Squiggle: a Semantic Search Engine at work

Irene Celino, Andrea Turati, Emanuele Della Valle and Dario Cerizza
CEFRIEL – Politecnico of Milano, Via Fucini 2, 20133 Milano, Italy
{irene.celino, andrea.turati, emanuele.dellavalle, dario.cerizza}@cefriel.it

Abstract

We present Squiggle*, a Semantic Web framework that eases the deployment of semantic search engines.*

Search engines are becoming such an easy way to find textual resources that we wish to use them also for multimedia content; however, syntactic techniques, even if promising, are not up to the task. With Squiggle *we prove that Semantic Web technologies provide real benefits to end users in terms of an easier and more effective access to information. The effectiveness of our approach is fully demonstrated by real-world deployments available on the web.*

1 Introduction and motivations

Searching everything everywhere is becoming our habit when we need to find something. However, *finding* what we need is often a hard job. Current search engine technology is very good in finding complete Web pages, but it lacks the desired precision[1] and recall[2] when searching for multimedia resources. For instance, searching "jaguar" in an image search engine results in a mix of felines and cars, which are difficult to tell apart.

Squiggle [2] is an extensible semantic search framework for the development of semantic search engines. By adding a conceptual flavor to the crawling and the indexing of resources, Squiggle can exploit ontological elements to improve and enrich searching time, without undermining the user experience. These features, together with the employment of SKOS [3] model, make Squiggle a powerful and reusable framework to build engines with both syntactic and semantic functionality.

2 Searching with Squiggle

The interaction of a final user with Squiggle is intuitive and very similar to the use of a traditional search engine; however, the results are better and more meaningful. In the following we provide examples of searches with Squiggle, explaining how it works.

1st step: Syntactic search and query analysis
The user is presented with a simple search form in which he can insert some keywords. Squiggle, like syntactic search engines, can immediately retrieve all results containing those keywords, since part of Squiggle is based on the well known search engine library Lucene[3] [6]. For example, a ski fan could search for images of "Herminator", the famous Austrian athlete Hermann Maier; using Squiggle, however, not only the user obtains syntactically-matching results (images tagged with the word "herminator"), but his query is also analyzed in order to identify its *meaning*. This is possible because of Squiggle *disambiguation* capabilities: the search engine can access an ontology of the domain (e.g., an ontology of the athletes in the skiing domain) and try to identify the concepts that could have some connections with the query (in the previous example, Squiggle identifies "herminator" as a nickname, an alternative label for Hermann Maier).

2nd step: Semantic search
The results of the previous *disambiguation* phase are displayed, together with the syntactic results, in a lateral box under the heading "Did you mean...?". Therefore, the user can manually choose the *meaning* of his query among the proposed ones. In response, he obtains more precise and numerous results; this is possible because, during the contents indexing, Squiggle is able to *semantically annotate* those resources with regards to the domain ontology. In the previous example, during the *conceptual indexing* phase, all the images whose syntactic annotations contained both Hermann Maier complete name (the concept's `skos:prefLabel`) and his nickname "herminator" (represented with a `skos:altLabel` relation) were annotated with his *concept* URI.

3rd step: Semantic suggestions
But there's more: accessing the domain ontology, Squiggle is able to exploit all its content, i.e. not only the alternative

[1]Precision is the proportion of relevant data of all data retrieved.
[2]Recall is the proportion of retrieved relevant data, out of all available relevant data.
[3]`http://lucene.apache.org/`

labels of its concepts, but also their relations. This capability lets Squiggle expand the user query, by following the relations between the concepts identified in it and other ontological elements, and propose to the user possible searches of his interest. For example, a fan of the Queen band, looking for audio files of their songs, could be presented with a lateral box suggesting an expansion of his query to include results related to Freddy Mercury (who could be in relation with the Queen band by a `skos:relatedPartOf` property in an ontology of the music domain). This *meaning suggestion*, as well as the disambiguation phase described before, is possible because Squiggle accesses a semantic repository built on Sesame[4] [1] that contains the domain ontology.

3 Squiggle real-world deployments

To prove our approach, we built two different domain-specific search engines with the Squiggle framework. The following sections explain some of their distinctive characteristics.

3.1 Squiggle Ski

CEFRIEL, as Official Supplier of Applied Academic Research for Torino 2006 Olympic Winter Games[5], caught the opportunity to demonstrate Squiggle by developing Squiggle Ski, which helps the international public in finding images of the athletes involved in the alpine skiing races.

In order to instantiate Squiggle Ski, we built a SKOS-based domain ontology partially by hand, developing a small multilingual taxonomy of the disciplines in the sectors of alpine skiing, and partially by collecting information on the FIS-Ski web site[6]. Then we built an experimental focused crawler that exploits the knowledge in the ski ontology to collect images of skiers from sport news Web sites all over the world. The awareness about all relevant terms (names of athletes, disciplines, places, etc. with possible alternative labels in different languages) helps both the focused crawler to filter the appropriate photos and the conceptual indexer to semantically annotate them before the indexing process.

Squiggle Ski is freely available at `http://squiggle.cefriel.it/ski`.

3.2 Squiggle Music

Squiggle Music is an instantiation of Squiggle framework that indexes audio files (mainly mp3 files) enriching them with information about artists, song titles and music genres.

We created a SKOS-based ontology by merging two freely available meta-databases: MusicMoz [4] and MusicBrainz [5]. For each song, MusicBrainz offers its TRM id[7], which is an audio fingerprinting unique for an audio track. Using tools like QuickNamer[8], that is able to calculate the TRM of a file, and searching in MusicBrainz database for matching, it's possible to put an mp3 file in relation with the song's metadata. Therefore, combining the *smart data* from MusicBrainz and MusicMoz with a *smart machine* like QuickNamer, we built an automatic semantic annotator that acts as a domain-dependent plug-in of Squiggle framework during the *conceptual indexing* phase. This annotator is therefore able to add to each file all its metadata (artist, song title, etc.).

Squiggle Music is on-line at `http://squiggle.cefriel.it/music`.

4 Conclusions

We enlightened how the employment of Semantic Web technologies to the development of search engines provides real benefits to end users, enabling an easier and more effective access to information; a semantic search engine like Squiggle appears to be more usable, in that users are supported with semantic "suggestions", as our test-beds demonstrate at a glance.

Moreover, we designed Squiggle keeping in mind the particular needs of searching when dealing with multimedia contents: the extensible nature of Squiggle fully enables the joint employment of smart machines to process media and domain ontologies.

References

[1] Arjohn Kampman, Frank van Harmelen, and Jeen Broekstra. Sesame: A generic architecture for storing and querying rdf and rdf schema. *in proceedings of ISWC 2002, October 7 - 10, Sardinia, Italy*, 2002.

[2] CEFRIEL Semantic Web Activities group. Squiggle home. `http://squiggle.cefriel.it/`, 2005.

[3] A. Miles and D. Brickley. SKOS Core Guide, W3C Working Draft. `http://www.w3.org/TR/swbp-skos-core-guide`, 2 November 2005.

[4] MusicBrainz. `http://www.musicbrainz.org/`, 2005.

[5] MusicMoz. `http://www.musicmoz.org/`, 2005.

[6] Otis Gospodnetic and Erik Hatcher. *Lucene in action*. Manning Publications, 2004.

[4] `http://openrdf.org/`

[5] `http://www.cefriel.it/press/olimpiadi2006.html`

[6] `http://www.fis-ski.com/`

[7] `http://www.relatable.com/tech/trm.html`

[8] `http://phonascus.sourceforge.net/`