A parametric architecture for tags clustering in folksonomic search engines

Nicola Raffaele Di Matteo
Dept. of Computer Science
University of Bologna
Bologna, Italy
dimatteo@cs.unibo.it

Silvio Peroni
Dept. of Computer Science
University of Bologna
Bologna, Italy
speroni@cs.unibo.it

Fabio Tamburini
Dept. of Linguistics and Oriental Studies
University of Bologna
Bologna, Italy
fabio.tamburini@unibo.it

Fabio Vitali
Dept. of Computer Science
University of Bologna
Bologna, Italy
fabio@cs.unibo.it

Abstract— **Semantic search engines rely on the existence of a rich set of semantic connections between the concepts associated to documents and those used for the queries. With folksonomies, this is not always guaranteed. Creating clusters of folksonomic tags around terms of controlled ontological vocabularies is a potentially sophisticated approach, but algorithms abound for this clustering and no clear cut winner exists. In this paper we introduce FolksEngine, a parametric search engine for folksonomies allowing to specify any clustering algorithm as a three step process: the user's query is expanded according to semantic rules associated to the terms of the query, the new query is then executed on the plain folksonomy search engine, and the results are ranked according to semantic rules associated to the folksonomic tags actually used for the documents.**

Keywords: FolksEngine; folksonomies; query expansion; semantic search engine

I. INTRODUCTION

In the last years, a lot of works on document search has been trying to introduce new approaches for searching based on the exploitation of semantic data. Usually, this kind of search is performed through content analysis, that is obviously imprecise, or by retrieving and analysing annotations added to documents.

These annotations could be added by classification professionals, able to categorize and describe the documents using rich and well-organized criteria, and yet hard to master, or by the collective action of a non-professional audience that describes documents in a free of constraint approach, in particular without using controlled and non-ambiguous vocabularies, i.e. by defining *folksonomies*.

Obviously, folksonomies do not generate a useful structure of concepts associated to the document content. Therefore, they introduce well-known problems such as terms' ambiguity or synonymity, that make it hard to offer a fully shareable and synthetic view of the content of the documents.

Literature exists that offers *a posteriori* semantic structuring of terms used in folksonomies, mostly based on ontologies, and contextualize and explain the terms that are used by folksonomies. But it is clear that is not possible to impose a precise semantic to them exactly because of their main benefit: they allow the collective making of complex vocabularies exactly by ignoring the issues of ambiguity and synonymicity.

An alternative path to imposing firm and definite semantics to terms used in folksonomies is to rather loosely associate them to well-defined terms in a predefined ontology, i.e to cluster folksonomic tags actually used in practice around concepts placed in a clear and well-defined semantic structure.

In this paper we present an architecture and a prototype for developing and testing different clustering approaches in order to verify their different effectiveness. To do that we need a flexible architecture, one that allows separating the essential and shared features necessary for clustering, and specifying case by case different clustering policies depending on the algorithm we are testing.

Furthermore, we want to understand how NLP techniques can be used to the analysis and clustering of folksonomic tag, applying well-studied algorithms originally created for wide document collections to the restricted domain defined by folksonomies.

The rest of the paper follows this structure: in section 2 we will introduce the current related research about query expansion and semantic enrichment. In section 3 we will talk about the key principles used in developing our framework, presented in section 4. Finally, we will introduce our ideas for future developments of folksonomic clustering algorithms (section 5), and provide the conclusions of this paper in section 6.

II. RELATED WORKS

Literature exists about two key aspects related to document annotation and retrieval: the semantic expansion

of queries and the semantic enrichment of flat structures of terms.

Various approaches can be pursued in order to expand the user query to enrich it with different tags or concepts.

The first approach uses lexical resources (such as WordNet [1]) and its ontological organization in order to extract concepts related to the terms inserted in the query. WordNet semantic relation such as synonymy, implicitly expressed in the *synsets* definition, hyponymy (type-subtype relation) and meronimy (part-whole relation) can be successfully used to add to the original query further concepts more or less directly related to those inserted by the user. The list of concepts connected to the original query can be expanded also considering measures of semantic relatedness between the lexical elements in WordNet [2].

The second approach to query expansion starts from a radically different view: instead of relying on a given lexical resource, concepts are expanded through the use of statistical and machine learning techniques applied on huge quantities of text resources. Words and multi-word expressions can be compared by means of distributional similarity measures and clustered accordingly using various techniques. See for example, [4] [5] [6] [7] [8] for methods based on latent semantic analysis, or n-gram extraction from large text corpora.

Some tools, frameworks and techniques have been developed to try to enrich in some way a particular tag set given in input. For example, the *FoLksonomy Ontology enRichment* tool (*FLOR*) [1] performs an automatic semantic enrichment of folksonomies, without any user contribution, using a three steps procedure that guarantees a lexical analysis, a thesaurus-based semantic expansion and a final semantic enrichment through ontologies.

III. PRINCIPLES

In this paper we introduce FolksEngine, a search engine framework created to exploit methods for clustering folksonomic tags around the controlled terms of a ontological thesaurus. The basic principles on which our framework is built impact on both design and theoretical aspects.

First of all, the flexibility of the infrastructure represent the main feature we wanted to guarantee. Our goal is to test a large number of clustering algorithms working on folksonomies in very different ways, in order to understand the effectiveness of each of them depending on the particular context considered.

So, we need to add easily new algorithms simply by extending the appropriate internal framework structures, without modifying the application core. Moreover, we need to be able to expand the current knowledge base with new folksonomic repositories and ontological thesaurus, in order to have new and up-to-date data and to guarantee an accurate analysis and evaluation of each tested algorithm.

Grounded on the basic design principle of flexibility, we have designed a general framework that allows development of folksonomy-based search engines organized around the following three steps:

- the user's query is expanded into multiple keywords depending on the algorithm specified;
- the new set of keywords is given as input to a general folksonomic search engine;
- the results are sorted by an algorithm, parametrically specified, depending on a particular ranking function.

The theoretical aspect we refer to concerns the query expansion technique, developed as first tentative for guaranteeing an acceptable document retrieval result.

Such approach is based on the semantic expansion of the query, according to semantic relations available in the thesaurus and optimized in order to avoid ambiguity among terms. For instance, rather than using different space-separated keywords, we suggest the use of "keytag:keyword" pairs in which the first term represents a sort of category for the latter. Through this, we expand the query *t:w* as *t:w, t w, t' w, t'' w, ... , t^n w*, where *t' ... t^n* are terms in semantic relation (synonymy, hyponymy, hyperonymy) with *t*.

These two fundamental principles are the basis of our framework called *FolksEngine*, derived from the contraction of the words "folksonomy" and "search engine".

IV. FRAMEWORK

The core of FolksEngine is composed of a set of cooperating classes that take care of different aspects of the documents search and retrieves from folksonomies.

The user's application is obtained by extending the framework core classes. Such user-defined classes contain only the specific problem description, while the framework deals with the relationships between classes and with their interactions. The core classes of the framework are organized following the Model - View – Controller pattern.

Our framework describes a classical information retrieval flow with a particular variant: the introduction of a semantic expansion of the user's query. Algorithms to expand queries – e.g., using semantic relations, clustered similarity, etc – are defined by the programmer simply by extending a core class. Both the application flow and the semantic distance algorithm (used for ranking the search results) are controlled by particular classes, easily extendible depending on the particular needs.

Our framework also provides facilities for semantic search engine using in WordNet in any part of the algorithms. The entire Wordnet thesaurus database is integrated using WordNet Sql Builder[2], and it is accessible directly from the framework. This is especially useful for the query expansion.

Obviously, it is possible to give access to other dictionaries, simply by creating classes either extending the service giving access to the database of terms relations or the class that connects to services on the Web.

By extending the *Model* abstract class we have created a search engine accessing directly Delicious[3]. The HTML

[1] http://wordnet.princeton.edu/

[2] http://wnsqlbuilder.sourceforge.net/

[3] http://delicious.com/

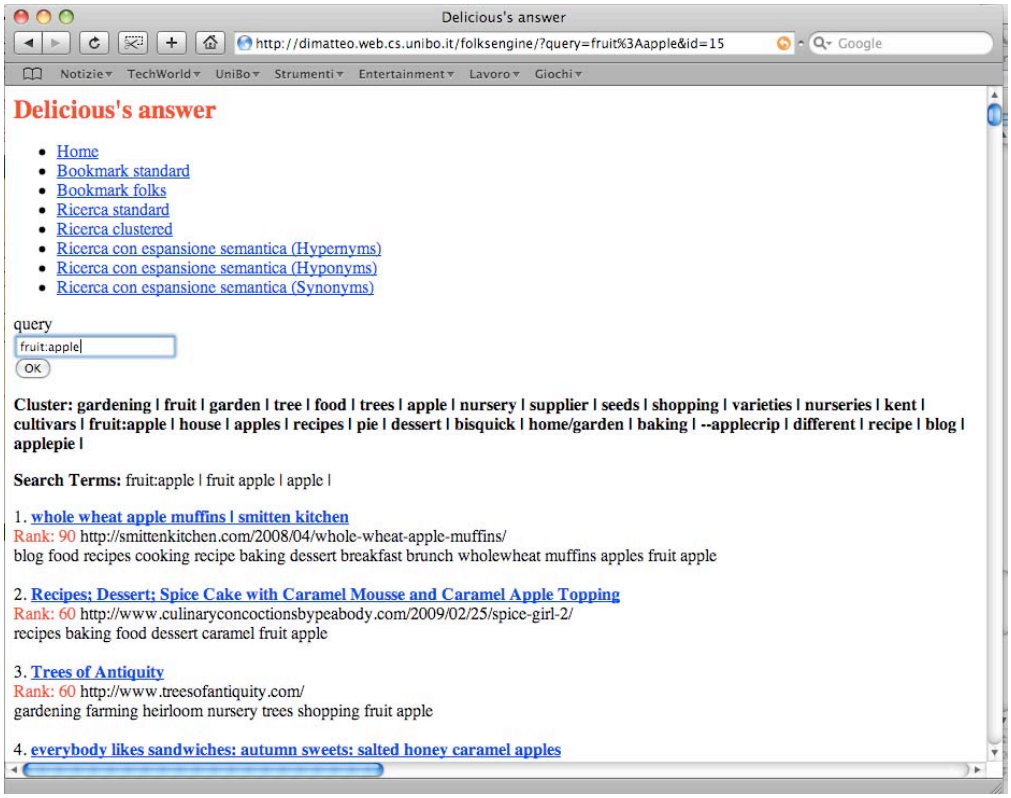

Fig. 1: the interface of the FolksEngine application

pages returned by Delicious are cleaned and transformed into a well-formed XML document, and it is then possible to use XPath to retrieve the data sought.

All the classes just introduced are instantiated in the main. It consists essentially of the following steps:

- instantiation of the search engines;
- instantiation of pages and forms;
- creation of the content array;
- instantiation of the controller and the viewer;
- calling of the controller.

The framework controller listens for requests from the user (i.e., inputs the queries) and dispatches them to the framework model. Once retrieved, the results are returned to the controller and finally shown by the viewer. Thus the typical process execution is "read the query, expand it, execute the search, rank the result".

The framework is developed as a server-side application, in order to have a better access to databases, parsers and other applications. It uses a pure object-oriented paradigm: the presence of code outside the developed classes is limited, while global variables are never used. It is developed in PHP5, in order to give the best portability in a Web context.

In fig. 1 we show the search engine expanding on the user's query "apple:fruit".

V. FUTURE WORKS

As we have seen before, in order to extract more specific data as a result of a user query, we need to expand the original query by adding further words, or better-structured concepts, that are related to the keywords and keytags specified in the query.

Finding words that are related together by some linguistic relation or that exhibit similar distributional behaviour is a field of research in NLP that is actively explored in the last few years.

The first line of research we aim to develop inside our framework concerns the use of lexical resources that are based on a formal ontology, not just WordNet, but also FrameNet[4], PropBank[5], and so on, and navigate through the lexical relations coded into the ontology to find concepts somehow related to the user's query.

Some basic relations, such as synonymy and hyponymy, have already been experimented with in our framework, but more complex metrics, for instance based on semantic relatedness between words and concepts (e.g., the lowest super-ordinate in taxonomic relation or overlapping measures between word definitions), can be used fruitfully to identify concepts that are related to the terms of the original query.

[4] http://framenet.icsi.berkeley.edu/

[5] http://verbs.colorado.edu/~mpalmer/projects/ace.html

Lexical resources can be very useful for these tasks, but, unfortunately, the availability of such resources is very limited, often restricted to the most common languages, mainly English, and, even in these cases, their coverage can not be adequate to allow a complete and useful expansion of the terms inserted by the user.

Beside the techniques described, we plan to implement a number of methodologies that, starting form a large amount of text corpora, such as the British National Corpus for English[6] or the CORIS/CODIS for Italian[7], try to identify the words that exhibit a similar distributional behaviour of the user's terms.

The first approach is grounded on the Latent Semantic Analysis firstly introduced by Deerwester, et al. [3]: representing the occurrence of each word as a vector of words co-occurring with it in the same context and collecting all vectors in a [word × context] matrix A, we can apply a singular value decomposition to A to map it into a reduced subspace. Each word is then mapped into a k-dimensional vector and can be compared to other word-vectors to find similarities using, for example, the cosine distance. The closer the vectors in the k-dimensional space, the most similar are the distributional behaviors of two words, thus appropriate clustering methods can identify sets of similar words.

Elaborating these methods as proposed, for example, by Widdows and Dorow [7], we can build small graphs for each target word showing the most similar words in a structured way and showing also the different senses the target word can have in real texts (see fig. 2).

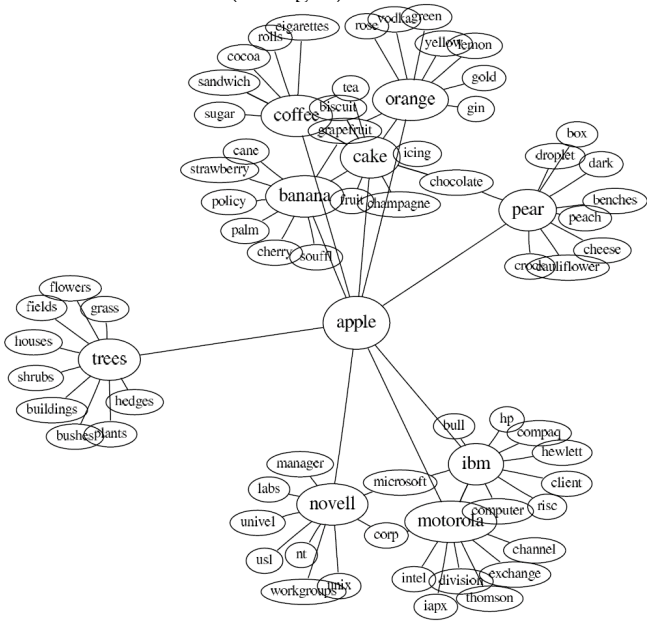

Fig. 2: Automatically generated graph showing the word apple and semantically related nouns. Note that the different senses of the word tend to cluster in different graph regions (figure taken from [7]).

A second methodology for driving the terms' expansion that we plan to implement in our system rely on a dependency-parsed corpus [4]. Triples (word, syntactic relation, word) are extracted and various statistical and stochastic measures are computed on them in order to define a similarity measure between two words. Again, a clustering algorithm relying on such similarity measures can group words exhibiting similar distributional and syntactic behaviour.

As outlined before, the research in this field of NLP community is very active, and a lot of alternative methods and techniques can be devised and successfully applied to this task.

VI. Conclusion

In this paper we have introduced a flexible PHP framework for folksonomy-based document search, actually developed as beta version and usable with a web browser at http://dimatteo.web.cs.unibo.it/folksengine.

Our ongoing studies concern the development of a systematic approach for the evaluation of different clustering algorithms, in order to prove in some concrete way the quality of the search results returned by each algorithms, and the implementation of other more sophisticated algorithms for query expansions and result ranking.

References

[1] S. Angeletou, "Semantic Enrichment of Folksonomy Tagspaces", Proc. of International Semantic Web Conference (ISWC'08), Doctoral Consortium. Karlsruhe, Germany, 2008.

[2] A. Budanitsky, G. Hirst, "Evaluating WordNet-based Measures of Lexical Semantic Relatedness", Computational Linguistics, vol. 32, pp. 13–47, 2006.

[3] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, R. Harshman, "Indexing by Latent Semantic Analysis", Journal of the American Society for Information Science, vol. 41, pp. 391–407, 1990.

[4] D. Lin , "Automatic retrieval and clustering of similar words", Proc. of the 17th international conference on Computational linguistics, Montreal, pp. 768–774, 1998.

[5] A. Purandare, T. Pedersen, "SenseClusters - Finding Clusters that Represent Word Senses", Proc. of the HLT-NAACL 2004: Demonstration Papers, Boston, MA, pp. 26-29, 2004.

[6] R. Sinha, R. Mihalcea, "Unsupervised Graph-based Word Sense Disambiguation Using Measures of Word Semantic Similarity", Proc. of the IEEE International Conference on Semantic Computing (ICSC 2007), Irvine, CA, pp. 363-369, 2007.

[7] D. Widdows, B. Dorow, "A Graph Model for Unsupervised Lexical Acquisition", Proc. of the 19th International Conference on Computational Linguistics, Taipei, pp. 1093-1099, 2002.

[8] D. Widdows, Geometry and Meaning. CSLI Publication, 2004

[6] http://www.natcorp.ox.ac.uk/

[7] http://sciling.dslo.unibo.it/accesso_coris_eng.html