The Workshops of the Tenth International AAAI Conference on Web and Social Media
Wiki: Technical Report WS-16-17

ENRICH: A Query Rewriting Service Powered by Wikipedia Graph Structure — Extended Abstract

Joan Guisado-Gámez
DAMA-UPC
Universitat Politècnica de Catalunya
joan@ac.upc.edu

David Tamayo-Domènech
DAMA-UPC
Universitat Politècnica de Catalunya
tamayo@ac.upc.edu

Jordi Urmeneta
Sparsity-Technologies
Barcelona
urmeneta@sparsity-technologies.com

Josep-Lluís Larriba-Pey
DAMA-UPC
Universitat Politècnica de Catalunya
larri@ac.upc.edu

Abstract

The search for relevant information in websites can be very frustrating for users who, unintentionally, use too general or inappropriate keywords to express their requests. To overcome this situation, query rewriting techniques aim at transforming the users requests to better describe the real intent of the users. However, to the best of our knowledge, current search tools either are too generic or require resources not available for everyone such as query log processors, natural language engines, etc. To supply this need, we present ENRICH, which is a query rewriting cloud service that is automatically tailored to each website and it is powered by an available, accessible and open resource: Wikipedia.

1 Introduction

Nowadays, all the institutions, most of large and small business and many people have their own websites, as it has become 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) (Google^{TM} 2016) is an integrated, all-in-one hardware and software, that provides 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 site[1].

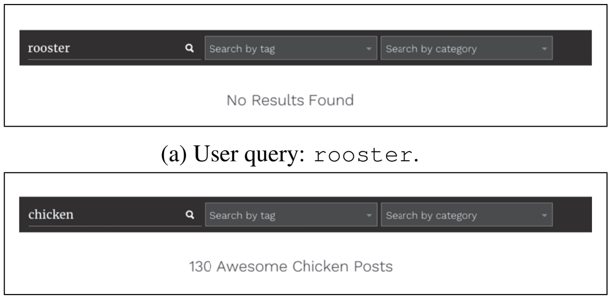
Previous research (Guisado-Gámez, Dominguez-Sal, and Larriba-Pey 2014; Guisado-Gámez and Prat-Pérez 2015; Guisado-Gámez, Prat-Pérez, and Larriba-Pey 2016) has shown that the graph structure of Wikipedia, which consists of articles and categories related to each other, encodes relevant information, which allows extracting reliable expansion features. Thanks to the support of the EU-Tetracom initiative, in this paper 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 rest of the paper is organized as follows. In Section 2 we give an overview of ENRICH. In Section 3 we provide details about the architecture behind ENRICH. Finally, in Section 4 we conclude and outline our future work.

2 ENRICH Overview

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.

[1] https://goo.gl/oQ2YoR.



rooster

No Results Found

(a) User query: rooster.

chicken

130 Awesome Chicken Posts

(b) ENRICH query: chicken.

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

ENRICH specializes its expansions for each site as opposed 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 alternative 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, ENRICH 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 returns a "No results found" message as depicted in Figure 1a. 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 overcoming 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.

In Snippet 1 we show a piece of the code that webmasters could use to capture the user's query and to send it to ENRICH. The user's query is the input of the function. In line 3, ENRICH is called by specifying its URL (enrichserver), the website id (11346) and its password (pwd). Once the function returns the rewritten query, line 7, the expansion features (finalQuery.expFeat) are sent to the search engine, in line 7.

```
1 $scope.queryExpansion = function(input){
2   $http({ method:'GET',
3         url: 'https://enrichserver/queryExpansion/11346/pwd',
4         params: {query:input}}).then(
5         function successCallback(response){
6           $scope.finalQuery = response;
7           $scope.search($scope.$finalQuery.expFeat); });
8   }
```

Snippet 1: JavaScript function that calls ENRICH.

3 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.

3.1 Wikipedia Graph Load

The goal of this block is to load Wikipedia into a Graph Database Management System (GDBMS) to easily exploit

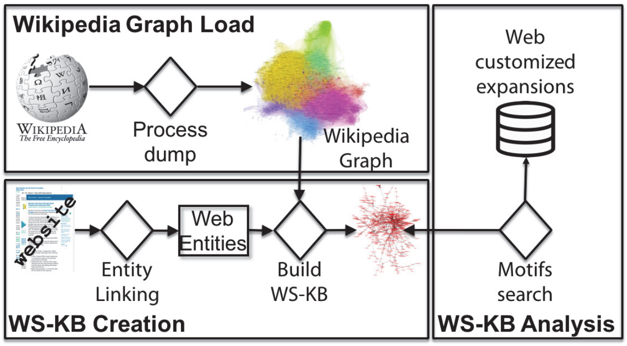


Figure 2: ENRICH architecture.

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 [2], which is a tool that parses the English Wikipedia to CSV. It requires i) pages-articles.xml, ii) page.sql and iii) categorylinks.sql Wikipedia's dump files to create 6 CSV files, each of which contains the information previously described. Notice that since ENRICH is based on Wikipedia's structural properties, the body of the articles is not required.

Then, we use the files to load Wikipedia into Sparksee (Martínez-Bazan and et al. 2012), which is a GDBMS that allows performing complex operations efficiently. To load the data, we discard all the relations with the hidden categories, which are a special kind of categories that are concerned with the maintenance of Wikipedia, rather than being part of the content of the encyclopedia. In our experience, the English Wikipedia, without the body of the articles, loaded in Sparksee requires 11Gb of disk.

This process is carried on whenever it is needed it, depending on the updates of Wikipedia that affect its overall structure. However, despite of the high frequency of updates in Wikipedia articles, and although, a more exhaustive analysis is required, we believe that the main structure of Wikipedia does not change dramatically in each dump.

3.2 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.

In order to create the WS-KB, first, we need to identify the website entities, and match them with the corresponding articles of the Wikipedia graph. Note that an entity is a concept, and its materialization in the Wikipedia graph is an article. We call the materialization of the website entities in the graph, **website articles**. For that purpose, we use Dexter (Ceccarelli et al. 2013) which is an open source project that actually recognizes entities in a given text and matches them with Wikipedia articles, providing its corresponding id, for each website entity. According to our experiments, which consisted in identifying and linking the entities in the queries of 3 datasets (ImageCLEF, CHiC 2012 and CHiC 2013), Dexter successes in 96% of the occasions. In this process, we also annotate its appearance name.

Second, although, the website articles constitute the core of the WS-KB, it is required that it contains more articles and categories, otherwise, we could not relate the appearance names of the website entities to other entities that do not appear in the website and their names. This would prevent ENRICH from overcoming the vocabulary mismatch problem. Our current proposal consists in building the WS-KB with the website articles, their redirects [3], their linked articles, their categories and the articles that belong to those categories. Note that we use the Wikipedia graph to identify the edges among the nodes and add them into the WS-KB.

The process of building the WS-KB is done the first time a webmaster installs ENRICH and each time that he/she considers that it is required to modify the web-customized rewriting file.

3.3 WS-KB Analysis

To identify the tightly linked articles in Wikipedia, we base our proposal on (Guisado-Gámez and Prat-Pérez 2015), where we analyzed relevant structures in Wikipedia that allow relating those articles that are close semantically with no need of any linguistic analysis. The analysis revealed that cycles (defined as a closed sequence of nodes, either articles or categories, with at least one edge among each pair of consecutive nodes) were important and relevant to relate them. Summarizing the characteristics that let us differentiate good from bad cycles, we have that:

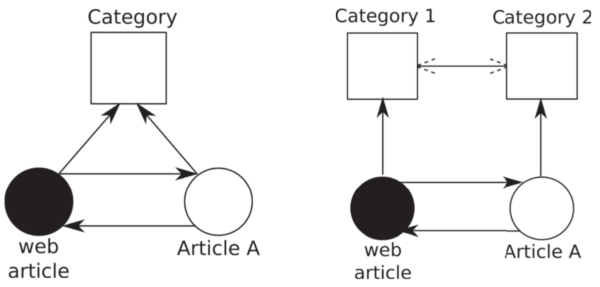
- Cycles of length 2 are not reliable.
- Cycles of length 3, 4 and 5 are to be trusted to reach articles that are strongly related with the website articles.
- Around a third of the nodes of cycles have to be categories. This ratio is expected to increase beyond the cycles of length 5.
- The expansion features obtained through the articles of dense cycles are capable of leading to better results.

Based on these characteristics we propose the motifs depicted in Figures 3a and 3b, which are based on cycles of length 3 and 4 respectively. The motif depicted in Figure 3a is called, from now on, **triangular motif**, while the one depicted in Figure 3b is called **square motif**. In the figures, the square nodes are categories, while round nodes are articles. The black round node is a website article, while the white round node is an article A, a new article selected as it forms a motif with the website article.

In the triangular motif we force the website article to be doubly linked with article A. That means that the website

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

[3] If the website article is a main article, we add all the redirects of this article, if it is a redirect articles, we add the corresponding main article, and also all its redirects.



(a) Triangular motif.

(b) Square motif.

Figure 3: Expansion motifs.

article actually links, in Wikipedia, to A, and A links, reciprocally, to the website article. Moreover, article A must belong to, at least, one of the categories of the website article. This structure guarantees a strong relation between the website article and article A. In the square motif of Figure 3b, the website article and article A must be also doubly linked. However, compared to the triangular motif, it is just required that at least one of the categories of the website article is linked with one of the categories of A, or *vice versa* (depicted with a dashed arrow in Figure 3b). This pattern still guarantees a strong relation between the website article and article A, but it is not as restrictive as the one represented in Figure 3a. Both patterns are chosen because they make sense from an intuitive point of view. It is expected that doubly linked articles that are also connected through categories, are also close semantically (although the system has not done any kind of semantic analysis), and the title of one can serve as an expansion feature of the other. Also because these cycles fulfill the edge density and ratio of categories requirements. To decide the length of the cycles that we base the motifs on, we ignore those of length 2, as they resulted not to be trustful to identify strongly related articles. Larger cycles, as those with a length larger or equal to 5, have been also avoided for performance reasons. The traversal of larger cycles expands too much the search space in the WS-KB, and would make it difficult to identify them in a reasonable time for query rewriting processes. Moreover, our previous results in (Guisado-Gámez, Prat-Pérez, and Larriba-Pey 2016) show that the motifs depicted in Figure 3 allow up to 150% improvement.

Given a website article, ENRICH identifies all its strongly related articles as those other articles that share, at least, one motif. This allows relating its appearance names (which we annotated at the beginning of the process) represented by a website article, with a set of articles, each of which have a title, and that may have several redirect articles. The article and redirects titles are used as the set of names recognized by ENRICH and that are translated into the appearance names. This constitutes the web-customized rewriting file.

Notice that in order to fulfill the performance requirements of a system like ENRICH, the access to the rewriting file must be done as fast as possible. For that purpose, we load this file into an indexed and in-memory structure that relates each recognized name (all the names of all the entities represented in the WS-KB) with the corresponding appearance name. Notice that the structure required only to represent the entities in Q1 and Q2 would consist of 10 entities, 138 recognized names (all the names of these entities) and 7 appearance names.

The process of analyzing the WS-KB is done always after the WS-KB is created, under webamaster's demand.

4 Conclusion & Future Work

In this paper we have presented the main ideas behind ENRICH, a query rewriting system. ENRICH differs from other current software in two main aspects: first, it specializes its query rewrites for each website, second, it uses the Wikipedia structure to identify the expansion features taking into account the website vocabulary. Most of the research regarding to Wikipedia is based on improving the methods for extracting and using its content. However, in previous research, we showed that exploiting exclusively the structure of Wikipedia, without using any kind of language analysis, allows achieving remarkable results.

ENRICH is still in its development phase, thus, we cannot provide any experimental result in this paper. Nonetheless, in our previous results (Guisado-Gámez, Prat-Pérez, and Larriba-Pey 2016), which did not include the rewriting index, we have shown that queries are rewritten in less than 0.2s, and achieved up to 150% improvements. We expect to drastically reduce the rewriting time due to the index. We also expect to improve the results due to the WS-KB tailored to each particular website.

Acknowledgments

The members of DAMA-UPC thank the Ministry of Science and Innovation of Spain and Generalitat de Catalunya, for grant numbers TIN2009-14560-C03-03 and SGR2014-890 respectively. The members of DAMA-UPC and Sparsity-Technologies also thank the EU-Tetracom for grant agreement number 609491.

References

Ceccarelli, D.; Lucchese, C.; Orlando, S.; Perego, R.; and Trani, S. 2013. Dexter: an open source framework for entity linking. In *ESAIR*, 17–20.

Google^{TM}. 2016. *Google Search Appliance, http://google.com/enterprise/gsa.*

Guisado-Gámez, J., and Prat-Pérez, A. 2015. Understanding graph structure of wikipedia for query expansion. In *GRADES*, 6:1–6:6.

Guisado-Gámez, J.; Dominguez-Sal, D.; and Larriba-Pey, J. 2014. Massive query expansion by exploiting graph knowledge bases for image retrieval. In *ICMR*, 33.

Guisado-Gámez, J.; Prat-Pérez, A.; and Larriba-Pey, J. 2016. Query expansion via structural motifs in wikipedia graph. *CoRR* abs/1602.07217.

Martínez-Bazan, N., and et al. 2012. Efficient graph management based on bitmap indices. In *IDEAS*, 110–119.