Abstract

Knowledge bases are playing an increasingly important role for integrating information between systems and over the Web. Today, most knowledge bases cover only specific domains, they are created by relatively small groups of knowledge engineers, and it is very cost intensive to keep them up-to-date as domains change. In parallel, Wikipedia has grown into one of the central knowledge sources of mankind and is maintained by thousands of contributors. The DBpedia (http://dbpedia.org) project makes use of this large collaboratively edited knowledge source by extracting structured content from it, interlinking it with other knowledge bases, and making the result publicly available. DBpedia had and has a great effect on the Web of Data and became a crystallization point for it. Furthermore, many companies and researchers use DBpedia and its public services to improve their applications and research approaches.

However, the DBpedia release process is heavy-weight and the releases are sometimes based on several months old data. Hence, a strategy to keep DBpedia always in synchronization with Wikipedia is highly required. In this thesis we propose the *DBpedia Live* framework, which reads a continuous stream of updated Wikipedia articles, and processes it. DBpedia Live processes that stream on-the-fly to obtain RDF data and updates the DBpedia knowledge base with the newly extracted data. DBpedia Live also publishes the newly added/deleted facts in files, in order to enable synchronization between our DBpedia endpoint and other DBpedia mirrors. Moreover, the new DBpedia Live framework incorporates several significant features, e.g. abstract extraction, ontology changes, and changesets publication.

Basically, knowledge bases, including DBpedia, are stored in triplestores in order to facilitate accessing and querying their respective data. Furthermore, the triplestores constitute the backbone of increasingly many Data Web applications. It is thus evident that the performance of those stores is mission critical for individual projects as well as for data integration on the Data Web in general. Consequently, it is of central importance during the implementation of any of these applications to have a clear picture of the weaknesses and strengths of current triplestore implementations. We introduce a generic SPARQL benchmark creation procedure, which we apply to the DBpedia knowledge base. Previous approaches often compared relational and triplestores and, thus, settled on measuring performance against a relational database which had been converted to RDF by using SQL-like queries. In contrast to those approaches, our benchmark is based on queries that were actually issued by humans and applications against existing RDF data not resembling a relational schema. Our generic procedure for benchmark creation is based on query-log mining, clustering and SPARQL feature analysis. We argue that a pure SPARQL benchmark is more useful to compare existing triplestores and provide results for the popular triplestore implementations Virtuoso, Sesame, Apache Jena-TDB, and BigOWLIM. The subsequent comparison of our results with other benchmark results indicates that the performance of triplestores is by far less homogeneous than suggested by previous benchmarks.

Further, one of the crucial tasks when creating and maintaining knowledge bases is validating their facts and maintaining the quality of their inherent data. This task include several subtasks, and in thesis we address two of those major subtasks, specifically fact validation and provenance, and data quality The subtask fact validation and provenance aim at providing sources for these facts in order to ensure correctness and traceability of the provided knowledge This subtask is often addressed by human curators in a three-step process: issuing appropriate keyword queries for the statement to check using standard search engines, retrieving potentially relevant documents and screening those documents for relevant content. The drawbacks of this process are manifold. Most importantly, it is very time-consuming as the experts have to carry out several search processes and must often read several documents. We present DeFacto (Deep Fact Validation), which is an algorithm for validating facts by finding trustworthy sources for it on the Web. DeFacto aims to provide an effective way of validating facts by supplying the user with relevant excerpts of webpages as well as useful additional information including a score for the confidence DeFacto has in the correctness of the input fact. On the other hand the subtask of data quality maintenance aims at evaluating and continuously improving the quality of data of the knowledge bases. We present a methodology for assessing the quality of knowledge bases' data, which comprises of a manual and a semi-automatic process. The first phase includes the detection of common quality problems and their representation in a quality problem taxonomy. In the manual process, the second phase comprises of the evaluation of a large number of individual resources, according to the quality problem taxonomy via crowdsourcing. This process is accompanied by a tool wherein a user assesses an individual resource and evaluates each fact for correctness. The semi-automatic process involves the generation and verification of schema axioms. We report the results obtained by applying this methodology to DBpedia.