: a free collaborative knowledgebase. *CACM*, 57(10), 2014.
[8] P. Wiemer-Hastings et al. Latent semantic analysis. In *IJCAI*, pages 1–14, 2004.