MidTerm Report

Self-Service Data Provisioning Through Semantic Enrichment of Data

Ahmad Assaf

EURECOM-Multimedia Communications
Institut Mines-Télécom

April 21st, 2014

Supervisors:	
Raphaël Troncy	**EURECOM**
Aline Senart	**SAP**

1. Introduction

Enterprises use a wide range of heterogeneous information systems in their business activities such as Enterprise Resource Planning (ERP), Customer Relationships Management (CRM) and Supply Chain Management (SCM) systems. An enterprise distributed IT landscape contains multiple systems using different technologies and data standards [1]. In addition to this heterogeneity, the amount of information in enterprise databases and on-line data stores expands exponentially each year. Enterprise Big Data isn't big in volume only, but in the associated file formats. The information is also often stored often in unstructured and unknown formats.

Data integration is the problem of combining data residing at different sources, and providing the user with a unified view of these data [2]. In large enterprises, it is a time and resource costly task. Various approaches have been introduced to solve this integration challenge. These approaches were primarily based on XML as the data representation syntax, Web Services to provide the data exchange protocols and Service Oriented Architecture (SOA) as a holistic approach for distributed systems architecture and communication [3][4]. However, it was found that these technologies are no sufficient to solve the integration problems in large enterprises. Recently, ontology-based data integration approaches have been suggested where ontologies are used to describe the data, queries and mappings between them [5]. A slightly different approach is the use of the Linked Data paradigm [6] for integrating enterprise data. Enterprises like Google and Microsoft are not only using the Linked Data integration paradigm for their information systems, but are also aiming at building enterprise knowledge bases (like the Google Freebase[1]) that will act as a crystallization point for their structured data.

Linked Open Data (LOD) movement has gained lots of momentum in the last years. From 12 datasets cataloged in 2007, the Linked Open Data has grown to almost 300 datasets containing almost 32 billion triples [7]. Data is being published by both public and private sectors and covers a diverse set of domains from life sciences to military. This success lies in the cooperation between data publishers and consumers. Users are empowered to find, share and combine information in their applications easily.

Despite the legal issues surrounding Linked Data licenses [8], it is still considered a gold mine for organizations who are trying to leverage external data sources in order to produce more informed business decisions [9]. In [10] the authors see the potential economic effect unfolding in education, transportation, consumer products, electricity, oil and gas, health care and consumer finance. They estimate the potential annual value enabled by Open Data in these domains to be 3 trillion US Dollars across seven domains.

Data becomes more useful when it is open, widely available and in shareable formats, and when advanced computing and analysis can yield from it. The quality and amount of structured knowledge available make it now feasible for companies to mine this huge amount of public data and integrate it in their next-generation enterprise information management systems. Analyzing this new type of data within the context of existing enterprise data should bring them new or more accurate business insights and allow better recognition of sales and market opportunities [11].

Business Intelligence has always been about creating new insight for business by converting data into meaning that can be shared between people to drive change in the organization. One key aspect of creating meaning is driving a common shared understanding of information also known as Semantics.

Classic BI and even the newer Agile Visualization tools focus much of their selling features on attractive and unique visualizations, but preparing data for those visualizations still remains the far more challenging task in most BI projects large and small. self-service data provisioning aims at tackling this problem by providing intuitive datasets discovery, acquisition and integration techniques intuitively to the end user.

[1] http://freebase.com

2. Challenges

In this thesis, we aim at creating a framework that leverages Semantic Web technologies in order to enrich enterprise data in general and Business Intelligence data in particular in order to facilitate self-service data provisioning. We investigate the following research challenges:

- **Dataset Integration and Enrichment:** The enterprise heterogeneous data sources raise tremendous challenges. They have inherently different file formats, access protocols or query languages. They possess their own data model with different ways of representing and storing the data. Data across these sources may be noisy (e.g. duplicate or inconsistent), uncertain or be semantically similar yet different [12]. Integration and provision of a unified view for these heterogeneous and complex data structures therefore require powerful tools to map and organize the data.
    - ∗ Attaching metadata and Semantic information to instances can be tricky. An entity is usually not associated with a single generic type in the knowledge base, but rather with a set of specific types which can be relevant or not given the context. The challenging task is finding the most relevant entity type within a given context.
    - ∗ Entities play a key role in knowledge bases in general and in the Web of Data in particular. Entities are generally described with a lot of properties, this is the case for DBpedia. It is, however, difficult to assess which ones are more "important" than others for particular tasks such data augmentation and visualizing the key facts of an entity.
    - ∗ Social Networks are not just gathering Internet users into groups of common interests, they are also helping people follow breaking news, contribute to online debates or learn from others. They are transforming Web usage in terms of users' initial entry point, search, browsing and purchasing behavior. Integrating information from these Social Networks can be tricky due to the vast amount of data available which makes hard to spot what is relevant in a timely manner.
- **Dataset Discovery:** Even though popular datasets like DBPedia[2] and Freebase are well known and widely used, there are other hidden useful datasets not being used. Indeed these datasets may be useful for specialized domains, however without proper registry of topics, it is difficult for users to find them [13].
- **Dataset Quality Control:** Linked Data consists of structured information supported by models, ontologies and vocabularies and contains query endpoints and links. This makes data quality assurance a challenge. Despite the fact that Linked Open Data quality is a trending and highly demanded topic, very few efforts are currently trying to standardize, track and formalize frameworks to issue scores or certificates that will help data consumers in their integration tasks.

3. Proposal

Linked Open Data datasets are described using either the Vocabulary of Interlinked Datasets (VOID) [14] or the Data Catalog Vocabulary (DCAT) [15]. With these standards, discovery and usage of linked datasets can be performed both effectively and efficiently. In our framework, we plan to use DCAT as the common standard for homogenizing description metadata of datasets indexed by our crawler. This choice came from the fact that the Open Data Support[3] is promoting the DCAT-AP (and consequently DCAT) as the standard for describing datasets and catalogs in Europe. In order to enable self-service data provisioning, we envisage building a framework (see Figure 1) that will be able to provide detailed DCAT descriptions for internal and external data sources. The framework is able to provide the following services:

[2] http://dbpedia.org
[3] http://opendatasupport.eu

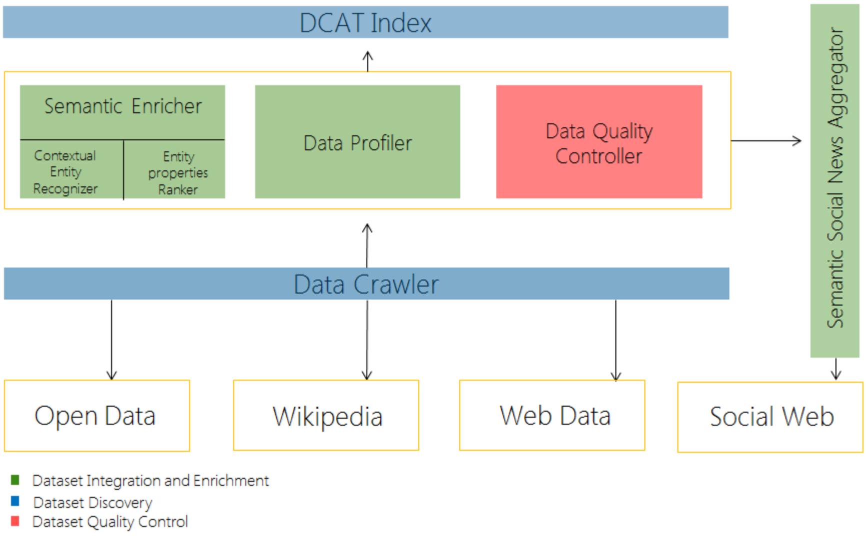

Figure 1. Overall Architecture

- **Data Acquisition**: Be able to sample data from the various structured data sources like Wikipedia tables, Open data portals, etc. While these are rich sources of information, sometimes live information streamed from the Social Networks is needed. As a result, we crawl Social Networks in order to aggregate semantically related information and connect it with the right resources.
- **Data Preparation**: This includes data profiling and validation, de-duplicating and enhancing relevant data sets with metadata. Profiling is used to examine data to understand its content, structure and data quality dependencies. The types of profiling tasks include:
  - Examining column data and getting statistical information such as min, max, average, median, null percentage, value distribution, pattern distribution.
  - Dependency tasks: Finds the values in one or more dependent columns that rely on values in a primary column
  - Redundancy tasks: Determine the degree of overlapping data values or duplication between two sets of columns
  - Uniqueness tasks: Returns the count and percentage of rows that contain non-unique data, for the set of column(s) selected.
  - Content type: Content type profiling provides suggested meaning based on the entities data in the columns.
  - Quality checks: An important aspect that we have to take into consideration while describing a dataset is its quality. For that, an objective Linked Data quality assessment framework should be created in order to issue quality profiles that extend the DCAT vocabulary. The framework helps on one hand data owners to rate the quality of their dataset and get some hints on possible improvements, and on the other hand data consumers to choose their data sources from a ranked set.
- **Dataset Classification**: Classify and organize datasets based on the input from the previous tasks.

3.1. Contributions on Dataset Integration and Enrichment

Regarding this aspect of our research, we have achieved the following tasks:

- Building RUBIX which is a framework enabling mash-up of potentially noisy enterprise and external data.

- RUBIX improves instance and schema matching by adding Semantic metadata to data at the instance level.
- RUBIX improves data integration techniques by enabling clean representation of the data regardless of the languages used, existence of abbreviations, synonyms and typos.
- Build a Social crawler that queries several Social endpoints and aggregates news based on a set of defined keywords.
- Build a common Semantic model to represent Web documents.
- Building a Semantic Social News Aggregating service (SNARC) [16] that aggregates relevant Social news with regards to a Web document.
- Reverse engineering of Google Knowledge Graph Panel in order to define the top properties of a selected entity.
- High traction from the BI organization to integrate it as a service into various offerings.
- We presented RUBIX at the First International Workshop on Open Data [17][18].
- SNARC has won the first place at the AI Mashup Challenge[4] at ESWC13.

3.2. Contributions on Dataset Quality Control

Concerning our contributions on Linked Data quality assessment, we have achieved the following tasks:

- We identified five principle classes to describe the quality of a particular linked dataset. For each class, we list the principles that are involved at all stages of the data management process.
- We have presented our Data quality principles at the Sixth IEEE International Conference on Semantic Computing [19].
- We have surveyed the landscape of Linked Data quality assessment frameworks.
- We have surveyed the landscape of Linked Data quality assessment tools.
- We have refined the five principles in [19] towards a more objective framework.
- We have evaluated the surveyed tools with regards to the suggested framework.

4. Dataset Integration and Enrichment

Tagging and attaching metadata is often seen as additional work for data publishers with few paybacks. Moreover, different data creators use different terminologies which means that the same object maybe be represented using different metadata descriptions [20]. Presenting and enterprise taxonomies requires a considerable amount of time and effort, at least in the initial creation steps [4].

4.1. What are the Important Properties of an Entity

In many knowledge bases, entities are described with numerous properties. However, not all properties have the same importance. Some properties are considered as keys for performing instance matching tasks while other properties are generally chosen for quickly providing a summary of the key facts attached to an entity. Our motivation is to provide a method enabling to select what properties should be used when depicting the summary of an entity, for example when augmenting extra columns into an existing dataset or when annotating instances with semantic tags.

Our approach consists in: (i) reverse engineering the Google Knowledge Panel by extracting the properties that Google considers as sufficiently important to show (Section 4.1.1), and (ii) analyzing users' preferences by conducting a user survey and comparing the results (Section 4.1.2).

[4] http://aimashup.org/aimashup13

We have shown that it is possible to reveal what are the "important" properties of entities in a large knowledge base by reverse engineering the choices made by Google when creating knowledge graph panels and by comparing with users preferences obtained from a user survey. Our motivation is to represent this choice explicitly, using the Fresnel vocabulary, so that any application could just read this configuration file for deciding which properties of an entity is worth to visualize. We are aware that this knowledge is highly dynamic, the Google Knowledge Graph panel differing from countries and varying along the time. We have provided the code that enables to perform new calculation at run time and we aim to study the temporal evolution of what are important properties on a longer period. This knowledge which has been captured will be made available shortly in a SPARQL endpoint. We are also investigating the use of Mechanical Turk to perform a larger survey for the complete set of DBpedia classes.

4.1.1. Reverse Engineering the Google KG Panel

Web scraping is a technique for extracting data from Web pages. We aim at capturing the properties depicted in the Google Knowledge Panel (GKP) that are injected in search result pages [21]. We have developed a Node.js application that queries all DBpedia concepts that have at least one instance which is `owl:sameAs` with a Freebase resource in order to increase the probability that the search engine result page (SERP) for this resource will contain a GKP. Moreover, we filter out generic concepts by excluding those who are direct subclasses of `owl:Thing` since they will trigger ambiguous queries. We obtained a list of 352 concepts[5].

For each of these concepts, we retrieve n instances. In our experiment, n was equal to 100 random instances. For each of these instances, we issue a search query to Google containing the instance label. Google does not serve the GKP for all user agents and we had to mimic a browser behavior by setting the $User - Agent$ to a particular browser. We use CSS selectors to check the existence of and to extract data from a GKP. An exemple of a query selector is $.\_om$ (all elements with class name $\_om$) which returns the property DOM element(s) for the concept described in the GKP. From our experiments, we found out that we do not always get a GKP in a SERP. If this happens, we try to disambiguate the instance by issuing a new query with the concept type attached. However, if no GKP was found again, we capture that for manual inspection later on. Listing 1 gives the high level algorithm for extracting the GKP. The full implementation can be found at `https://github.com/ahmadassaf/KBE`.

Algorithm 1 Google Knowledge Panel reverse engineering Algorithm

```
INITIALIZE $equivalentClasses(DBpedia, Freebase)$ AS $vectorClasses$
Upload $vectorClasses$ for querying processing
Set $n$ AS number-of-instances-to-query
for each $conceptType \in vectorClasses$ do
    SELECT $n$ instances
    $listInstances \leftarrow$ SELECT-SPARQL($conceptType$, $n$)
    for each $instance \in listInstances$ do
        CALL http://www.google.com/search?q=$instance$
        if $knowledgePanel$ exists then
            SCRAP GOOGLE KNOWLEDGE PANEL
        else
            CALL http://www.google.com/search?q=$instance + conceptType$
            SCRAP GOOGLE KNOWLEDGE PANEL
        end if
        $gkpProperties \leftarrow$ GetData(DOM, EXIST(GKP))
    end for
    COMPUTE ocurrences for each $prop \in gkpProperties$
end for
return $gkpProperties$
```

4.1.2. Evaluation

We conducted a user survey in order to compare what users think should be the important properties to display for a particular entity and what the GKP shows.

[5] SPARQL query: `http://goo.gl/EYuGm1`

4.1.2.1 User survey

We set up a survey[6] on February 25th, 2014 and for three weeks in order to collect the preferences of users in term of the properties they would like to be shown for a particular entity. We select one representative entity for nine classes: `TennisPlayer`, `Museum`, `Politician`, `Company`, `Country`, `City`, `Film`, `SoccerClub` and `Book`. 152 participants have provided answers, 72% from academia, 20% coming from the industry and 8% having not declared their affiliation. 94% of the respondents have heard about the Semantic Web while 35% were not familiar with specific visualization tools. The detailed results[7] show the ranking of the top properties for each entity. We only keep the properties having received at least 10% votes for comparing with the properties depicted in a KGP. Hence, users do not seem to be interested in the `INSEE code` identifying a French city while they expect to see the `population` or the `points of interest` of this city.

4.1.2.2 Comparison with the Knowledge Graphs.

The results of the Google Knowledge Panel (GKP) extraction[8] clearly show a long tail distribution of the properties depicted by Google, with a top N properties (N being 4, 5 or 6 depending on the entity) counting for 98% of the properties shown for this type. We compare those properties with the ones revealed by the user study. Table 1 shows the agreement between the users and the choices made by Google in the GKP for the 9 classes. The highest agreement concerns the type `Museum` (66.97%) while the lowest one is for the `TennisPlayer` (20%) concept.

Classes	TennisPlayer	Museum	Politician	Company	Country	City	Film	SoccerClub	Book
Agr.	20%	66.97%	50%	40%	60%	60%	60%	50%	60%

Table 1

Agreement on properties between the users and the Knowledge Graph Panel

With this set of 9 concepts, we are covering 301,189 DBpedia entities that have an existence in Freebase, and for each of them, we can now empirically define the most important properties when there is an agreement between one of the biggest knowledge base (Google) and users preferences.

4.1.2.3 Modeling the preferred properties with Fresnel.

Fresnel[9] is a presentation vocabulary for displaying RDF data. It specifies *what* information contained in an RDF graph should be presented with the core concept `fresnel:Lens` [22]. We use the Fresnel and PROV-O ontologies[10] to explicitly represent what properties should be depicted when displaying an entity.

```
:tennisPlayerGKPDefaultLens rdf:type fresnel:Lens ;
fresnel:purpose fresnel:defaultLens ;
fresnel:classLensDomain dbpedia-owl:TennisPlayer ;
fresnel:group :tennisPlayerGroup ;
fresnel:showProperties (dbpedia-owl:abstract dbpedia-owl:birthDate
 dbpedia-owl:birthPlace dbpprop:height dbpprop:weight
 dbpprop:turnedpro dbpprop:siblings) ;
prov:wasDerivedFrom
  <http://www.google.com/insidesearch/features/search/knowledge.html> .
```

[6] The survey is at `http://eSurv.org?u=entityviz`

[7] `https://github.com/ahmadassaf/KBE/blob/master/results/agreement-gkp-users.xls`

[8] `https://github.com/ahmadassaf/KBE/blob/master/results/survey.json`

[9] `http://www.w3.org/2005/04/fresnel-info/`

[10] `http://www.w3.org/TR/prov-o/`

4.2. RUBIX to Enhance Schema Matching

RUBIX is our approach to bootstrap the process of attaching meta information to data objects. We leverage DBpedia and Freebase as knowledge bases for our annotation process. In RUBIX we assign a vector of Semantic types for every object at the instance level. For example, Orange will be represented by a vector of rich types that contain `(Organization, Organism Classification, Place ...)`. Currently, we rely on Freebase API to identify these rich types, but we have already started the effort to build an entity type ranking tool inspired by [23].

4.3. RUBIX to Enhance Schema Matching

In the past, some work has tried to improve existing data schema [24] but literature mainly covers automatic or semi-automatic labeling of anonymous data sets through Web extraction. Examples include [25] that automatically labels news articles with a tree structure analysis or [26] that defines heuristics based on distance and alignment of a data value and its label. These approaches are however restricting label candidates to Web content from which the data was extracted. [27] goes a step further by launching speculative queries to standard Web search engines to enlarge the set of potential candidate labels. More recently, [28] applies machine learning techniques to respectively annotate table rows as entities, columns as their types and pairs of columns as relationships, referring to the YAGO ontology. The work presented aims however at leveraging such annotations to assist semantic search queries construction and not at improving schema matching. With the emergence of the Semantic Web, new work in the area has tried to exploit Linked Data repositories. The authors of [29] present techniques to automatically infer a semantic model on tabular data by getting top candidates from Wikitology [30] and classifying them with the Google page ranking algorithm. Since the authors' goal is to export the resulting table data as Linked Data and not to improve schema matching, some columns can be labeled incorrectly, and acronyms and languages are not well handled [29]. In the Helix project [31], a tagging mechanism is used to add semantic information on tabular data. A sample of instances values for each column is taken and a set of tags with scores are gathered from online sources such as Freebase. Tags are then correlated to infer annotations for the column. The mechanism is quite similar to ours but the resulting tags for the column are independent of the existing column name and sampling might not always provide a representative population of the instance values.

In RUBIX we have implemented several matching algorithms (Cosine Similarity, Pearson Product-Moment Correlation Coefficient (PPMCC) and Spearman's Rank Correlation Coefficient) in order to calculate similarity between data based on their rich types population. Our preliminary evaluation shows that for datasets where mappings were relevant yet not proposed, our framework provides higher quality matching results. Additionally, the number of matches discovered is increased when Linked Data is used in most datasets.

4.4. Dataset Annotation and Domain Identification

The increasing diversity of the datasets makes it difficult to annotate them with a fixed number of pre-defined tags. Moreover, manually entered tags are subjective and may not capture the essence and breadth of the dataset [13]. RUBIX can be used to Annotate datasets with meta information based on examining the data instances themselves. In addition to that, we can enhance our approach by applying techniques similar to [13] in order to identify topical domains with fine-grained classification.

4.5. Semantic Social News Aggregation (SNARC)

The Internet has created a paradigm shift in how we consume and disseminate information. Data nowadays is spread over heterogeneous silos of archived and live data. People willingly share data on social media by posting news, views, presentations, pictures and videos. SNARC is a service that uses semantic web technology and combines services available on the web to aggregate social news. SNARC brings live and archived information to the user that is directly related to his active page. The key advantage is an instantaneous access to complementary information without the need to dig for it. Information appears when it is relevant enabling the user to focus on what is really important.

Crawling data from these heterogeneous platforms implies the studying of related API specifications which differ in terms of use, privacy policy and described methods. Harvesting content spread over multiple platforms is a challenging task that has to ensure an easy and flexible way for integrating several social Web APIs. Tools such as API Blender [32] or Media Finder [33] provide such interface with the aim to save developers' efforts for learning each API specification. However, for in SNARC we needed services that are not available in the mentioned services. As a result, we have implemented our own Social crawler for Twitter[11], Google+[12], YouTube[13], Vimeo[14], Slideshare[15], Stack Exchange [16]and the Web.

4.5.1. Underlying Mechanism

The back-end of SNARC consists of three major components: a document handler that creates a S̈emantic Modelẗhat represents any web resource, a query layer that is responsible for disseminating queries to the supported social services and a data parser which processes the search results, wraps them in a common social model and generates the desired output.

4.5.1.1 Document Handler

The main idea behind SNARC is to provide a uniform model for web entities, whether they are blog entries, multimedia objects or micro-posts. To do so, SNARC creates a S̈emantic Modelc̈ontaining all the annotations and meta-data needed to query and reconcile social results.

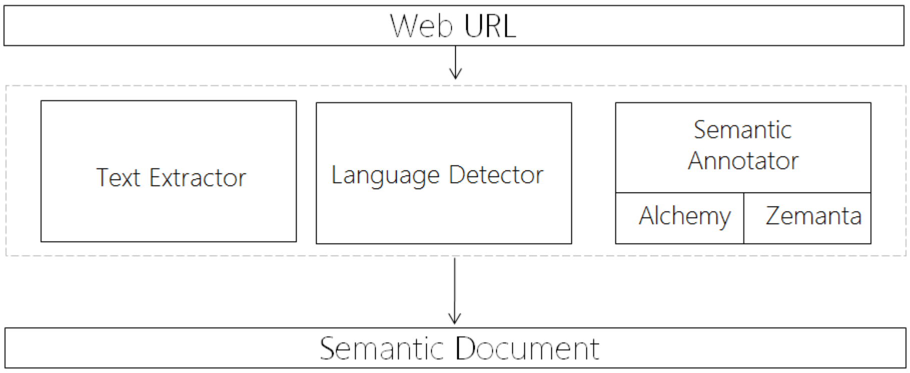

Figure 2. SNARC's Document Handler

The Semantic Model is created by the Document Handler (see Figure 1) which receives a web page URL and performs these three main steps:

[11] http://www.twitter.com
[12] http://plus.google.com
[13] http://www.youtube.com
[14] http://www.vimeo.com
[15] http://www.slideshare.com
[16] http://stackexchange.com/

1. **Text Extraction:** Fetch the webpage that corresponds to the received URL and extract the textual content using a set of heuristics. These latter identify the main content of the page by stripping unwanted HTML tags and rank the different sections based on their semantics, class names and order. In the beginning we have used Alchemy API[17] to perform text extraction; but we have chosen to implement a simpler method ourselves which saved us an extra API call.
2. **Language Detection:** Detect the web page language using the Language Detection service of Alchemy API. This is necessary to match the desired language with compatible services like Twitter, YouTube, etc.
3. **Semantic Annotation:** Annotating the extracted text is the most important step in this process. We use Zemanta Suggest[18] and Alchemy API in order to extract:

   – **Tags:** These are the finest-grained queryable keywords that we use to retrieve the social results. From our experiments, combining tags results in better findings than using entities or concepts. However, we plan to evaluate the combination of keywords, entities and concepts in order to find the top-queryable terms that will retrieve the most relevant results on different abstraction levels. Tags retrieved from these services are ranked by confidence values calculated by their internal algorithms, these values are normalized for each service. According to our experiments we have found that Alchemy's Keywords Extraction API returns a large set of closely related keywords (i.e. Android, Android Phone, Android Tablet, ...). To construct a good query we therefore need to provide a certain level of abstraction.We perform a cleaning process on those keywords by applying the Levenshtein distance to rule out closely related keywords by disregarding those with lower confidences. We perform a similar process on the result of the union between the keywords returned by Alchemy and Zemanta to ensure a sparse keywords set.

   – **Semantic Entities:** Entities provide a higher abstraction level of the document. They are used to reconcile the social results in order to maintain relevancy with the document. Similar to the keywords extraction services, the entities retrieved are ranked and contain outbound links to the matched entity on dbpedia, Wikipedia, Freebase, etc. A union is made between the results from Alchemy and Zemanta to ensure a wider coverage of entities. When a match is found, we merge the links from the two sources to ensure that we include all the resources that can be used to augment extra information about that entity in the document.

   – **Categories:** These are high-level taxonomies that can generally describe the document's content. A taxonomy is used to narrow down our query scope when targeting services like YouTube. In our Semantic Document model we define two possible category sets, one retrieved from Alchemy's Text Categorization API[19] and the other retrieved from Zemanta Suggest API that follows the DMOZ categorization scheme[20].

At the end of this process, we will have constructed the needed elements (keywords, entities and high level categories) wrapped in our Semantic Model to be passed to the query generator. For example, a summary of the Semantic Model for a web page titled T̈urkey protests: Erdogan in f́inalẃarning[21]l̈ooks like:

1. **Categories:** Culture_Politics, Regional and Society
2. **Keywords:** Taksim Square, Protesters, Gezi Park, Mr Erdogan, Istanbul ...
3. **Entities:** Gezi Park, Recep Tayyip Erdogan, Taksim Square, Justice and Development Party (Turkey), Police of Turkey ...

[17] http://www.alchemyapi.com
[18] http://developer.zemanta.com/docs/suggest/
[19] http://www.alchemyapi.com/api/categ/categs.html
[20] http://www.dmoz.org/desc/Top
[21] http://www.bbc.co.uk/news/world-europe-22889060

4.5.1.2 Query Layer

In this component, the calls to the social services are made. SNARC uses the extracted keywords from the Semantic Document in order to construct the queries and disseminate them to the appropriate services. Figure 2 shows the different steps in order to retrieve a set of social results.

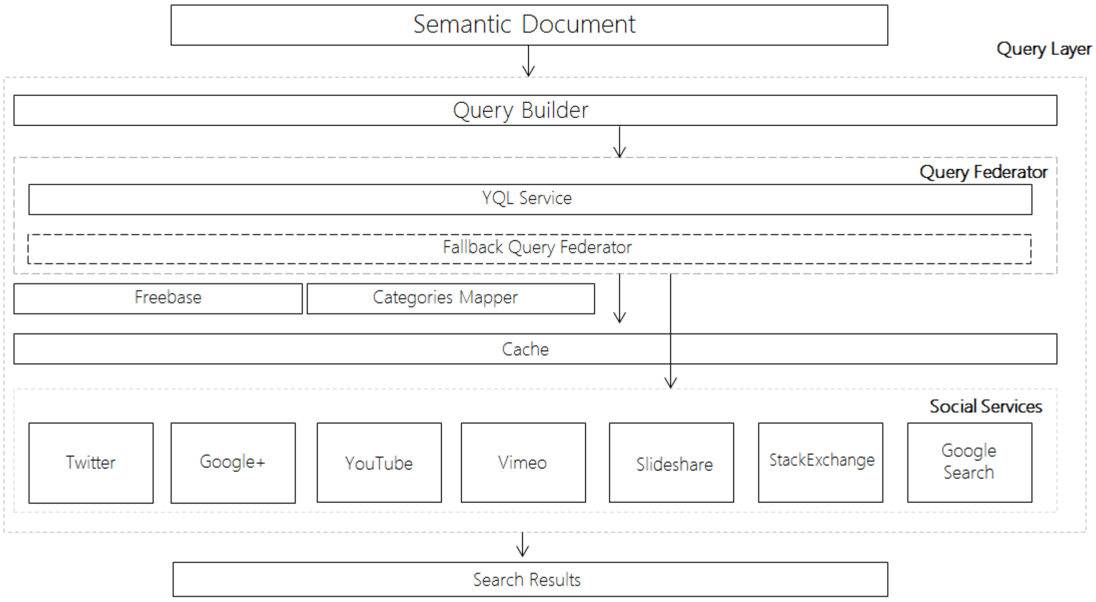

Figure 3. SNARC's Query Layer

1. **Query Builder:** Responsible for identifying targeted services and building tailored queries for each service. For example, if the processed document is categorized as a computer or technology related one, Stackoverflow service will be targeted with the queries constructed. However, other categories will correspond to different services from the Stack Exchange websites[22].
2. **Query Federator:** Responsible for federating the queries identified in the previous step to the corresponding services. To enhance performance, we tried to reduce the number of external calls. Yahoo Query Language (YQL)[23] helped us in minimizing the number of calls and batching them into a single one. It is an expressive SQL-like language that lets you query, filter, and join data across Web services. However, we have found that we cannot fully rely on YQL due to their API calls limit and the restriction on the query execution time that is set to 30 seconds. To overcome this, we have implemented a fallback mechanism that federates the queries to the selected social services and groups the result to be passed afterwards to the parser.
   To further optimize the number of calls, we have decided to take the top two ranked keywords. We do not apply logical operator (AND/OR) in our queries; instead, we perform one-to-one mapping between each keyword and query. Indeed, we have found that gathering keywords even if semantically related might bring up noise in the results. However, as mentioned earlier, a part of the future work will be investigating the best method to construct the most relevant queryable entity using different logical operators.

[22] http://stackexchange.com/sites
[23] http://developer.yahoo.com/yql/

3. **Caching:**The main setback in the query layer was the variable limited number of calls we can make to external APIs. To overcome this, we have implemented a simple cache mechanism that saves the results on disk up to an hour. There are several cache levels; the first is a URL level one where the results of the parsed queries are cached. For example, if a user visited a certain article on the CNN webpage the results might take up to 15 seconds to appear, whereas a second user visiting the same article minutes afterwards will have the cached results in few seconds. The second level is keyword and service specific. This can be very helpful as users generally browse articles of related topics or interests (semantic concepts), so for each user we can end up with the same high level concepts being requested frequently. An important thing to note is that the caching is done on the server side and is disk-based.

The social services queried can be grouped as follows:

1. **Multimedia Services:** They include Slideshare, Vimeo and YouTube. Slideshare and YouTube allow the results to be fetched in a specific language that was detected in the previous step. In addition to that, YouTube search services are called twice; the first call is done to the YouTube V2 API[24] where we specify in addition to the keywords a high level category to be targeted. To do so, we have manually created a category mapping file that maps the high-levels categories of Alchemys API and DMOZ to those provided by YouTube. The second call is done to YouTube V3 API[25]. The new feature provided by Google in this version is the ability to search using a semantic concept that corresponds to a Freebase concept ID; it proves to retrieve better results that the normal search. Freebase concept calls are cached for longer periods as they are less prone to changes.
2. **Micro-posts Services:** They include Twitter, Google+ and Stackoverflow. Language filtering is done where applicable.
3. **General Search:** This includes similar results found via Google search or those retrieved from the Zemanta API call. They are general articles or blog posts related to the current active page.

4.5.1.3 Data Parser

This is the last step where the results and unified and wrapped in a single social model. Figure 3 shows the different steps needed to produce the final parsed results that will be pushed back to the front-end.

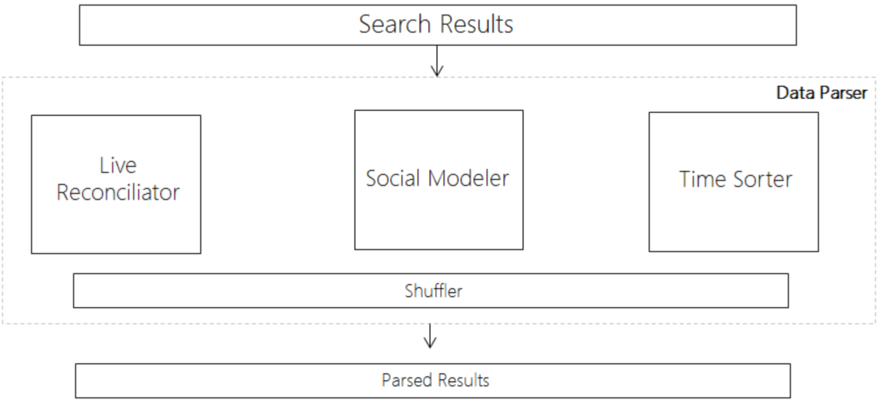

Figure 4. SNARC's Data Parser

[24] https://developers.google.com/youtube/2.0/
[25] https://developers.google.com/youtube/v3/

1. **Live Reconciliator:** Social (or folksonomic) tagging has become a trending method to describe, search and discover content on the web. Folksonomies empower users by giving them total freedom in choosing their categories and keywords that they think describe best the content. This contrasts with taxonomies that over-impose hierarchical categorization of content [34]. However, in services like Twitter and Google+, tagging has been abused in a way that increased noise in the stream of results. To overcome this problem, we align the incoming stream of posts with the set of semantic concepts or keywords that describe the document. There are several approaches and tools like [35,36,37,34] that aim at solving this problem. In SNARC we rely on two levels of reconciliation: one uses the high-level taxonomy (categories); and the other uses the vector of entities defined in the Semantic Document. For example, if SNARC wants to reconcile a blog post result retrieved from a general search, it constructs a Semantic Document Model for that result and applies the Cosine Similarity on the vector of ranked entities for each Semantic Model. Currently, we only reconcile against blog posts as it is very straightforward to construct a Semantic Document Model for them. However, an integral part of the future work will be the integration of SNARC's model to micro-posts and video search services.
2. **Social Modeler:** Every social network has its own underlying data model. To overcome this problem, we need to present the social results in a common wrapper. To do so, we have created an optimized universal social model that contains all the necessary data to model social information and can be reused in other projects. The model contains service related attributes like the service name and type, general post information like the author's name, profile link, image and geo-location information and post-specific information like the title, thumbnail, embed code, main content and link.
3. **Time Sorter and Results Shuffler:** To better display the results on the front-end, we unify the time representation and sort the results based on it. Afterwards we pick the top N results and shuffle them to generate a random order.

5. Dataset Quality Control

We are entering an era where open is the new default. Governments, universities, organizations and even individuals are publicly publishing huge amounts of open data. This openness should be accompanied with a certain level of trust or guarantees about the quality of data. To our knowledge, only one certificate is available to data publishers to assess the quality level of their datasets, the ODI certificate[26].

This certificate provides a description of the published data quality in plain English. It aspires to act as a mark of approval that helps publishers understand how to publish good open data and users how to use it. It gives publishers the ability to provide assurance and support on their data while encouraging further improvements through an ascending scale.

ODI comes as an online and free questionnaire for data publishers focusing on certain characteristics about their data. The questions are classified into the following categories: general information (about dataset, publisher and type of release), legal information (e.g., rights to publish), licensing, privacy (e.g., whether individuals can be identified), practical information (e.g., how to reach the data), quality, reliability, technical information (e.g., format and type of data) and social information (e.g., contacts, communities, etc.). Based on the information provided by the data publisher, a certificate is created with one of four different ratings.

Although ODI is a great initiative, the issued certificates are self-certified. ODI does not verify or review submissions but retains the right to revoke a certificate at any time. The dynamicity of Linked Data makes it also very difficult to update the certificates manually, especially when these changes are frequent and

[26] https://certificates.theodi.org/

affect multiple categories. There is clearly a need for automatic certification which can be supplemented with some manual input for categories that cannot be processed by machines.

The emerging critical need for large, distributed, heterogeneous, and complex structured datasets identified the necessity to establish industry cooperation between vendors of RDF and Graph database technologies in developing, endorsing, and publishing reliable and insightful benchmark results. The Linked Data Benchmark Council (LDBC)[27] aims to bridge the gap between the industry and the new trending stack of semantic technologies and their vendors.
LDBC more specifically aims at developing new benchmarks that will lead to significant progress in scalability, storage, indexing and query optimization techniques to become the de facto standard for publishing performance results. LDBC is promising initiative, but it is still work in progress with the final report expected on the first quarter of 2015.

In addition to the initiatives mentioned above, there exists a number of data quality frameworks and tools that are either standalone or implemented as modules in data integration tools.

LODGRefine[28] is the Open Refine[29] of Linked Data. It does not act as a quality assessment tool, but it is powerful in cleaning and refining raw instance data. LODGRefine can help detect duplicates, empty values, spot inconsistencies, extract Named Entities, discover patterns and more. LODGRefine helps in improving the quality of the dataset by improving the quality of the data at the instance level.

PROLOD [38] is also not a quality assessment tool. It is a Linked Data profiling tool that provides clustering and labeling capabilities, schema discovery and statistics about data types and patterns. The statistics are about properties distribution, link-to-literal ratio, number of entities and RDF triples, average properties per entity and average error. PROLOD had been tested with DBpedia but the authors plan to improve its scalability to larger datasets.

Sieve [39] is framework for expressing quality assessment and fusion methods. It is implemented as a component of the Linked Data Integration Framework (LDIF)[30]. Sieve leverages the LDIF provenance metadata as quality indicators to produce quality assessment scores. However, despite its nice features, it is only targeted to perform data fusion based on user-configurable conflict resolution tasks. Moreover, since Sieve main input is provenance metadata, it is only limited to domains that can provide such metadata associated with their data.

Quality Assessment of Data Sources (Flemming's Data Quality Assessment Tool)[31] calculates data quality scores based on manual user input. The user should assign weights to the predefined quality metrics and answer a series of questions regarding the dataset. These include, for example, the use of obsolete classes and properties by defining the number of described entities that are assigned disjoint classes, the usage of stable URIs and whether the publisher provides a mailing list for the dataset. The main disadvantage for using this tool is the manual intervention which requires deep knowledge in the dataset examined. Moreover, the tool lacks support for several quality concerns like completeness or consistency.

SWIQA [40] is composed of three layers: data acquisition, query and ontology layers. It uses query templates based on the SPARQL Inferencing Notation (SPIN)[32] to express quality requirements. The

[27] http://ldbc.eu/

[28] http://code.zemanta.com/sparkica/

[29] http://openrefine.org/

[30] http://ldif.wbsg.de/

[31] http://linkeddata.informatik.hu-berlin.de/LDSrcAss/datenquelle.php

[32] http://spinrdf.org/

queries are built to compute weighted and unweighted quality scores. At the end of the assessment, it uses vocabulary elements to annotate important values of properties and classes, assigning inferred quality scores to ontology elements and classifying the identified data quality problems.

Despite all the recent efforts in providing frameworks and tools for data quality in Linked Open Data, there is still no framework for the objective assessment of such quality taking into account all aspects of Linked Open Data.

In our previous work [19] we have identified 24 different Linked Data quality attributes. We have refined these attributes into a condensed framework of 13 objective attributes. Since these attributes are rather abstract, we should rely on quality indicators that reflect data quality [41]. In this paper, we transform the quality indicators presented as a set of questions in [19] into more concrete quality indicator metrics. We extend them with the the objective quality indicators listed in the systematic review done in [42].

Table(1) lists the refined attributes along with their quality indicators. These attributes are:

5.1. Objective Linked Data Quality Classification

5.1.1. Completeness

Data completeness can be judged in the presence of a task where the ideal set of attributes and objects are known. An entity is considered to be complete if it contains all the attributes needed for a given task, has complete language coverage [43] and has documentation properties [44][43].

A dataset is considered to be complete if it contains all the necessary objects for a given task [39], contains supporting structured metadata [45], contains links for external data providers, supports providing data in multiple serializations [42], includes the correct MIME-type for the content [45], contains appropriate volume of data for a particular task [42], has different queryable endpoints to access the data (i.e. SPARQL endpoint, RDF Dump, REST API, etc.) [42], has been checked against syntactic errors [45], uses datasets description vocabularies like DCAT[33] or VOID[34] and if the publishers provide descriptions about the size (using void:statItem, void:numberOfTriples or void:numberOfDocuments) and categorization (using dcterms:subject) of the dataset.

Links are considered to be complete if all the in-bound and out-bound links are dereferenceable [45][43][46] and have the linkage information represented in the metadata [45].

Models are considered to be complete if they have a complete set of values [43] and do not contain disconnected graph clusters [43]. Disconnected graphs are the result of incomplete data acquisition or accidental deletion of terms that leads to deprecated terms. In addition to that, models are considered to be complete if they do not contain omitted top concepts or unidirectional related concepts [45] and if there exists some metadata about the kind and number of used vocabularies [42].

5.1.2. Availability

A dataset is considered to be available if the publishers provide an RDF dump that can be downloaded by users [41][45] and if its queryable endpoints respond to direct queries.

5.1.3. Correctness

Data correctness is related to the validity of its entities. An entity is considered to be correct if there are no missing or empty labels [47][43], no incorrect data type for typed literals [45][47], no omitted or invalid languages tags [48][43] and does not contain “orphan terms” (orphan terms are terms without any associative or hierarchical relationships [49]).

Links are considered to be correct if they actually show related content to the subject of the RDF triple [48][47] and are syntactically correct.

[33] http://www.w3.org/TR/vocab-dcat/

[34] http://www.w3.org/TR/void/

Models are considered to be correct if the top concepts are marked and do not have broader concepts (for example having incoming hasTopConcept or outgoing topConceptOf relationships) [43].

5.1.4. Conciseness

Extensional conciseness measures the number of unique objects in relation to the overall number of objects representation in the dataset. Intensional conciseness measures the number of unique attributes of a dataset in relation to the overall number of attributes in a target schema [50].

An entity is considered to be concise if it has intensional conciseness (it does not contain redundant attributes, which means that there is no equivalent attributes with different names) [39] and uses short URIs [42] that follow the HTTP URI scheme [51][52].

A dataset is considered to be concise if it has extensional conciseness (it does not contain redundant objects, which means that there is no equivalent objects with different identifiers) [39].

5.1.5. Consistency

An entity is considered to be consistent if it does not contain overlapping labels such as two concepts have the same preferred lexical label in a given language when they belong to the same schema [53][43]. Moreover, an entity is considered to be consistent if it does not contain disjoint labels [43], extra white spaces in labels [48] and does not contain inconsistent preferred labels per language tag and no more than one value of skos:prefLabel without a language tag [43][48].

A dataset is considered to be consistent if it is free of conflicting information. This can be measured by considering properties with cardinality 1 that contain more than one distinct value [39].

Models are considered to be consistent if they do not include atypical use of collections, containers and reification [45], overlapping usage of owl:sameAs and owl:differentFrom [45], overlapping usage of owl:AllDifferent and owl:distinctMembers [45], asserted members of owl:Nothing and membership violation for disjoint classes [45].

5.1.6. Coherence

Coherence is the ability to interpret data as expected by the publisher or vocabulary maintainer [45]. It is mainly associated with the modeling quality.

A model is considered to be coherent when it does not contain:

- Usage of undefined classes and properties [45]. Many errors that are due to spelling or syntactic mistakes are resolvable through minor fixes via ontology checkers tools. However, for new terms, [45] suggests to have them defined in a separate namespace in order to allow reuse [43].
- Usage of blank nodes as they affect merging data from different sources [51].
- Misplaced or deprecated classes or properties [45].
- Misuse of the owl:DataTypeProperty or owl:ObjectProperty [45].
- Relations and mappings clashes [48].
- Invalid inverse-functional values [45].
- Cyclic hierarchical relations [54][48][43].
- Incomplete literals with datatype range [45].
- Solely transitive related concepts [43].
- Redefinitions of existing vocabularies [45].
- Valueless associative relations [43].

5.1.7. Efficiency

Dataset efficiency is calculated by measuring how fast it can be identified [55]. A dataset is considered to be efficient if it satisfies the following performance metrics:

- No usage of slash-URIs where large amounts of data is provided [42].
- Acceptable delay between the request and its response [56].
- Low Latency HTTP requests (average answer time of one second) [42].
- Scalable such that the time to answer an amount of ten requests divided by ten is not longer than the time it takes to answer one request [42].

5.1.8. Freshness

Freshness is a measure for the recency of data. The basic assumption is that old information is more likely to be outdated and unreliable [57]. Entity freshness can be identified if it contains timestamps that can keep track of its modifications.

5.1.9. Accuracy

Accuracy describes the proximity of data value representations of an object related to their real world states [40]. A dataset is considered to be accurate when it does not contain outliers and attributes that do not contain useful values for data entries [42].

5.1.10. Provenance

Entity level provenance can be calculated by constructing decision networks informed by provenance graphs [58]. The accuracy of computing trust between two entities [42] can be computed by calculating an aggregate trust value based on the combination of the propagation and aggregation algorithms on weighted mechanism [59]. Provenance can be achieved at the dataset level by including metadata that describes its authoritative information (title, content and URI), ensuring the reliability and trustworthiness of the publisher [57], verifying if the dataset uses a provenance vocabulary like PROV [60] and uses digital signatures [42].
Models provenance can be achieved by ensuring the trustworthiness of RDF statements [61].

5.1.11. Security

Security is a quality attribute that is measured on the dataset level. It is identified if the publishers use login credentials, SSL or SSH to provide access to their dataset, or if they only grant access to specific users [42].

5.1.12. Licensing

Licensing is a quality attribute that is measured on the dataset level. It includes the availability of machine readable license information [51], human readable license information in the documentation of the dataset or its source [51] and the indication of permissions, copyrights and attributions specified by the author [42].

5.1.13. Comprehensibility

Comprehensibility is identified if the publisher indicates at least one exemplary URI and SPARQL query, regular expression pattern that matches the URIs of a dataset [42] and if he provides a list of used vocabularies and an active mailing list or message board for the dataset [41].

Table 2: Objective Assessment Framework for Linked Data Quality

Quality Attribute	**Quality Category**	**ID**	**Quality Indicator**
Completeness	Entity Level	QI.1	Covers of all the attributes needed for a given task [39]
		QI.2	Has complete language coverage [43]
		QI.3	Existence of documentation properties [44][43]
	Dataset Level	QI.4	Existence of all the necessary objects for a given task [39]
		QI.5	Existence of supporting structured metadata [45]
		QI.6	Supports multiple serializations [42]
		QI.7	Includes the correct MIME-type for the content [45]
		QI.8	Contains appropriate volume of data for a particular task [42]
		QI.9	Has different queryable endpoints to access the data [42]
		QI.10	Checked against syntactic errors [45]
		QI.11	Usage of datasets description vocabularies
		QI.12	Existence of descriptions about its size and categorization
	Links Level	QI.13	Existence of complete dereferenceable in-bound and out-bound links [45][43][46]
		QI.14	Existence of supporting linkage metadata [45]
	Model Level	QI.15	Covers the complete set of values [43]
		QI.16	Absence of disconnected graph clusters [43]
		QI.17	Absence of omitted top concept [45]
		QI.18	Absence of unidirectional related concepts [45]
		QI.19	Existence of supporting metadata about the kind and number of used vocabularies [42]
Availability	Dataset Level	QI.20	Existence of an RDF dump that can be downloaded by users [41][45]
		QI.21	Existence of queryable endpoints that respond to direct queries
Correctness	Entity Level	QI.22	Absence of missing or empty labels [47][43]
		QI.23	Absence of incorrect data type for typed literals [45][47]
		QI.24	Absence of omitted or invalid languages tags [48][43]
		QI.25	Absence of terms without any associative or hierarchical relationships [49]
	Links Level	QI.26	Existence of content related to the subject of the RDF triple [48][47]
		QI.27	Absence of syntactic errors [52]
	Model Level	QI.28	Contains marked top concepts [43]
		QI.29	Absence of broader concepts for top concepts [43]
Conciseness	Entity Level	QI.30	Absence of redundant attributes [39]
		QI.31	Existence of short URIs [42]
	Dataset Level	QI.32	Absence of redundant objects [39]
		QI.33	Follows the HTTP URI scheme [51][52]
Security	Dataset Level	QI.34	Uses login credentials to restrict access [42]
		QI.35	Uses SSL or SSH to provide access to their dataset [42]
		QI.36	Grants access to specific users [42]
Freshness	Entity Level	QI.37	Existence of timestamps that can keep track of its modifications [57]
Licensing	Dataset Level	QI.38	Existence of machine readable license information [51]
		QI.39	Existence of human readable license information [51]
		QI.40	Specifies permissions, copyrights and attributions [42]

Continued on next page

Table 2 Objective Assessment Framework for Linked Data Quality

Quality Attribute	Quality Category	ID	Quality Indicator
Comprehensibility	Dataset Level	QI.41	Existence of at least one exemplary URI [42]
		QI.42	Existence of at least one exemplary SPARQL query [42]
		QI.43	Existence of regular expression pattern that matches the URIs of a dataset [42]
		QI.44	Existence of a list of used vocabularies
		QI.45	Existence of a mailing list or message board [41]
Consistency	Entity Level	QI.46	Existence of consistent preferred labels per language tag [53][43]
		QI.47	Absence of overlapping labels
		QI.48	Absence of disjoint labels [43]
		QI.49	Absence of extra white spaces in labels [48]
		QI.50	Existence of only one value of skos:prefLabel without a language tag [43][48]
	Dataset Level	QI.51	Absence of conflicting information [39]
	Model Level	QI.52	Absence of atypical use of collections, containers and reification [45]
		QI.53	Absence of overlapping usage of owl:sameAs and owl:differentFrom [45]
		QI.54	Absence of overlapping usage of owl:AllDifferent and owl:distinctMembers [45]
		QI.55	Absence of asserted members of owl:Nothing [45]
		QI.56	Absence of membership violations for disjoint classes [45]
Coherence	Model Level	QI.57	Absence of misplaced or deprecated classes or properties [45]
		QI.58	Absence of misused owl:DataTypeProperty or owl:ObjectProperty [45]
		QI.59	Absence of relation and mappings clashes [48]
		QI.60	Absence or minimal usage of blank nodes [51]
		QI.61	Absence of invalid inverse-functional values [45]
		QI.62	Absence of cyclic hierarchical relations [54][48][43]
		QI.63	Absence of undefined classes and properties usage [45]
		QI.64	Absence of solely transitive related concepts [43]
		QI.65	Absence of redefinitions of existing vocabularies [45]
		QI.66	Absence of valueless associative relations [43]
		QI.67	Absence of incomplete literals with datatype range [45]
Efficiency	Dataset Level	QI.68	Absence of slash-URIs [42]
		QI.69	Acceptable delay between the request and its response [56]
		QI.70	Low Latency HTTP requests [42]
		QI.71	Ability to scale [42]
Accuracy	Dataset Level	QI.72	Absence of outliers [42]
		QI.73	Absence of attributes that do not contain useful values for data entries [42]
Provenance	Entity Level	QI.74	Ability to construct decision networks informed by provenance graphs [58]
	Dataset Level	QI.75	Existence of metadata that describes its authoritative information [57]
		QI.76	Reliability and Trustworthiness of the publisher [57]
		QI.77	Usage of a provenance vocabulary
		QI.78	Usage of digital signatures [42]
	Model Level	QI.79	Trustworthiness of RDF statements [61]

5.2. Linked Data Quality Tools

In this section, we present the results of our survey on the Linked Data quality tools that reflect the different aspects of LOD: modeling, ontologies and vocabularies, dataset and SPARQL end-points.

5.2.1. Information Modeling Quality

RDF is the standard to model information in the Semantic Web. Linked Data publishers can pick from a plethora of tools that can check their RDF files for quality problems[35]. Syntactic RDF checkers are able to detect errors in RDF documents like the W3C RDF Validator[36], RDF:about validator and Converter[37] and The Validating RDF Parser (VRP)[38]. The RDF Triple-Checker[39] is an online tool that helps find typos and common errors in RDF data. Vapour[40] [62] is a validation service to check whether semantic Web data is correctly published according to the current best practices[63].

5.2.2. Ontologies and Vocabularies Quality

Reusing existing ontologies is a common practice that Linked Data publishers are always trying to adopt. However, ontologies and vocabularies development is often a long error-prone process especially when many contributors are working consecutively or collaboratively [52]. This can introduce deficiencies such as redundant concepts or conflicting relationships [64]. Getting to choose the right ontology or vocabulary is vital to ensure modeling correctness and consistency.

qSKOS[41] [43] scans SKOS vocabularies to provide reports on vocabulary resources and relations that are problematic. Skosify [48] supports OWL and RDFS ontologies by converting them into well-structured SKOS vocabularies. Skosify includes automatic correction abilities for quality issues that have been observed by reviewing vocabularies on the Web. The OOPS! pitfall scanner [65] evaluates OWL ontologies against a rules catalog and provides the user with a set of guidelines to solve them. PoolParty checker[42] highlights quality issues in OWL, RDFS and SKOS ontologies and vocabularies, the latest version supports qSKOS to indicate the quality of controlled vocabularies on the Web. ASKOSI[43] retrieves vocabularies from different sources, stores and displays the usage frequency of the different concepts used by different applications. It promotes reusing existing information systems by providing better management and presentation tools.

Some errors in RDF will only appear after reasoning (incorrect inferences). In [66][67] the authors perform quality checking on OWL ontologies using integrity constraints involving the Unique Name Assumption (UNA) and the Closed World Assumption (CWA). Pellet[44] provides reasoning services for OWL ontologies. It incorporates a number of heuristics to detect and repair quality issues among disjoint properties, negative property assertions and reflexive, irreflexive, symmetric, and anti-symmetric properties. Eyeball[45] provides quality inspection for RDF models (including OWL). It provides checks for a variety of problems including the usage of unknown predicates, classes, poorly formed namespaces, literal syntax validation, type consistency and other heuristics. RDF:Alerts[46] provides validation for many issues highlighted in [45] like misplaced, undefined or deprecated classes or properties.

[35] http://www.w3.org/2001/sw/wiki/SWValidators

[36] http://www.w3.org/RDF/Validator/

[37] http://rdfabout.com/demo/validator/

[38] http://139.91.183.30:9090/RDF/VRP/index.html

[39] http://graphite.ecs.soton.ac.uk/checker/

[40] http://validator.linkeddata.org/vapour

[41] https://github.com/cmader/qSKOS

[42] http://www.poolparty.biz/

[43] http://www.w3.org/2001/sw/wiki/ASKOSI

[44] http://clarkparsia.com/pellet

[45] http://jena.sourceforge.net/Eyeball/

[46] http://swse.deri.org/RDFAlerts/

5.2.3. Dataset Quality

Considering the large amount of available datasets in the Linked Open Data, users have a hard time trying to identify appropriate datasets that suit certain tasks. There are two approaches to rank datasets, a manual and an automatic approach. The manual approach depends on the wisdom of the crowd to highlight specific quality issues. The automatic approach has several implementations. The most adopted ones are based on link assessment. Provenance-based approaches and entity-based approaches are also used to compute not only dataset rankings, but also rankings on the entity level.

Manual Ranking Tools

There are several quality issues that can be difficult to spot and fix automatically. In [47] the authors highlight the fact that the RDFification process of some data can be more challenging than others, leading to errors in the Linked Data provisioning process that needs manual intervention. This can be more visible in datasets that have been semi-automatically translated to RDF from their primary source (the best example for this case is DBpedia [68]). The authors introduce a methodology to adjust crowdsourcing input from two types of audience: 1) Linked Data experts, researchers and enthusiasts through a contest to find and classify erroneous RDF triples and 2) Crowd-sourcing through the Amazon Mechanical Turk[47].
TripleCheckMate [69] is the tool used by the authors to run out their assessment. The tool allows users to select resources, identify and classify possible issues according to a pre-defined taxonomy of quality problems. It measures inter-rater agreements, meaning that the resources defined are checked multiple times. These features turn out to be extremely useful to analyze the performance of users and allow better identification of potential quality problems. TripleCheckMate is used to identify accuracy issues in the object extraction (completeness of the extraction value for object values and datatypes), relevancy of the extracted information, representational consistency and interlinking with other datasets.

Automatic Ranking Tools

Links Based Approach

The basic idea behind link assessment tools is to provide rankings for datasets based on the cardinality and types of the relationships with other datasets. Traditional link analysis has proven to be an effective way to measure the quality of Web documents search. Algorithms like PageRank [70] and HITS [71] became successful based on the assumption that a certain Web document is considered to have higher importance or rank if it has more incoming links that other Web documents [72][73].
However, the basic assumption that links are equivalent does not suit the heterogeneous nature of links in the Linked Open Data. Thus, the previous approaches fall short to provide reliable rankings as the types of the links can have a direct impact on the ranking computation [55].
The first adaption of PageRank for Semantic Web resources was the Ontology Rank algorithm implemented in the Swoogle search engine [74]. They use a rational random surfing model that takes into account the different types of links between discovered sets and compute rankings based on three levels of granularity: documents, terms and RDF graphs. ReConRank [75] rankings are computed at query time based on two levels of granularity: resources and context graphs. DING [55] adapted the PageRank to rank datasets based on their interconnections. DING can also automatically assign weights to different link types based on the nature of the predicate involved in the link.
Broken links are a major threat to Linked Data. They occur when resources are removed, moved or updated. DSNotify[48][76] is a framework that informs data consumers about the various types of events that occur on data sources. Their approach is based on an indexing infrastructure that extracts feature vectors and stores them to an index. A monitoring module detects events on sources and write them to a central

[47] https://www.mturk.com/
[48] http://www.cibiv.at/ niko/dsnotify/

event log which pushes notifications to registered applications.

Provenance-based Approach

Provenance-based assessment methods are an important step towards transparency of data quality in the Semantic Web. In [61] the authors use a provenance model as an assessment method to evaluate the timeliness of Web data. Their model identifies types of "provenance elements" and the relationships between them. Provenance elements are classified into three types: actors, executions and artifacts. The assessment procedure is divided into three steps: 1) Creating provenance graph based on the defined model 2) Annotating the graph with impact values 3) Calculating the information quality score. In [57] the authors describe a set of provenance-based assessment metrics to support quality assessment and repair in Linked Open Data. They rely on both data and metadata and use indicators like the source reputation, freshness and plausibility. In [77] the authors introduce the notion of naming authority which connects an identifier with the source to establish a connection to it's provenance. They construct a naming authority graph that acts as input to derive PageRank scores for the data sources.

Entity-based Approach

Sindice [78] uses a set of techniques to rank Web data. They use a combination of query dependent and query independent rankings implemented in the Semantic Information Retrieval Engine (SIREn)[49] to produce a final entity rank. Their query dependent approach rates individual entities by aggregating the the score of the matching terms with a term frequency - inverse subject frequency (tf-isf) algorithm. Their query independent ranking is done using hierarchical links analysis algorithms [79]. The combination of these two approaches is used to generate a global weighted rank based on the dataset, entities and links ranks.

5.2.4. Queryable End-point Quality

The availability of Linked Data is highly dependent on the performance qualities of its queryable end-points. The standard query language for Semantic Web resources is SPARQL, thus SPARQL endpoints are the main focus. In [80] the authors present their findings to measure the discoverability of SPARQL endpoints by analyzing how they are located and the metadata used to describe them. In addition to that, they also analyze endpoints interoperability by identifying features of SPARQL 1.0 and SPARQL 1.1 that are supported. The authors tackled the endpoints efficiency by testing the time taken to answer generic, content-agnostic SPARQL queries over HTTP. Finally, the authors measured endpoints reliability by monitoring the uptime of public SPARQL endpoints on a course of 27 months. The results for this work can be accessed online via the SPARQL Endpoints Status tool [50] and is queryable using their public SPARQL endpoint[51].

5.3. Linked Data Quality Tools Evaluation

In this section, we present the results for our evaluation of the various Linked Data tools mentioned with regards to the presented framework (Table 2). Due to space limitation, we have only shown the tools that can directly or indirectly assess one or more of the quality indicators listed in Table 1. Moreover, we have also grouped the Syntax Validation tools (W3C RDF Validator, RDF:about, VRP, The RDF Triple-Checker and Vapour) under one approach. We should highlight that the PoolParty tool includes

[49] http://siren.sindice.com/
[50] http://labs.mondeca.com/sparqlEndpointsStatus/
[51] http://labs.mondeca.com/sparqlEndpointsStatus/endpoint/endpoint.html

qSKOS in its implementation, thus they are grouped in one tool which is the PoolParty.

Table 3: Evaluation of Linked Data quality tools for each quality indicator

	PROLOD	Sieve	Flemming's Tool	SWIQA	Syntax Validators	Skosify	OOPS!	PoolParty	ASKOSI	Pellet	Eyeball	RDF:Alerts	TripleCheckmate	DING	DSNotify	SPARQL Endpoints-Status
QI.1		✓		✓									✓			
QI.2			✓					✓	✓							
QI.3			✓					✓								
QI.4		✓		✓									✓			
QI.5			✓													
QI.6			✓											✓		
QI.7			✓													
QI.8	✓		✓													
QI.9			✓													
QI.10			✓	✓	✓							✓				
QI.11														✓		✓
QI.12	✓		✓											✓		
QI.13	✓		✓	✓				✓				✓			✓	
QI.14															✓	✓
QI.15		✓		✓												
QI.16								✓								
QI.17						✓		✓								
QI.18						✓		✓								
QI.19			✓													
QI.20			✓													
QI.21			✓													✓
QI.22	✓					✓		✓	✓				✓			
QI.23							✓			✓	✓	✓	✓			
QI.24						✓		✓								
QI.25								✓	✓							
QI.26																
QI.27		✓									✓	✓			✓	
QI.28						✓		✓								
QI.29							✓	✓	✓		✓					
QI.30		✓		✓			✓		✓							
QI.31																
QI.32		✓		✓												
QI.33								✓				✓				
QI.34																
QI.35																
QI.36																
QI.37		✓		✓												
QI.38			✓													
QI.39			✓													

Continued on next page

Table 3 Evaluation of Linked Data quality tools for each quality indicator

	PROLOD	Sieve	Flemming's Tool	SWIQA	Syntax Validators	Skosify	OOPS!	PoolParty	ASKOSI	Pellet	Eyeball	RDF:Alerts	TripleCheckmate	DING	DSNotify	SPARQL Endpoints-Status
QI.40			✓													
QI.41			✓													
QI.42			✓													
QI.43			✓													
QI.44			✓													
QI.45			✓													
QI.46						✓		✓	✓							
QI.47							✓	✓			✓					
QI.48						✓	✓	✓			✓					
QI.49						✓					✓					
QI.50									✓							
QI.51		✓								✓	✓		✓			
QI.52											✓					
QI.53							✓			✓						
QI.54										✓						
QI.55										✓						
QI.56			✓			✓	✓	✓		✓	✓	✓				
QI.57							✓	✓			✓	✓				
QI.58							✓			✓	✓					
QI.59								✓								
QI.60																
QI.61			✓				✓			✓		✓				
QI.62						✓	✓	✓	✓							
QI.63			✓				✓	✓		✓	✓					
QI.64						✓	✓	✓								
QI.65			✓				✓			✓	✓					
QI.66								✓								
QI.67							✓	✓		✓	✓	✓				
QI.68			✓													
QI.69																✓
QI.70			✓													✓
QI.71																✓
QI.72													✓			
QI.73													✓			
QI.74																
QI.75			✓													
QI.76																
QI.77			✓													
QI.78			✓													
QI.79																

As a result, we have identified the need for a complete quality framework that can assess all the quality indicators. Most of the tools were designed with limited coverage to certain aspects, for example, ontology and vocabulary checkers focus mainly on the coherence, completeness and correctness at the modeling level. Flemming's tool covers several attributes like the completeness, correctness, conciseness, security, licensing and comprehensibility, but it falls short in measuring the consistency, coherence, efficiency and provenance.

We have also noticed the lack of tools to measure certain quality indicators like the dataset's security. Moreover, to our knowledge, there are no tools that can measure all the provenance quality indicators (except for Flemming's tool that is able to check for the use of digital signatures) although the literature covers several approaches to achieve that [61][57][77].

6. Conclusions and Future work

We have presented in this document our main contributions in some issues around Enriching Enterprise Data Towards self-service Data Provisioning. We have first focused on the aspect of data profiling through RUBIX. We plan to extend RUBIX to be able to work with DBpedia leveraging the planned entity type ranking module. We also plan to include data mining techniques to profile numerical data and provide statistical insights about data distribution.

Regarding the Linked Data quality module, we plan to develop a comprehensive objective Linked Data quality evaluation tool. The tool will be able to automatically measure the various quality indicators listed in this paper, introduce a scoring function with different weights for the various quality attributes and issue a quality certificate.

Regarding the Social integration, we would like to test SNARC on business web application, check if our annotations can be used to successfully query and attach relevant social snippets to the data.

We also plan to build our Linked Data crawler that will be responsible for the data acquisition phase which is the entry point for the work done so far. We also plan to investigate possible extensions to the current data description vocabularies to allow more comprehensive datasets categorization.

References

[1] Nandana Mihindukulasooriya, Raul Garcia-Castro, and Miguel Esteban Gutiérrez. Linked data platform as a novel approach for enterprise application integration. In *COLD*, 2013.

[2] Maurizio Lenzerini. Data integration: A theoretical perspective. In *Proceedings of the Twenty-first ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '02, pages 233–246, New York, NY, USA, 2002. ACM.

[3] Philipp Frischmuth, Sren Auer, Sebastian Tramp, Jrg Unbehauen, Kai Holzweiig, and Car-Martin Marquardt. Towards linked data based enterprise information integration. In *Proceedings of the Workshop on Semantic Web Enterprise Adoption and Best Practice (WASABI) 2013*, 2013.

[4] Philipp Frischmuth, Jakub Klmek, Sren Auer, Sebastian Tramp, Jrg Unbehauen, Kai Holzweiig, and Carl-Martin Marquardt. Linked data in enterprise information integration. 2012.

[5] H. Wache, T. Vgele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, and S. Hbner. Ontology-based integration of information - a survey of existing approaches. pages 108–117, 2001.

[6] C. Bizer, T. Heath, and T. Berners-Lee. Linked data - the story so far. *Int. J. Semantic Web Inf. Syst.*, 5(3):122, 2009.

[7] Chris Bizer, Anja Jentzsch, and Richard Cyganiak. State of the lod cloud, 2011.

[8] Krzysztof Janowicz Prateek Jain, Pascal Hitzler and Chitra Venkatramani. Theres no money in linked data. 2013.

[9] D Boyd and Kate Crawford. Six provocations for big data. *A Decade in Internet Time: Symposium on the Dynamics of the Internet and Society*, pages 1–17, 2011.

[10] Diana Farrell Steve Van Kuiken Peter Groves James Manyika, Michael Chui and Elizabeth Almasi Doshi. Open data: Unlocking innovation and performance with liquid information. *McKinsey Business Technology Office*, 2013.

[11] Rebecca Shockley Michael S. Hopkins Steve LaValle, Eric Lesser and Nina Kruschwitz. Big data, analytics and the path from insights to value. *MIT Sloan Management Review*, 2011.

[12] G. Sudha; Shenoy Sangeetha N avitha, C.; Sadasivam. Ontology based semantic integration of heterogeneous databases. *European Journal of Scientific Research;11/13/2011,,*, Vol. 64(Issue 1):p115, 2011.

[13] Pascal Hitzler Amit Sheth Sarasi Lalithsena, Prateek Jain. Automatic domain identification for linked open data. *IEEE/WIC/ACM International Conference on Web Intelligence*, 2013.

[14] Richard Cyganiak, Jun Zhao, Keith Alexander, and Michael Hausenblas. Describing linked datasets with the VoID vocabulary. W3C note, W3C, March 2011. http://www.w3.org/TR/2011/NOTE-void-20110303/.

[15] Fadi Maali and John Erickson. Data catalog vocabulary (DCAT). Last call WD, W3C, August 2013. http://www.w3.org/TR/2013/WD-vocab-dcat-20130801/.

[16] Ahmad Assaf, Aline Senart, and Raphaël Troncy. Snarc - an approach for aggregating and recommending contextualized social content. In *ESWC (Satellite Events)*, pages 319–326, 2013.

[17] Ahmad Assaf, Eldad Louw, Aline Senart, Corentin Follenfant, Raphaël Troncy, and David Trastour. Rubix: a framework for improving data integration with linked data. In *Proceedings of the First International Workshop on Open Data*, WOD '12, pages 13–21, New York, NY, USA, 2012. ACM.

[18] Ahmad Assaf, Eldad Louw, Aline Senart, Corentin Follenfant, Raphaël Troncy, and David Trastour. Improving schema matching with linked data. *CoRR*, abs/1205.2691, 2012.

[19] Ahmad Assaf and Aline Senart. Data quality principles in the semantic web. *CoRR*, abs/1305.4054, 2013.

[20] G. W. Furnas, T. K. Landauer, L. M. Gomez, and S. T. Dumais. The vocabulary problem in human-system communication. *COMMUNICATIONS OF THE ACM*, 30(11):964–971, 1987.

[21] Mike Bergman. Deconstructing the Google Knowledge Graph. `http://www.mkbergman.com/1009/deconstructing-the-google-knowledge-graph`.

[22] Emmanuel Pietriga, Christian Bizer, David Karger, and Ryan Lee. Fresnel: A Browser-Independent Presentation Vocabulary for RDF. In *5^{th} International Semantic Web Conference (ISWC'06)*, pages 158–171, 2006.

[23] Alberto Tonon, Michele Catasta, Gianluca Demartini, Philippe Cudré-Mauroux, and Karl Aberer. Trank: Ranking entity types using the web of data. In *ISWC*, 2013.

[24] Renée J. Miller and Periklis Andritsos. Schema discovery. *IEEE Data Eng. Bull.*, 26(3):40–45, 2003.

[25] Davi de Castro Reis, Paulo B. Golgher, Altigran S. da Silva, and Alberto H. F. Laender. Automatic web news extraction using tree edit distance. In Stuart I. Feldman, Mike Uretsky, Marc Najork, and Craig E. Wills, editors, *Proceedings of the Thirteenth International World Wide Web Conference*, pages 502–601, New York, NY, May 2004. ACM Press.

[26] Jiying Wang and Frederick H Lochovsky. Data Extraction and Label Assignment for Web Databases. *The World Wide Web Conference*, pages 187–196, 2003.

[27] Altigran Soares da Silva, Denilson Barbosa, João M. B. Cavalcanti, and Marco A. S. Sevalho. Labeling data extracted from the web. In *OTM Conferences (1)*, pages 1099–1116, 2007.

[28] Girija Limaye, Sunita Sarawagi, and Soumen Chakrabarti. Annotating and searching web tables using entities, types and relationships. *Proc. VLDB Endow.*, 3(1-2):1338–1347, September 2010.

[29] Zareen Syed, Tim Finin, Varish Mulwad, and Anupam Joshi. Exploiting a Web of Semantic Data for Interpreting Tables. In *Proceedings of the Second Web Science Conference*, April 2010.

[30] Tim Finin, Zareen Syed, James Mayfield, Paul McNamee, and Christine D. Piatko. Using wikitology for cross-document entity coreference resolution. In *AAAI Spring Symposium: Learning by Reading and Learning to Read*, pages 29–35. AAAI, 2009.

[31] Oktie Hassanzadeh, Songyun Duan, Achille Fokoue, Anastasios Kementsietsidis, Kavitha Srinivas, and Michael J. Ward. Helix: online enterprise data analytics. In *WWW (Companion Volume)*, pages 225–228. ACM, 2011.

[32] Georges Gouriten and Pierre Senellart. Api blender: A uniform interface to social platform apis. *CoRR*, abs/1301.2086, 2013.

[33] Hyunmo Kang and Ben Shneiderman. Mediafinder: an interface for dynamic personal media management with semantic regions. In Gilbert Cockton and Panu Korhonen, editors, *CHI Extended Abstracts*, pages 764–765. ACM, 2003.

[34] Valentina Zanardi and L Capra. Social ranking: uncovering relevant content using tag-based recommender systems. *ACM conference on Recommender systems*, 2008.

[35] Iván Cantador and Alejandro Bellogín. Semantic contextualisation of social tag-based profiles and item recommendations. *E-Commerce and Web . . .*, (2), 2011.

[36] Ernesto Diaz-Aviles, Lucas Drumond, Lars Schmidt-Thieme, and Wolfgang Nejdl. Real-time top-n recommendation in social streams. In *Proceedings of the sixth ACM conference on Recommender systems - RecSys '12*, page 59, New York, New York, USA, 2012. ACM Press.

[37] D Preotiuc-Pietro and S Samangooei. Trendminer: An architecture for real time analysis of social media text. *AAAI Publications, Sixth International AAAI Conference on Weblogs and Social Media*, pages 4–7, 2012.

[38] Christophe Bohm, Felix Naumann, Ziawasch Abedjan, Fenz Dandy, Toni Grutze, Daniel Hefenbrock, Matthias Pohl, and David Sonnabend. Proling Linked Open Data with ProLOD.PDF. *ICDE 2010*, 2010.

[39] PN Mendes, Hannes Mühleisen, and Christian Bizer. Sieve: linked data quality assessment and fusion. *LWDM2012 - Proceedings of the 2012 Joint EDBT*, 2012.

[40] C Fürber and M Hepp. SWIQAA Semantic Web information quality assessment framework. *ECIS 2011*, 2011.

[41] A Flemming. Quality characteristics of linked data publishing datasources, 2010.

[42] Conceptual Framework, Amrapali Zaveri, Anisa Rula, Andrea Maurino, Ricardo Pietrobon, and Jens Lehmann. Quality

Assessment Methodologies for Linked Open Data. *Under review, Semantic Web Journal*, 1:1–5, 2012.

[43] Christian Mader, Bernhard Haslhofer, and Antoine Isaac. Finding quality issues in SKOS vocabularies. *Theory and Practice of Digital Libraries*, 2012.

[44] A. Miles and S. Bechhofer. Skos simple knowledge organization system reference. w3c recommendation 18 august 2009., 2009.

[45] Aidan Hogan, Andreas Harth, Alexandre Passant, Stefan Decker, and Axel Polleres. Weaving the pedantic web. *LDOW 2010*, 2010.

[46] Christophe Guéret, Paul T. Groth, Claus Stadler, and Jens Lehmann. Assessing linked data mappings using network measures. In *Proceedings of the 9th Extended Semantic Web Conference*, volume 7295 of *Lecture Notes in Computer Science*, pages 87–102. Springer, 2012.

[47] Maribel Acosta, Amrapali Zaveri, Elena Simperl, and Dimitris Kontokostas. Crowdsourcing Linked Data quality assessment. *ISWC 2013*, 2013.

[48] Osma Suominen and Eero Hyvönen. Improving the quality of skos vocabularies with skosify. In *Proceedings of the 18th international conference on Knowledge Engineering and Knowledge Management*, EKAW'12, pages 383–397, Berlin, Heidelberg, 2012. Springer-Verlag.

[49] Henry Living. Review of: Hedden, heather. the accidental taxonomist medford, nj: Information today, inc., 2010. *Inf. Res.*, 15(2), 2010.

[50] Jens Bleiholder and Felix Naumann. Data fusion. *ACM Comput. Surv.*, 41(1):1:1–1:41, January 2009.

[51] Aidan Hogan, JüRgen Umbrich, Andreas Harth, Richard Cyganiak, Axel Polleres, and Stefan Decker. An empirical survey of linked data conformance. *Web Semant.*, 14:14–44, July 2012.

[52] Osma Suominen and Christian Mader. Assessing and Improving the Quality of SKOS Vocabularies. *Journal on Data Semantics*, June 2013.

[53] Antoine Isaac and Ed Summers. Skos simple knowledge organization system primer. World Wide Web Consortium, Working Draft WD-skos-primer-20080829, August 2008.

[54] Dagobert Soergel. Thesauri and ontologies in digital libraries. In Mary Marlino, Tamara Sumner, and Frank M. Shipman III, editors, *JCDL*, page 421. ACM, 2005.

[55] Nickolai Toupikov, J Umbrich, and Renaud Delbru. DING! Dataset ranking using formal descriptions. *WWW09*, 2009.

[56] Christian Bizer. *Quality-Driven Information Filtering in the Context of Web-Based Information Systems*. PhD thesis, March 2007.

[57] Giorgos Flouris, Yannis Roussakis, and M Poveda-Villalón. Using provenance for quality assessment and repair in linked open data. pages 1–12, 2012.

[58] Matthew Gamble. Quality, Trust, and Utility of Scientic Data on the Web: Towards a Joint Model.pdf. *WebSci'11*, 2011.

[59] Saeedeh Shekarpour and S.D. Katebi. Modeling and evaluation of trust with an extension in semantic web. *Web Semantics: Science, Services and Agents on the World Wide Web*, 8(1), 2010.

[60] Khalid Belhajjame, James Cheney, David Corsar, Daniel Garijo, Stian Soiland-Reyes, Stephan Zednik, and Jun Zhao. Prov-o: The prov ontology. Technical report, 2012.

[61] Olaf Hartig and Jun Zhao. Using web data provenance for quality assessment. In *In: Proc. of the Workshop on Semantic Web and Provenance Management at ISWC*, 2009.

[62] Diego Berrueta, Sergio Fernndez, and Ivn Frade. Cooking http content negotiation with vapour. In *In Proceedings of 4th workshop on Scripting for the Semantic Web 2008 (SFSW2008). co-located with ESWC2008*, 2008.

[63] Berners-Lee Tim. Linked data. Technical report, W3C, July 2006. http://www.w3.org/DesignIssues/LinkedData.html.

[64] Patricia Harpring. *Introduction to Controlled Vocabularies: Terminology for Art, Architecture, and Other Cultural Works*. Getty Research Institute, Los Angeles, 2010.

[65] Mara Poveda-Villaln, MariCarmen Surez-Figueroa, and Asuncin Gmez-Prez. Validating ontologies with oops! In Annette Teije, Johanna Vlker, Siegfried Handschuh, Heiner Stuckenschmidt, Mathieu dAcquin, Andriy Nikolov, Nathalie Aussenac-Gilles, and Nathalie Hernandez, editors, *Knowledge Engineering and Knowledge Management*, volume 7603 of *Lecture Notes in Computer Science*, pages 267–281. Springer Berlin Heidelberg, 2012.

[66] Evren Sirin, Michael Smith, and Evan Wallace. Opening, closing worlds - on integrity constraints. In Catherine Dolbear, Alan Ruttenberg, and Ulrike Sattler, editors, *OWLED*, volume 432 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008.

[67] Jiao Tao, Li Ding, and Deborah L. McGuinness. Instance data evaluation for semantic web-based knowledge management systems. In *HICSS*, pages 1–10. IEEE Computer Society, 2009.

[68] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sren Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. DBpedia - a crystallization point for the web of data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3):154–165, September 2009.

[69] Dimitris Kontokostas, Amrapali Zaveri, S Auer, and J Lehmann. TripleCheckMate: A Tool for Crowdsourcing the Quality Assessment of Linked Data. *4th Conference on Knowledge Engineering and Semantic Web*, pages 1–8, 2013.

[70] Page Lawrence, Brin Sergey, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford University, 1998.

[71] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, September 1999.

[72] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the seventh international conference on World Wide Web 7*, WWW7, pages 107–117, Amsterdam, The Netherlands, The Netherlands, 1998. Elsevier Science Publishers B. V.

[73] Soumen Chakrabarti, Byron E. Dom, S. Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, Andrew Tomkins, David Gibson, and Jon Kleinberg. Mining the web's link structure, 1999.

[74] L Ding, Tim Finin, A Joshi, R Pan, and RS Cost. Swoogle: A semantic web search and metadata engine. *CIKM04*, 2004.

[75] Aidan Hogan, Andreas Harth, and Stefan Decker. Reconrank: A scalable ranking method for semantic web data with context. In *In 2nd Workshop on Scalable Semantic Web Knowledge Base Systems*, 2006.

[76] Bernhard Haslhofer and Niko Popitsch. Dsnotify: Detecting and fixing broken links in linked data sets. In *8th International Workshop on Web Semantics (WebS &#8217;09), co-located with DEXA 2009*, Berlin, Heidelberg, August 2009. Springer.

[77] Andreas Harth, Sheila Kinsella, and Stefan Decker. Using naming authority to rank data and ontologies for web search. *ISWC 2009*, 2, 2009.

[78] Renaud Delbru. Sindice at SemSearch 2010. *WWW10*, 2010.

[79] Renaud Delbru, Nickolai Toupikov, and Michele Catasta. Hierarchical link analysis for ranking web data. *The Semantic Web: Research and Applications*, 2010.

[80] C Buil-Aranda and Aidan Hogan. SPARQL Web-Querying Infrastructure: Ready for Action? *International ...*, 2013.

[81] Besiki Stvilia, Les Gasser, Michael B. Twidale, and Linda C. Smith. A framework for information quality assessment. *Journal of the American Society for Information Science and Technology*, 58(12):1720–1733, October 2007.

[82] Mamoru Ohtai, Kouji Kozaki and Riichiro Mizoguchi. A Quality Assurance Framework for Ontology Construction and Renement.pdf. *Web Intelligence Conference (AWIC2011)*, 2011.

[83] RY Wang and DM Strong. Beyond accuracy: What data quality means to data consumers. *Journal of management information systems*, 1996.

[84] Allen Moulton, S Madnick, and M Siegel. Cross-organizational data quality and semantic integrity: learning and reasoning about data semantics with context interchange mediation. *MIT Sloan*, III(1):1–4, 2001.

[85] Barbara Pernici and Monica Scannapieco. Data quality in web information systems. *Journal on Data Semantics I*, pages 48–68, 2003.

[86] Anisa Rula. DC proposal: towards linked data assessment and linking temporal facts. *ISWC 2011*, 2011.

[87] M D'Aquin. Formally measuring agreement and disagreement in ontologies. *K-CAP 09*, 2009.

[88] Stuart Madnick and Hongwei Zhu. Improving data quality through effective use of data semantics. *Data & Knowledge Engineering*, (October):1–19, 2006.

[89] Astrid Duque-ramos, Jesualdo Tomás Fernández-breis, Robert Stevens, and Nathalie Aussenac-gilles. OQuaRE : A SQuaRE-based Approach for Evaluating the Quality of Ontologies. *Journal of Research and Practice in Information Technology, Software Engineering and Semantic Web Technologies*, 43(2), 2011.

[90] Christian Mader and Bernhard Haslhofer. Perception and Relevance of Quality Issues in Web Vocabularies. *I-SEMANTICS 2013*, 2013.

[91] Stephen Wood. The Equation between Semantics and Data Quality. *Other Conferences*, 2010.

[92] Christian Fürber and Martin Hepp. Towards a vocabulary for data quality management in semantic web architectures. *Proceedings of the 1st International Workshop on Linked Web Data Management - LWDM '11*, page 1, 2011.

[93] Graeme Shanks and B Corbitt. Understanding data quality: Social and cultural aspects. *Proceedings of the 10th Australasian Conference on ...*, (1998):785–797, 1999.

[94] Christian Fürber and Martin Hepp. Using semantic web resources for data quality management. *Knowledge Engineering and Management by the ...*, pages 1–15, 2010.

[95] Christian Fürber and Martin Hepp. Using SPARQL and SPIN for Data Quality Management on the Semantic Web. *Business Information Systems*, (1):1–12, 2010.

[96] Li Ding, Rong Pan, Tim Finin, and Anupam Joshi. Finding and ranking knowledge on the semantic web. *ISWC 2005*, (November), 2005.

[97] Beverly K. Kahn, Diane M. Strong, and Richard Y. Wang. Information quality benchmarks: product and service performance. *Communications of the ACM*, 45(4ve):184–192, April 2002.

[98] PY Vandenbussche, CB Aranda, Aidan Hogan, and J Umbrich. Monitoring the Status of SPARQL Endpoints. *ISWC 2013*, 1380(3130617):3–6, 2013.

[99] Samir Tartir, I Budak Arpinar, Michael Moore, Amit P Sheth, and Boanerges Aleman-meza. OntoQA : Metric-Based Ontology Quality Analysis University of Georgia. *IEEE Workshop on Knowledge Acquisition from Distributed, Autonomous, Semantically Heterogeneous Data and Knowledge Sources*, 2005.

[100] Christian Fürber and Martin Hepp. Towards a vocabulary for data quality management in semantic web architectures. *Proceedings of the 1st International Workshop on Linked Web Data Management - LWDM '11*, page 1, 2011.

[101] Li Ding, Tim Finin, and Yun Peng. Tracking rdf graph provenance using rdf molecules. *ISWC 2005*, pages 1–2, 2005.

[102] Christian Fürber and Martin Hepp. Using SPARQL and SPIN for data quality management on the Semantic Web. *Business Information Systems*, (1):1–12, 2010.

[103] JLG Sánchez, Roberto García, and JM Brunetti. Using SWET-QUM to Compare the Quality in Use of Semantic Web

Exploration Tools. *Journal of Universal Computer Science*, 19(8):1025–1045, 2013.

[104] Joseph. M. Juran and A. Blanton Godfrey. *Juran's quality handbook*. Juran's quality handbook, 5e. McGraw Hill, 1999.

[105] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules. In *Proc. of 20th Intl. Conf. on VLDB*, pages 487–499, 1994.

[106] Yuangui Lei, Victoria Uren, and Enrico Motta. A framework for evaluating semantic metadata. In *Proceedings of the 4th international conference on Knowledge capture*, K-CAP '07, pages 135–142, New York, NY, USA, 2007. ACM.

[107] Eyal Oren, Renaud Delbru, Michele Catasta, Richard Cyganiak, and Giovanni Tummarello. Sindice.com: A document-oriented lookup index for open linked data. *International Journal of Metadata, Semantics and Ontologies*, 3:2008.

[108] N.I.S. Organization and National Information Standards Organization (U.S.). *Guidelines for the Construction, Format, and Management of Monolingual Controlled Vocabularies*. National information standards series. NISO Press, 2005.

[109] Eytan Bakshy, Itamar Rosenn, Cameron Marlow, and Lada Adamic. The role of social networks in information diffusion. In *Proceedings of the 21st international conference on World Wide Web - WWW '12*, page 519, New York, New York, USA, 2012. ACM Press.

[110] Marco Pennacchiotti and Siva Gurumurthy. Investigating topic models for social media user recommendation. In *WWW11*, page 101, New York, New York, USA, 2011. ACM Press.

[111] Houda Khrouf, Ghislain Atemezing, and Giuseppe Rizzo. Aggregating Social Media for Enhancing Conference Experiences. *Proceedings of the 1st Int. Workshop on Real-Time Analysis and Mining of Social Streams*, 2012.

[112] Thomas Steiner and Stefan Mirea. SEKI@home or Crowdsourcing an Open Knowledge Graph. In *1^{st} International Workshop on Knowledge Extraction & Consolidation from Social Media (KECSM'12)*, Boston, USA, 2012.

[113] Ben Shneiderman. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In *IEEE Symposium on Visual Languages*, pages 336–343, 1996.