Technology Transfer in Computing Systems

D3.24: Individual TTP24 abstract

Project no.:	609491
Funding scheme:	Collaborative project
Start date of the project:	1st September 2013
Duration:	36 months
Work programme topic:	FP7-ICT-2013-10
Deliverable type:	Report
Deliverable reference number:	ICT-609491 / D3.24
WP and tasks contributing:	WP 3 / all
Due date:	30/04/2016
Actual submission date:	03/05/2016
Responsible Organization:	CIT UPC
Dissemination Level:	Public
Revision:	1.0

TETRACOM D3.24: EnrichData: Providing richer search environments for search engines

Jordi Urmeneta (Sparsity Technologies) - Joan Guisado-Gámez, David Tamayo-Domènech, Josep Lluís Larriba-Pey (DAMA-UPC)

Nowadays, all the institutions, most of large and small business and many people have their own websites, as it has be- come one of the most common ways to disseminate information. However, the process of searching for information in each of those sites can be a tedious task for users who often obtain a “No results found” message. It may happen that, despite the message, the site has information about the topic the user is looking for, but the vocabulary used in the website is different from the user’s. This phenomenon is called vocabulary mismatch and it is common in the usage of natural language processes. Also, the topic inexperience of the users, which is caused by the lack of familiarity with the vocabulary, entails that not all the interesting documents of the site are retrieved.

Query rewriting techniques aim at improving the results achieved by the user search by means of introducing new terms, commonly called expansion features and/or removing terms from the original query. Thus, the challenge is to select those expansion features that are capable of improving the results the most. However, it is difficult for institutions, small business or people to have the technology to implement such techniques. As a response to this

need, specialized companies in information retrieval have become third parties that offer search solutions. For example, Google Search Appliance (GSA) is an integrated, all-in-one hardware and software, that pro- vides Google search technology for organizations. However, this technology is thought and designed for large organizations that can afford it. Moreover, for GSA to exploit its full potential, and to retrieve qualitative results, it is suggested to manually create files of customized expansion terms for the specific vocabulary of the site1 .

Previous research [1] [2] has shown that the graph structure of Wikipedia, which consists of articles and categories related to each other, encodes relevant information, which allows extracting reliable expansion features. We present ENRICH, which is a collaborative task between academia and industry to take advantage of previous research findings. ENRICH is a query rewriting system service that specializes its expansions for each particular website. It uses Wikipedia as a generic knowledge base (KB) out of which it derives a website-specific knowledge base (WS-KB), the structure of which is exploited to identify strongly related concepts that are good candidates to be used as expansion features.

The main goal of ENRICH is to improve the search experience of users, offering a query rewriting service in the cloud that is based on Wikipedia and really easy for webmasters to integrate it in their sites.

ENRICH specializes its expansions for each site as op- posed to general query rewriting techniques, which offer generic solutions independently of the website topic and vocabulary. For that purpose, ENRICH analyzes each website and uses Wikipedia to identify its entities, which are defined as the real world concepts. It also identifies the way they are referred in the website. Notice that the same entity can have several names, for example, car, auto, automobile are alter- native names for the same entity. We call the set of entities that appear in the website, website entities, and their names (those that are used in the website), appearance names.

Notice that search engines only will retrieve documents if the user’s query matches any of the appearance names. To increase the hit rate of the search engine, ENRICH automatically builds a website-customized rewriting file that allows translating the user queries into a set of appearance names. In order to do that, ENRICH follows two strategies: First, for each website entity, it finds the rest of its names besides its appearance names. Second, for each website entity it finds a set of strongly related entities, so that, their names can be translated into the appearance names of the website entity. To illustrate this second strategy, imagine the scenario in which car is a website entity, but vehicle is not (i.e. there is no website page in which it appears). Since car and vehicle represent two strongly related entities, EN- RICH would translate the latter into the former in a way that the search engine could retrieve car-about pages. In order to follow this strategy,

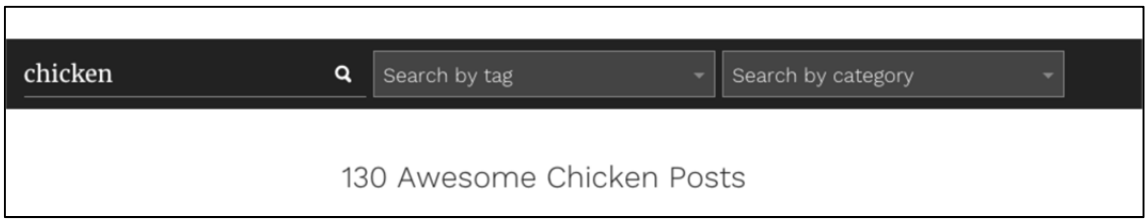
ENRICH uses Wikipedia to build, for each website, a specific knowledge base (WS-KB). Then, the structure of the WS-KB is analyzed to identify strongly related entities.

As an example of ENRICH capabilities, we have applied it to http://iamafoodblog.com. We show two examples of ENRICH rewriting a query for this site:

- Query 1 (Q1): Rooster ENRICH query: Chicken.
- Query 2 (Q2): Sausage ENRICH query: Sausage, hot dog, chorizo, Chinese sausage, sausage roll, merguez,...

In Q1 the user is looking for posts that talk about roosters. However, the website does not contain any post that uses that particular term, therefore, the search engine re- turns a “No results found” message as depicted in Figure 1a.

(a) User query: rooster



(b) ENRICH query: chicken

Figure 1: ENRICH in http://iamafoodblog.com

Thanks to ENRICH, the query is rewritten as chicken, which allows the search engine to return 130 posts as shown in Figure 1b. This example shows that ENRICH is capable of overcoming the vocabulary mismatch problem. In Q2, the user is looking for posts talking about sausages. Although there are up to 31 posts that talk about sausages, the results can be improved if they are combined with those obtained by more specific queries, such as hot dog, chorizo, which is a Spanish sausage, and merguez, which is a typical sausage from Maghreb, etc. This situation shows a scenario of topic inexperience that ENRICH is capable of over- coming by adding strongly related website entities’ names.

Notice that the use of ENRICH is completely transparent for website users, who are not conscious, in any case, of the system working for the particular website they are querying. A user would simply introduce the query in a typical search box, as the ones depicted in Figure 1, and the website would return the results. Nonetheless, to make ENRICH work properly, the webmaster has to modify the website code of its site to integrate it. The modifications are minor and consist in capturing the user’s query and sending it to ENRICH via a REST API. Once ENRICH receives a request, it identifies the entities within the user’s query, accesses the web-customized rewriting file, and returns the corresponding appearance names. The result is in the form of a JSON text that contains 2 fields, the appearance names that are explicitly in the user’s query, and the set of appearance names that are introduced due to the analysis of its WS-KB. It is the responsibility of the webmaster to use the names in the returned JSON to send the rewritten query to the search engine.

ENRICH Architecture

In Figure 2 we schematically show the architecture behind ENRICH. We distinguish three main blocks, which consist in i) loading the Wikipedia graph, ii) building the WS-KB and iii) analyzing it. In the rest of this section we explain in detail each of these blocks.

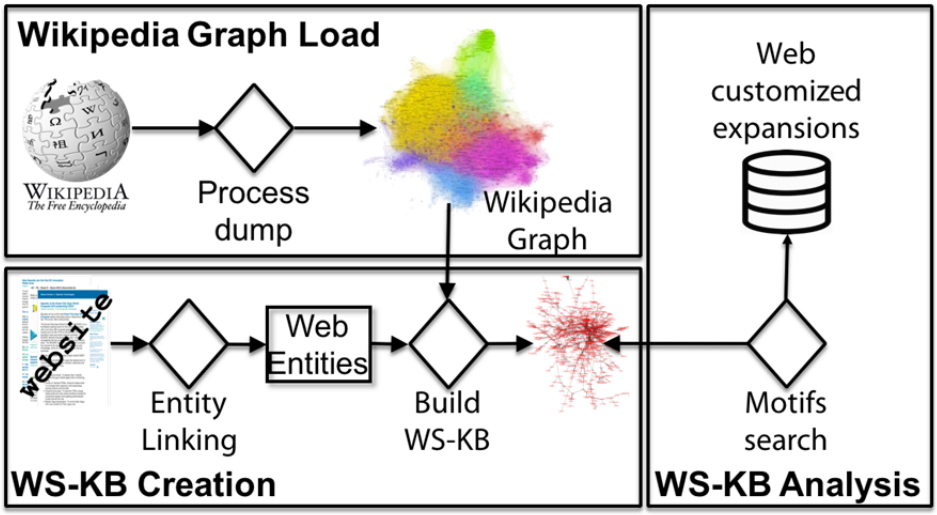


Figure 2: Enrich architecture.

Wikipedia Graph Load

The goal of this block is to load Wikipedia into a Graph Database Management System (GDBMS) to easily exploit its structural properties. For that purpose, we need to parse the Wikipedia dump to obtain i) article ids and titles, ii) category ids and names, iii) article redirections, iv) links among articles, v) links among categories and vi) links among articles and categories. For that purpose, we have developed WikiParser[1], which is a tool that parses the English Wikipedia to CSV.

WS-KB Creation

This block consists in building the specific knowledge base for each site and identifying the strongly related articles. Notice that although Wikipedia acts as a generic KB in the form of a graph, each WS-KB is a subgraph of Wikipedia that includes the web's topics.

WS-KB Analysis

To identify the tightly linked articles in Wikipedia, we base our proposal on [2] where we analyzed relevant structures in Wikipedia that allow relating those articles that are close semantically with no need of any linguistic analysis.

Bibliography

[1] J. Guisado-Gámez, D. Domíguez-Sal und J.-L. Larriba-Pey, Massive query expansion by exploiting graph knowledge bases for image retrieval, Glasgow: ICMR, 2014.

[2] J. Guisado-Gámez und A. Prat-Pérez, Understanding graph structure of wikipedia for query expansion, Melbourne: GRADES, 2015.

[1] https://github.com/DAMA-UPC/WikiParser