Investigating the Semantic Gap through Query Log Analysis

Peter Mika[1], Edgar Meij[2], and Hugo Zaragoza[1]

[1] Yahoo Research
Diagonal 177, 08018 Barcelona, Spain
{pmika,hugoz@yahoo-inc.com}
[2] ISLA, University of Amsterdam
Sciencepark 107, 1098 XG Amsterdam
{edgar.meij@uva.nl}

Abstract. Significant efforts have focused in the past years on bringing large amounts of metadata online and the success of these efforts can be seen by the impressive number of web sites exposing data in RDFa or RDF/XML. However, little is known about the extent to which this data fits the needs of ordinary web users with everyday information needs. In this paper we study what we perceive as the semantic gap between the supply of data on the Semantic Web and the needs of web users as expressed in the queries submitted to a major Web search engine. We perform our analysis on both the level of instances and ontologies. First, we first look at how much data is actually relevant to Web queries and what kind of data is it. Second, we provide a generic method to extract the attributes that Web users are searching for regarding particular classes of entities. This method allows to contrast class definitions found in Semantic Web vocabularies with the attributes of objects that users are interested in. Our findings are crucial to measuring the potential of semantic search, but also speak to the state of the Semantic Web in general.

1 Introduction

Semantic search is by its broadest definition a collection of approaches that aim at matching the Web's content with the information need of Web users at a semantic level. Most of the work in this area has focused on the *supply-side* of semantic search, in particular elevating Web content to the semantic level by relying on methods of information extraction [4] or working with explicit metadata embedded inside or linked to Web resources. With respect to explicit metadata, several studies have been done on the adoption of Semantic Web formats in the wild, mostly based on statistics from the crawls of Semantic Web search engines [8, 7, 6, 14, 10]. Much less effort has focused on the *demand-side* of semantic search, i.e. interpreting queries at the semantic level and studying the information needs of web users in terms of semantic categories. Conversely, little is known as to how much the supply of metadata actually matches the

demand for information from ordinary web users, i.e. how large is the semantic gap between supply and demand on the Semantic Web. This question is central to the success of semantic search, but also to the success of the public Semantic Web in general.

In this paper, we divide our analysis in two parts and provide methods and tools for studying the semantic gap at both the level of instance data and vocabularies[3]. Section 3 covers our analysis of metadata on the Semantic Web. The question we seek to answer is to what extent data on the Semantic Web matches the information needs of the average Web search users as evidenced by search sessions sampled from the query log of a Web search engine. In addition, we look at how information extracted from individual sites with significant influence on the Web could be effective in filling in missing data. Last, we also investigate the particular categories of queries for which there is already metadata on the Web. These questions are pertinent because the success of semantic search hinges on the availability of data that covers user needs.

In Section 4, we address the problem of studying the information need of Web searchers at an ontological level, i.e., in terms of the particular attributes of objects they are interested in. We describe a set of methods for extracting the context words appearing in queries next to the instances of certain classes of objects. We implement these methods in an interactive tool called the Semantic Search Assist. The original purpose of this tool was to generate type-based query suggestions when there is not enough statistical evidence for entity-based query suggestions. However, from an ontology engineering perspective, this tool answers the question of what attributes a class of objects would have if the ontology for it was engineered purely based on the information needs of Web search users. As such it allows us to reflect on the gap between the properties defined in Semantic Web vocabularies and the attributes of objects that people are searching for on the Web. We evaluate our tool by measuring its predictive power on the query log itself.

Our main contribution is thus the usage-based perspective we take on analyzing metadata and vocabularies. We provide a set of methods and their implementation in tools for measuring the Semantic Web from this perspective. The results we provide can be independently validated and we plan to publish some of the detailed analysis in an online form. We conclude and summarize future work in Section 5.

2 Related Work

This work lies at the intersection of two separate streams of research on analyzing Semantic Web data and understanding user queries at a semantic level. In the first area, a number of studies have been done based on the crawls of Semantic Web search engines [8, 7, 6, 14, 10], although these studies have focused on data

[3] In the following, we will use the term vocabulary instead of the term ontology when we want to put the emphasis on the surface forms of ontological elements. Otherwise we will use the two terms interchangeably.

quality based on principles such as ontology reuse and interlinking, irrespective of particular applications of the data. These studies also have not touched upon embedded metadata (RDFa or microformat data), which are likely to have different characteristics, especially when it comes to user-generated content.

Analyzing query logs as a source of semantics bears many resemblances to mining semantics from folksonomies. Some of the related work use methods of networks analysis and unsupervised methods of data mining such as frequent itemset mining and hierarchical clustering among others. [13, 11, 15]. Krause et al. perform a network analysis of a 'logsonomy' which emerges by looking at queries as tags of clicked URLs and conclude that folksonomies and logsonomies share similar characteristics [12]. Francisco et al. generate a similar network and carry out clustering to mine semantically related queries [9]. Our analysis is different in that we are decomposing queries into entities and their context, and we use background knowledge in the form of entity to type mappings to associate queries.

3 The Data Gap

Discussions around the growth and adoption of the Semantic Web often revolve around the observable size of the Semantic Web, whether it's the number and size of datasets in the Linked Data cloud[4] or the number of pages annotated with microformats or using RDFa. As an example of this sort of analysis, Figure 1 shows the percentage of pages with certain types of microformats or RDFa data as observed in September, 2008 and March, 2009. We can read the growth rates from the chart, and observe, for example, that roughly 2% of webpages contained hCard data by the end of the observed period. But we have to ask ourselves the questions: how useful is this analysis? Just how big the Semantic Web should be?

One possible answer is that the Semantic Web should be *just big enough* to answer all the questions that we may want to ask.[5] The questions that we may want to ask to the data may depend on the particular application but considering semantic search on the public web, one valuable source of information are the logs collected by Web search engines. Due to the widespread, everyday use of search engines query logs provide an excellent record of the information needs of the collective of Web users.

Given a set of queries, the effectiveness of search still depends on the corpus as well as the search engine. Since our goal is not to evaluate semantic search engines, but to evaluate data, we fix the search engine in question by relying on Yahoo Search to retrieve web pages and look at the metadata associated with the results returned. Thus we assume that the current text search engine is a good approximation for a semantic search engine. We believe this is a reasonable

[4] see `http://linkeddata.org`

[5] There is ample evidence that the Web is bigger than just enough: the three largest search engines crawl, index and query different parts of the Web and yet come up with qualitatively similar answers.

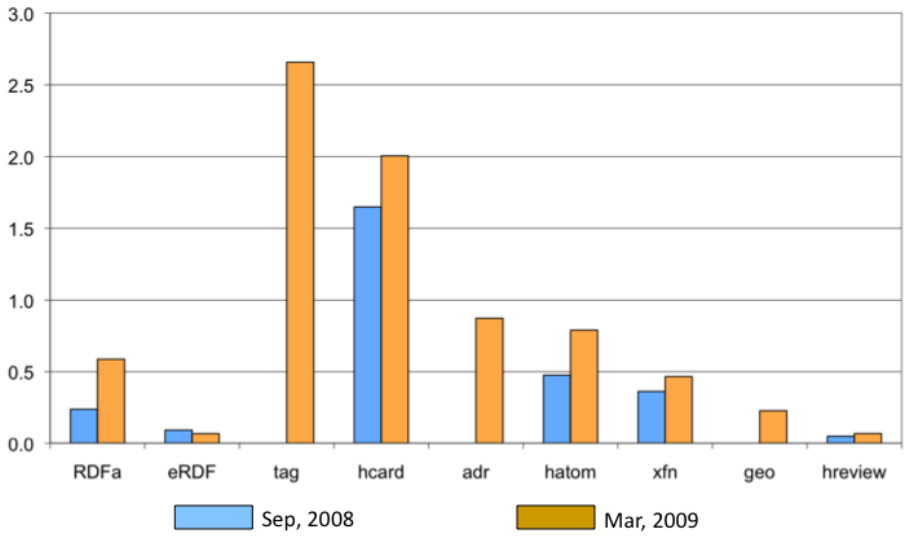

Fig. 1. Percentage of URLs with eRDF, RDFa data and certain popular microformats.

assumption for embedded metadata (microformats, RDFa) where the metadata is typically a structured representation of the main object presented in the page.[6] We also fix the corpus in the sense that we are naturally limited to the part of the Semantic Web that is crawled and indexed by Yahoo, which includes various microformats and RDFa data, but doesn't include Linked Data in general (e.g. RDF/XML documents).

In summary, we are interested in the forms and quantities of metadata that are returned with search results based on the behavior of the average Web user.

3.1 Methodology

For our analysis, we have taken a uniform sample of search sessions appearing in Yahoo's US query logs for the month of January, 2009.[7] This sample contained 10699 queries, with 7081 unique queries with at least one result. (We obtain the list of unique queries by taking each query once, no matter how often it occurred in the query log.) The distribution of queries follows the typical 'very long tailed'

[6] However, we do not know of any studies that would have verified this commonly held assumption.

[7] The difference between sampling sessions and sampling queries directly is negligible for our analysis.

distribution observed in query logs [2]: 64% of the unique queries appear only once in the sample.[8] This is mostly a result of the fact that the same query can be written in multiple ways. Again, we rely on the search engine to return comparable results for equivalent queries.

We executed the queries in the log using the Yahoo BOSS search API, which has been recently extended to return embedded metadata with each search result.[9] The metadata is returned either as RDF/XML or in DataRSS format, where the RDF triples are grouped into 'adjuncts' based on the source of the metadata, i.e. RDFa or one of the recognized microformats.[10] DataRSS is a proprietary serialization format, but one that is fully compatible with RDFa, which means that the actual RDF triples can be extracted with any RDFa parser.

format	1	2	3	4	5	6	7	8	9	10	TI	ATI
hcard	1457	370	93	11	3	0	0	0	0	0	2535	0.36
rel-tag	1317	350	95	44	14	8	6	3	1	1	2681	0.38
adr	456	77	21	6	1	0	0	0	0	0	702	0.10
hatom	450	52	8	1	0	0	0	0	0	0	582	0.08
license	359	21	1	1	0	0	0	0	0	0	408	0.06
xfn	339	26	1	1	0	0	0	1	0	0	406	0.06
RDFa	176	2	0	0	0	0	0	0	0	0	180	0.03
Any	2127	1164	492	244	85	24	10	5	3	1	7623	1.08

Table 1. The number of queries that return 1 to 10 results with metadata in particular formats, plus the total impressions for the entire set of queries and the average total impressions per query.

Using the adjunct ids returned for each query we can count the number of results with embedded metadata and the source of the information. The rows of Table 1 show the results for each format (RDFa or microformat) separately and for considering any format (Any). Each row shows the number of queries with 1 to 10 results with embedded metadata, the total impressions (TI) that is the total number of returned URLs that contained metadata, and the average total impressions (ATI), which is the average number of impressions per query.

As an example on how to read this table, the number 370 in the row labeled 'hcard' and the column labeled '2' shows that 370 queries returned two results with hCard data. In the same row, the column TI shows that 2535 of the returned results for all the queries contained hCard data, which makes an average of 0.36 results with hCard data per query. Note that the last row is not a total because

[8] In other words, this is the percentage of queries on the list of unique queries that appeared only once in the original sample. This is different from the ratio of unique queries, i.e. the percentage of queries that occur only once when counting with multiplicity.

[9] see `http://developer.yahoo.com/search/boss/structureddata.html`

[10] Yahoo Search converts microformats to RDF during indexing.

it's not a simple sum of the rows above: a single page may contain multiple types of microformats, or a combination of microformats and RDFa.

Based on the results, we observe that 59% of the queries have at least one search result with metadata, with an average of about one search result with metadata. (Note that taking ten search results for each query, the ATI has a maximum value of 10.) hCard and rel-tag each appear on every third search result page on average, while other microformats appear a lot less frequently (the numbers in the last column are decreasing quickly). An RDFa enabled result would appear only for every 40th query at the time of the analysis (March, 2009).[11]

3.2 The role of popular sites

It is a well-known phenomenon in Web search that the size of a web site doesn't necessarily correlate with its usefulness as determined by users. On the one hand, a web site doesn't have to be large to be popular with users: a well-known example is Wikipedia, which contains relatively a small amount, but diverse and high quality content, and as a result dominates search result pages beyond its size. At the other extreme, a large part of Web pages that are crawled are never returned by search engines. One can say that these pages are useful to search engine users only to the extent that they are linked or otherwise findable from pages that are being returned.

We are interested in measuring the extent to which large sites dominate search results, and consequently the importance of the data they provide compared to the numerous but smaller contributions of average websites. To achieve this, we counted unique host names in search results exactly as we counted unique formats appearing. Table 2 shows the results in the same format as the previous table. Note that as a general rule Yahoo does not return more than two results from the same host except when the query is a URL or site query.

These results are illuminating in the sense that they show a surprisingly large influence of some websites. For example, if YouTube would introduce an entirely new microformat or one would extract information from this particular Web site, from the perspective of search users this data alone would be more significant than the total amount of XFN information on the Web contributed by millions of hosts. We also see that the most of the importance we can attribute to RDFa data comes from the adoption of RDFa by a single large site, myspace.com. We expect the relative importance of large sites to diminish over time, but it seems characteristic for the current early adoption phase of the Semantic Web.

[11] Note that we don't count as RDFa triples in the XHTML namespace such as those generated by <link> elements with a rel attribute of icon or stylesheet. We choose to ignore these triples because they have no value for a semantic search engine. The only frequent-enough property in the XHTML namespace that does have a semantic value is xhtml:license, which we account for under rel-license.

host name	1	2	3	4	5	6	7	8	9	10	TI	ATI
en.wikipedia.org	1676	1	0	0	0	0	0	0	1	0	1687	0.24
www.youtube.com	475	1	0	0	0	0	0	2	0	0	493	0.07
www.amazon.com	345	3	0	0	0	0	1	0	0	0	358	0.05
www.answers.com	294	0	0	0	0	0	0	0	0	0	294	0.04
www.geocities.com	263	2	0	0	0	0	0	0	0	0	267	0.04
www.yellowpages.com	233	0	0	0	0	0	0	0	0	0	233	0.03
blog.360.yahoo.com	228	0	0	0	0	0	0	0	0	0	228	0.03
local.yahoo.com	220	1	0	0	0	0	0	0	0	0	222	0.03
www.imdb.com	197	0	0	0	0	0	0	0	0	0	197	0.03
www.myspace.com	163	0	0	2	0	0	0	0	0	0	171	0.02

Table 2. Most popular hostnames in search results by total impression

3.3 The influence of the query category

While query logs in general cover the breadth of information needs, we might be interested in measuring the potential of semantic search for particular categories of queries. The performance of current Web search technology in general strongly depends on the type of query (e.g. short queries vs. long queries, navigational vs. non-navigational) or domain of queries (e.g. person queries vs. product queries). Thus the potential for improvement using semantic technologies is consequently larger for certain kind of queries than others. Another reason to break down the results might be that certain kinds of queries are more important from the perspective of search advertising.

Given any classification of queries, the results of the analysis above can be easily broken down by category. The categories used to classify queries will depend on the type of application. For demonstration purposes, we show how the results break down for a small number of query categories defined by ourselves and used to categorize a set of 1000 queries.

Organization		Location		Person		Recent event	
hcard	0.40	hcard	0.7	rel-tag	0.65	hcard	0.65
rel-tag	0.35	adr	0.55	hcard	0.54	en.wikipedia.org	0.48
adr	0.23	local.yahoo.com	0.33	en.wikipedia.org	0.23	rel-tag	0.43
en.wikipedia.org	0.21	geo	0.31	hcalendar	0.16	hcalendar	0.40
geo	0.10	yelp.com	0.21	hatom	0.14	answers.com	0.15
local.yahoo.com	0.09	rel-tag	0.16	youtube.com	0.12	imdb.com	0.15
yelp.com	0.08	yellowpages.com	0.15	answers.com	0.10	myspace.com	0.15
hatom	0.08	en.wikipedia.org	0.11	facebook-video	0.07	hatom	0.12
Any	1.14	Any	1.31	Any	1.52	Any	1.66

Table 3. Average Total Impression (ATI) values for particular formats when restricting the query set by query category.

Table 3 shows the ATI values for different formats and for the top four categories that surfaced most of the metadata: queries containing organizations, location names and person names, and recency sensitive queries, i.e. queries referring to news or events. There are again a number of noteworthy observations. As stipulated, and as shown by the last row, each of these restrictions of the data set resulted in returning more metadata per query than for the general case (where the ATI measure was 1.08), i.e. there is indeed more metadata to be exploited for particular classes of queries. We can also see significant changes in the relative importance of particular sites and different types of metadata. For example, Wikipedia's importance is significantly diminished for queries containing locations, which points to the fact that Wikipedia is rather incomplete when it comes to articles about places. Metadata in Facebook Share format[12] describing videos is not relevant for queries in general, but it has a relative importance to queries related to people (in particular, celebrities). Similarly, hCalendar did not appear in Table 1 because its significance was below that of the last entry (XFN). However, hCalendar data seems very significant to queries about events.

We will return to some of the limitations of this analysis in our conclusions in Section 5. In the second part of our paper we look at a parallel problem of measuring the semantic gap between the information needs (and corresponding vocabulary) of web searchers and the information captured in ontologies.

4 The Vocabulary Gap

We begin by observing that in Web search query logs and in particular for queries that contain a named entity, the *class* of the entity that the user is looking for often determines the query context, i.e., the terms written before (prefix) or after the name (suffix) of an entity, respectively. Put differently, entities of the same class often occur in the context of similar words, representing specific information users are interested in with respect to that particular class of entities. Table 4 shows some examples of queries with class-based contexts.

Query	Entity	Context	Class
aspirin side effects	ASPIRIN	+*side effects*	Anti-inflammatory drugs
how to take ibuprofen	IBUPROFEN	–*how to take*	Anti-inflammatory drugs
britney spears video	BRITNEY SPEARS	+*video*	American film actors
britney spears shaves her head	BRITNEY SPEARS	+*shaves her head*	American film actors

Table 4. Example queries, extracted entities, completions, and types.

[12] `http://www.facebook.com/share_partners.php`

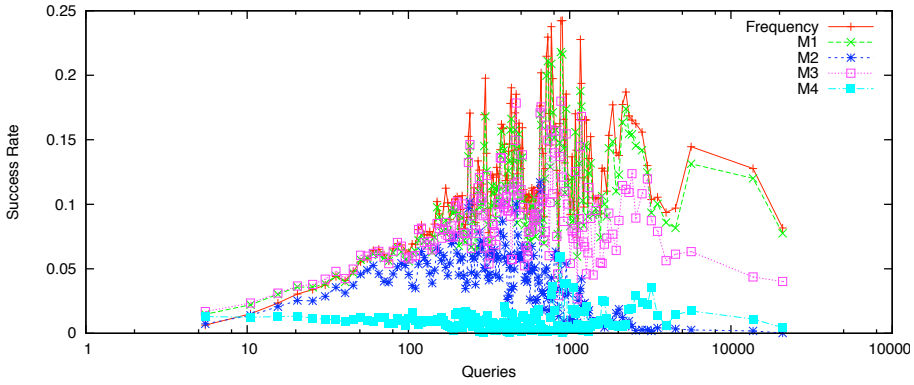

Fig. 2. Success rate for queries. The x-axis shows the queries which are averaged and binned by frequency of occurrence. On the left we see rare queries, on the right popular ones.

In this section, we look at how at a method to mine common attributes of classes of objects using query logs and class-membership information as background knowledge. The original use case of this analysis was to provide search suggestions based on the type of entity the user is looking, which is useful in situations where no good suggestions are available for the entity itself. However, the resulting structures are also interesting to compare to the explicit conceptualizations found in Web ontologies.

We start by selecting those queries from the query logs which have a named entity in them. Given this subset of the query logs, we assume queries can be decomposed in an entity part e and a context part f and, further, that entities can be assigned a type T. In case the query contains a pre- and suffix, we treat it as two separate queries.

We then determine the matrix $N = (n_{ef})_{e,f}$, where n_{ef} is the number of times we see f with e. By grouping all entities of a certain type we can, for example, compute $n_{Tf} := \sum_{e \in T} n_{ef}$ which is the number of times we see completion f with an entity of type T. Using N, we can readily estimate probabilities such as $P(f)$, $P(f|e)$, $P(f|T)$, and $P(e|f,T)$ which we use to implement several intuitions regarding semantic query completion.

4.1 Extraction Methods

Imagine that a user is typing a query and we recognise what she has typed so far as an entity with a corresponding type. The most naive approach (and the one that is taken by most Web search engines) would be to suggest the most frequent completions for the current entity (**M0**): $score_{M0}(f,e) = P(f|e)$. Given an infinite amount of data this should suffice. However, it will probably fail for rare entities since we will have none or very few completions for them. For this

reason we turn to to the entity type and smooth the entity distribution with the type distribution.

M1 aggregates completions over types and looks at the most likely completion for the current type:

$$score_{M1}(f,T) = P(f|T). \tag{1}$$

Another desirable property a completion should have, is being rare over all types. **M2** rewards such completions:

$$score_{M2}(f,T) = \frac{P(f|T)}{P(f)}. \tag{2}$$

Another intuition is that completions which are frequent as well as evenly distributed among the entities in the type should be rewarded (**M3**):

$$score_{M3}(f,T) = G(f|T) = \left(\prod_{e \in T} n_{ef}\right)^{1/|T|}. \tag{3}$$

The final method (**M4**) only considers the distribution of completions within the type:

$$score_{M4}(f,T) = H(\theta_{f|T}), \tag{4}$$

where H is the entropy of the multinomial $\theta_{f|T} = (P(e|f,T))_{e \in T}$.

To detect entities in queries, we require an ontology with a set of classes and a set of instances for each class. Consequently, we match the largest substring common to the query and the label of an instance.[13] In our experiments, we use DBpedia [4] considering both templates and categories as classes. Wikipedia entries can belong to many categories (e.g. 34 for Madonna) and reference a number of templates. We choose only one and try to select the *best* entity type. This is a challenging research question in itself; trivial methods such as choosing the most frequent or rarest type did not work well. Instead, we apply M1 on the training data and evaluate the performance of all possible types of each entity. We then choose the type that led to the best performance on the training set. For entities not present in the training set we select the type with the most entities.

4.2 Evaluation of Type-based Context Prediction

We evaluate the success of our context extraction by measuring its predictive power. In particular, we compare the highest scoring completions of the various

[13] We remove any disambiguation part in the entry title. This has the adverse effect of introducing noise, e.g. collapsing Madonna (art) and Madonna (entertainer). Disambiguating such queries is beyond the scope of the current work but could, e.g., be achieved by leveraging a user's history [3].

methods with the actual observed remainder of the queries in the test set. We use 6 consecutive days of query logs which we split equally into a training and a test set. We analyze each query and if it contains an entity we keep it. This results in 1,681,753 queries for training and 1,644,033 for testing. For each query, we compute the top $K = 10$ completions predicted by each method using postfixes only. The correct completion for that query is the one typed by the user. We are interested in two evaluation measures: (i) Success Rate @ K (SR), i.e. whether the completion is correctly predicted and (ii) Mean Reciprocal Rank @ K (MRR), i.e. the mean of the inverse of the ranks at which the completion was found, up to K.

	M0	M1	M2	M3	M4
MRR	0.081	0.068	0.014	0.046	0.006
SR	0.118	0.104	0.041	0.088	0.010

Table 5. Aggregated results over all queries.

infobox_settlement	infobox_musical_artist	drugbox	infobox_football_club
hotels	lyrics	buy	forum
map	buy	what is	news
map of	**pictures of**	tablets	website
weather	what is	what is	homepage
weather in	video	side effects of	tickets
flights to	download	hydrochloride	official website
weather	hotel	online	**badge**
hotel	dvd	overdose	fixtures
property in	mp3	capsules	free
cheap flights to	best	addiction	**logo**

Table 6. Top ten prefixes and postfixes using our model M4 and Wikipedia templates as classes.

Table 5 shows the results over all test queries. As is clear from the low absolute scores, the task of suggesting the correct completion is a difficult one. The highest obtained MRR lies around 0.18 for M0 with queries that occur around 1000 times. M0 outperforms the type-based methods on almost all queries and measures. However, as indicated by Figure 2, the type-based methods, in particular M1 and M3, perform slightly better than M0 for less frequent queries (occurring 40 times or less and making up 12.7% of the total query volume). For other queries, M0 outperforms all other methods although the difference with M1 is usually small. The reason for the lower scores at the most frequently occurring queries is that these mostly consist of entities such as “in”, “to”, and “uk” (which are actual Wikipedia entries).

In the future, we plan to complement this evaluation with a user study as we feel that some of the models might achieve a high prediction accuracy by over-fitting popular entities. There are also many query contexts that are particular to the specific entity (e.g. *britney spears shaves her head*) but a user is likely to accept other reasonable suggestions based on the type (e.g. *britney spears videos*) when offered a choice.

4.3 Qualitative Analysis

We have implemented the above methods in a tool that can be used to dynamically query for the most common context words of an entity of a certain type. Shown in Figure 3, the tool allows to search for all entities using a text box that performs autocompletion. Once the user has selected an entity, the relevant types are retrieved, and the user can chose one of the available types. Based on the selected entity and the type, the tool shows both the entity-based and type-based context words. The tool relies on a number of indices built from the query log using DBpedia as background knowledge. The tool could be used to perform the analysis for any other Semantic Web ontology by rebuilding the underlying indices.

In the following, we compare the results of context mining and the attributes found in DBpedia itself. This analysis is necessarily manual and qualitative because we would like to accept the situation where there is a semantic equivalence. For example, the users may be looking for 'pictures' while the ontology may contain a 'photo' property.

Table 6 shows the most common contexts (prefixes or suffixes) for five different Wikipedia templates, computed using method M4. We have chosen this particular model over our other models because it seems to give better type-specific results: even though our M1 has higher predictive power, at the same time it is over-fitting popular entities in the class. We have chosen these five templates because they vary in size from 43225 entities for *infobox_settlement* to 998 entities for *infobox_football_club*.

We show in **bold** the prefixes or suffixes that match an infobox property, i.e., where the user's query is likely to be satisfied by infobox data (assuming that the particular property is defined for the particular entity the user is searching for, i.e. that the infobox has been completed for this property). It is immediately obvious that there are very few of these. In fact, it seems the majority of these popular information needs cannot even be possibly satisfied by factual data. We leave it for further investigation to study whether it is the case that factual questions –which may be individually uncommon– would still make up a substantial portion of query volume.

It is interesting to note that there are also information needs where the answer could be relatively concise and expressed in a single sentence or paragraph. This is often reflected in the structure of articles, i.e. the division of information into sections. For example, articles on drugs often have sections titled 'Overdose' and 'Side Effects'. Even if the answer to a query such as *aspirin overdose* can not be answered by a single fact, the information the user is looking for may come

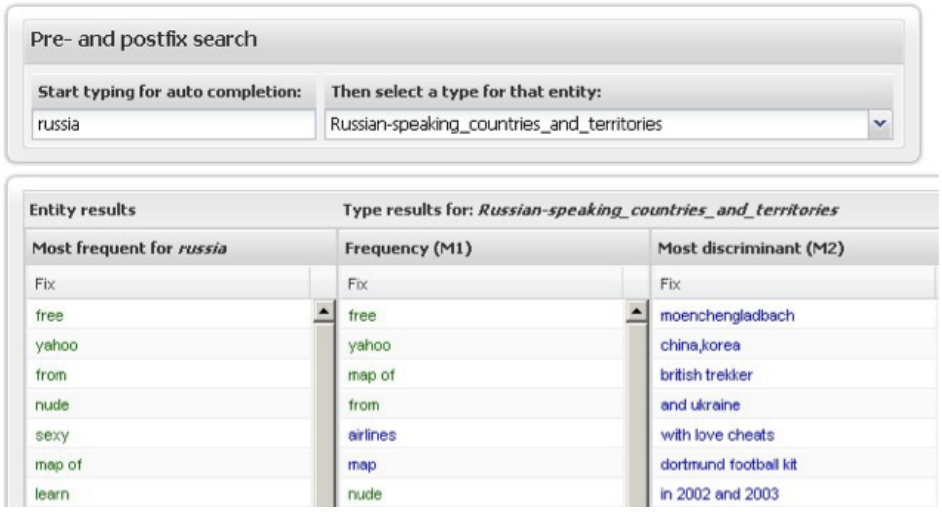

Fig. 3. Interactive search tool for the most common pre- and postfixes given an entity and a type.

from a single section or even a single paragraph within the Wikipedia article. This warrants further investigation of exploiting article structure when searching Wikipedia.

In summary, it is clear that if infobox data would be geared toward answering popular information needs as surfaced by our tool, the infoboxes would need to contain different information at different levels of granularity. This suggests that for answering these 'head' queries one may need to merge the methods of data retrieval with methods of structured retrieval and unstructured retrieval. Put differently, for using the output of our tool as an input for ontology engineering, the list of context words extracted will need to be filtered to those representing attributes of objects, i.e. properties that can be filled with simple values. Although this out of the scope for our current work, ontology learning in this context would be similar to ontology learning in folksonomies [5, 13, 1].

5 Conclusion and Future Work

Ultimately, the success of the Semantic Web depends not only on technology, but also how well the knowledge captured using microformats or Linked Data satisfies the needs of ordinary users. The two main factors in this respect are the coverage and quality of data and ontologies. In this paper, we have looked at the issue of coverage, in particular to what extent data on the Semantic Web is

potentially useful in resolving queries and how well the vocabularies used match the implicit vocabularies of users as expressed by their queries.

We have presented methods of analysis and discussed the results of our evaluation. We plan to repeat these evaluations as the evolution of the semantic gap is just as interesting as a static picture of it. We have chosen Web search as our target domain, but the general ideas represented by these methods are equally applicable to vertical, enterprise or desktop search scenarios. The particular experiments we have performed can be reproduced using the BOSS API, which provides access to Web metadata crawled by Yahoo.

In terms of measuring the relevance of Semantic Web data to Web search, we have shown how we can measure the contributions of various forms of data by effectively replaying a large number of sessions sampled from query logs. We posit that just like in the case of the HTML web where often relatively small, but popular or qualitative websites serve a large number of user needs (such as Wikipedia), the Semantic Web also looks very different when looking at it from the perspective of user queries, instead of just to gauging the number of triples in public datasets. In fact, we find that popular sites have also a lot to contribute to the Semantic Web from this perspective, possibly just as much as the long tail of web sites. Last, we found useful breaking down the analysis into query categories, since such breakdown significantly influences the results and may point to query types where the Semantic Web has a particular potential.

We have also presented a number of methods and their implementation in an online tool for mining type-based query context information, i.e. query prefixes and postfixes that are common to a class of entities, while uncommon to other entities outside of their class. Postulating that these context words represent aspects of entities that search engine users are interested in, we proceeded to investigate on the case of Wikipedia the extent to which this schema of information needs matches the schema of available structured data. We find that at least for the most common context words the overlap is very low as the most common queries are not specific enough to be answered by factual data. We suggest that our tool could be used in the future to analyze, extend or create new ontologies based on the information needs extracted from query logs. In this case it is left to the ontology developer to consider which context words signify relevant attributes of objects to be included in an ontology.

The reader may note that throughout our analysis we attribute the same value to each query and to each piece of data. There might be very good reasons to attribute different value to different queries, for example, because the queries can be monetized to different extents and URLs may have different visibility in the search result page (e.g. top three positions vs. the rest). Certain data sets or combinations of data sets may provide extraordinary value to a small number of users. For example, a biomedical database may provide significant value to a researcher in biomedicine. This is not reflected in our average-value analysis. It is part of the future work to extend our analysis to weighted query sets.

Another limitation of our analysis is that we rely on existing query methods. One might argue that semantic search engines will allow the users to express

different forms of queries (natural language queries, SPARQL queries, etc.) and the mere possibility to address information needs in a different form or the fact that semantic search engines will successfully answer new types of queries will change user behavior. Indeed, there are plenty of latent queries that users do not enter into Web search engines because they have learned they would not be answered. Often, these queries are turned into navigational queries, e.g. a user interested in flights from boston to san francisco would simply type in the name of an airline, knowing the search engine itself would not be able to return flight information directly. While such a transition toward rich, semantic queries may happen in the future, this change in user behavior will take some time. Similarly, as the Semantic Web grows and sees more usage in general, data may be more aligned with general information needs of Web users. In the meantime, semantic search engines will have to cope with the substantial gap in both data and vocabularies.

References

1. S. Angeletou, M. Sabou, and E. Motta. Folksonomy Enrichment and Search. In L. Aroyo, P. Traverso, F. Ciravegna, P. Cimiano, T. Heath, E. Hyvonen, R. Mizoguchi, E. Oren, M. Sabou, and E. P. B. Simperl, editors, *ESWC*, volume 5554 of *Lecture Notes in Computer Science*, pages 801–805. Springer, 2009.
2. R. Baeza-Yates, A. Gionis, F. Junqueira, V. Murdock, V. Plachouras, and F. Silvestri. The impact of caching on search engines. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 183–190, New York, NY, USA, 2007. ACM.
3. J. Bai and J.-Y. Nie. Adapting information retrieval to query contexts. *IPM*, 44(6):1901–1922, 2008.
4. C. Bizer. DBPedia: Querying Wikipedia Like a Database. In *WWW '07*, 2007.
5. P. Brusilovsky and H. C. Davis, editors. *HYPERTEXT 2008, Proceedings of the 19th ACM Conference on Hypertext and Hypermedia, Pittsburgh, PA, USA, June 19-21, 2008*. ACM, 2008.
6. M. d'Aquin, C. Baldassarre, L. Gridinoc, S. Angeletou, M. Sabou, and E. Motta. Characterizing Knowledge on the Semantic Web with Watson. In *EON*, 2007.
7. L. Ding and T. Finin. Characterizing the Semantic Web on the Web. In *ISWC '06*, 2006.
8. L. Ding, L. Zhou, T. Finin, and A. Joshi. How the Semantic Web is Being Used: An Analysis of FOAF Documents. In *HICSS '05*, 2005.
9. A. P. Francisco, R. A. Baeza-Yates, and A. L. Oliveira. Clique Analysis of Query Log Graphs. In A. Amir, A. Turpin, and A. Moffat, editors, *SPIRE*, volume 5280 of *Lecture Notes in Computer Science*, pages 188–199. Springer, 2008.
10. M. Hausenblas, W. Halb, Y. Raimond, and T. Heath. What is the Size of the Semantic Web? In *I-Semantics '08*, Graz, Austria, 2008.
11. R. Jäschke, A. Hotho, C. Schmitz, B. Ganter, and G. Stumme. Discovering shared conceptualizations in folksonomies. *J. Web Sem.*, 6(1):38–53, 2008.
12. B. Krause, R. Jäschke, A. Hotho, and G. Stumme. Logsonomy - social information retrieval with logdata. In Brusilovsky and Davis [5], pages 157–166.
13. P. Mika. Ontologies Are Us: A Unified Model of Social Networks and Semantics. In Y. Gil, E. Motta, V. R. Benjamins, and M. A. Musen, editors, *International*

Semantic Web Conference, volume 3729 of *Lecture Notes in Computer Science*, pages 522–536. Springer, 2005.
14. G. Tummarello, R. Delbru, and E. Oren. Sindice.com: Weaving the Open Linked Data. *The Semantic Web*, pages 552–565, 2008.
15. M. Zhou, S. Bao, X. Wu, and Y. Yu. An Unsupervised Model for Exploring Hierarchical Semantics from Social Annotations. In K. Aberer, K.-S. Choi, N. F. Noy, D. Allemang, K.-I. Lee, L. J. B. Nixon, J. Golbeck, P. Mika, D. Maynard, R. Mizoguchi, G. Schreiber, and P. Cudré-Mauroux, editors, *ISWC/ASWC*, volume 4825 of *Lecture Notes in Computer Science*, pages 680–693. Springer, 2007.