

Faceted Wikipedia Search

Rasmus Hahn¹, Christian Bizer², Christopher Sahnwaldt¹, Christian Herta¹,
Scott Robinson¹, Michaela Bürgle¹, Holger Düwiger¹, Ulrich Scheel¹

¹ neofonie GmbH

`firstname.lastname@neofonie.de`

<http://www.neofonie.de>

Robert-Koch-Platz 4, 10115 Berlin, Germany

² Freie Universität Berlin

`chris@bizer.de`

<http://www.wiwiss.fu-berlin.de/en/institute/pwo/bizer/>

Garystraße 21, 14195 Berlin, Germany

Abstract. Wikipedia articles contain, besides free text, various types of structured information in the form of wiki markup. The type of wiki content that is most valuable for search are Wikipedia infoboxes, which display an article's most relevant facts as a table of attribute-value pairs on the top right-hand side of the Wikipedia page. Infobox data is not used by Wikipedia's own search engine. Standard Web search engines like Google or Yahoo also do not take advantage of the data. In this paper, we present Faceted Wikipedia Search, an alternative search interface for Wikipedia, which facilitates infobox data in order to enable users to ask complex questions against Wikipedia knowledge. By allowing users to query Wikipedia like a structured database, Faceted Wikipedia Search helps them to truly exploit Wikipedia's collective intelligence.

Key words: faceted search, faceted classification, Wikipedia, DBpedia, knowledge representation

1 Introduction

This paper presents *Faceted Wikipedia Search*, an alternative search interface for the English edition of Wikipedia. *Faceted Wikipedia Search* allows users to ask complex questions, like “Which rivers flow into the Rhine and are longer than 50 kilometers?” or “Which skyscrapers in China have more than 50 floors and were constructed before the year 2000?” against Wikipedia knowledge. Such questions cannot be answered using keyword-based search as provided by Google, Yahoo or Wikipedia's own search engine.

In order to answer such questions, a search engine must facilitate structured knowledge which needs to be extracted from the underlying articles. On the user interface side, a search engine requires an interaction paradigm that enables inexperienced users to express complex questions against a heterogeneous information space in an exploratory fashion.

For formulating queries, *Faceted Wikipedia Search* relies on the faceted search paradigm. Faceted search enables users to navigate a heterogeneous information space by combining text search with a progressive narrowing of choices along multiple dimensions [6, 7, 5]. The user subdivides an entity set into multiple subsets. Each subset is defined by an additional restriction on a property. These properties are called the facets. For example, facets of an entity “person” could be “nationality” and “year-of-birth”. By selecting multiple facets, the user progressively expresses the different aspects that make up his overall question. Realizing a faceted search interface for Wikipedia poses three challenges:

1. Structured knowledge needs to be extracted from Wikipedia with precision and recall that are high enough to meaningfully answer complex queries.
2. As Wikipedia describes a wide range of different types of entities, a search engine must be able to deal with a large number of different facets. As the number of facets per entity type may also be high, the search engine must apply smart heuristics to display only the facets that are likely to be relevant to the user.
3. Wikipedia describes millions of entities. In order to keep response times low, a search engine must be able to efficiently deal with large amounts of entity data.

Faceted Wikipedia Search addresses these challenges by relying on two software components: The *DBpedia Information Extraction Framework* is used to extract structured knowledge from Wikipedia [4]. *neofonie search*, a commercial search engine, is used as an efficient faceted search implementation.

This paper is structured as follows: Section 2 describes the *Faceted Wikipedia Search* user interface and explains how facets are used for navigating and filtering Wikipedia knowledge. Section 3 gives an overview of the *DBpedia Information Extraction Framework* and the resulting DBpedia knowledge base. Section 4 describes how the efficient handling of facets is realized inside *neofonie search*. Section 5 compares *Faceted Wikipedia Search* with related work.

2 User Interface

This section describes how queries are formulated as a series of refinements within the *Faceted Wikipedia Search* user interface. *Faceted Wikipedia Search* is publicly accessible at <http://dbpedia.neofonie.de>. Several example queries are found at <http://wiki.dbpedia.org/FacetedSearch>. Figure 1 shows a screen shot of the interface. The main elements of the interface are:

1. *Text Search*: Free-text search terms can be entered into this search field.
2. *Faceted Navigation*: The most frequent values of the relevant facets are displayed in the faceted navigation. The user can define filters by selecting or entering values.

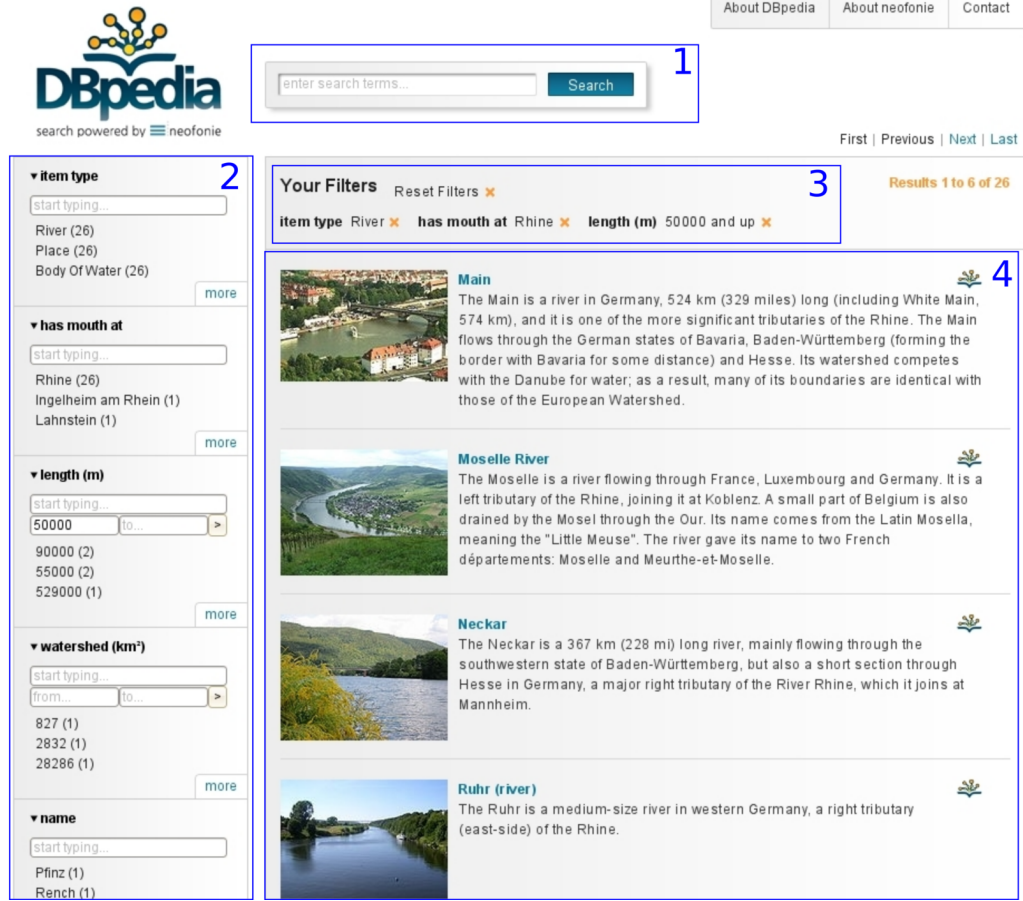


Fig. 1. Screen shot of the *Faceted Wikipedia Search* user interface. The facets are shown in the leftmost area of the screen. The numbers in brackets are the number of results corresponding with each facet value. *Your Filters:* (Area 3) displays the breadcrumb of the selected facet values: *item type River* with properties *has mouth at Rhine* and *length more than 50000*.

3. *Your Filters:* A breadcrumb navigation displays the selected facet values and search terms. Facets and search terms can be disabled independently of each other by clicking on the corresponding delete button.
4. *Search Results:* The search results contain the titles of the matching Wikipedia articles, a teaser of each articles' text, and an image from each article (if existent).

To formulate the question "Which rivers flow into the Rhine and are longer than 50 kilometers?", a user would go through the following steps:

1. On the start page of the *Wikipedia Faceted Browser*, the user would type the value “River” into the facet `item type`. As a result, 12,432 “River” entities are shown.
2. With the selection of “More Facets”, the `has mouth at` facet will be displayed. The user types “Rhine” into “has mouth at” entry field, which restricts the results to the 32 rivers which flow into the Rhine.
3. To define the numeric-range constraint, he types 50000 in the “from” field of the facet `length (m)`. As result 26 entities which match the complete query are returned.

In addition to the exploration of the entity space using facets, users can also mix full-text search with facet selection.

3 DBpedia

Faceted Wikipedia Search relies on the *DBpedia knowledge base* to answer queries. The knowledge base is provided by the DBpedia project [4], a community effort to extract structured information from Wikipedia and to make this information available on the Web under an open license. This section describes the information extraction framework that is used to generate the *DBpedia knowledge base* as well as the knowledge base itself.

3.1 The DBpedia Extraction Framework

Wikipedia articles consist mostly of free text, but also contain various types of structured information in the form of wiki markup. Such information includes infobox templates, categorization information, images, geo-coordinates, links to external Web pages, disambiguation pages, redirects between pages, and links across different language editions of Wikipedia. The DBpedia project extracts this structured information from Wikipedia and turns it into an RDF knowledge base [9].

The type of Wikipedia content that is most valuable for the DBpedia extraction are infoboxes. Infoboxes display an article’s most relevant facts as a table of attribute-value pairs on the top right-hand side of the Wikipedia page. The Wikipedia infobox template system has evolved over time without central coordination. Therefore, different communities of Wikipedia editors use different templates to describe the same types of things (e.g. `infobox_city_japan`, `infobox_swiss_town` and `infobox_town_de`). Different templates use different names for the same property (e.g. `birthplace` and `place-of-birth`). As many Wikipedia editors do not strictly follow the recommendations given on the page that describes a template, property values are expressed using a wide range of different formats and units of measurement.

In order to deal with the problems of synonymous attribute names and multiple templates being used for the same type of things, the DBpedia project maps Wikipedia templates onto an ontology using a custom mapping language.

This ontology was created by manually arranging the 550 most commonly used infobox templates within the English edition of Wikipedia into a subsumption hierarchy consisting of 205 classes and by mapping mapping 3200 infobox attributes to 1843 properties of these classes. The property mappings define fine-grained rules on how to parse infobox values and define target datatypes, which help the parsers to process property values. For instance, if a mapping defines the target datatype to be a list of links, the parser will ignore additional text which may be present in the property value. The ontology currently uses 55 different datatypes. Deviant units of measurement are normalized to one of these datatypes.

3.2 The DBpedia Knowledge Base

The DBpedia knowledge base currently consists of around 479 million RDF triples, which have been extracted from the English, German, French, Spanish, Italian, Portuguese, Polish, Swedish, Dutch, Japanese, Chinese, Russian, Finnish, Norwegian, Catalan, Ukrainian, Turkish, Czech, Hungarian, Romanian, Volapük, Esperanto, Danish, Slovak, Indonesian, Arabic, Korean, Hebrew, Lithuanian, Vietnamese, Slovenian, Serbian, Bulgarian, Estonian, and Welsh versions of Wikipedia. The knowledge base describes more than 2.9 million entities. For 1.1 million out of these entities, the knowledge base contains clean infobox data which has been extracted using the mapping-based approach described above. The knowledge base features labels and short abstracts in 30 different languages; 609,000 links to images; 3,150,000 links to external web pages; 415,000 Wikipedia categories, and 286,000 YAGO categories [12]. Table 1 gives an overview of common DBpedia classes, and shows the number of instances and some example properties for each class.

Besides being provided for download in the form of RDF dumps, the DBpedia knowledge base is also accessible on the Web via a public SPARQL endpoint and is served as Linked Data [2]. In order to enable DBpedia users to discover further information, the DBpedia knowledge base is interlinked with various other data sources on the Web according to the Linked Data principles [2]. The knowledge base currently contains 4.9 million outgoing data links that point at complementary data about DBpedia entities, as well as meta-information about media items depicting an entity. Altogether, the Web of interlinked data around DBpedia provides approximately 13.1 billion pieces of information (RDF triples) and covers domains such as geographic information, people, companies, films, music, genes, drugs, books, and scientific publications [1].

In the future, the data links between DBpedia and the external databases will allow applications like *Faceted Wikipedia Search* to answer queries based not only on Wikipedia knowledge but based on a world wide web of databases.

Ontology Class	Instances	Example Properties
Person	282,000	name, birthdate, birthplace, employer, spouse
Artist	54,262	activeyears, awards, occupation, genre
Actor	26,009	academyaward, goldenglobeaward, activeyears
MusicalArtist	19,535	genre, instrument, label, voiceType
Athlete	74,832	currentTeam, currentPosition, currentNumber
Politician	12,874	predecessor, successor, party
Place	339,000	lat, long
Building	23,304	architect, location, openingdate, style
Airport	7,971	location, owner, IATA, lat, long
Bridge	1,420	crosses, mainspan, openingdate, length
Skyscraper	2,028	developer, engineer, height, architect, cost
PopulatedPlace	241,847	foundingdate, language, area, population
River	12,432	sourceMountain, length, mouth, maxDepth
Organisation	119,000	location, foundationdate, keyperson
Band	14,952	currentMembers, foundation, homeTown, label
Company	20,173	industry, products, netincome, revenue
Educ.Institution	29,154	dean, director, graduates, staff, students
Work	189,620	author, genre, language
Book	15,677	isbn, publisher, pages, author, mediatype
Film	44,680	director, producer, starring, budget, released
MusicalWork	101,985	runtime, artist, label, producer
Album	74,055	artist, label, genre, runtime, producer, cover
Single	24,597	album, format, releaseDate, band, runtime
Software	5,652	developer, language, platform, license
TelevisionShow	10,169	network, producer, episodenummer, theme

Table 1. Common DBpedia classes with the number of their instances and example properties.

4 Faceted Search Implementation

This section gives an overview of the requirements that had to be met by the *Faceted Wikipedia Search* implementation as well as the approach that is used to select the potentially relevant subset of facets that is displayed to the user and the approach that is used to represent facet values in memory.

4.1 Requirements

In *Faceted Wikipedia Search*, each document is ordered to an item type, which the facets are then assigned to. For example, a document about a person may have a property *nationality*, but this property makes little sense when ordered to a document about a celestial body. But, a facet *age* would make sense for both documents. Therefore, a collection of documents consisting of a large variety of themes, like Wikipedia, will need a large total number of facets, but only a small number of facets per document will be needed. The statistical characteristics of the documents are shown in Table 2. In other scenarios, for example, an online

shop, a much smaller number of total facets would be required, but documents would have more facets in common, e.g. price.

Property	Value
number of documents	1134853
number of types	205
number of different facets	1843
average number of unique facets per document	14.8
number of values	24368625
number of unique values	5569464
average number of values per document	21.5
average number of values per facet	13222

Table 2. Statistical characteristics of the documents of *Faceted Wikipedia Search*.

For the user, only two aspects of a faceted-search system are readily apparent:

- *Facet Display*: For any set of documents (e.g. search result set) the facets and facet values are displayed. For example, a user is returned a set of documents, some of which correspond to the item type person. The user would be presented the *first-name* facet of the document set. The user would then see that there are 53 persons named *John*, and 63 people named *James*, etc.
- *Faceted Search*: The user can narrow the set of chosen documents based on the value of one or more facets by selecting a facet value. Technically, this is the intersection of one set of documents with another set which has the specific, selected values to the corresponding facets. Generally, for an implementation of a faceted-search system, this means that when the user selects a facet value, the search results reflect this selection.

4.2 Actual Implementation

In our implementation of faceted search these two aspects were implemented independently from one another to a large extent. This is mainly due to the fact that both aspects were implemented in an existing information retrieval engine¹, which already had various methods for document selection. Most notably, the possibility of performing boolean queries [10] is a previously existing feature of the search engine. Some adaptations were necessary to index facet values within documents (no normalization, special tokenization). In the case of faceted search, however, the user only selects the values that are presented to him, without requiring him to enter keywords. Therefore, there is no need for normalization and tokenization.

¹ We used our proprietary full text retrieval system for the implementation

Facet Display

The selection of the facet values which are to be displayed is dependent on the number of corresponding documents in the currently displayed document set. This set is determined by a previous search. *Wikipedia Faceted Search* offers the user two possibilities of search: first, through the selection of facet values and second, through a traditional full-text search.

The facet values are presented to the user as a list. The order of the values presented is dependent on the number of documents concerning a particular facet value. That means, for the selected document set the number of documents with the same facet value for any facet is counted. The values are then ordered by the absolute number of documents corresponding to a particular facet.

This ordering of values by number of occurrences of facet values is not necessarily the only or most comprehensible for the user; there are many facets which have a natural order (mainly numeric facets like e.g. *year*), but in the DBpedia Search we do not use this.

Due to limitations of the user interface and diversity of documents, not all facets and their values can be presented in a user friendly way. Therefore, the facet values which are displayed are limited to a set that can clearly be represented. The set of facets which is retrieved from the system is preselected at the time of the query. These we define as the target facets. This selection is primarily done to keep the number of round trips, and the amount of data transferred, small. This issue is readily apparent in the DBpedia system, as the documents are heterogeneous, i.e. many facets are only defined for a small subset of documents and only a few facets are shared between all documents.

To determine the target facets which are queried, we distinguish between three cases:

1. At the start of the user-session (without any search or selection) only the item type facet is displayed.
2. If there is no *item type* selected, the most generic facets, *item-type*, *location* and *year-of-appearance* etc. are target facets, since these are facets of the majority of documents.
3. If the user has selected an *item type*, the most frequent facets of the *item type* are target facets.

The resulting target facets (of 2, 3) are ranked according to their most frequent facet value. Only the target facets with the highest value frequencies are displayed.

Faceted Search

Conceptually, the facet information in the data is a set of tuples of document, facet and value, where a tuple (f, d, v) represents that a document d has a value v in the facet f . After the selection of a subset of documents D_q as a result of a query q , and a choice of the facets F_q , the set of resulting facet values must be calculated. For each facet and value the number of documents for this

combination is returned as a response to the query. That is, given a subset of documents D_q and a subset of facets F_q we must calculate $|\{(f, d, v) | d \in D_q\}|$ for each v and $f \in F_q$ efficiently. We do not know the set of documents D_q in advance, which leads to the difficulty in calculating the number of these tuples. We also do not know the set of facets F_q in advance, but as there are not many facets in total, this does not pose much of a problem. However, as we have a sizable amount of documents, this forces us to use a data representation which allows us to represent the facet values efficiently. To accomplish this, we use a (sparse) tree.

In the tree, the facet values are stored in a hierarchical data structure with three levels, the first level being the facets, the second level being the documents and the third level the values. This particular ordering is not strictly mandatory, but since the query output is ordered by facet first, it is more memory-efficient to reflect this in the data-structure, since each facet can then be calculated independently. This also allows for more efficient memory usage in the case that not all facets are queried, as we only need facets to be in the memory when they are being used.

As it turns out, this design is especially useful for the DBpedia use case where there is a large number of facets in relation to the number of documents. By having the facets in the first level of the tree structure, the amount of data to be examined is efficiently reduced.

5 Related Work

The following section is dedicated to discussing a sample of the related work on faceted search and Wikipedia information extraction.

Faceted Search. An early faceted search prototype was the “flamenco” [5] system developed at the University of California. “flamenco” is implemented on top of a SQL-database and uses the `group by`-command and specific optimizations [3]. This setup was developed without a full-text search engine.

In Yitzhaz et. al. [14], a hierarchically faceted search implementation in the *Lucene* search library is described. The facet value-information for the documents is stored in the *Payload* of a dedicated posting list (*FacetInfo*). The values are counted with a customized `HitCollector` for the target facets. The main difference to our approach is that their method aggregates by document first while our approach aggregates by facet. In our opinion, our implementation is better suited for the Wikipedia document collection (see 4.2).

Today, many commercial websites use faceted search. Examples include eBay and Amazon. A faceted search system that works on similar content as Faceted Wikipedia Search is Freebase Parallax². Parallax focuses on extending faceted search to a chained-sets navigation paradigm, while Faceted Wikipedia Search aims at providing a simple, self-explanatory search interface for Wikipedia.

² <http://www.freebase.com/labs/parallax/>

Extraction of structured Wikipedia content. A second Wikipedia knowledge extraction effort is the Freebase Wikipedia Extraction (WEX) [11]. Freebase³ is a commercial company that builds a huge online database which users can edit in a similar way as editing Wikipedia articles. Freebase employs Wikipedia knowledge as initial content for their database that will afterwards be edited by Freebase users. By synchronizing the DBpedia knowledge base with Wikipedia, DBpedia in contrast relies on the existing Wikipedia community to update content. Since November 2008, Freebase is published as Linked Data, and DBpedia as well as Freebase include data links pointing to corresponding entities in the respective other data source. These links allow applications to fuse DBpedia and Freebase knowledge.

A third project that extracts structured knowledge from Wikipedia is the YAGO project [12]. YAGO extracts 14 relationship types, such as `subClassOf`, `type`, `diedInYear`, `bornInYear`, `locatedIn` etc. from Wikipedia category system and from Wikipedia redirects. YAGO does not perform an infobox extraction like DBpedia. The YAGO and DBpedia projects cooperate and we serve the resulting YAGO classification together with the DBpedia knowledge base.

In [13] the KOG system is presented, which refines existing Wikipedia infoboxes based on machine learning techniques using both SVMs and a more powerful joint-inference approach expressed in Markov Logic Networks. In conjunction with DBpedia, KOG gives Wikipedia authors valuable insights about inconsistencies and possible improvements of infobox data.

NLP-based knowledge extraction. There is a vast number of approaches employing natural language processing techniques to obtain semantics from Wikipedia. Yahoo! Research Barcelona, for example, published a semantically annotated snapshot of Wikipedia⁴, which is used by Yahoo for entity ranking [15]. A commercial venture, in this context, is the Powerset search engine⁵ which uses NLP for both understanding queries in natural language as well retrieving relevant information from Wikipedia. Further potential for the DBpedia extraction as well as for the NLP-field in general lies in the idea of using huge bodies of background knowledge — like DBpedia — to improve the results of NLP-algorithms [8].

6 Conclusion

We have presented *Faceted Wikipedia Search*, an alternative search interface for Wikipedia, which facilitates infobox data in order to enable users to ask complex queries against Wikipedia. The answers to these queries are not generated using key word matching like the search engines Google or Yahoo, but are generated based on structured knowledge that has been extracted and combined from many different Wikipedia articles.

³ <http://www.freebase.com>

⁴ http://www.yr-bcn.es/dokuwiki/doku.php?id=semantically_annotated_snapshot_of_wikipedia

⁵ <http://www.powerset.com>

In future projects, we plan to extend the user interface of *Faceted Wikipedia Search* with more sophisticated facet value selection components like maps, timeline widgets and the automatic binning of numerical and date values. We also plan to complement and extend the application’s knowledge base by fusing Wikipedia infobox data with additional data from external Linked Data sources.

References

1. Christian Bizer. The emerging web of linked data. *IEEE Intelligent Systems*, 24:87–92, 2009.
2. Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data - the story so far. *Int. J. Semantic Web Inf. Syst.*, 5(3):1–22, 2009.
3. Kevin Chen. Computing query previews in the flamenco system. Technical report, University of Berkeley, 2004.
4. Christian Bizer et al. Dbpedia - a crystallization point for the web of data. *Journal of Web Semantics*, 7(3):154–165, 2009.
5. Jennifer English, Marti Hearst, Rashmi Sinha, Kirsten Swearingen, and Ka-Ping Yee. Flexible search and navigation using faceted metadata. Technical report, University of Berkeley, 2002.
6. Marti Hearst, Ame Elliott, Jennifer English, Rashmi Sinha, Kirsten Swearingen, and Ka-Ping Yee. Finding the flow in web site search. *Commun. ACM*, 45(9):42–49, 2002.
7. Marti A. Hearst. Uis for faceted navigation: Recent advances and remaining open problems. In *HCIR08 Second Workshop on Human-Computer Interaction and Information Retrieval*. Microsoft, October 2008.
8. Junichi Kazama and Kentaro Torisawa. Exploiting wikipedia as external knowledge for named entity recognition. In *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 2007.
9. Graham Klyne and Jeremy Carroll. Resource description framework (rdf): Concepts and abstract syntax - w3c recommendation. <http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/>, 2004.
10. Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.
11. Metaweb Technologies. Freebase wikipedia extraction (wex). <http://download.freebase.com/wex/>, 2009.
12. Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: A large ontology from wikipedia and wordnet. *Journal of Web Semantics*, 6(3):203–217, 2008.
13. Fei Wu and Daniel Weld. Automatically Refining the Wikipedia Infobox Ontology. In *Proceedings of the 17th World Wide Web Conference*, 2008.
14. Ori B. Yitzhak, Nadav Golbandi, Nadav Har’el, Ronny Lempel, Andreas Neumann, Shila O. Koifman, Dafna Sheinwald, Eugene Shekita, Benjamin Sznajder, and Sivan Yogev. Beyond basic faceted search. In *WSDM ’08: Proceedings of the international conference on Web search and web data mining*, pages 33–44, New York, NY, USA, 2008. ACM.
15. Hugo Zaragoza, Henning Rode, Peter Mika, Jordi Atserias, Massimiliano Ciaramita, and Giuseppe Attardi. Ranking very many typed entities on wikipedia. In *CIKM ’07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 1015–1018, New York, NY, USA, 2007. ACM.