CISWeb 2008

ESWC 2008 Workshop on Collective Semantics:

Collective Intelligence & the Semantic Web

http://mklab.iti.gr/CISWeb

June 2, 2008
Tenerife, Spain

Preface

This volume includes the papers presented at the 1st International Workshop on "Collective Semantics: Collective Intelligence & the Semantic Web (CISWeb 2008)", which was hosted by the *5th European Semantic Web Conference (ESWC-08)*, in Tenerife, Spain, June 2nd, 2008.

Web 2.0 technologies have introduced new information sharing practices which favor mass users participation and aim at improving quality of information content and information organization. It is challenging to dynamically capture knowledge that emerges as the outcome of the interactions of masses of users in social networks, since difficulties are posed by the heterogeneous data sources, the large information scale and the huge amount of information postings. Semantic Web may contribute by providing language basis, structuring help from distributed ad-hoc ontologies, and by offering new ways of exploring the information space.

In this context, CISWeb 2008 Workshop has attracted very interesting work which covers crucial and emerging research topics such as using and enriching ontologies, semantically enhancing folksonomies and webspaces, social data management, , and interrelating Web 2.0 to Semantic Web. More specifically, interesting ideas were presented at the Workshop for ontology matching via knowledge extracted by multiple ontologies, enriching ontological user profiles with tagging history, merging Web 2.0 and the Semantic Web by (semi-) automated content tagging, semantically enriching folsonomies and tagging. Most of these efforts were experimented and validated under popular datasets and testbeds (such as Wikipedia, Flickr, LycosIQ).

There were 11 submissions from 9 countries, and three reviewers were assigned to each paper. The program committee has finally selected 5 regular papers and 3 poster papers for presentation at the workshop. We would like to thank all the program committee members for their dedicated effort to review papers in their area of expertise and on a timely manner. Their effort was valuable to accommodate high quality papers in the CISWeb 2008 program.

The research work presented at CISWeb 2008 was very interesting and exciting and the Workshop involved live discussions and fruitful comments. Moreover, the program included a very interesting invited talk by Prof. Bettina Hoser, from the Universität Karlsruhe, who presented "Information Retrieval versus Knowledge Retrieval: A social network pespective", a topic which is emerging and of wide interest. We are grateful to Prof. Bettina Hoser for her insightful presentation.

Special thanks are ought to Eirini Giannakidou, PhD graduate student from the CERTH Research Institute, for her technical support to CISWeb 2008 organization. The workshop has been held in cooperation with the European Commission and WeKnowIt Integrated Project, and we are indebted for their contributions and financial support.

CISWeb 2008 Co-Chairs

Dr. Yannis Avrithis, National Technical University of Athens, Greece
Dr. Yiannis Kompatsiaris, CERTH-ITI, Greece
Prof. Steffen Staab, University of Koblenz-Landau, Germany
Prof. Athena Vakali, Aristotle University of Thessaloniki, Greece

Conference Organization

Programme Chairs

Yannis Avrithis
Ioannis Kompatsiaris
Steffen Staab
Athena Vakali

Programme Committee

Harith Alani
Andrea Baldassarri
Nick Bassiliades
Susanne Boll
Ciro Cattuto
Thierry Declerck
Ying Ding
William Grosky
Harry Halpin
Andreas Hotho
Paul Lewis
Jose Martinez
Phivos Mylonas
Lyndon Nixon
Noel O'Connor
Raphael Troncy

External Reviewers

Eirini Giannakidou
Gianluca Correndo
Ioannis Katakis
Georgios Meditskos

Author Index

Table of Contents

Information Retrieval vs Knowledge Retrieval: A Social Network Perspective

Bettina Hoser

Information Services and Electronic Markets,
Institute of Information Engineering and Management,
Department of Economics and Business Engineering,
Universität Karlsruhe (TH)
Germany
hoser@iism.uni-karlsruhe.de

Bettina Hoser
Institute for Information services and management
Department for Information Services and Electronic Markets
Universität Karlsruhe (TH)
Kaiserstrasse 12
D-76128 Karlsruhe
Germany

Information Retrieval vs Knowledge Retrieval: A Social Network Perspective

Abstract. Information Retrieval, especially in connection with the internet, is a well known research field. But as the technologies used for the internet become more and more elaborate, so grows the need to not only find (retrieve) already available information, but to generate knowledge. In this paper the emergence of collective intelligence from information retrieval and social network analysis will be presented. The main point will be how information gained from e.g. websites can be enhanced to become knowledge using methods and models from the research field of Social Network Analysis. Such an approach is very context sensitive, so two examples will be presented.

1 Introduction

When is a trend a trend? When 'the right people' initialize it. This is very well known from the world of fashion. In the world of news, research and technology this may translate to the fact that a trend is a trend when 'relevant' people or websites take up the topic. But how can the relevant people or websites be distinguished from the less relevant? How can 'relevant' be defined? How can one detect really 'relevant' trends? 'Relevant' is always a reflective approach. It is dependent on the circumstances. Thus it is, e.g. in the case of fashion or news, a social context.

As an example for the question discussed here take a high tech company (e.g. mobile phones) or a reinsurance company. For both it is essential that they see trends before the competitors or the possible clients see it. In the case of the high tech company, for example, it is crucial to know what the potential customers are interested in, or which features in the current product are not accepted and why. For the reinsurance company, it is necessary to know which hazards, e.g. in health care, are being discussed, so that the company may prepare its policy accordingly. As an illustration of that point take the discussion on obeisity in children and subsequent health problems in adults.

2 Information retrieval

As companies look for ways to find trends as shown above they used to look for example at newspapers. Nowadays the internet with its chat rooms, newsgroups, social networking sites and blogs offers a wide area of information, which had not been accessible before. To gather this information various methods have been devised.

Text analysis is one of the methods often used to extract information from a text source. There is a large body of research literature, see e.g. [FNR03], in the fields of linguistics, information science or classification on diverse ways

to extract keywords, key phrases, etc. from websites and other text sources. In these research fields models have been built to explain how the context sensitive relevance of words, phrases etc. can be defined. Just think about classifications like e.g. the ACM Classification System. Some of these methods lead to lists of possible topics listed by relative relevance according to their usage of phrases in text.

Another approach is to use additional information like keyword or tags to enhance the information retrieved by classifying it. This has grown into the research fields on folksonomies, tagging, semantic web, etc.

At this point though what is known is that these phrases or words are often used. What is not known is who used them. Or to put it precisely, whether the user is a 'relevant' user in the context. This is a question that has been at the heart of the research field of Social Network Analysis.

3 Social Network Analysis

Social Network Analysis (SNA) is a research area that tries to analyze and model actor behavior based on his or her connections or relations to other members of a group. Fur further reference see [WF99]. An actor is thus seen as restricted or empowered by his or her connection to others. The basis of this structural approach is given by models about group interaction. The first research questions were posed to define roles to actors given a social context. Thus e.g. leadership of a group is such a role. There are also models about the power to manipulate. Thus a person in such a context may be called relevant, or central, if he or she is positioned in such a way in the group's network that all information exchanged between any two actors has to pass through this 'central' actor. He or she can thus manipulate the group.

Thus the question of who is relevant within a group is one of the research questions with SNA. Based on graph theory this can be analyzed by using different so called centrality indices. Some of them are inuitive, like e.g. degree centrality, other are more elaborate like e.g. betweenness centrality or eigenvector centrality. But always the question is: given a clearly defined context, who within a group is relevant, who is not, how are the actors in the group connected and what, if any, predicitions can be made for the future deveploment of the group structure.

Thus, this analysis approach can be used to find the 'relevant' people or websites needed to enhance the information found by text retrieval.

4 Knowledge Retrieval

The idea to retrieve knowledge means not only to gather the information available but to enrich it with other information to gain knowledge about a topic. In the case proposed here this means to use results from SNA to enrich the information gathered by text analysis to find whether the topics found by information retrieval are 'really hot topics' because 'relevant people' talk about it, or

whether it is just 'small talk' by 'bystanders'. In a conceptual study [HSGS+07] we used such an approach to look for socially enriched information about mobile phones within a newsgroup.

The idea proposed here is based on following information fusion approach: First a text corpus and a group are defined. Then the text corpus is analyzed and the group structure is evaluated. As a last step these two results are combined to gain knowledge. This is just a very crude and short description of the procedure. One major challenge here is to to define the group. Depending on the area of interest this can be a very large group or a collection of websites corresponding to a group. Sometimes this may not even be a well defined group. Thus biases can be introduced by choice of actors (or websites). But once the group is defined, there is also the question of the appropriate text analysis method. Questions like scalability and validity have to be answered here. As a last step, the interpretation of the combined results have to be validated before any measures should be taken.

But even with regard to the aforementioned challenges this approach seems to yield deeper insights into topics and trends, since it includes the social component of trends.

5 Outlook

The potential of such an approach is very high. Not only are companies interested in such a kind of knowledge gained from different 'news'-sources, weighted by the social impact, but also the average internet user. If one takes a look at communites of diverse interests such as travel or such necesseties as emergencies, it is not only valuable to have information at hand gathered from collective sites, but also to know who gave the information and whether the source can be viewed as 'relevant' in the given context. In the context of emergencies, this may save lives.

References

[FNR03] J. Franke, G. Nakhaeizadeh, and I. Renz, editors. *Text Mining, Theoretical Aspects and Applications.* Advances in soft computing. Physica-Verlag, Heidelberg, 2003.

[HSGS+07] B. Hoser, J. Schröder, A. Geyer-Schulz, M. Viermetz, and M. Skubacz. Topic trend detection in newsgroups. *Künstliche Intelligenz*, (3), 2007.

[WF99] S. Wasserman and K. Faust. *Social Network Analysis*, volume 8 of *Structural Analysis in the Social Sciences.* Cambridge University Press, Cambridge, 1 edition, 1999.

Enriching Ontological User Profiles with Tagging History for Multi-Domain Recommendations

Iván Cantador[1], Martin Szomszor[2], Harith Alani[2],
Miriam Fernández[1], Pablo Castells[1]

[1] Escuela Politécnica Superior
Universidad Autónoma de Madrid
28049 Madrid, Spain
{ivan.cantador, miriam.fernandez, pablo.castells}@uam.es

[2] School of Electronics and Computer Science
University of Southampton
SO17 1BJ Southampton, United Kingdom
{mns2, ha}@ ecs.soton.ac.uk

Abstract. Many advanced recommendation frameworks employ ontologies of various complexities to model individuals and items, providing a mechanism for the expression of user interests and the representation of item attributes. As a result, complex matching techniques can be applied to support individuals in the discovery of items according to explicit and implicit user preferences. Recently, the rapid adoption of Web2.0, and the proliferation of social networking sites, has resulted in more and more users providing an increasing amount of information about themselves that could be exploited for recommendation purposes. However, the unification of personal information with ontologies using the contemporary knowledge representation methods often associated with Web2.0 applications, such as community tagging, is a non-trivial task. In this paper, we propose a method for the unification of tags with ontologies by grounding tags to a shared representation in the form of Wordnet and Wikipedia. We incorporate individuals' tagging history into their ontological profiles by matching tags with ontology concepts. This approach is preliminary evaluated by extending an existing news recommendation system with user tagging histories harvested from popular social networking sites.

Keywords: social tagging, web 2.0, ontology, semantic web, user modelling, recommender systems.

1 Introduction

The increasing proliferation of Web2.0 style sharing platforms, coupled with the rapid development of novel ways to exploit them, is paving the way for new paradigms in Web usage. Virtual communities and on-line services such as social networking, folksonomies, blogs, and wikis, are fostering an increase in user participation, engaging users and encouraging them to share more and more information, resources, and opinions. The huge amount of information resulting from this emerging phenomenon gives rise to excellent opportunities to investigate, understand, and exploit the knowledge about the users' interests, preferences and needs. However, the current infrastructure of the Web does not provide the mechanisms necessary to

consolidate this wealth of personal data since they are spread over many unconnected, heterogeneous sources.

Community tagging sites, and their respective folksonomies, are a clear example of this situation: users have access to a plethora of web sites that allow them to annotate and share many types of resources. For example, they can organise and make photos available on Flickr[1], classify and share bookmarks using del.icio.us[2], communicate and share resources with friends using Facebook[3]. Through personal tags, users implicitly declare different facets of their personalities, such as their favourite book subjects on LibraryThing[4], movie preferences on IMDb[5], music tastes on Last.fm[6], and so forth. Therefore, the domains covered by social tagging applications are both disparate and divergent, creating considerably complex and extensive descriptions of user profiles.

In the current Web2.0 landscape, there is a distinct lack of tools to support users with meaningful ways to query and retrieve resources spread over disparate end-points: users should be able to search consistently across a broad range of sites for diverse media types such as articles, reviews, videos, and photos. Furthermore, such sites could be used to support the recommendation of new resources belonging to multiple domains based on tags from different sites. As a step towards making this vision a reality, we explore the use of syntactic and semantic based technologies for the combination, communication and exploitation of information from different social systems.

In this paper, we present an approach for the consolidation of social tagging information from multiple sources into ontologies that describe the domains of interest covered by the tags. Ontology-based user profiles enable rich comparisons of user interests against semantic annotations of resources, in order to make personal recommendations. This principle has already been tested by the authors in different personalised information retrieval frameworks, such as semantic query-based searching [4], personalised context-aware content retrieval [13], group-oriented profiling [3], and multi-facet hybrid recommendations [2].

We propose to feed the previous strategies with user profiles built from personal tag clouds obtained from Flickr and del.icio.us web sites. The mapping of those social tags to our ontological structures involve three steps: the filtering of tags, the acquisition of semantic information from the Web to map the remaining tags into a common vocabulary, and the categorisation of the obtained concepts according to the existing ontology classes.

An application of the above techniques has been tested in News@hand, a news recommender system which integrates our different ontology-based recommendation approaches. In this system, ontological knowledge bases and user profiles are generated from public social tagging information, using the aforementioned techniques. The News@hand system, along with the automatic acquisition of news articles from the Web, and the automatic semantic annotation of these items using Natural Language Processing tools [1] and the Lucene[7] indexer shall also be described.

---

[1] Flickr, Photo Sharing, http://www.flickr.com/
[2] del.icio.us, Social Bookmark manager, http://del.icio.us/
[3] Facebook, Social Networking, http://www.facebook.com/
[4] LibraryThing, Personal Online Book Catalogues, http://www.librarything.com/
[5] IMDb, Internet Movie Database, http://imdb.com/
[6] Last.fm, The Social Music Revolution, http://www.last.fm/
[7] Lucene, An Open Source Information Retrieval Library, http://lucene.apache.org/

The structure of the paper is the following. Section 2 briefly describes our approach for representing user preferences and item features using ontology-based knowledge structures, and how they are exploited by several recommendation models. Section 3 explains mechanisms to automatically relate and transform social tagging and external semantic information into our ontological knowledge structures. A real implementation and evaluation of the previous tag transformation and recommendation processes within a news recommender system are presented in section 4. Finally, section 5 proclaims some conclusions and future research lines.

2 Hybrid recommendations

In this section, we summarise the ontology-based knowledge representation and recommendation models in which filtered social tags are proposed to be integrated and exploited.

2.1 Ontology-based representation of item features and user preferences

In the knowledge representation we propose [4, 13], user preferences are described as vectors $\mathbf{u}_m = (u_{m,1}, u_{m,2}, ..., u_{m,K})$ where $u_{m,k} \in [0,1]$ measures the intensity of the interest of user $u_m \in \mathcal{U}$ for concept $c_k \in \mathcal{O}$ (a class or an instance) in a domain ontology $\mathcal{O}$, K being the total number of concepts in the ontology. Similarly, items $d_n \in \mathcal{D}$ are assumed to be annotated by vectors $\mathbf{d_n} = (d_{n,1}, d_{n,2}, ..., d_{n,K})$ of concept weights, in the same vector-space as user preferences.

The main advantages of this knowledge representation are its portability, thanks to the XML-based Semantic Web standards, the domain independency of the subsequent content retrieval and recommendation algorithms, and the multi-source nature of the proposal (different types of media could be annotated: texts, images, videos).

2.2 Personalised content retrieval

Our notion of content retrieval is based on a matching algorithm that provides a personal relevance measure $pref(d_n, u_m)$ of an item d_n for a user u_m. This measure is set according to semantic preferences of the user and semantic annotations of the item, and is based on a cosine vector similarity $\cos(\mathbf{d}_n, \mathbf{u}_m)$. The obtained similarity values (Personalised Ranking module of Figure 1) can be combined with query-based scores without personalisation $sim(d_n, q)$ and semantic context information (Item Retrieving module of Figure 1), to produce combined rankings [13].

To overcome the existence of *sparsity* in user profiles, we propose a preference spreading mechanism, which expands the initial set of preferences stored in user profiles through explicit semantic relations with other concepts in the ontology. Our approach is based on Constrained Spreading Activation (CSA), and is self-controlled by applying a decay factor to the intensity of preference each time a relation is traversed. We have empirically demonstrated [3, 13] that preference extension improves retrieval precision and recall. It also helps to mitigate other well-known limitations of recommender systems such as the cold-start, overspecialisation and portfolio effects.

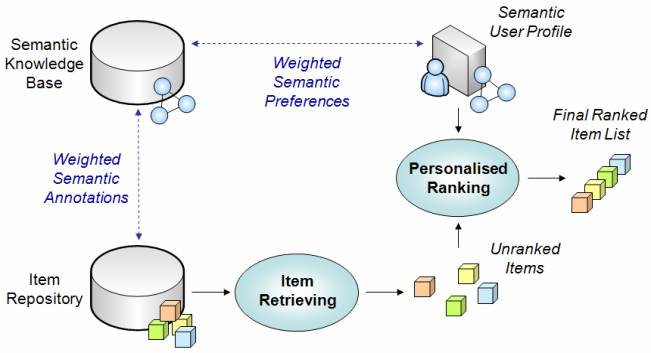

Figure 1. Ontology-based personalised content retrieval

2.3 Context-aware recommendations

The context is represented in our approach [13] as a set of weighted ontology concepts. This set is obtained by collecting the concepts that have been involved in the interaction of the user (e.g. accessed items) during a session. It is built in such a way that the importance of concepts fades away with time by a decay factor. Once the context is built, a contextual activation of user preferences is achieved by finding semantic paths linking preferences to context. These paths are made of existing relations between concepts in the ontologies, following the spreading technique mentioned in section 2.2.

2.4 Group-oriented recommendations

The presented user profile representation allows us to easily model groups of users. We have explored the combination of the ontology-based profiles to meet this purpose [3], on a per concept basis, following different strategies from social choice theory. In our approach, user profiles are merged to form a shared group profile, so that common content recommendations are generated according to this new profile.

2.5 Multi-facet hybrid recommendations

In order to make hybrid recommendations we cluster the semantic space based on the correlation of concepts appearing in the profiles of individual users. The obtained clusters C_q represent groups of preferences (topics of interests) shared by a significant number of users. Using these clusters profiles are partitioned into semantic segments. Each of these segments corresponds to a cluster and represents a subset of the user interests that is shared by the users who contributed to the clustering process. By thus introducing further structure in user profiles, we define relations among users at different levels, obtaining multilayered communities of interest.

Exploiting the relations of the communities which emerge from the users' interests, and combining them with item semantic information, we have presented in [2] several recommendation models that compare the current user interests with those of the others users in a double way. First, according to item characteristics, and second, according to connections among user interests, in both cases at different semantic layers.

$$pref\left(d_n,u_m\right)=\sum_q nsim\left(d_n,C_q\right)\sum_i nsim_q\left(u_m,u_i\right)\cdot sim_q\left(d_n,u_i\right)$$

3 Relating social tags to ontological information

Parallel to the proliferation and growth of social tagging systems, the research community is increasing its efforts to analyse the complex dynamics underlying folksonomies, and investigate the exploitation of this phenomenon in multiple domains. Results reported in [5] suggest that users of social systems share behaviours which appear to follow simple tagging activity patterns. Understanding, predicting and controlling the semiotic dynamics of online social systems are the base pillars for a wide variety of applications.

For these purposes, the establishment of a common vocabulary (set of tags) shared by users in different social systems is a desirable situation. Indeed, recent works have focused on the improvement of tagging functionalities to generate tag datasets in a controlled, coordinated way. P-TAG [6] is a method that automatically generates personalised tags for web pages, producing keywords relevant both to their textual content and to data collected from the user's browsing. In [8], an adaptation of user-based collaborative filtering and a graph-based recommender is presented as a tag recommendation mechanism that eases the process of finding good tags for a resource, and consolidating the creation of a consistent tag vocabulary across users.

The integration of folksonomies and the Semantic Web has been envisioned as an alternative approach to the collaborative organisation of shared tagging information. The proposal presented in [11] uses a combination of pre-processing strategies and statistical techniques together with knowledge provided by ontologies for making explicit the semantics behind the tag space in social tagging systems.

In the work presented herein, we propose the use of knowledge structures defined by multiple domain ontologies as a common semantic layer to unify and classify social tags from several Web 2.0 sites. More specifically, we propose a mechanism for the creation of ontology instances for the gathered tags, according to semantic information collected from the Web. Tagging information is linked to ontological structures by our method through a sequence comprising three processing steps:

- *Filtering social tags*: To facilitate the integration of information from different social sources as well as the subsequent translation of that information into ontological knowledge, a pre-processing of the tags is needed, associating them to a common vocabulary, shared by the different involved applications. Morphologic and semantic transformations of tags are performed at this stage based on the WordNet English dictionary [9], the Wikipedia[8] encyclopaedia and the Google[9] web search engine.
- *Obtaining semantic information about social tags*: The shared vocabulary is created with the use of Wikipedia, which provides semantic information about millions of concepts.
- *Categorisation of social tags into ontology classes*: Once the tags have been filtered and mapped to a shared vocabulary, they are automatically converted into instances of classes of domain ontologies. Again, semantic categorisation information available in Wikipedia is exploited in this process.

These steps are explained in more detail in the next subsections.

[8] Wikipedia, The Free Encyclopaedia, http://en.wikipedia.org/
[9] Google, Web Search Engine, http://www.google.com/

3.1 Filtering social tags

Raw tagging information can be noisy and inconsistent. When manual tags are introduced with a non-controlled tagging mechanism, people often make grammatical mistakes (e.g. *barclona* instead of *barcelona*), tag concepts indistinctly in singular, plural or derived forms (*blog*, *blogs*, *blogging*), sometimes add adjectives, adverbs, prepositions or pronouns to the main concept of the tag (*beautiful car*, *to read*), or use synonyms and acronyms that could be converted into a single tag (*biscuit* and *cookie*, *ny* and *new york*). Moreover, the tag encoding and storage mechanisms used by social systems often alter the tags introduced by the users: they may transform white spaces (*san francisco*, *san-francisco*, *san_francisco*, *sanfrancisco*) and special characters in the tags (*los angeles* for *los ángeles*, *zurich* instead of *zürich*), etc.

Thus, while it is possible to gather information from multiple folksonomy sites, such as Flickr or del.icio.us, inconsistency will lead to confusion and loss of information when tagging data is compared. For example, if a user has tagged photos from a recent holiday in New York with nyc, but also bookmarked relevant pages in del.icio.us with new_york, the correlation will be lost. In order to facilitate the folksonomy data analysis and integration, tags have to be filtered and mapped to a shared vocabulary. Here, we present a tag filtering architecture that makes use of external knowledge resources such as the WordNet dictionary, Wikipedia encyclopaedia and Google web search engine.

The filtering process is a sequential execution where the output from one filtering step is used as input to the next. The output of the entire filtering process is a set of new tags that correspond to an agreed representation. As will be explained below, this is achieved by correlating tags to entries in two large knowledge resources: Wordnet and Wikipedia. Wordnet is a lexical database and thesaurus that group English words into sets of cognitive synonyms called synsets, providing definitions of terms, and modelling various semantic relations between concepts: synonym, hypernym, hyponym, among others. Wikipedia is a multilingual, open-access, free-content encyclopaedia on the Internet. Using a wiki style of collaborative content writing, is has grown to become one of the largest reference Web sites with over 75,000 active contributors, maintaining approximately 9,000,000 articles in over 250 languages (as of February 2008). Wikipedia contains collaboratively generated categories that classify and relate entries, and also supports term disambiguation and dereferencing of acronyms.

Figure 2 provides a visual representation of the filtering process where a set of raw tags are transformed into a set of filtered tags and a set of discarded tags. Each of the numbers in the diagram corresponds to a step outlined below.

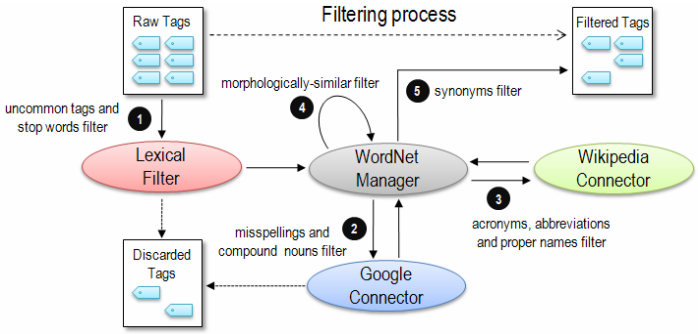

Figure 2. The tag filtering process

For this work, tags from public available user accounts from Flickr and del.icio.us sites have been collected and filtered. A total of 1004 user profiles have been gathered from these two systems, providing 149,529 and 84,851 distinct tags respectively. Initially, the intersection between both datasets was 28,550 common tags.

Step 1: Lexical filtering

After raw tags have been harvested from different folksonomy sites, they are passed to the *Lexical Filter*, which applies several filtering operations. Tags that are too small (with length = 1) or too large (length > 25) are removed, resulting in a discarding rate of approximately 3% of the initial dataset. In addition, considering the discrepancies in the use of special characters (such as accents, dieresis and caret symbol), we convert such special characters to a base form (e.g., the characters à, á, â, ã, ä, å are converted to a).

Tags containing numbers are also filtered based on a set of custom heuristics. For example, to maintain salient numbers, such as dates (*2006*, *2007*, etc), common references (*911*, *360*, *666*, etc), or combinations of alphanumeric characters (*7 up*, *4 x 4*, *35 mm*), we discard unpopular tags below a certain global tag frequency threshold. Finally, common stop-words, such as pronouns, articles, prepositions and conjunctions are removed. After lexical filtering, tags are passed on to the *Wordnet Manager*. If a tag has an exact match in Wordnet, we pass it on directly to the set of filtered tags, to save further unnecessary processing.

Step 2: Compound nouns and misspellings

If a tag is not found in Wordnet, we consider possible misspellings and compound nouns. Motivated by [11], to solve these problems, we make use of the Google "did you mean" mechanism. When a search term is entered, the Google engine checks whether more relevant search results are found with an alternative spelling. Because Google's spell check is based on occurrences of all words on the Internet, it is able to suggest common spellings for proper nouns that would not appear in a standard dictionary.

The Google "did you mean" mechanism also provides an excellent way to resolve compound nouns. Since most tagging systems prevent users from entering white spaces into the tag value, users create compound nouns by concatenating nouns together or delimiting them with a non-alphanumeric character such as _ or -, which introduces an obvious source of complication when aligning folksonomies. By sending compound nouns to Google, we easily resolve the tag into its constituent parts. This mechanism works well for compound nouns with two terms, but is likely to fail if more than two terms are used. For example, the tag *sanfrancisco* is corrected to *san francisco*, but the tag *unitedkingdomsouthampton* is not resolved by Google.

We have thus developed a complementary algorithm that quickly and accurately splits compound nouns of three or more terms. The main idea is to firstly sort the tags in alphabetical order, and secondly process the generated tag list sequentially. By caching previous lookups, and matching the first shared characters of the current tag string, we are able to split it into a prefix (previously resolved by Google) and a postfix. A second lookup is then made using the postfix to seek further possible matches. The process is iteratively repeated until no splits are obtained from our *Google Connector*. Compared to a bespoke string-splitting heuristic, this process has a very low computational cost. This mechanism successfully recognizes long compound

nouns such as *war of the worlds*, *lord of the rings*, and *martin luther king jr*.

Similarly to Step 1, after using Google to check for misspellings and compound nouns, the results are validated against the *Wordnet Manager*. Unprocessed tags are added to the pending tag stack, and unmatched tags are discarded.

Step 3: Wikipedia correlation

Many of the popular tags occurring in community tagging systems do not appear in grammar dictionaries, such as Wordnet, because they correspond to proper names (such as famous people, places, or companies), contemporary terminology (such as *web2.0* and *podcast*), or are widely used acronyms (such as *asap* and *diy*).

In order to provide an agreed representation for such tags, we correlate tags to their appropriate Wikipedia entries. For example, when searching the tag *nyc* in Wikipedia, the entry for New York City is returned. The advantage of using Wikipedia to agree on tags from folksonomies is that Wikipedia is a community-driven knowledge base, much like folksonomies are, so that it rapidly adapts to accommodate new terminology.

Apart from consolidating agreed terms for the filtered tags, our *Wikipedia Connector* retrieves semantic information about each obtained entry. Specifically, it extracts ambiguous concepts (e.g., "java programming language" and "java island" for the entry "java"), and collaboratively generated categories (e.g., "living people", "film actors" and "american male models" for the entry "brad pitt"). This information is exploited by the ontology population and annotation processes described below.

Step 4: Morphologically similar terms

An additional issue to be considered during the filtering process is that users often use morphologically similar terms to refer to the same concept. One very common example of this is the no discrepancy between singular and plural terms, such as *blog* and *blogs*, and other morphological deviations (e.g. *blogging*). In this step, using a custom singularisation algorithm, and the stemming functions provided by the Snowball library[10], we reduce morphologically similar tags to a single tag. For each group of similar tags, the shortest term found in Wordnet is used as the representative tag.

Step 5: WordNet synonyms

When people communicate a certain concept, they often use synonyms, i.e., terms that have the same meaning, but with different morphological forms. A natural filtering step is the simplification of the tag sets by merging pairs of synonyms into single terms.

WordNet provides synonym relations between synsets of the terms. However, due to ambiguous meanings of the tags, not all of them can be taken into consideration, and the filtering process must be very carefully executed. Our merging process comprises three stages. In the first stage, a matrix of synonym relations is created by using Wordnet. In the second stage, according to the number of synonym relations found for each tag, we identify the non-ambiguous synonym pairs, and finally, stage three replaces each of the synonym pairs by the term that is most popular. Examples of thus processed synonym pairs are *android* and *humanoid*, *thesis* and *dissertation*, *funicular* and *cable railway*, *stein* and *beer mug*, or *poinsettia* and *christmas flower*.

[10] Snowball, String-handling Language, http://snowball.tartarus.org/

3.2 Obtaining semantic information about social tags

In order to populate ontologies with concepts associated to the filtered social tags, general multi-domain semantic knowledge is needed. In this work, as mentioned before, we propose to extract that information from Wikipedia. The Wikipedia articles describe a number of different types of entities: people, places, companies, etc., providing descriptions, references, and even images about the described entities.

Many of these entities are ambiguous, having several meanings for different contexts. For instance, the same tag "java" could be assigned to a Flickr picture of the Pacific island, or a del.icio.us page about the programming language. One approach to address tag disambiguation is by using the information available in Wikipedia. A Wikipedia article is fairly structured: the title of the page is the entity name itself (as found in Wikipedia), the content is divided into well delimited sections, and a first paragraph is dedicated to possible disambiguation options for the corresponding term. For example, the page of the entry "apple" starts as follows:

- "This article is about the *fruit*…"
- "For the *Beatles multimedia corporation*, see…"
- "For *the technology company*, see…"

Apart from these elements, every article contains a set of collaboratively generated categories. Hence, for example, the categories created for the concept "teide" are: world heritage sites in spain, tenerife, mountains of spain, volcanoes of spain, national parks of spain, stratovolcanoes, hotspot volcanoes, and decade volcanoes. Processing somehow the previous information, we might infer that "teide" is a volcano in Spain.

Disambiguation and categorisation information have been therefore extracted from Wikipedia for every concept appearing in our social tag datasets. Once the most suitable category for a term is determined, we match its relevant categories to classes defined in the domain ontologies, as explained next.

3.3 Categorisation of social tags into ontology classes

The assignment of an ontology class to a Wikipedia entry is based on a morphologic matching between the name and the categories of the entry, and the names of the ontology classes. The ontology classes with most similar names to the name and categories of the entry are chosen as the classes whereof the corresponding individual (instance) is to be created. The created instances are assigned a URI containing the entry name, and are given RDFS labels with the Wikipedia categories.

To better explain the proposed matching method, let us consider the following example. Let "brad pitt" be the concept we wish to instantiate. If we look up this concept in Wikipedia, a page with information about the actor is returned. At the end of the page, several categories are shown: "action film actors", "american film actors", "american television actors", "best supporting actor golden globe (film)", "living people", "missouri actors", "oklahoma (state) actors", "american male models", etc.

After retrieving that information, all the terms (tokens) that appear in the name and categories of the entry (which we will henceforth refer to as entry terms) are morphologically compared with the names of the ontology classes (assuming that a class-label mapping is available, as it is usually the case). Computing the Levenshtein distance, and applying singularisation and stemming mechanisms, only the entry terms that match

some class name, above a certain distance threshold, are kept, and the rest are discarded. For instance, suppose that "action", "actor", "film", "people", and "television" are the ones sufficiently close to some ontology class. To select the most appropriate ontology class among the matching ones, we firstly create a vector whose coordinates correspond to the filtered entry terms, taking as value the number of times the term appears in the entry name and categories together. In the example, the vector might be as follows: {(action, 1), (actor, 6), (film, 3), (people, 1), (television, 1)}, assuming that "actor" appears in six categories of the Wikipedia entry "brad pitt", and so forth.

Once this vector has been created, one or more ontology classes are selected by the following heuristic:

1. If a single coordinate holds the maximum value in the vector, we select the ontology class that matches the corresponding term.
2. In case of a tie between several coordinates having the maximum value, a new vector is created, containing the matched classes plus their taxonomic ancestor classes in the ontologies. Then the weight of each component is computed as the number of times the corresponding class is found in this step. Finally, the original classes that have the highliest valued ancestor in the new vector are selected.

Here "ontology class" and "ancestor" denote a loose notion admitting a broad range of taxonomic constructs, ranging from informally built subject hierarchies (such as the ones defined in the Open Directory tree or, in our experiments, the IPTC Subjects), to pure ontology classes in a strict Description Logic sense.

In our example, the weight for the term "actor" is the highest, so we select its matching class as the category of the entry. Thus, assuming that the class matching this term was "Actor", we finally define "Brad Pitt" as an instance of "Actor".

Now suppose that, instead, the vector for Brad Pitt was {(actor, 1), (film, 1), (people, 1)}. In that case, there would be a tie in the matching classes, and we would apply the second case of the heuristic. We take the ancestor classes, which could be e.g. "cinema industry" for "actor", "cinema industry" for "film", and "mammal" for "person", and create a weighted list with the original and ancestor classes. Then we count the number of times each class appears in the previous list, and create the new vector: {(actor, 1), (film, 1), (person, 1), (cinema industry, 2), (mammal, 1)}. Since the class "cinema industry" has the highest weight, we finally select its sub-classes "actor" and "film" as the classes of the instance "brad pitt".

We must note that our ontology population mechanism does not necessarily generate individuals following a strict semantic "is-a" schema, but a more relaxed semantic "is-related-to" association principle. This is not a problem for our final purposes in personalised content retrieval, since the annotation and recommendation methods in that area are themselves rooted on models of inherently approximated nature, e.g. regarding the relationships between concepts and item contents.

4 Preliminary evaluations

Recent works show an increasing interest in using social tagging information to enhance personalised content retrieval and recommendation. FolkRank [7] is a search algorithm that exploits the structure of folksonomies to find communities and organise search results. The recommender system presented in [10] suggests web pages available on the Internet, by using folksonomy and social bookmarking information. The movie

recommender proposed in [12] is built on keywords assigned to movies via collaborative tagging, and demonstrates the feasibility of making accurate recommendations based on the similarity of item keywords to those of the user's rating tag-clouds.

In the following, we present and preliminary evaluate how our ontological knowledge representation, recommendation models, and tag filtering and matching strategies are integrated in News@hand, a news recommender system.

4.1 News@hand

News@hand is a news recommender system that describes news contents and user preferences with a controlled and structured vocabulary, using semantic-based technologies, and integrating the recommendation models described in section 2. Figure 3 depicts how ontology-based item descriptions and user profiles are created and exploited by the system.

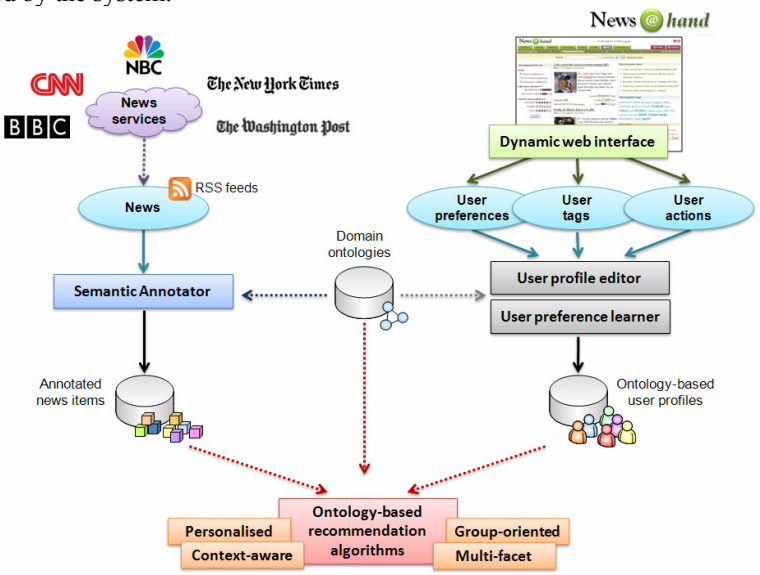

Figure 3. Architecture of News@hand

News items are automatically and periodically retrieved from several on-line news services via RSS feeds. The title and summary of the retrieved news are annotated with concepts of the domain ontologies. A dynamic graphic interface allows the system to automatically retrieve all the users' inputs in order to analyse their behaviour with the system, update their preferences, and adjust the recommendations in real time.

Figure 4 shows a screenshot of a typical news recommendation page in News@hand. The news items are classified into eight different sections: headlines, world, business, technology, science, health, sports and entertainment. When the user is not logged in the system, s/he can browse any of the previous sections, but the items are listed without any personalised criterion. On the other hand, when the user is logged in the system, recommendation and profile edition functionalities are activated, and the user can browse the news according to his and others' preferences in different ways. Click history is used to detect the short term user interests, which represent the dynamic semantic context exploited by our personalised content retrieval mechanism.

The terms occurring in the title and summary that are associated to semantic annotations of the contents, the user profile, and the current context are highlighted with different colours. A collaborative rating is shown on a 0 to 5 star scale, and two coloured bars indicate the relevance of the item for the profile and the context. The user has the possibility adding comments, tags and ratings to the article. S/he also can set parameters for single or group-oriented recommendations, such as the activation or deactivation of his/her individual preferences, those of his/her contacts and/or all other users, the weight that the dynamic context should have over the profile, and the weight of multiple rating criteria.

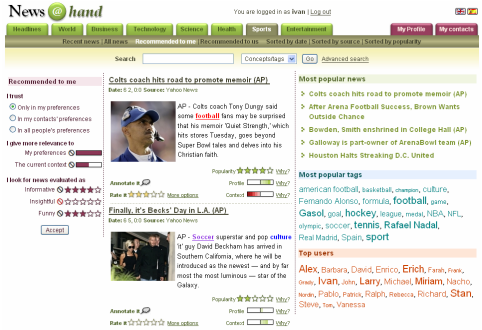

Figure 4. Item recommendation page of News@hand

4.2 Knowledge base

A total of 17 ontologies have been used for the current version of the system. They are adaptations of the IPTC ontology[11], which contains concepts of multiple domains such as education, culture, politics, religion, science, technology, business, health, entertainment, sports, weather, etc. They have been populated with semantic information about the tags we extracted from Flickr and del.icio.us web sites, applying the population mechanism explained in Section 3. A total of 137,254 Wikipedia entries were used to populate 744 ontology classes with 121,135 instances. Table 1 describes the characteristics of the obtained knowledge base.

In order to evaluate the ontology population process, we asked 20 users to randomly select, and manually assess 25 instances of each ontology. They were undergraduate and PhD students of our department, half of them with experience on ontological engineering. They were requested to declare whether each instance was assigned to its correct class, to a less correct class but belonging to a suitable ontology, or to an incorrect class/ontology. The table shows the average accuracy values for all the users considering correct class and correct ontology assignments.

These preliminary results demonstrate the feasibility of our ontology population mechanism. The average accuracy for class assignment is 69.9%, and the average accuracy for ontology assignment arises to 84.4%. Improvements in our mapping heuristics can be investigated. Nevertheless, we presume they are good enough for our recommendation goals. In general, the main common concepts are correctly instantiated, and the effect of an isolated incorrect annotation in a news item is mitigated by the domain/s of the rest of the correct annotations.

[11] IPTC ontology, http://nets.ii.uam.es/mesh/news-at-hand/news-at-hand_iptc-kb_v01.zip

Table 1. Number of classes and instances of News@hand KB, and average population accuracy

Ontology	#classes	#instances	Avg. #instances/class	Avg. accuracy
arts, culture, entertainment	87	33,278	383	78.7 / 93.3
crime, law, justice	22	971	44	62.7 / 73.3
disasters, accidents	16	287	18	74.7 / 84.0
economy, business, finance	161	25,345	157	69.3 / 80.0
education	20	3,542	177	57.5 / 76.7
environmental issues	41	20,581	502	72.0 / 85.3
health	26	1,078	41	65.3 / 89.3
human interests	6	576	96	64.0 / 84.0
labour	6	133	22	70.7 / 78.7
lifestyle, leisure	29	4,895	169	72.0 / 90.7
politics	54	3,206	59	60.0 / 81.3
religion, belief	31	3,248	105	84.0 / 90.7
science, technology	50	7,869	157	68.0 / 86.7
social issues	39	8,673	222	70.7 / 85.3
sports	124	5,567	45	72.0 / 86.7
unrests, conflicts, wars	23	1,820	79	61.3 / 80.0
weather	9	66	7	69.7 / 89.5
	744	121,135	163 (avg.)	**69.9** / **84.4**

4.3 Semantic annotation of news

News@hand periodically retrieves news items from the websites of well-known news media, such as BBC, CNN, NBC, The New York Times, and The Washington Post. These items are obtained via RSS feeds, and contain information of published news articles: their title, summary of content, publication date, hyperlinks to the full texts and related on-line images. The system analyses and automatically annotates the textual information (title and summary) of the RSS feeds (Figure 5).

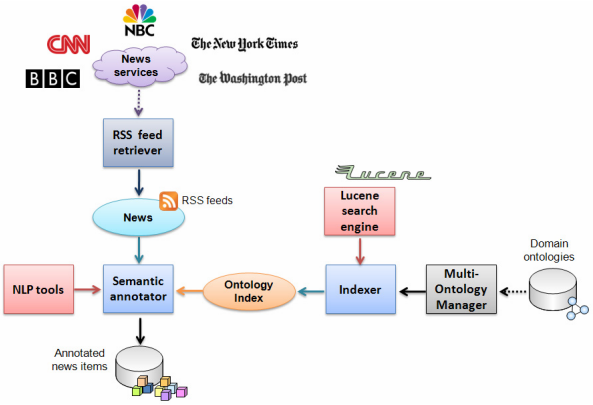

Figure 5. Automatic RSS feeds extraction and semantic annotation processes in News@hand

Using a set of Natural Language Processing tools [1], an annotation module removes stop words and extracts relevant (simple and compound) terms, categorised according to their Part of Speech (PoS): nouns, verbs, adjectives or adverbs. Then, nouns are morphologically compared with the names of the classes and instances of the domain ontologies. The comparisons are done using an ontology index created with Lucene, and

according to fuzzy metrics based on the Levenshtein distance. For each term, if similarities above a certain threshold are found, the most similar semantic concept (class or instance) is chosen and added as an annotation of the news item. After all the annotations are created, a TF-IDF based technique computes and assigns them weights.

For 2 months, since 1st January 2008, we have been daily gathering RSS feeds. A total of 9,698 news items were stored. For this dataset, we run our semantic annotation mechanism, and a total of 66,378 annotations were obtained. Table 2 shows a summary of the average number of annotations per news item generated with our system. Similarly to the experiments conducted for our ontology population strategy, we asked the 20 students to evaluate 5 news items from each of the 8 topic sections of News@hand, giving ratings with values from 0 to 10. The annotation accuracies for each topic are also presented in the table. An average accuracy of 74.8% was obtained.

Table 2. Average number of annotations per news item, and average annotation accuracies

	headlines	**world**	**business**	**technology**	**science**	**health**	**sports**	**entertainment**
#news items	2,660	2,200	1,739	303	346	803	603	1,044
#annotations	18,210	17,767	13,090	2,154	2,487	4,874	2,453	5,343
#annotations/item	7	8	8	7	7	6	4	5
Avg. accuracy	71.4	72.7	79.2	76.3	74.1	73.1	75.8	76.0

4.4 Personalised news recommendations

Our 20 experimenters were requested to evaluate news recommendations according to 10 user profiles obtained from Flickr and del.icio.us datasets. Using News@hand and its recommendation algorithms, they had to evaluate the 5 top ranked news items for each user/topic, specifying whether a recommended item would be relevant or not for the users taking into account their profiles. Table 3 shows the average results. Each value represents the percentage of evaluated news items that were marked as relevant. The results are compared with those obtained with a classic keyword-based algorithm [4] applied to the initial folksonomy-based user profiles.

Table 3. Average relevance values for the 5 top ranked news items recommended by News@hand

	headlines	**world**	**business**	**technology**	**science**	**health**	**sports**	**entertainment**
keyword-based	46.3	34.3	39.0	43.5	35.9	21.1	58.0	33.5
News@hand	57.0	53.2	72.8	94.0	60.9	40.6	98.2	60.4

5 Conclusions and future work

The combination of folksonomy information with knowledge available in the Semantic Web is in our opinion a powerful and promising approach to provide flexible, multi-domain collaborative recommendations. It benefits from two major issues: the easy adaptation to new vocabularies, and the supervised representation of semantic knowledge. Folksonomies and Wikipedia repositories continuously and collaboratively grow, providing consensual up-to-date semantic information about user preferences and items. On the other hand, ontologies allow us to describe and organise the above information, so that relations between concepts can be defined and used by fine-grained content retrieval and recommendation strategies.

We have presented techniques that filter personal tags, and integrate them into multi-domain ontological structures considering semantic information extracted from Wikipedia. Annotating item contents with concepts of the same knowledge bases, we relate user profiles and item descriptions under a common semantic concept space, fact that is exploited by several ontology-based recommendation algorithms. We have conducted preliminary evaluations of the above techniques obtaining favourable results. However, more detailed experimentation should be done in order to obtain founded conclusions about the benefits of our proposals.

Acknowledgements

This research has been supported by the European Commission (FP6-027685 – MESH, IST-34721 – TAGora). The expressed content is the view of the authors but not necessarily the view of the MESH and TAGora projects as a whole.

References

1. Alfonseca, E. Moreno-Sandoval, A., Guirao, J. M., Ruiz-Casado, M. (2006). *The Wraetlic NLP Suite*. Proc. of the 5th Int. Conf. on Language Resources and Evaluation (LREC 2006).
2. Cantador, I., Bellogín, A., Castells, P. (2008). *A Multilayer Ontology-based Hybrid Recommendation Model*. AI Communications, special issue on Recommender Systems. In press.
3. Cantador, I., Castells, P. (2008). *Extracting Multilayered Semantic Communities of Interest from Ontology-based User Profiles: Application to Group Modelling and Hybrid Recommendations*. Computers in Human Behavior, special issue on Advances of Knowledge Management and the Semantic Web for Social Networks. Elsevier. In press.
4. Castells, P., Fernández, M., Vallet, D. (2007). *An Adaptation of the Vector-Space Model for Ontology-based Information Retrieval*. IEEE Transactions on Knowledge and Data Engineering, 19 (2), pp. 261-272.
5. Cattuto, C., Loreto, V., Pietronero, L. (2007). *Collaborative Tagging and Semiotic Dynamics*. Proceedings of the National Academy of Sciences 104(1461).
6. Chirita, P. A., Costache, S., Handschuh, S., Nejdl, W. (2007). *PTAG - Large Scale Automatic Generation of Personalized Annotation Tags for the Web*. Proceedings of the 16th international conference on World Wide Web (WWW 2007). Banff, Alberta, Canada.
7. Hotho, A., Jäschke, R., Schmitz, C., Stumme, G. (2006). *Information Retrieval in Folksonomies: Search and Ranking*. Proceedings of the 3rd European Semantic Web Conference (ESWC 2006). Budva, Montenegro.
8. Jäschke, R., Marinho, L., Hotho, A., Schmidt-Thieme, L., Stumme, G. (2007). *Tag Recommendations in Folksonomies*. Knowledge Discovery in Databases 2007, pp. 506-514.
9. Miller, G. A. (1995). *WordNet: A Lexical Database for English*. Communications of the Association for Computing Machinery, 38(11), pp. 39-41.
10. Niwa, S., Doi, T., Honiden, S. (2006). *Web Page Recommender System based on Folksonomy Mining for ITNG'06 Submissions*. Proceedings of the 3rd International Conference on Information Technology (ITNG 2006). Las Vegas, Nevada, USA.
11. Specia, L., Motta, E. (2007). *Integrating Folksonomies with the Semantic Web*. Proc. of the 4th European Web Semantic Conference (ESWC 2007). Innsbruck, Austria.
12. Szomszor, M., Cattuto, C., Alani, H., O'Hara, K., Baldassarri, A., Loreto, V., Servedio, V. D. P. (2007). *Folksonomies, the Semantic Web, and Movie Recommendation*. Proc. of the 1st Int. Workshop "Bridging the Gap between Semantic Web and Web 2.0". Innsbruck, Austria.
13. Vallet, D., Castells, P., Fernández, M., Mylonas, P., Avrithis, Y. (2007). *Personalised Content Retrieval in Context Using Ontological Knowledge*. IEEE Transactions on Circuits and Systems for Video Technology, 17 (3), pp. 336-346.

From Web 2.0 to Semantic Web: A Semi-Automated Approach

Andreas Heß, Christian Maaß and Francis Dierick

Lycos Europe GmbH, Gütersloh, Germany
{andreas.hess, christian.maass, francis.dierick}@lycos-europe.com

Abstract. Web 2.0 and Semantic Web are regarded as two complementary paradigms that will probably converge in the future. However, whereas the Semantic Web is an established field of research, there has been little analysis devoted to Web 2.0 applications. For this reason it remains unclear how the advantages of both paradigms could be merged. In this paper we make three contributions in this direction. First, we discuss why merging Web 2.0 and the Semantic Web is beneficial and propose five approaches. Second, we show that (semi-) automated tagging of content improves the quality of annotations. Third, we present an automatic approach for improving the tag quality by using duplicate detection techniques. We verify our approach on a large-scale data set from the social search service Lycos IQ.

1 Introduction

The Semantic Web promises an easy access to information sources using a machine-understandable (not only machine-readable) representation of knowledge. This requires web resources be annotated with machine-understandable meta-data. Presently, the primary approach to achieve this is to first define an ontology and then use the ontology to add semantic markup for web resources. However, at present only a fraction of web users can take part in the process of building ontologies. Ontology tools and ontology languages impose high entrance barriers for potential users [11]. This is likely to contribute to the fact that the most popular approach of creating ontologies is engineering-oriented, i.e. a small number of individuals carefully construct the representation of the domain of discourse and release the results at some point in time to a wider community of users. In other words, the ontology evolution is not under full control of the users. For example, missing entries cannot be added by any user who finds the need for a new concept, but have to be added by the small group of creators. In natural language, in comparison, the evolution of the vocabulary is under the control of the user community. Anybody can invent and define a new word or concept in the course of communication.

Against this background there was a large debate in academic literature on how the process of meta-data generation can be automated to address the problem of cost-intensive ontology construction and generation of meta-data, e.g. [3, 11]. This is an important research question as several researchers agree

that without the proliferation of formal semantic annotations the Semantic Web is certainly doomed to failure [3].

One way to lower the threshold for a user to enter the Semantic Web is to move away from strict 'heavy' ontologies towards light-weight ontologies, folksonomies or tagging. With the proliferation and growth of so-called Web 2.0 sites, tag-based folksonomies promise to be a useful tool for search and navigation. Unlike an ontology, which is usually defined as a "specification of a conceptualisation" [9] and due to its formal nature created by trained experts, a folksonomy is "a type of distributed classification system" and "usually created by a group of individuals, typically the resource users" [10].

However, as the user and content base of a site grows, folksonomies tend to become more diffuse and imprecise. A certain degree of automation would help to make the maintenance of folksonomies as well as the annotation of content easier. In spite of the obvious need for this, up to now only a few automated approaches have been presented.

Even though tagging is not comparable to annotations using a full ontology, it can be regarded as a first step towards the Semantic Web. Even more important, tagging is already used every day by millions of web users posting blog entries. It is accepted that automated tagging algorithms, unlike manually tagged data, can have significant levels of mis-classification [5]. In a semi-automatic setting predicted tags do not have to be perfectly accurate in order to be useful. It is still easier for a user to browse through a list of only a few possible tags than to enter their own free-text tags. Furthermore, entering new tags is error-prone, since synonyms or spelling mistakes are not always detected. If the tags are drawn from a full ontology or at least a controlled vocabulary, looking only at a few suggestions is easier than looking at a complete ontology with possibly thousands of concepts.

In the remainder of this paper we will make three contributions to address the question, how the top-down approach of ontology engineering could be merged with a bottom-up approach that is typical for so-called Web 2.0 applications. First, we provide an overview of the advantages and disadvantages of ontologies and folksonomies and why a merging of these concepts is beneficial. Second, we will focus on the semi-automated classification or tagging of content. We believe that when using algorithmic assistance for annotating text, the quality of annotations will increase. We present a machine-learning-based classification algorithm that is tailored for use with short texts and folksonomies. We show that part-of-speech-tagging can be used in text classification and retrieval to dramatically reduce the dimensionality of the corpus without affecting performance. The algorithm is fast enough for interactive use. Third, we present an automatic approach for tag merging and correction. This is an important issue as folksonomies usually do not consist of a limited number of well-written tags. Rather, almost every Web 2.0 application faces the problem that tags are misspelled or are redundant. To address these problems, we present a method for detecting different (mis-)spellings of a tag that is based on a spell-checker in con-

junction with string edit distance metrics. We use a rule learner for fine-tuning the parameters of the algorithm.

1.1 Folksonomies: Usage Scenario

Despite their complementary nature, currently folksonomies (or tag clouds) and ontologies are used in quite distinct usage scenarios. Folksonomies are mainly used for tagging in Web 2.0 applications, the main use cases being search, navigation and recommendation. For social bookmarking sites such as `del.icio.us` tagging is an essential part of these processes: links are annotated and thereby sorted into categories, a user can search by category, etc. For applications like photo sharing, tagging is a prerequisite for effective searching, since a picture cannot easily be searched by its actual content. By arranging related tags (e.g. by co-occurrence) folksonomies also allow for browsing through different categories. While folksonomies have the clear advantage of being cheap and reflecting the language of the user, there are certain problems related to their use: if tags are not drawn from a controlled vocabulary but are just plain text keywords, several issues that affect the usefulness of tagging will arise. Especially inexperienced users tend to assign tags that are not meaningful to other users or to assign no tags at all. For example, as a result of an analysis of the leading social bookmarking system `del.icio.us`, Lee points out that about 20% of its users do not annotate or tag any of their bookmarks [15]. Moreover, different spellings and subjective combinations of tags lead to more or less diffuse folksonomies. Therefore, errors occur frequently while searching for related issues and subjects. Some users are well aware that this is a problem and that there should be some tagging guidelines. This problem and whether tools should be used is discussed both in the blogosphere[1] and by scientists [21]. We argue that by moving from folksonomies towards ontologies, the usefulness for Web 2.0 scenarios will improve as well. Therefore, we investigate the use of semi-automatic techniques for assisting the user in annotating content and cleaning existing tag clouds and thus moving to a more structured representation.

1.2 Proposed Approaches

Making the transition from Web 2.0 to Semantic Web smooth and user-friendly is a difficult task. We propose a combination of five approaches to address the different aspects of the problem:

Semi-Automated Tagging As a first step, we believe it is very important to reduce the uncontrolled growth of tag clouds due to usage of synonyms, misspelled words and inconsistent tagging. We propose using semi-automated tag suggestions based on text classification to guide users towards consistent tagging.

[1] e.g. `http://paolo.evectors.it/2005/05/24.html#a2532` or `http://ross.typepad.com/blog/2005/05/tags_and_simple.html`

We believe that users will more likely choose from a list of suggested tags than entering new tags. As a consequence the quality of annotations will increase. In section 2 we present our algorithm for semi-automated tagging in detail.

Tag Merging We believe that merging of synonyms and misspelled tags will increase the quality of annotations in the same way as automated tagging. The result of merging similar tags denoting the same concept will be a more consistent tagging. We discuss our algorithm for tag merging in section 3.

Identification of Related Tags Based on co-tagging (i.e. tags used together to annotate the same content) it is possible to identify a network of relationships between tags. Approaches that involve further analysis of such a network of tags have been discussed, e.g. [13]. However, this topic is out of the scope of our own research area.

Tag Rating For the combination of tags drawn from a folksonomy and concepts in an ontology we follow a layer concept: User-entered tags are located in an outer layer, whereas concepts in an ontology that is maintained by experts are located in an inner core. When tags in the outer layer are identified as being consistent and precise and there is no equivalent concept in the ontology already, these tags should be included as concepts. We propose that the user community can rate annotations. When a tag gets a high number of good ratings it should be recommended to the experts for inclusion in the core ontology. We will investigate using such a rating mechanism in future work.

Information Extraction We propose to make extensive use of information extraction techniques to fill the core ontology with facts. DBPedia [1] is an approach for extracting facts from Wikipedia articles. Furthermore, a large amount of previous work on information extraction from websites (e.g. [14]) and free form text (e.g. [7]) exists. Some approaches are targeted directly towards use in the Semantic Web area, e.g. [4]. In [16] an approach for finding relationships via a web search is presented. We are currently researching in the same area. In our approach, we are combining results from information extraction sources with a web search in order to identify the type of relation between two persons and to either confirm or disprove whether such a connection exists. We will present this approach in greater detail in a future publication.

1.3 Case Study: Lycos iQ

Lycos iQ is a question-and-answer community web site. Q&A communities try to deliver answers where algorithmic search engines fail to generate high quality results by activating users from the Web 2.0 community. For example, current search technologies have problems to answer search enquiries just like “Who is the Swedish singer that sounds like Heather Nova”. These kinds of questions

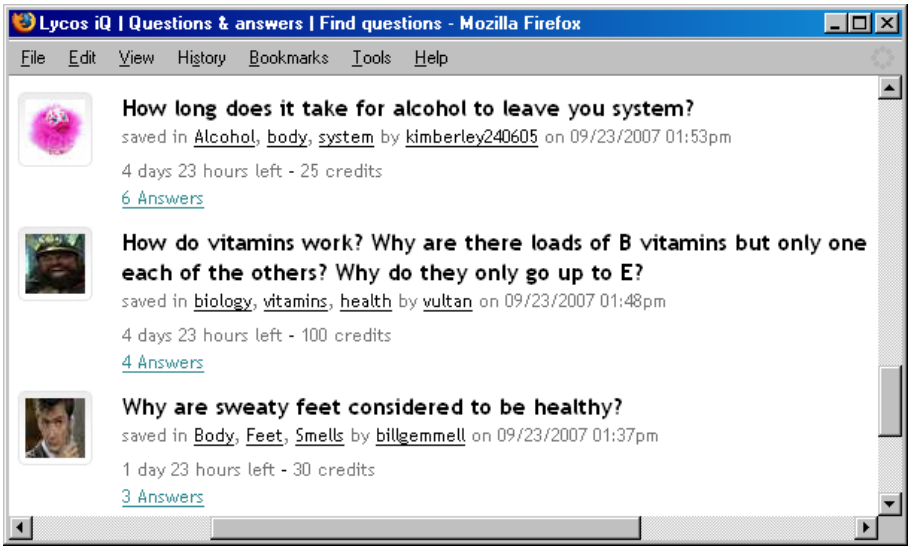

Fig. 1. Screenshot of the Lycos iQ website, illustrating our dataset

could only be answered by humans or by a group of users. Q&A services provide the necessary infrastructure to discuss such questions and enable a broad community to share their knowledge. Figure 1 shows a screenshot from the Lycos iQ website[2]. Q&A communities such as Lycos iQ or Yahoo Answers must not be confused with Q&A systems as known in Information Retrieval.

Tags are a certain form of meta-data that serves as a description for a particular content. For example, a question like the above could be published with the tags 'music', 'Sweden', 'singer' or 'songwriter'. Through these tags the posting is associated with the topics 'music' and 'songwriter', although the terms are not explicitly mentioned in the text.

From the technical point of view the 'tagging' of questions is the key to success for these kinds of services: Based on tags, expert users that can answer questions from a specific topic can be identified and brought together with users seeking information. The quality of tags is a crucial element in the functionality of such a service. However, as social bookmark systems have attracted a great attention [15, 8], there has been little analysis devoted to Q&A- and other communities.

2 Semi-Automated Tagging

2.1 Related Work

Although text classification is an old and well-researched topic in information retrieval and machine learning (e.g. [17]) it has not been widely used for au-

[2] `http://iq.lycos.co.uk/`

tomatic tagging in Web 2.0-applications yet. An exception is AutoTag [19], a system that uses a k-nearest-neighbour classifier for automated tagging of blog posts. This work is closely related to ours. We will highlight the differences in the next section. A more complex case-based system for semi-automated tagging is TagAssist [22]. As far as we are aware, these systems have not been deployed in an environment outside of the research community. No well-known Web 2.0 sites actually use semi-automated tagging based on learning or natural language techniques.

2.2 Problem Formulation

Formally, semi-automated tagging is a multi-value text classification problem.[3] Most machine learning algorithms can only handle single-value classification. Therefore it is common practise that single-value classification algorithms are adapted by means of some combination method; see [23] for a recent survey. However, these strategies are out of the scope of this paper: given that in our scenario many annotations are plausible and the user is involved in the classification process, it is not necessary that the algorithm predicts the exact set of true annotations and presenting a ranked list is acceptable. Considering the high number of classes and the need for an incremental learning algorithm, using vector space classification algorithms such as kNN or Rocchio is a logical choice. AutoTag [19] uses a search engine to locate similar blog posts. The search query is derived from the text that is to be classified using statistical query rewriting techniques. In the next step, tags from the search results are aggregated and re-ranked using information about the user. Yet, this method of predicting tags for posts has a disadvantage. Re-writing the query at classification time is computationally costly. In a semi-automatic setting, where users can annotate their content online immediately after they type, response time is critical. Therefore, to avoid the need for query rewriting, we decided to perform a feature selection at training time. First we tokenised the short headline of the questions. This headline usually consists of just one sentence. We applied part-of-speech tagging and kept only nouns and proper nouns. Although the headlines in Lycos iQ are much shorter than blog posts (usually just one sentence) and we did not use re-ranking, we achieved similar performance results in preliminary experiments conducted with the current live version of Lycos iQ.

2.3 kNN classification vs. Rocchio

For comparison, we implemented two classifiers, both based on an index created using the dimensionality reduction method described above (POS tagging). First, we implemented a kNN classifier with $k = 10$ that queries the index for the ten nearest questions (postings) in the database and aggregates the tags from

[3] For the remainder of this paper, we will use the terms 'class' as in text classification and 'tag' as synonyms.

the results. Preliminary tests showed that IDF weighting does not improve classification results in this setting, so we decided not to use it in our experiments.

Second, we implemented a Rocchio-style classifier. Rocchio classification is based on Rocchio relevance feedback [20]. The centroids for each class were simply computed as a big bag of words containing all tokens from all posts labelled with this specific class (tag). Note that if a posting is tagged with more than one tag the tokens in the post are indexed for all its classes. As opposed to the kNN classifier, we found out that IDF weighting slightly improves performance, so we used it in our experiments. Although the difference in accuracy between using IDF weighting and not using it were rather low, we decided to use the best setting for each of the two approaches to achieve a fair comparison.

Given the nature of the two algorithms, we expect that Rocchio classification will be faster at classification time, a factor that is very important in an interactive setting, when users are not willing to accept long response times.

When comparing the classification performance of Rocchio vs. kNN, Rocchio has some known disadvantages: It becomes inaccurate, when classes are not spheres of similar size in vector space, and it does not handle non-spherical classes (e.g. multi-modal classes that consist of more than one cluster) very well. However, we argue that these properties of Rocchio classifiers will not affect performance in our setup, because of three main reasons: First, while the dataset is indeed skewed and classes have very different size, we expect that most of the classes will be mono-thematic. Second, while large classes will be preferred by the classifier, this is not a disadvantage in our scenario. When considering semi-automated tagging of postings, it is actually an advantage when we direct users towards choosing more popular tags, as explained in section 1. Third, since our scenario is a multi-value classification problem where not only a single prediction is correct, overlap between classes does not necessarily influence accuracy negatively.

2.4 Evaluation

To evaluate our approach, we used a corpus of 116417 question from the knowledge community web site Lycos iQ. Figure 1, showing a screenshot from the Lycos iQ website, gives a good impression of what our dataset looks like. Note the different levels of tags the users assigned and the style of the texts. We use only the headlines as shown on this screenshot for classification. After tokenisation and POS tagging, there were 89275 distinct tokens. After deleting frequent tokens according to Zipf's law, this number was further reduced to 27309, leading to an average of only 2.63 tokens per question. Questions were tagged with 49836 distinct tags that follow the usual distribution with some frequently used tags followed by a long tail. It should be noted that the number of classes exceeds the number of tokens. We expect that tags drawn from the long tail will be suggested very rarely if at all. However, this does not pose a problem, since guiding the user towards common tags accepted by many users is one of the goals of our approach. Suggesting tags that were initially only used very few times or even only once does not make sense. It is not useful to include text

from answers in the classification. When the user is tagging a question, there are no answers available yet. The text from the answers is not helpful for classifying the questions, since the words are too different. We evaluated both the kNN approach and the Rocchio approach automatically using the 111629 questions that had user-assigned tags, comparing the predicted tags with the manually assigned tags. We used the well-known leave-one-out cross-validation scheme for evaluation. After preprocessing we encountered the fact that some questions were not assigned any tokens. These questions were, however, left in the corpus and thus affect the overall performance of our algorithm negatively.

Methodology It is important to note that the tags assigned by users should not be regarded as a gold standard. Tags are not drawn from an ontology, taxonomy or controlled vocabulary, but are free text entered by users and thus prone to spelling mistakes. Also, inexperienced users tend to assign either no tags at all, only very few tags or they tag inconsistently. Given the large number of users, we also expect that users use different synonyms to denote the same concept. Due to these ambiguities and inconsistencies we expect that the accuracy of any automated approach is considerably lower than its true usefulness.

To overcome this problem in our empirical evaluation, we distributed questionnaires and had test persons check the plausibility of tags suggested by our semi-automated approach. To reduce the workload for the test persons and because it outperformed the kNN classifier in the automated tests, we decided to test only the Rocchio-style approach. For comparison, we also had the test persons check the precision of the user-assigned tags, since we assumed many nonsensical or inconsistent tags among them. Every test person was given one or two chunks of 100 out of a random sample of 200 questions that were either machine-tagged or hand-tagged. Every question was checked by four persons to average out disagreement about the sensibility of tags.

Automatic Evaluation For the classification results evaluated against the user-assigned tags, we report precision, recall and accuracy. Precision is defined as the number of predicted tags that also occur in the set of manually assigned tags divided by the number of predicted tags that were considered. We report precision only for the top three predicted tags. Since most questions are only tagged with up to three tags, it does not make sense to report precision when allowing for more predicted tags. Recall is defined as the number of tags in the set of user-assigned tags that also occur in the set of predicted tags divided by the number of user-assigned tags. We report recall for the top ten predicted tags. With this definition of precision and recall we follow the generally accepted definition in information retrieval and machine learning. Furthermore, we report the fraction of questions where there is at least one overlap between the assigned and predicted tags as accuracy. We regard it is already useful, if only some or even one sensible suggestion is among the top suggested tags, since the user can quickly browse through a short list of suggestions and select or deselect tags. We expect that accuracy and recall will go up if we include more suggested tags.

Top n tags	1	2	3	4	**5**	6	7	8	9	10
kNN Precision	0.26	0.24	0.20	—	—	—	—	—	—	—
kNN Recall	0.26	0.23	0.21	0.23	**0.24**	0.25	0.26	0.27	0.27	0.28
kNN Accuracy	0.23	0.29	0.32	0.34	**0.35**	0.36	0.37	0.38	0.39	0.39
Rocchio Precision	0.32	0.31	0.27	—	—	—	—	—	—	—
Rocchio Recall	0.32	0.30	0.28	0.31	**0.33**	0.35	0.37	0.38	0.39	0.40
Rocchio Accuracy	0.28	0.36	0.41	0.43	**0.45**	0.47	0.48	0.49	0.49	0.50

Table 1. Results of the automatic evaluation

Table 1 shows the empirical results for precision, recall and accuracy for the two different proposed methods. For recall and accuracy, we highlight the value at five suggested tags, because we believe that a user will not accept a longer list. As Miller pointed out in [18], most people can process five items at once. We measured the classification time per instance for both approaches on an Intel Core 2 machine with 1.86 GHz and 1 GB RAM. As expected, Rocchio classification was much faster than kNN. The classification time for each instance was 155 ms for kNN and 57 ms for Rocchio.

Manual Evaluation As expected, we could observe that there was a big disagreement among the test persons and the users who originally tagged the questions as well as between the test persons themselves. For the manual evaluation, we checked only the Rocchio classifier because it performed better in the automatic test. As explained above, the total 200 questions that were evaluated were split in two sets of 100 questions, yielding four different questionnaires (two for the original user-assigned tags and two for machine-annotated tags) and each chunk of 100 questions was checked by four persons. Each test person was checking at most two sets of questions. To highlight the huge difference of the several test persons, we report the individual results in the table below. For the human-annotated tags, we evaluated precision, defined as the number of useful tags divided by the total number of assigned tags. For the machine-assigned tags, we report accuracy as well, with the same definition of accuracy as in the automatic test. Questions that had no assigned tags were ignored for evaluating accuracy. Among the 200 randomly selected question were 6 that had no assigned tags, leading to 194 questions evaluated for accuracy.

For all manual tests, we evaluated the algorithms with five suggested tags only. We believe that in a real-world semi-automated setting, we cannot assume that an inexperienced user is willing to look at more than five tags. The questions that were manually tagged had mostly three tags each, some of them only two and very few questions had more than three tags.

As expected, there was a large disagreement between different persons both on the human-annotated tags as well as the machine-annotated tags (see tables 2). It is interesting to note when looking at the second set of questions, that, although the human annotations on this set of 100 questions were rated worse than those from the first set, the tags suggested by our algorithm were

Test	TP	TP+FP	avg. Prec.
assigned tags	1535	1856	0.83
suggested tags	1866	3360	0.56

Test	Person 1	Person 2	Person 3	Person 4
Set 1, assigned tags, prec.	0.89	0.89	0.93	0.96
Set 2, assigned tags, prec.	0.52	0.73	0.73	0.87
Set 1, suggested tags, prec.	0.41	0.52	0.53	0.71
Set 2, suggested tags, prec.	0.51	0.54	0.59	0.65
Set 1, accuracy	0.84	0.84	0.86	0.87
Set 2, accuracy	0.87	0.87	0.91	0.91

Table 2. Results of the manual evaluation

on average rated even slightly better. Keeping in mind that we envision a semi-automated scenario with human intervention, we see this as a confirmation that automatically suggested tags can help to improve the quality of tagging.

When looking at macro-averaged precision, it is obvious that a classification system is still not good enough for fully automated tagging. However, it is important to note that even the human-annotated questions were rated far below 100% correct by the test persons. More than half of the suggested tags were rated as useful by the test persons. We believe that this is certainly good enough for a semi-automated scenario, were users are presented a small number of tags to choose from. In absolute numbers, interestingly, the automatic classifier produced more helpful tags than were assigned by users, even compared to the number of all user-assigned tags, not just the ones perceived as helpful by the test persons. We believe that this confirms our hypothesis that users will assign more tags when they are supported by a suggestion system. However, this can only be finally answered with a user study done with a live system.

Finally, the high accuracy of the classifier (with accuracy being defined as in section 2.4) underlines our conclusion that semi-automated tagging is good enough to be implemented in a production environment. In almost nine out of ten cases there was at least one helpful tag among the suggestions.

3 Tag Merging

3.1 Problem Formulation

When considering a social tagging system with a potentially high number of users, one of the most common problems is inconsistent tagging. One common facet is that users tend to misspell tags . While we can guide the user towards consistent tagging by suggesting tags, a complementary approach is to clean existing tag clouds after the annotation phase by identifying tags that should be merged. This merging can be based both on string similarity to detect misspellings or based on a thesaurus to detect synonyms. The first approach is well known as duplicate detection or record linkage. Yet, little attention has been

devoted to using duplicate detection techniques for improving the quality of tag clouds. Unlike semi-automated annotation, tag duplicate detection must have a precision that is high enough to be run unsupervised, with only little manual correction. An interactive review is infeasible when merging possibly several thousand misspelled tags. Therefore, the duplicate detection algorithm should be biased towards high precision.

3.2 Parameter Tuning

The obvious approach to address the tag duplicate detection problem is to use string similarity metrics. Preliminary experiments, however, showed that the performance of using a single string distance metric is not sufficient to be employed in an automatic setting in terms of precision. Therefore, we decided to combine multiple similarity measures and use a machine learning algorithm to fine-tune the exact setting. A similar approach has been used by Bilenko and Mooney [2].

We use a two-step approach. First, we check whether a tag should be considered for merging. Second, if the tag is suitable, we use an off-the-shelf spell-checking tool to identify possible tags that are candidates for merging. To be able to use a standard spell checker for our purpose, we do not use a natural language dictionary but the list of all tags in the system instead. Therefore, suggestions by the spell checker are tags that are textually similar. While an efficient implementation of a spell checker will return a result very quickly, these results are not accurate enough for a fully automated merging process. To increase precision, we perform further checks. Pairwise calculation of all of these features for all tags would be computationally too expensive and therefore prohibitive.

For the first step, checking whether a tag should be considered for merging, we used a number of features such as frequency, number of related tags, string length and the number of tokens (a tag can consist of multiple words). Tags are called related tags, when they are used together to annotate the same content. Related tags of second order are tags that are related to the same other tag (e.g. if there are relationship between “house” and “door” and between “building” and “door”, then “house” and “building” have a second order relationship). Frequency is defined as the number of times the tag has been used. For the second step, checking whether two tags should be merged, we considered additional features such as Levenshtein edit distance, Jaro-Winkler string similarity, Monge-Elkan string similarity and Smith-Waterman string similarity. The relationship strength is defined as the number of co-occurrences. The relationship strength of second order is defined as the sum of the strength of the connecting first order relations.

To tune the thresholds for the merging system we created training sets with the features explained above and exported them into ARFF format for processing with WEKA [24]. Because of the desired area of use and the simplicity of implementation of the classifier, learners that output rules seemed an obvious choice. We experimented with different rule learning and decision tree algorithms

including RIPPER and C4.5 as implemented in WEKA.[4] Since our goal is to implement an automatic, unsupervised tag merger, we consider precision on the class of merged tags as more important than recall and accuracy. A classifier with high precision leads to a merger that will make few mistakes at the price of missing some tags that should be merged. To bias the learning algorithm towards precision on one class, we experimented with biased classifiers from the implementation of the Triskel algorithm [12]. Biasing methods include under-sampling (randomly dropping training instances from one class) and over-weighting (assigning a higher weight to instances from one class). It has been shown that even dropping as much as 90% of instances leads to a good classifier with high precision. After looking at the resulting models, it turned out that, as expected, some of the features were redundant.

3.3 Evaluation

To evaluate the tag merger, we used the same dataset as described in section 2, with 49836 distinct tags in the database. Running the merging algorithm generated 4320 sets of merged tags with 10245 tags in total, yielding an average 2.4 tags per set. This means that after merging, about one fifth of tags are part of a set of merged tags, and when only keeping one tag per set, the amount of tags available will be reduced by around 11.9%.

To measure the precision of the merger, we examined a sample of 100 tag sets containing 248 distinct tags. Of these 100 sets, 3 contained one tag that was not correctly merged, 1 set contained two tags that did not fit and in 2 cases it could be considered questionable whether or not the tags should be merged. This leads to an average precision of 94% resp. 96% (depending on whether the two tags in doubt are considered correctly merged). When putting the errors in relation to individual tags instead of sets (keeping in mind that a set can contain more than two tags), precision is approx. 97.98% resp. 97.18%.

We believe that the precision of this approach is high enough to allow elimination of misspelled tags or merging of tags in their singular or plural form in a fully automated setting. Even with very conservative settings yielding a high precision, there is already a significant amount of tags that can be merged or discarded.

4 Conclusion

4.1 Summary

In this paper, we have made three contributions: first, we have proposed five approaches to address some of the problems when moving from a folksonomy towards an ontology. We argue that this move brings benefits, although we have to keep in mind that heavy ontologies tend to become too complicated for the broad mass of inexperienced users. Therefore, we advocate for (semi-) automated

[4] The WEKA implementations of RIPPER and C4.5 are called JRip and J4.8

measures to guide users towards more structured tagging. We regard this as a first step on the move from Web 2.0 to the Semantic Web.

Second, we have presented a classification algorithm that has been shown to perform well in terms of accuracy even when using very short text snippets. We have shown that a drastically reduced dimensionality of the text corpus can be achieved when using part-of-speech tagging. We conclude from our empirical study based on a dataset with questions from the Lycos iQ web site that the performance of our classifier is good enough to be employed in a semi-automatic setting and that it scales well enough to be used in a production system with a large number of instances.

Third, as a complementary measure to the proposed semi-automated tagging, we have shortly described an algorithm for automatically merging misspelled or similar tags and have shown that its precision is high enough for unsupervised operation while still eliminating a high number of tags. We believe that both ad-hoc interactive tagging suggestions as well as post-hoc offline merging are effective approaches to improve the quality of a folksonomy as a prerequisite towards moving to more structured semantic networks and ontologies.

4.2 Generalisation

In recent work that is beyond the scope of this paper we have tested our text classification system on the well-known Reuters dataset. We have adapted our algorithm to output a true multi-value classification instead of a ranked list. We introduce a new method for multi-value-classification that is related to stacking. The performance of our algorithm is comparable to the older results presented in [6], although we use only the headlines of the articles as opposed to the full text. In [6], a macro-averaged accuracy for the top 10 classes of 63.7% was reported for the Rocchio classification algorithm in a one-classifier-per-class scheme. We achieved a 67.1% accuracy using our stacking approach. Due to the different focus of these experiments and space restrictions, we will present these results in a separate publication. Even though we focused on Rocchio and kNN classification due to performance requirements related to interactive use, using SVMs should be further evaluated.

Second, we applied our tag merging algorithm to the task of aligning headlines of movie reviews, news articles and movie titles from the cinema programme. The dataset was taken from the Lycos movie portal site. The setup of these experiments was slightly different: a pre-selection using a fast implementation of a spell checking algorithm was not needed due to the smaller dataset size. We removed domain-specific stop-words. Due to the different nature of the dataset, co-occurrence information was not available. As an additional distance metric, a soft-token TFIDF metric was used. As in the experiments described here, we used a rule learner to identify combination rules for the different similarity metrics and to learn a threshold. Additionally, we experimented with an SVM classifier. In these experiments we could achieve a precision of almost 100%.

4.3 Future Work

One of the next big challenges will be to align folksonomies with existing ontologies. As mentioned in section 1.2, we are currently investigating using web search to identify relationships between persons. In future work, we want to investigate how ratings by users can be used to identify high-quality tags. These tags could then be presented to a group of moderators or administrators to decide whether or not they should be added as concepts into a more stable ontology.

One of the clear hurdles towards a widespread adoption of ontologies have been usability problems related to visualising and interacting with such complex datasets. Specifically, we hope to guide the user towards real semantic tagging without alienating them. This process of gradually increasing ontology complexity has been described in literature as ontology evolution.

4.4 Acknowledgements

The research presented in this paper was partially funded by the German Federal Ministry of Economy and Technology (BMWi) under grant number 01MQ07008. The authors are solely responsible for the contents of this work.

We thank our colleagues at Lycos Europe who helped us with the manual evaluation of our tagging system.

References

1. Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *The Semantic Web*, volume 4825 of *Lecture Notes in Computer Science*, pages 722–735, Berlin/Heidelberg, 2008. Springer.
2. Mikhail Bilenko and Raymond J. Mooney. Learning to combine trained distance metrics for duplicate detection in databases. Technical report, Dept. of Computer Science, University of Texas, Austin, Texas, USA, February 2002.
3. Philipp Cimiano, Günter Ladwig, and Steffen Staab. Gimme' the context: Context-driven automatic semantic annotation with c-pankow. In *Proc. of the 14th Int. World Wide Web Conference*, 2005.
4. Fabio Ciravegna, Sam Chapman, Alexiei Dingli, and Yorick Wilks. Learning to harvest information for the semantic web. In *Proceedings of the 1st European Semantic Web Symposium*, Heraklion, Greec, May 2004.
5. Stephen Dill, Nadav Eiron, David Gibson, Daniel Gruhl, R. Guha, Anant Jhingran, Tapas Kanungi, Sridhar Rajagopalan, Andrew Tomkins, John Tomlin, and Jason Zien. Semtag and seeker: Bootstrapping the semantic web via automated semantic annotation. In *Proc. of the 12th Int. World Wide Web Conference*, 2003.
6. Susan Dumais, John Platt, David Heckerman, and Mehran Sahami. Inductive learning algorithms and representations for text categorization. In *CIKM '98: Proceedings of the seventh international conference on Information and knowledge management*, New York, NY, USA, 1998. ACM.
7. Aidan Finn and Nicholas Kushmerick. Multi-level boundary classification for information extraction. In *Proc. European Conf. on Machine Learning*. Springer, 2004.

8. Gernot Gräfe, Christian Maaß, and Andreas Heß. Alternative searching services: Seven theses on the importance of social bookmarking. In *Proc. of the 1st Conf. on Social Semantic Web*, 2007.
9. Tom R. Gruber. A translation approach to portable ontologies. *Knowledge Acquisition*, 5(2):199–220, 1993.
10. Marieke Guy and Emma Tonkin. Folksonomies – tidying up tags? *D-Lib Magazine*, 12(1), January 2006.
11. M. Hepp, D. Bachlechner, and K. Siorpaes. Ontowiki: Community-driven ontology engineering and ontology usage based on wikis. In *Proceedings of the 2006 international symposium on Wikis*, 2006.
12. Andreas Heß, Rinat Khoussainov, and Nicholas Kushmerick. Ensemble learning with biased classifiers: The triskel algorithm. In *Proceedings of the 6th International Workshop on Multiple Classifier Systems*, Monterey Bay, California, USA, 2005.
13. Andreas Hotho, Robert Jäschke, Christoph Schmitz, and Gerd Stumme. Emergent semantics in bibsonomy. In Christian Hochberger and Rdiger Liskowsky, editors, *GI Jahrestagung*, volume 94 of *LNI*, pages 305–312, 2006.
14. Nicholas Kushmerick. *Wrapper induction for information extraction.* PhD thesis, University of Washington, 1997.
15. K. Lee. What goes around comes around: An analysis of del.icio.us as social space. In *Proc. of the 20th conf. on Computer supported cooperative work*, 2006.
16. Gang Luo, Chunqiang Tang, and Ying li Tian. Answering relationship queries on the web. In *Proc. of the 16th Int. World Wide Web Conference*, New York, NY, USA, 2007. ACM.
17. Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval.* Cambridge University Press, 2008.
18. G. A. Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *The Psychological Review*, 1(63):81–97, 1956.
19. Gilad Mishne. Autotag: a collaborative approach to automated tag assignment for weblog posts. In *Proc. of the 15th Int. World Wide Web Conference*, New York, NY, USA, 2006. ACM Press.
20. J. J. Rocchio. *The SMART Retrieval System: Experiments in Automatic Document Processing*, chapter Relevance feedback in information retrieval, pages 313–323. Prentice-Hall, Englewood Cliffs, NJ, USA, 1971.
21. Clay Shirky. Ontology is overrated: Categories, media & community. http://shirky.com/writings/ontology_overrated.html, 2006.
22. Sanjay C. Sood, Sra H. Owsley, Kristian J. Hammond, and Larry Birnbaum. Tagassist: Automatic tag suggestion for blog posts. In *Proceedings of the International Conference on Weblogs and Social Media*, Boulder, Colorado, USA, March 2007.
23. Grigorios Tsoumakas and Ioannis Katakis. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining*, 3(3):1–13, 2007.
24. I. H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools with Java Implementations.* Morgan Kaufmann, San Francisco, 1999.

Using multiple ontologies as background knowledge in ontology matching

Zharko Aleksovski[1], Warner ten Kate[1], and Frank van Harmelen[2]

[1] Philips Research, Eindhoven,
[2] Vrije Universiteit, Amsterdam

Abstract. Using ontology as a background knowledge in ontology matching is being actively investigated. Recently the idea attracted attention because of the growing number of available ontologies, which in turn opens up new opportunities, and reduces the problem of finding candidate background knowledge. Particularly interesting is the approach of using *multiple ontologies as background knowledge*, which we explore in this paper. We report on an experimental study conducted using real-life ontologies published online.
The first contribution of this paper is an exploration about how the matching performance behaves when multiple background ontologies are used cumulatively. As a second contribution, we analyze the impact that different types of background ontologies have to the matching performance. With respect to the precision and recall, more background knowledge monotonically increases the recall, while the precision depends on the quality of the added background ontology, with high quality tending to increase, and the low quality tending to decrease the precision.

1 Introduction

Ontology matching is regarded as one of the most urgent and most important problems in the Semantic Web. It is scientifically challenging and inherently very difficult problem [1–6]. It generated a lot of research in the past years, which resulted in many different solution methods proposed. Good surveys of the existing ontology matching methods can be found in [2, 3, 7]. According to [7] they can be divided into four categories: *terminological* that use lexical similarities between names, comments etc., *structural* that use the similarities in the structure of the matching ontologies, *instance-based* that use the classified instance-data in the ontologies, and *using background knowledge* that rely on external structured resources to find matching entities across different ontologies. In this paper, we focus on the last category - using ontologies as background knowledge in the matching.

Background knowledge in matching has been used in different ways [8–10]. In this study we use a very simple approach. We try to match each pair of concepts from the matching ontologies in two steps - *anchoring* and *deriving relations*. In the anchoring, we look if the matching concepts can be themselves matched to the background knowledge, and in the deriving relations we check

if they match to background concepts which are related to one another. If they are, then we report that the testing pair of concepts are matched. This type of match we call an *indirect match* because it is being discovered indirectly through the background knowledge ontology.

In respect to the matching success, regardless of the choice, no background knowledge ontology is likely to provide all the matches we would like to find. Instead, it is reasonable to expect that matches missed by one background ontology can be found using some other. Hence, hoping to find more of the matches we desire to find, we can use *multiple ontologies as background knowledge*. The question we face now is how the characteristics of the background ontologies will impact the matching performance. As discussed in the study of [11], the landscape of the online published ontologies is very diverse.

We set to investigate the feasibility of the matching when multiple background ontologies are used. To stress the paradigm, we present the results of several experiments in which we set our objectives as follows: *(i)* the anchoring to the background knowledge is a simple lexical matching technique, i.e. we only use simple matching as needed to obtain relatively successful anchoring (see Section 3 for further explanation), *(ii)* the background knowledge candidates are relatively large sized ontologies[3], and *(iii)* there is lexical overlap between the matching ontologies and the background knowledge, that is, lexical match is possible between the matching ontologies and the background knowledge. We use multiple background knowledge ontologies by using each ontology separately, and then combining the sets of obtained matches. We are interested to see how do the background ontologies perform together as compared to how each of them performs alone.

Multiple ontologies as background knowledge have already been used [9]. Contribution of this paper is that we study the contribution of each background ontology individually as compared to their cumulative contribution, and we also study the effects of the different types of ontologies when used as background knowledge. All the test data was selected from online published ontologies, and it consisted of two ontologies which we matched to one another and six other that we used as background knowledge. Having selected our data from online published ontologies, our results reflect on the current state of the published ontologies.

The experiments revealed that using background ontologies published on the semantic web provide low but stable increase of recall as compared to a simple direct matching. Multiple background ontologies find almost disjoint sets of matches, and hence result in cumulative increase of recall. The precision of background-based matching mostly, but not entirely depends on the quality of the background ontologies. The low quality ontologies increase the recall, but they reduce the precision. The high-quality background knowledge ontologies

[3] Ontologies of size around 30 concepts are common as demonstration examples, however, they are trivial to analyze and do not provide well-grounded empirical insight. Hence, we focused our attention on ontologies of larger size with at least couple of hundreds of concepts as more interesting candidates.

also find wrong matches, but these are mainly caused by the different context of the knowledge, not by mistakes in the ontologies.

The rest of the paper is organized as follows: in Section 2 we will describe our approach to using background knowledge in detail, in Section 3 we will describe our case study with the experimental data and the results, in Section 4 we will discuss the findings of the experiments, and finally with Section 5 we will conclude the paper.

2 Using background knowledge in ontology matching

In our approach we match two ontologies while using a third one as background knowledge. We call the ontologies being matched the source and the target, however, this naming is not discriminative - the matching algorithm treats them equally, and swapping their places will only invert the result set[4]. As we mentioned in the introduction, the algorithm proceeds in two steps - anchoring and deriving relations. Its scheme is depicted on Figure 1.

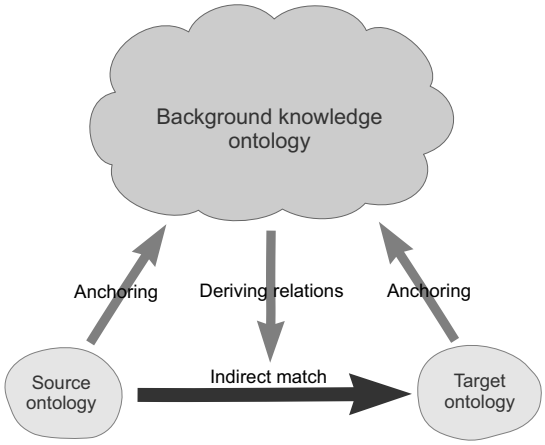

Fig. 1. Scheme of ontology matching using background knowledge.

Anchoring is matching the source and target concepts to the background knowledge. In general, this process can be performed by using any existing ontology matching technique. In our case we only use a simple lexical matching. Using other methods can make it difficult to explain the experimental results, because they may produce wrong matches, and simple technique while being rigid it is very precise and allows us to concentrate on the use of the background knowledge itself.

[4] The source and the target concept in each match on the result set will have their places swapped as well

Deriving relations is the process of discovering relations between source and target concepts by looking for relations between their anchored concepts in the background knowledge. Both the source and target concepts anchors are part of the background knowledge, and checking if they are related means using the reasoning service in the background knowledge ontology. Combining the anchor matches with the relations between the background knowledge concepts derives the match between source and target concepts, which is what we are looking for.

To explain this process in the context of life-sciences ontologies, we can see a realistic example on Figure 2: the source concept SRC: Brain is anchored to background knowledge concept BK: Brain, and the target concept TAR: Head is anchored to a background knowledge concept BK: Head. The background knowledge reveals a relation BK: Brain part-of BK: Head, and we derive a relation that source concept SRC: Brain has a narrower meaning than the target concept TAR: Head. Using background knowledge was crucial in this case; the match was not found by directly matching the source to the target ontology, SRC: Brain is classified under SRC: Central nervous system which is in no way related to the concept TAR: Head.

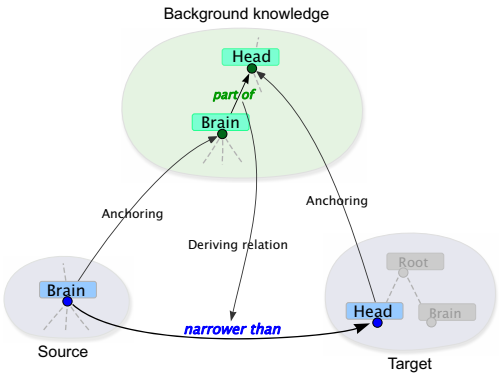

Fig. 2. Example using background knowledge in the matching process.

As suggested by the example above, of particular interest in our approach is exploiting the structure of the background knowledge ontology. It is done in the deriving relations step, when checking for relatedness between the anchored concepts in the background knowledge ontology. Before moving to the experimental part of the work, we will first introduce the formal definitions of all the components in this framework, which we will later use in the experimental part.

2.1 Formal framework

Concept is a class of things grouped together due to some shared property. It is named with a label, and sometimes with additional alternative names (syn-

onyms). Besides the name(s), the meaning of a concept is determined by its semantic neighborhood, that is how it is related to the other concepts in the ontology. We will refer to concepts with capital italic letters X, Y..., with X^{ONT} to a concept from specific ontology, and we will also use the concept's label (in Sans Serif font), like Temporal lobe, or ONT: Temporal lobe, for the concept from the particular ontology.

Relation instance (also called just relation) is a triple $(X \ \mathit{relation} \ Y)$, where X and Y are concepts, and $\mathit{relation} \in \mathcal{T}$ is a relation type. $\mathcal{T}$ is the universal set of all relation types. The relation instance $(X \ \mathit{relation} \ Y)$ is interpreted as the concept X is related through the relation type *relation* to the concept Y. When clear from the context we will call the relation instances simply relations.

Ontology is a pair of sets: $\mathsf{ONT}(\mathcal{C}, \mathcal{R})$. $\mathcal{C}$ is a set of concepts, $\mathcal{R}$ is the set of relations among these concepts. We will refer to ontologies with their full name, like Foundational Model of Anatomy (or the name in italic *Foundational Model of Anatomy*), with short form of the name in Sans Serif font, like ONT for an arbitrary ontology, or FMA for the particular ontology *Foundational Model of Anatomy*

Ontology match between two ontologies S and T is a set of relation instances:

$$M \ \subseteq \ \mathcal{C}^{\mathsf{S}} \times \mathcal{T} \times \mathcal{C}^{\mathsf{T}} \tag{1}$$

Each element in this set $(X \ r \ Y) : \ X \in \mathcal{C}^{\mathsf{S}}, \ r \in \mathcal{T}, \ Y \in \mathcal{C}^{\mathsf{T}}$ we call a *match* between X and Y, or, X is matched to Y, through the relation type r. We will write it as $X \xrightarrow{r} Y$, or, $X \rightarrow Y$ when the relation type of the match is known from the context.

An ontology match is the result of any ontology matching technique. In practice, it plays the role of a bridge between different ontologies. Two specific ontology matches are of particular interest to our approach. They correspond to the two phases of the matching - anchoring and deriving relations.

2.2 Evaluation

To characterize the degree of success for matching we adopt two notions from the information retrieval field: *precision* and *recall*. In Information Retrieval (IR) the precision and recall are measures on performance of document retrieval [12]. They rely on a collection of documents and a query for which the relevancy of the document is known, assuming binary relevancy: a document is either relevant or non-relevant. In the ontology matching we define these measures through two sets - *desired* matches, and matches *found* by a matching method.

Precision is the proportion of desired and found matches, to all the found matches:

$$\textbf{Precision} = \frac{|\mathit{Desired} \cap \mathit{Found}|}{|\mathit{Found}|} \tag{2}$$

Recall is the proportion of desired and found matches, to all the desired matches:

$$\textbf{Recall} = \frac{|Desired \cap Found|}{|Desired|} \quad (3)$$

The precision represents the quality or the preciseness of the matches - what portion of the found matches are correct, and the recall represents the completeness of the matches - how many of the matches we want to find were actually found. The precision and recall have values between 0 and 1 inclusive. In practice they are often expressed in terms of percentage, ranging from 0% to 100%.

3 Case study

In our case study we matched two ontologies from the agricultural domain using six other ontologies as background knowledge. Motivated by the variety of ontologies that exist online, we decided to use background knowledge ontologies with varying origin. We investigated three different types of ontologies: different but related domain ontologies, general knowledge ontologies, and ontologies of an unknown origin. We set simple direct matching as a baseline to evaluate the matching performance, and we analyzed the matching performance by observing the precision and recall. All the test data was extracted in March 2007.

Matching ontologies The source ontology was NALT[5] and the target Agrovoc[6]. They both describe the domain of life sciences and agriculture. Agrovoc, as stated on the description provided on its homepage[7], I quote "is a multilingual, structured and controlled vocabulary designed to cover the terminology of all subject fields in agriculture, forestry, fisheries, food and related domains (e.g. environment)." NALT, as described on its homepage [8], I quote: "The NALT is primarily used for indexing and for improving retrieval of agricultural information. Currently, the NALT is the indexing vocabulary for NAL's bibliographic database of citations to agricultural resources, AGRICOLA[9]. The Food Safety Research Information Office[10] (FSRIO) and Agricultural Network Information Center[11] (AgNIC) also use the NALT as the indexing vocabulary for their information systems. In addition, the NALT is used as an aid for locating information on the ARS[12] and AgNIC web sites." In the experiments we used the versions of the OAEI 2006[13], which are publicly available. They contain 41,577 concepts NALT, and 28,174 concepts Agrovoc. Many of the concepts besides the labels are additionally described with synonyms.

[5] http://agclass.nal.usda.gov/agt
[6] http://www.fao.org/agrovoc
[7] http://www.fao.org/aims/ag_intro.htm
[8] http://agclass.nal.usda.gov/about.shtml
[9] http://agricola.nal.usda.gov/
[10] http://www.nal.usda.gov/foodsafety/
[11] http://www.agnic.org/
[12] http://www.ars.usda.gov/
[13] Published on http://www.few.vu.nl/\~wrvhage/oaei2006/

Background knowledge We selected the background knowledge ontologies to faithfully represent the types of background knowledge we set to investigate. We used the Watson[14] ontology search engine to find them. We queried Watson for concept labels from the matching ontologies which are common English terms like *meat*, *animal*, *food*, etc. and selected six ontologies which frequently occurred in the retrieved results and also seemed like reasonable choice for the goal we set to analyze, that is exploring the different background knowledge types. Note that the choice of the search engine is not any special, in other studies different search engines have been successfully used for the same purpose, [9] used Swoogle to dynamically select background ontologies for an ontology matching task.

The selected six ontologies were the following: Economy which models a different but related domain as the matching ontologies; Mid-level, Sumo and Tap which are general knowledge ontologies; and A.com and Surrey which are ontologies of an unknown origin.

- *Background knowledge 1*: Economy ontology is described at `www.daml.org`, I quote: "is based on CIA World Fact Book (2002). Some industry concepts are based on the North American Classification System ('NAICS') - online at `http://www.census.gov/rpcd/www/naics.html`." As its name indicates, it intends to formally describe the domain of economy. It was engineered by Teknowledge Corporation[15] and submitted to the collection of ontologies gathered at `www.daml.org`. The size is 323 concepts.
- *Background knowledge 2*: Mid-level is constructed to play the role of bridge between the Sumo abstract level ontology, and the different varieties of Sumo domain-specific ontologies[16]. It is not domain-specific, and contains 1773 concepts.
- *Background knowledge 3*: Sumo (Suggested Upper Merged Ontology) is being created as part of the IEEE Standard Upper Ontology Working Group. It contains 576 concepts.
- *Background knowledge 4*: Tap as described in [13] is a shallow but broad knowledge base containing basic lexical and taxonomic information about a wide range of popular objects. It is claimed to be independent of a domain, however, a manual inspection indicated that it mainly covers the chemical, machine and electronic industry domains. It contains 5488 concepts.
- *Background knowledge 5*: A.com is an ontology with an unknown origin. By browsing it we got the impression that it has been produced as a result of merging several ontologies. In addition, noticeable are surprising relations such as:

  $$\text{Volume} \preceq \text{Pollution}$$

  which can be seen as an indication that some form of directory structure was the origin of the data. It seems to cover various domains, and its size is 5624 concepts.

[14] `http://watson.kmi.open.ac.uk/WatsonWUI/`

[15] `http://www.teknowledge.com/`

[16] `http://ontology.teknowledge.com/`

– *Background knowledge 6*: Surrey ontology, according to the Watson search engine, originates from the web site `www.surrey.co.uk`. In our analysis we did not manage to trace back its source, the download link does not work and on the web site the ontology is not available. Similarly as in the previous case, parts of its content gave the impression that it was created by transforming a directory structure into an ontology in a straight-forward way. Having no available documentation about how it was created, we treated it as an unknown origin ontology. Its size is 672 concepts.

Background knowledge ontology	Type of ontology	Size in number of concepts
BK_1: Economy	Different domain	323
BK_2: Mid-level	General knowledge	1773
BK_3: Sumo	General knowledge	576
BK_4: Tap	General knowledge	5488
BK_5: A.Com	Unknown origin	5624
BK_6: Surrey	Unknown origin	672

Fig. 3. Properties of the background knowledge ontologies

The six background knowledge ontologies with their properties are summarized on the table in Figure 3. With respect to the common ontology sizes found online [11], they are large sized ontologies.

Evaluation We manually evaluated the results of the matching experiments. As a reference use-case we set the task of document reclassification, which is realistic in this context because the matching ontologies are used for classifying books and articles.

3.1 Experiments

We performed seven experiments in which we matched NALT to Agrovoc. In the first experiment, which served as baseline, we matched the ontologies directly, and in the other six we matched them indirectly using the six previously described background ontologies, one per experiment.

Direct matching (Experiment 1) In the direct matching we combined lexical and structural matching. In the lexical phase the labels were normalized by discarding stop words (*the*, *and*, *an*, *a*) and interpunction, and then matched to one another accounting for different word order and plural/singular form of the words. As a result, the lexical phase produced list of pairs of equivalent concepts. In the structural phase the hierarchical structure of the ontologies was used to induce further matches. The direct matching algorithm is shown in Figure 4.

The set of direct matches is empty in the beginning
1 dmatches $:= \emptyset$

Lexical phase: find equivalent lexical matches
2 **for** every concept pair $X \in \mathcal{C}^{\mathsf{SRC}},\ Y \in \mathcal{C}^{\mathsf{TAR}}$ **do**
3 **if** FullLexMatch(X, Y) **then**
4 dmatches $\Leftarrow (X \xrightarrow{\equiv} Y)$
5 **end for**

Structural phase: use the structure to find more matches
6 **for** every two relations $(X_1 \preceq X_2) \in \mathcal{R}^{\mathsf{SRC}}$, $(Y_1 \preceq Y_2) \in \mathcal{R}^{\mathsf{TAR}}$
7 **if** $(X_2 \xrightarrow{\equiv} Y_1) \in$ dmatches **then**
8 dmatches $\Leftarrow (X_1 \xrightarrow{\preceq} Y_2)$
9 **for** every two relations $(X_1 \succeq X_2) \in \mathcal{R}^{\mathsf{SRC}}$, $(Y_1 \succeq Y_2) \in \mathcal{R}^{\mathsf{TAR}}$
10 **if** $(X_2 \xrightarrow{\equiv} Y_1) \in$ dmatches **then**
11 dmatches $\Leftarrow (X_1 \xrightarrow{\succeq} Y_2)$

Fig. 4. Algorithm for matching ontologies directly.

Even though the direct matching was done using such a simple and rigid technique (no edit distance, or other form of approximation), it produced 6,437 matches between NALT and Agrovoc. This number is comparable to the numbers obtained in the OAEI 2006 [5] on the same test data, where most of the participating matching systems produced between 5000 and 10,000 matches. Hence, our direct matching can be considered relatively successful, and a good base-line to measure the added value of the background knowledge.

Indirect matching (Experiments 2 - 7) In the indirect matching we lexically anchored the matching ontologies to the background knowledge, and then used the hierarchies of the background knowledge to induce the indirect matches. In other words, the indirect matching algorithm can be explained as follows: for two matching concepts we first find their equivalent concepts in the background knowledge (if possible), then check if these background concepts are hierarchically related, and if they are we report an indirect match between the matching concepts. The indirect matching algorithm is shown on Figure 5.

The table on Figure 6 summarizes the results of the anchoring phase showing the number of source and target anchors (NALT and Agrovoc, respectively) established to each background knowledge ontology. The Economy ontology has the highest number of anchors as compared to its size, roughly to about one third of its concepts there are anchors established from the matching ontologies. Contrary, ACom has much fewer anchors relatively to its size, roughly one out of each 90 concepts has anchor established to it. We can also observe from the table that this ratio is variable for the background ontologies of the same origin.

Generally, the number of anchors is much smaller than the sizes of the matching ontologies NALT and Agrovoc, which count in tens of thousands. However,

The set of indirect matches is empty in the beginning
1 $\mathsf{imatches} := \emptyset$

Anchoring phase: anchor SRC *and* TAR *to* BK *using direct matching*
2 $\mathsf{anch}^{S \mapsto B} := \textsc{MatchDirectly}(\mathsf{SRC}, \mathsf{BK})$
3 $\mathsf{anch}^{T \mapsto B} := \textsc{MatchDirectly}(\mathsf{TAR}, \mathsf{BK})$

Deriving relations phase: find indirect matches using the anchors and BK
4 **for** every two anchors $(X \overset{\preceq}{\longmapsto} Z_1) \in \mathsf{anch}^{S \mapsto B}$, $(Y \overset{\succeq}{\longmapsto} Z_2) \in \mathsf{anch}^{T \mapsto B}$
5 **if** $(Z_1 \preceq Z_2)$ **then**
6 $\mathsf{imatches} \Leftarrow (X \xrightarrow{\preceq} Y)$
7 **for** every two anchors $(X \overset{\succeq}{\longmapsto} Z_1) \in \mathsf{anch}^{S \mapsto B}$, $(Y \overset{\preceq}{\longmapsto} Z_2) \in \mathsf{anch}^{T \mapsto B}$
8 **if** $(Z_1 \succeq Z_2)$ **then**
9 $\mathsf{imatches} \Leftarrow (X \xrightarrow{\succeq} Y)$

Fig. 5. Algorithm for matching SRC to TAR indirectly through BK as a background knowledge.

Background knowledge	BK_i size	Source anchors	Target anchors
BK_1: Economy	323	121	106
BK_2: MidLevel	1773	330	271
BK_3: Sumo	576	79	72
BK_4: Tap	5488	367	227
BK_5: ACom	5624	66	69
BK_6: Surrey	672	102	95

Fig. 6. Overview of the anchoring results.

given the sizes of the background ontologies and the fact that they are not agriculture-specific, this anchoring result is not surprising.

The table in Figure 7 gives an overview on the indirect matching results. The third and fourth column show the number of indirect matches, and the number of additional indirect matches which were not found in the baseline direct matching. Each row in the table corresponds to one background knowledge ontology, except for the last one which shows the *cumulative* number of matches (union). Note that these cumulative numbers are not simple sums of the numbers above them, for example for the indirect matches the sum is 2287 and the cumulative number of matches is 2183. They are different because some of the matches are found by more than one background knowledge ontology. Similarly, the sum of the additional indirect matches is 1462 whereas the cumulative number is 1428. We see that the sum and the cumulative number are close to one another, which reveals very important and attractive behavior of using multiple background knowledge ontologies. Namely, different ontologies produce nearly disjoint sets of indirect matches. This means that the more ontologies we use - the more matches we will find. If we look at the cumulative matches, the additional indirect

Background ontology	BK_i size	Indirect matches	Additional matches on top of direct matches
BK_1: Economy	323	259	85
BK_2: MidLevel	1773	200	81
BK_3: Sumo	576	115	57
BK_4: Tap	5488	1003	625
BK_5: ACom	5624	87	71
BK_6: Surrey	672	623	543
Cumulatively all BK_i		2183	1428

Fig. 7. Overview of the indirect matching results, the number of matches established using each background ontology

matches represent 66% of all the indirect matches, which in turn means that an arbitrary indirect match has higher chances to be an addition to the baseline matches. However, these numbers say nothing about the quality of the matches, as a next step we will evaluate their correctness.

3.2 Evaluation

In order to get better insight in the matching process we decided to undertake the effort of manually assessing the matches. As a natural reference we choose the task of document reclassification: the obtained matches are expected to faithfully reclassify the documents from the source to the target ontology, ideally, in the same way as a human would do.

For the precision we did the evaluation as follows: each match was checked for validity, if the correctness was not obvious then Google was used as reference by querying for *define: label* to find the definition of the term *label*. The evaluation of the precision proceeded in two phases: first evaluate the direct and then the indirect matches. For the direct matching which produced more than 6000 matches, we choose the random sampling method. After drawing a random sample of 10% (640 matches), we manually assessed these matches as described above. For the indirect matches, which were in total little bit more than 2000, we took the effort to manually assess all of them.

The recall was hard to estimate because it requires all the correct matches between the matching ontologies available, which we don't have. Therefore we set to observe the change in recall between different experiments instead of estimating the achieved recall.

The evaluation revealed that the direct matching achieved 100% precision, i.e. all the matches in the evaluation sample were correct. The precision of the indirect matching and the change in recall are shown in the table on Figure 8.

Matching experiment	**Precision** indir. matches	**Precision** addit. matches	**ΔRecall**
Exp.2: BK$_1$: Economy	84.17%	51.76%	0.68%
Exp.3: BK$_2$: Mid-level	97.00%	92.59%	1.17%
Exp.4: BK$_3$: Sumo	76.52%	52.63%	0.47%
Exp.5: BK$_4$: Tap	57.23%	31.36%	3.04%
Exp.6: BK$_5$: A.Com	36.78%	22.54%	0.25%
Exp.7: BK$_6$: Surrey	35.63%	26.15%	2.21%
Cumulativly BK$_1$-BK$_6$	57.63%	35.22%	7.81%

Fig. 8. Performance of the indirect matching experiments

4 Analysis

First general observation on the matches (all the matches from all the seven experiments) is that they were established between a small subset of the matching concepts: 2241 in NALT, and 1757 concepts in Agrovoc participated in the matches, as compared to the size of NALT which is 41,577 concepts and Agrovoc 28,174 concepts. The number of concepts which participated in the matching results were in the order of about 5% of the size of the matching ontologies. But, this effect is not peculiarity of our experimental data, in other studies [14, 15] similar effect was noticed when matching the FMA and GALEN ontologies which model the human anatomy. These ontologies have 59,000 and 24,000 concepts respectively, and the number of matched concepts reported in the studies is in the order of 10% of the ontology sizes. It seems that this effect occurs when matching large ontologies even though they model the same domain. Most likely explanation for this is that for the general concepts there is much better naming agreement, while for the more specific ones, which represent the majority, there is almost no agreement. In such a situation the labeling problem is solved by using many words to name a single concept. As an example, in NALT there is a concept named Salmonella choleraesuis subsp.choleraesuis serovar Paratyphi A.

Precision and recall The table on Figure 8 shows the precision of the indirect matches, and the increase of recall with respect to the baseline direct matching. Each row corresponds to one background ontology, except for the last which shows the results for the cumulative use of all the background ontologies together.

All the indirect matches which were also found in the baseline were correct, incorrect matches only appeared when they were not found in the baseline matching. Hence, the precision of the additional indirect matches is lower than the precision of the indirect matches.

The Tap ontology resulted in 57.23% precision, however, a special situation had reduced the precision of this ontology. Many of the matches were wrongly established to the target concept called Node. The root concept in TAP is called Node, and the target concept anchored to it was found related to any source concept anchored in Tap. When these wrong matches are not taken into account,

the precision of Tap is calculated to 92.13%. This example gave a very important insight, the indirect matching can be very sensitive to mistakes which are high in the background knowledge hierarchy. The fact that the root concept of Tap was named Node caused drastic change in the results when we used it as background knowledge.

The first four background ontologies which are expert-created exhibit high precision in the indirect matches (more than 75%), and relatively high precision in the additional indirect matches (more than 50%). On the other hand, the unknown-origin ontologies show lower precision which is not a surprising thing given the low quality of their content.

Observing the recall we see that Tap provides the highest increase of recall, shown in the third column, but the Surrey ontology is the second next to the Tap ontology in the recall increase. While the ontologies of an unknown origin might show low precision, that does not prevent the recall being increased considerably. We also see that Surrey is much smaller than Midlevel, Tap and ACom, which is an empirical proof that small size does not immediately imply low recall.

Causes of wrong matches For the first four background knowledge ontologies there were two main causes for wrong matches: contextual problems and relatively small mistakes. Examples of matches caused by contextual problems are the following:

$$\text{NALT: Meat} \xrightarrow{\preceq} \text{Agrovoc: Product}$$
$$\text{NALT: Vehicle} \xrightarrow{\preceq} \text{Agrovoc: Product}$$
$$\text{NALT: Organism} \xrightarrow{\preceq} \text{Agrovoc: Agent}$$

Meat can be seen as a kind of product in the domain of economy, however, for our matching task this was not a desirable match. These matches can be seen as relations establishing roles, meat and vehicles can have the role of a product, and organism can have the role of an agent. Such modeling is apparently good for the contexts of these background ontologies. For discussions related to the context issues in knowledge representation the reader is referred to the Cyc Knowledge Base [16] and the study of [17][17]. In addition to the context problems, few of the wrong matches were caused by relatively small mistakes, such examples are the matches:

$$\text{NALT: Marine invertebrae} \xrightarrow{\preceq} \text{Agrovoc: Fish}$$
$$\text{NALT: Herbivore} \xrightarrow{\preceq} \text{Agrovoc: Mammals}$$

Jellyfish are kind of Marine invertebrae but they are not fish, and some kinds of birds are herbivore but not mammals. These relations come close to generally accepted claims like "birds fly" while exceptions exist: "penguins are birds, and

[17] The study argues that the knowledge representation issues and the functionality of the system are intrinsically tied to one another, I quote: "Representation and reasoning are inextricably intertwined: we cannot talk about one without also, unavoidably, discussing the other. We argue as well that the attempt to deal with representation as knowledge content alone leads to an incomplete conception of the task of building an intelligent reasoner."

yet they do not fly". We stress here that there were no different causes for wrong matches between Economy and the other three general-knowledge ontologies. The high-quality ontologies, whether they model different domain or are general-knowledge, the same reasons caused them to produce wrong matches when they were applied as background knowledge.

For the last two ontologies, which have unknown origin, mistakes were the cause for the wrong matches. For example:

$$\text{NALT: Gas} \xrightarrow{\prec} \text{Agrovoc: Turbines}$$

$$\text{NALT: Waste} \xrightarrow{\prec} \text{Agrovoc: Water}$$

The concepts in these wrong matches are semantically related, however, no strict relation can be established. These matches are clearly wrong. This suggests that ACom and Surrey were obtained by straight-forward transformation of a directory structure into an ontology.

5 Conclusions

Based on the work presented in this paper, we conclude that using multiple ontologies as background knowledge in ontology matching is useful and practically feasable. Our experiments indicated the key factors that influence the matching performance. The recall increases monotonically with adding more background ontologies. This is an important property because the recall increase is seen as bigger challenge for the current matching systems. For the precision, the success primarially depends on the quality of the background ontologies.

Observing the precision, the expert-created ontologies such as Economy, Mid-level, Sumo and Tap resulted in relatively high precision (more than 75%), and the main causes of wrong matches were contextual differences with the matching ontologies and small mistakes. The ontologies of unknown origin like ACom and Surrey resulted in lower precision (less than 40%) and the main cause of wrong matches were mistakes. This makes the expert-created ontologies more trustworthy and clearly preferable background knowledge candidates over the unknown-origin ontologies with respect to the precision.

All the background ontologies together provided relatively small increase in the recall of about 8% in addition to the direct matching. However, they resulted in nearly disjoint sets of matches, which means that new ontologies are likely to provide new additional matches and further increase the recall.

Furthermore, the expert-created ontologies, regardless whether they modeled different domain from the matching ontologies (Economy) or they were general-knowledge (Mid-level, Sumo and Tap), they resulted in similar matching qualities. On our experimental data we could not discriminate by the precision or the recall increase, and all of them had the same causes of wrong matches.

Finally, the Tap ontology showed that the matching process can be very sensitive to mistakes high in the background knowledge hierarchy. Other mistakes also resulted in wrong matches, but the mistake in Tap with the root concept being labeled Node seriously affected the precision when applying this background ontology.

References

1. Rahm, E., Bernstein, P.A.: A Survey of Approaches to Automatic Schema Matching. VLDB Journal **10**(4) (2001)
2. Shvaiko, P., Euzenat, J.: A survey of schema-based matching approaches. Journal on Data Semantics **4** (2005) 146–171
3. Kalfoglou, Y., Schorlemmer, M.: Ontology mapping: the state of the art. Knowl. Eng. Rev. **18**(1) (2003) 1–31
4. Euzenat, J., Isaac, A., Meilicke, C., Shvaiko, P., Stuckenschmidt, H., Svab, O., Svatek, V., van Hage, W.R., Yatskevich, M.: First results of the ontology alignment evaluation initiative 2007. In: Ontology Matching Workshop at International Semantic Web Conference (ISWC). (2007)
5. Euzenat, J., Mochol, M., Shvaiko, P., Stuckenschmidt, H., Svab, O., Svatek, V., van Hage, W.R., Yatskevich, M.: Results of the ontology alignment evaluation initiative 2006. In: Ontology Matching Workshop at International Semantic Web Conference (ISWC). (2006)
6. Euzenat, J., Stuckenschmidt, H., Yatskevich, M.: Introduction to the ontology alignment evaluation 2005. In: Integrating Ontologies. (2005)
7. Euzenat, J., Le Bach, T., Paolo, Debo, J., Kuntz, R.D., Ehrig, M., Hauswirth, M., Jarrar, M., Lara, R., Dianamaynard, Napoli, A., Stamou, G., Stuckenschmidt, H., Shvaiko, P., Sergiotessaris, Van Acker, S., Zaihrayeu, I.: State of the art on ontology alignment. deliverable d2.2.3 (2004)
8. Aleksovski, Z., Klein, M., ten Kate, W., van Harmelen, F.: Matching unstructured vocabularies using a background ontology. In: Proceedings of Knowledge Engineering and Knowledge Management (EKAW). (2006) 182–197
9. Sabou, M., dAquin, M., Motta, E.: Using the semantic web as background knowledge for ontology mapping. In: Proceedings of Ontology Matching Workshop. (2006)
10. Bouquet, P., Serafini, L., Zanobini, S.: Semantic coordination: A new approach and an application. In: International Semantic Web Conference (ISWC). (2003) 130–145
11. Wang, T.D., Parsia, B., Hendler, J.A.: A survey of the web ontology landscape. In Cruz, I.F., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L., eds.: International Semantic Web Conference. Volume 4273 of Lecture Notes in Computer Science., Springer (2006) 682–694
12. Baeza-Yates, R., Ribeiro-Neto, B., et al.: Modern information retrieval. Addison-Wesley Harlow, England (1999)
13. Guha, R., McCool, R.: Tap: A semantic web toolkit. Semantic Web Journal **1**(1) (2003) 81–87
14. Zhang, S., Bodenreider, O.: Aligning representations of anatomy using lexical and structural methods. AMIA Symposium (2003) 753–757
15. Mork, P., Pottinger, R.A., Bernstein, P.A.: Challenges in precisely aligning models of human anatomy using generic schema matching. Technical report, Microsoft Research (2004)
16. Lenat, D.: Cyc: a large-scale investment in knowledge infrastructure. Communications of the ACM **38**(11) (1995) 33–38
17. Davis, R., Shrobe, H., Szolovits, P.: What is a knowledge representation? AI Magazine **14**(1) (1993) 17–33

Flickring Our World: An Approach for a Graph Based Exploration of the Flickr Community

João Tojo, Jorge Sousa and Paulo Gomes

{tojo, pelaio}@student.dei.uc.pt
pgomes@dei.uc.pt

CISUC – DEI/FCTUC University of Coimbra
Polo II – Coimbra – Portugal

Abstract. When using the Web, one begins to understand and acknowledge the existence of the "other side of the mirror". The Internet has become a means of inter-relationship between its users, usually forming complex and emergent relationships, providing a handful of knowledge waiting to be extracted. In this paper, we present a tool that analyzes the Flickr community using semantic web technologies. It uses the FOAF ontology to present the result of the analysis as graphs based on the relations between the users of this community.

Keywords: Social Networks, Flickr, FOAF, Semantic Connections, Web 2.0, Semantic Web.

1 Introduction

The Web, as we know it today, is made of – and by – everything. By everything we mean *everything*: plain text, pictures, movies, and so on. However, one thing that is missing is meaning. Every one of these things has the meaning that we provide to them, but none has the underlying meaning that can be recognizable by machines, necessary in order to do some computation and reason about the single most powerful media mechanism known to men. The science of semantic web [1] has this role, to provide a means to compute web resources, providing them the semantics and the structures in order to use them in an intelligent fashion.

One of the main focus of this fairly new science is the contribution to the study of social sciences [2]. The communities that originate from the main web concentration points are very rich in semantic information, as they self organize themselves and their interests by using a tagging mechanism [24]. These virtual societies, by using tags, provide an adaptive taxonomy – the folksonomies [3] – that allows the emergence of social trends, by analyzing the movement and creation of clouds of information. These clouds, composed by terms exclusively created by users, provide a way to verify trends, relations and future directions of people, businesses, products, and so on, thus providing a way of analyzing the hidden features and relations of the web. Our work will focus on developing a tool that allows the analysis of such phenomenon regarding the Flickr environment.

Using Flickr [4] – the famous photo blogging site – as our base of information, we will make use of the API [5] provided by the site to compute users, the pictures they upload and, most of all, the tags they associate with each picture. This information will be structured using a specific ontology [6] named FOAF [7, 8, 9] – friend-of-a-friend –, used mostly on web analysis of societies, with the focus on connections between users. We will treat this information as RDF graphs [25] and try to make a simplified approach to the, so called, second generation clouds [10, 11], by presenting visual, interactive graphs representing the semantic landscape created by the users of the community.

In the present paper, we begin by introducing Flickr and its community, discussing about the concepts behind the creation and usage of the site. We will then make a brief introduction to social network analysis, referring the history of the field and its area of work, and exposing some key concepts behind a social analysis of a network. The following section will introduce our approach to the developing of the tool, by discussing the main structure, followed by the explanation of the algorithms using during the graph computation/creation. We will then expose some of the experimentations made, including its results and observations, and address some future work to be made related to this work. In the end, we round up some conclusions to the work performed and the results obtained.

2 The Flickr World

Developed by Ludicorp [12, 13], the Flickr project was an evolution of a project named Game Neverending [12, 13], a massive multiplayer online game developed in order to provide a platform for an enhanced real time online interaction, in the form of a role playing game based on social interaction. As the project was cancelled, its ideias and tools were used in order to build the Flickr website. Nowadays owned by Yahoo, Ludicorp developed what is now called one the first Web 2.0 applications [12] and, certainly, one of the most famous.

The Flickr website provides a way for people to, not only share their photos with the other users of the site, but also for these users to collaborate in the definition of the meaning of these photos, making use of the tagging phenomenon. As such, one can use Flickr to search for photos depicting a certain color, place or whatever other characteristics, by searching these tags. This tagging is one of the best examples of the use of folksonomies (although there are opposing opinions [14, 15] of this view regarding the Flickr context) , where the users define, evolve and adapt the taxonomy used for a determined context, instead of using a predetermined, fixed and static set of terms.

Although the Flickr website provides for the identification of the users (direct) relatedness, either by being able to (co)exist in each others contact lists or by belonging to the same group, this tagging mechanism may allow for a higher level connectivity assessment approach. Regarding the growing relevance of the Flickr website, and the also growing diversity and number of people participating both in sharing the photos and labeling them, we see the Flickr community as a good possible source of social network analysis.

3 Social Network Theory and the Web

Social science [18] is a thoroughly studied field, regarding what the humans do as specie. Focusing on scientific methods, based on psychology, sociology, and so on, the social sciences are aimed at studying and understanding the human behavior [18]. However, although the traditional view of social sciences focuses on the rational choices made by individuals, it does not regard the aspects of interrelation between those actors [16].

As such, a new field emerged in order to address this issue, named social network theory (or analysis). This field focuses mainly on the social context of the actors and the behavior of their relationships [16], identifying the underlying patterns on these relationships. There is actually a debate [16] whether the social network analysis should be an independent, self contained field or, on the other hand, a subset of social studies, a set of collection methods and studies in order to use in the study of social, human relations (centered on the individual). However, the complexity of the structures of these social networks may induce that it is, indeed, a whole new science [19], that although has its roots and some of its foundations in the traditional social sciences, tries to form its own theories regarding a complex concept.

Social network analysis (SNA) is based on the concept that there are determinable structures behind the formation of the network of agents and their relationships [17]. This concept has originated many theories regarding people and how they organize themselves in networks, being the most famous the “six degrees of separation” theory [20], with roots on the small world models [21].

The social networks field had three main influences [16]: sociometric analysis (graph theory models), mathematical analysis and the anthropology view on the structure of communities. During the 60’s and the 70’s, however, a Harvard congregation on these influences created the general field known as SNA.

SNA regards three main points of investigation [16]:

1) The total structure;
2) The subsets of a determined group;
3) The individuals (as “points”, “vertices” or “nodes”) that form the network;

In order to address these points, SNA also defines a set of concepts regarding the study of networks [16]:

- *Dyad*: Two actors how have a connection, meaning they have a relationship;
- *Clique*: A subset of actors within a network who have ties with all other actors between the subset;
- *Density*: The proportion of total available ties connecting actors;
- *Centralization*: The fraction of main actors within a network;
- *Reachability*: The number of ties connecting actors;
- *Connectedness*: The ability of the actors to reach one another reciprocally, that is, the ability to choose a relationship between both parties;

- *Asymmetry*: The ratio of reciprocal relationships – the mutual relationships – to total relationships within the network;
- *Balance*: The extent to which ties in the network are direct and reciprocated;

Regarding the study of individuals in the network, there are a few more concepts to consider [16]:

- *Centrality*: The degree to which an actor is in a central role in the network;
- *Homophily*: The degree to which similar actors in similar roles share information;
- *Isolate*: An actor with no ties to other actors;
- *Gatekeeper*: An actor who connects the network to outside influences;
- *Cutpoint*: An actor whose removal results in unconnected paths in the network;

3.1 The Web and its Dual Relation with SNA

The Internet boom has provided a very rich and diverse arena for the study of social networks. In fact, it was the development of the World Wide Web (WWW) that provided for a wider application of the field, accompanied by further developments on theories, models and concepts. The presence of logs, blogs, forums and registers of all kinds provide for very useful data to be used in SNA [17].

SNA has provided some interesting studies (and consequent results), using traditional methods and targets of study. The Internet has evolved this study and its scope one step further: the globalization of the population has provided for a much wider sense of connections, regarding aspects from preferences of music, food and films to ideologies and ways of living. In a sense, we can say we are in the presence of the *meta level* of social networks, where the localization of the agents no longer matter, due to the world virtualization of the WWW [23].

If, at first, there was an underlying fear of the eventually ephemeral Internet base connections, the appearance of Internet communities proved that the aforementioned relationship arena was strong, cohesive and robust, and, as such, ready and perfectly suited for the SNA field [22, 23]. The fact that online communities often change its paradigms (the *social trends* shifting) provides yet another dimension for this social analysis, as the networks *evolve* in time (and space, eventually). As such, SNA has another characteristic to study (evolution of online networks) and a playground to test and infer theories.

On the other hand, social network concepts became so famous that, nowadays, we have companies and businesses based solely on them, regarding networking sites such as Orkut, Flickr, MySpace, etc. [16].

4 Developing a Tool for Social Network Analysis in Flickr

In this section, we describe our work presenting several key points of the project, such as the ontology used, the architecture built and the algorithm and structures included for the generation of graphs. The goal is the development of a tool that allows for the modulation, parameterization and navigation on graph representations of the Flickr social networks.

4.1 Architecture

In order to build a tool for both constructing and analyze the graphs representing the connections between the Flickr users, we need to perform the task of data extraction from the Flickr database. After this preliminary step, we needed some way of constructing the models in order to apply the algorithms, and then a dynamic fashion on representing the graphs. As so, the application consists, mainly, on three different tiers, as represented on figure 1.

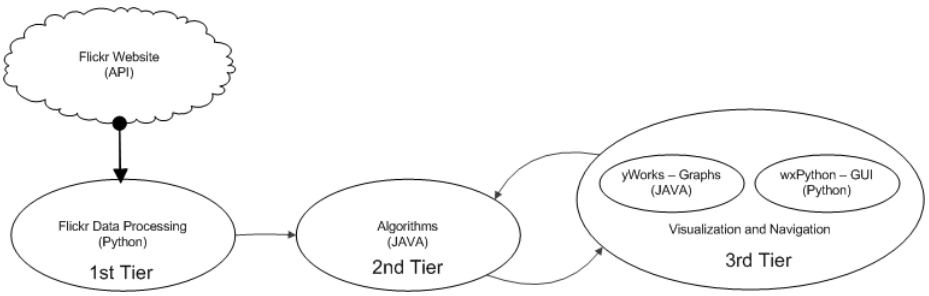

Figure 1: Overall architecture of the tool

The first tier (Flickr Data Processing) is responsible for querying the Flickr API service, while storing this data in a local MySQL database. The algorithms for implementing these queries are built using the Python language [31], and the focus is based mostly on the task of processing the XML data originated by the Flickr API upon a data request.

The second tier (Algorithms) is in charge of all data manipulation and algorithmic functions. This tier uses the data retrieved from the first tier for the construction and storing of RDF [25] graphs, which are then used for SPARQL [26] querying. These queries are built upon the needs of the data used in the main algorithms, responsible for performing the assessment of the relations between the Flickr users. Related to this tier is the creation of a virtual matrix representing all the connections between the Flickr users (stored in the local database). This structure will be used in the third tier. This second tier is implemented using a Java application, including the JENA [27] package in order to perform the RDF construction and querying.

The third tier (Visualization and Navigation) uses the previously mentioned virtual matrix to perform the mapping of the matrix to a visual graph. Using the YWorks

Java package [28], we (re)present the results derived from the implemented (and selected) algorithms to a graph of nodes representing the users. The layout is built in order to depict visually the connectivities between the Flickr users in the graph. This visualization is inserted in an interface allowing the user to define certain parameters of the presented graph. An example of a graph is shown in figure 2. This third tier comprises another visual element, allowing the algorithms to be selected, as previously mentioned. As the goal of this work is to build a tool for social graph analysis, we developed a Graphic User Interface (GUI), allowing the user to both choose from a different set of algorithms and insert the desired parameter values regarding the formulas used in the assessment of the relatedness of the Flickr users. This GUI is built using a Python graphic package, wxPython [29].

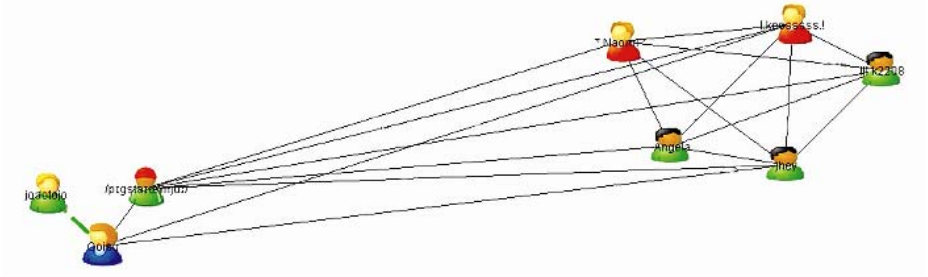

Figure 2: An example of a graph created with yWorks and a subset of Flickr data.

4.2 Ontology

In order to analyze the Flickr data, we use a base structure that can provide for the needs on asserting relationships between actors, which can be either persons or objects. Not only for that reason, but also because of principles of good engineering, the use of an ontology is an obvious choice. For that we used the FOAF [7] (friend-of-a-friend) ontology, because it provides not only defined structures for persons, activities and properties, but also the means of performing simple and direct connections between them and their related objects. This ontology enables the identification of relationships within groups of people and relations between people and resources (locations, films, photos), and people and their activities (blogging, tagging). FOAF ontology not only provides these direct assertions, but also lays the foundation for reasoning about some higher level relationships, not so clear at a first glance, such as the relation between actors based on the properties of their activities, or the objects they create.

It is with one of these higher level relationship identifications in mind (namely, the tagging relationship) that we use the FOAF ontology. However, in order to use the ontology within our project, we had to perform some modifications, for example, including specific terms related to our work and concerning the Flickr activities and properties.

As stated above, the use of FOAF ontology is considered to be important from a conceptual point of view.

Despite that, the technological perspective should not be forgotten as it provides the possibility to make direct reasoning with the Flickr data and prepares it for future applications in terms of scalability. The ontology used in the present work is depicted on figure 3.

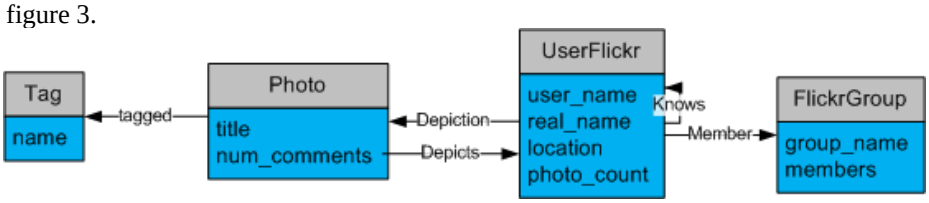

Figure 3: Part of the ontology used in the current work.

4.3 Algorithms for Asserting Connections on the Network Creation

We have used a formula to compute the proximity between users, which takes in consideration 3 factors:

4) If the users are contacts of each other;
5) If the users are in the same group;
6) The similarity asserted from the tags related to each of the users;

As so, the main formula is presented on formula 1.

$$con(u1,u2) = w1 * ContactS(u1,u2) + w2 * GroupS(u1,u2) + w3 * TagS(u1,u2) \quad (1)$$

With: ContactS = Similarity as a contact, GroupS = Similarity as in the same group, TagS = Similarity by tags and, preferentially, w1+w2+w3 = 1, in order to normalize values.

4.3.1 Contact Similarity

on the contact similarity is computed based on the user's contacts. However, this relation may not be bidirectional, as the user can have a contact that doesn't have the user as its own contact. As such, we specify this option as a parameter (p1), so it can be chosen how to deal with this relation. If p1 is true, the relation is assumed to be directional, otherwise it is assumed bidirectional. The following formulas, 2 and 3, represent, respectively, both these cases.

$$p1 = false \rightarrow \quad contactS = u1 has(u2) \quad (2)$$

$$p1 = true \rightarrow \quad contactS = (u1 has(u2) + u2 has(u1)) / 2 \quad (3)$$

With: u1has(u2) = Verification if u1 has u2 on its contact list (equals 1) or not (equals 0) and u1has(u2) = (0;1).

4.3.2 Group Similarity

As a user can belong to several different groups, we've considered this value to represent a higher relation between users if the number of similar groups of the users is higher. So, the formula has in consideration not only the number of similar groups between the users, but also the amount of groups of the first user, as shown in formula 4.

$$groupS(u1,u2) = \frac{\sum similarGroups(u1,u2)}{\sum groups(u1)} \quad (4)$$

With: similarGroups(u1, u2) = Number of groups similar between u1 and u2, and groups(u1) = Number of total groups of u1.

4.3.3 Tag Similarity

Tag similarity is the most complex relation between the users, as it is the only one that is not directly retained when consulting the data. We could have used several approaches to tackle this issue: we have chosen, however, one that simplifies this relation by relating the tags associated with the photos of each user.

The algorithm runs each user list of tags present in her/his photos, comparing the values and counting the similar cases. We've also implemented a step deeper, by using the Flickr API function, which returns the related tags of a specific tag. We add this into the equation by, whenever a tag between two users does not relate, discovering the related tags (as "stated" by Flickr) and finding a relation between them. As this isn't a direct relation, and regarding the fact that Flickr already provides in its response for a classification of the most related, we infer a linearly descendant weight for each tag on the list of similar.

Two formulas were implemented for the direct relation between tags and one for the indirect relation. The weight of these two relations can be user defined.

For the direct tag similarity, we have formulas 5 (Record Semantic Proximity (RSP) [30]) and 6 (self named Empiric Semantic Proximity (ESP)).

$$tagS(u1,u2) = \frac{\sum similarTags(u1,u2)}{\sum tags(u1) + \sum tags(u2) - \sum similarTags(u1,u2)} \quad (5)$$

$$tagS(u1,u2) = \frac{\frac{\sum similarTags(u1,u2)}{\sum tags(u1)}}{\frac{1-\sum tags(u1)-\sum tags(u2)}{\sum tags(u1)+\sum tags(u2)}} \quad (6)$$

With: similarTags(u1, u2) = Number of equal tags in tag lists of user 1 and user 2, and tags(uX) = Number of total tags in tag list of user X.

The difference between RSP and ESP resides mostly on its concept. While RSP accounts for the similarity of both the users on a joint context, asserting it as a whole, ESP looks at this similarity from a "point of view" of u1, giving the tag relation a slight different meaning.

Regarding the indirect tag relationship, the value is computed according to the following steps:

1) For every tag " tagU" from user1, if it's not on user2 tag list, for every "tagY" in user2 tag list, get "tagY" list of similar tags (according to Flickr), checking if this list contains "tagU";
2) If so, add to the counter the weight related to the position of the tag on the list. For example (assuming a base value of 0.8 and a threshold of values 0.01):
    a. ['cat':0.4, 'dog':0.2, 'bird':0.05, 'meow':0.025]
3) Divide the sum value obtained for the multiplication between the number of sub lists compared with and the maximum value of each sub list. See formula 7, for the given example.

$$indirectRelation = sum / (0.4 * nAnalyzed) \quad (7)$$

With: sum = All the "contributions" from the different sub lists. And nAnalyzed = the number of sub lists analyzed.

As a final result, both the direct and indirect tag relations are taken into account (if that is the desire of the user), regarding formula 8.

$$tagS = w1 * DirectSim(u1,u2) + w2 * IndirectSim(u1,u2) \quad (8)$$

With: DirectSim(u1, u2) = The value of the direct tag relations (formula 5 or 6), IndirectSim(u1, u2) = The value of indirect tag relation and, preferentially, w1+w2 = 1, in order to normalize values.

4.4 Adjacency Matrix

The result of the previous formulas is a value for each pair (UserX, UserY). The value of the pair (User1, User2) *may not be the same* as the value of the pair (User2, User1). For instance, if it is decided not to use the complete approach when asserting the users contact similarity, and User1 has User2 as a contact while the reverse is not true, then User1 will have a weight of 1 in that relationship, while User2 will have a weight of 0 in the same calculus. As the graphs to be constructed are spatially visualized, we had to, somehow, normalize this distances. As so, we opted for the following way: for each pair (of pairs) ((UserX, UserY), (UserY, UserX)), we assign the mean value, regarding, each sub pair. For example, let's say that con(User1, User2) = 0.83 and con(User2, User1) = 0.25. The assigned value for this (double) relation would be (0.83+0.25)/2 = 0.54.

In the end, the matrix will look like the one represented in table 1.

Table 1. The Adjacency Matrix

P1\P2	Person1	Person2	Person3	Person4	Person5	Person6	Person…
Person1	NA	0.54	0.67	0.04	0.08	0.34	…
Person2	0.54	NA	0.78	0.23	0.51	0.001	…
Person3	0.67	0.78	NA	0.02	0.045	0.43	…
Person4	0.04	0.23	0.02	NA	0.78	0.12	…
Person5	0.08	0.51	0.045	0.78	NA	0.28	…
Person6	0.34	0.001	0.43	0.12	0.28	NA	…
Person…	…	…	…	…	…	…	…

Note: "NA" means the value is "not assigned", as it wouldn't make sense to connect a person to itself.

5 Experimentations

In order to create a reasonable experience, where we could visually control and manage the data with ease, we've extracted around 200 Flickr users, chosen randomly, with their tags, contacts and photos. From that information, we created a set of graphs, modifying some of the parameters.

The base parameters, used in formula 1, are, respectively, 0.2, 0.2 and 0.6. This means that the contact relation has a weight of 0.2, the same for the group relation. The tag relation, the one we intend to perceive the most, has a weight of 0.6. Figures 4

and 5 depict the graphs obtained, with both the RSP formula and the ESP formula. The indirect relation from tags was not used in these computations.

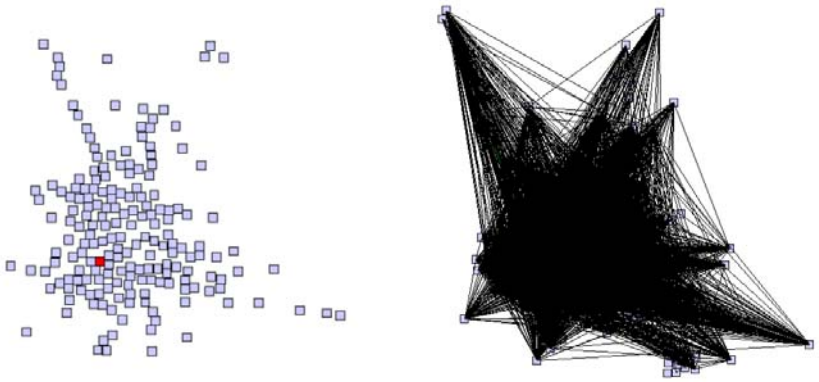

Figure 4: Graph generated with the RSP formula. Graph with the connections on the right and just the nodes on the left.

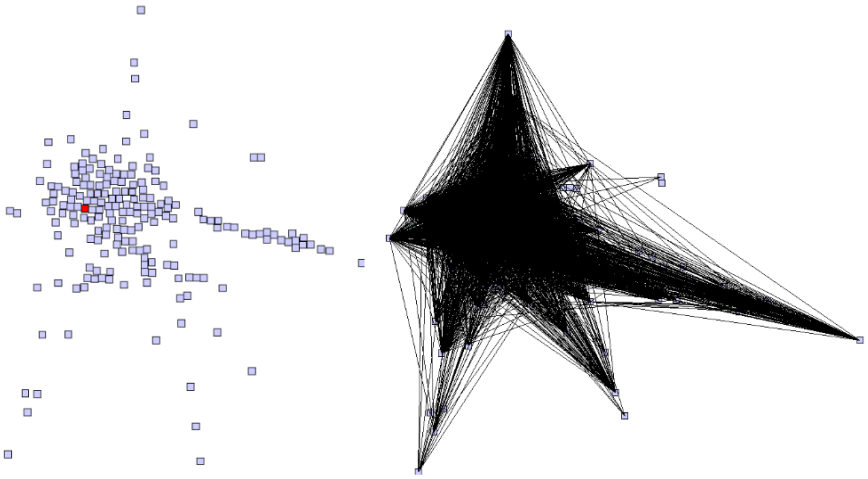

Figure 5: Graph generated with the ESP formula. Graph with the connections on the right and just the nodes on the left.

We varied the base parameters in order to use the values 1.0, 0.0 and 0.0, meaning that only the contact factor is represented in the graph. The resulting graph is shown in figure 6.

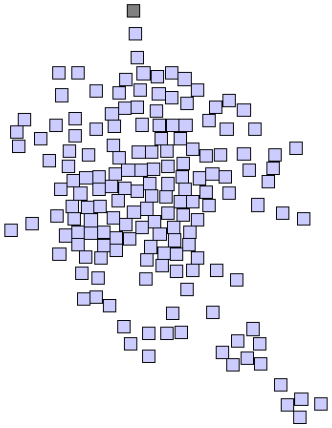

Figure 6: Graph generated by giving only significance to the contacts relation.

6 Observations

Regarding our experimentations, we can observe some relevant aspects:

- *Graph Modulation*: Both figures 4, 5, and 6 show that we can modify the topology of a graph representing the same data, by varying the parameters that construct that graph. This aspect allows graph modifications related to the aspects more relevant to the people analyzing the graphs. In the present case, one could prefer to study the dynamics of a network fully based on Flickr contact connections (figure 6), while, on the other hand, a study could be performed where all the variables are inserted into the graph construction (figures 4 and 5).

- *Visual properties*: If we observe the graphs on the previous section, we can easily identify a few characteristics of networks. In all the examples (figures 4, 5 and 6) we are able to visually identify sets of people grouped together, forming a subset of the network – *clusters* – we can also spot some subsets further away and separated from the others, forming some sort of *islands*; some of the persons are farther away from the main group of people, forming a set of individuals that we can call of *outliers*.

- *Graph navigation*: Another goal of the project was over the second generation view on this sort of graphs. In figure 7 is shown an example of navigability on the graph. By selecting a node in the graph, one can perform a sort of navigation by "travelling" through the neighbors of each node. When a node is selected, its directed connections will be highlighted giving a visual output of the "road". By performing this sort of navigation, we enhance the study of networks on the individual level [16], allowing the possible analysis of *individual's connections*

and characteristics of a determined individual, either as a “real” person or as a role on the network.

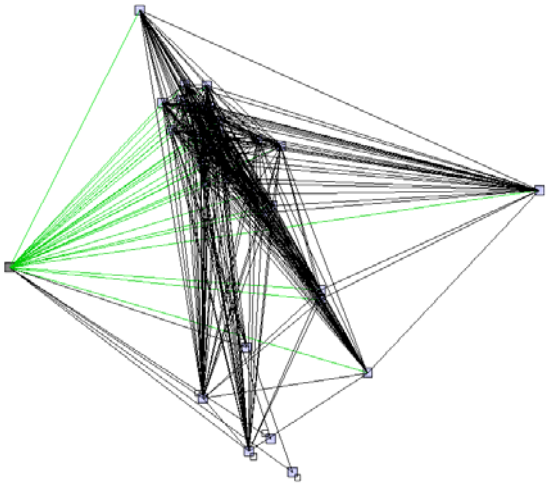

Figure 7: Example of navigability on the graph, while selection the leftmost node.

7 Future Work

The current work, presented in this article, was a first step towards an interactive tool for the science of SNA. The principal basis of the work is already implemented, with a project that allows for the creation and visualization of graphs representing social networks. However, and precisely because this work is a basis for future work, there are a few features that can be enhanced in the project.

One of the main augmentations of the work will be the inclusion of methods for explicit social network analysis, where we are able to withdraw statistics for inference on characteristics like density or reachability of the graph. On the other hand, the inclusion of different layers on the second generation aspect will allow for a concrete visualization of the proximity based on specific aspects. One example on the current work is the inclusion of tag clouds, where the location of the persons on the network is accompanied by a set of tags, representing the semantic landscape on another level. These two additions to the current work will allow for a deeper theoretical validation on the graphs constructed with the tool,

Regarding the construction of the graph, other elements can be included into the calculations of proximities. One can include another level of tagging, by considering the tags related by proximity (Flickr API provides this list), or by including metadata from the pictures (when the API provides methods for that goal). Other formulas can also be included, in order to experiment with different topologies.

As we validate the constructed graphs, trying larger sets of data may provide for richer network environments. We believe that the built tool will handle these larger sets of data without much effort, thus achieving a good level of scalability.

8 Conclusion

The work performed and described in this article provides a step further on the SNA. Using the basis of semantic web, we've developed a project that reveals as a core engine for an auxiliary tool for SNA. We've proved the creation of the graphs depending on the desired set of parameters is achieved. At the same time, the graph navigation is already possible, as the main work is already produced; the additional methods that allow for a specific analysis can be included in the tool.

Several questions also emerge from the produced work: what other elements from web 2.0 can be used along with the web semantics in order to produce valid results and conclusions? What metadata is valid, and what isn't? On the other hand, how can this SNA and its networks enhance the web 2.0 experience? Although the answers to questions of these sorts are not evident, the solution to them is to investigate and aid in the investigation of SNA in the context of the Internet. And tools as the one described in this work are a precious aid in this goal.

References

1. W3C Semantic Web Activity, 1994-2008, accessed 10 January 2008, <http://www.w3.org/2001/sw/>
2. Mika, P.: Social Networks and the Semantic Web, Springer (2007)
3. Mathes, A.: Folksonomies - Cooperative Classification and Communication Through Shared Metadata, University of Illinois Urbana – Champaign, December 2004
4. Flickr website, 2004-2008, accessed 12 February 2008, <http://www.flickr.com>
5. Flickr services API, accessed 12 February 2008, <http://www.flickr.com/services/api/>
6. Gruber, "Ontology", *Encyclopedia of Database Systems*, 2008, Springer, accessed 1 January 2008, <http://tomgruber.org/writing/ontology-definition-2007.htm>
7. FOAF Project, 2000-2008, accessed 8 December 2007, <http://www.foaf-project.org/>
8. FOAF vocabulary specification, Vers. 0.91 on 2 November 2007, accessed 8 December 2007, http://xmlns.com/foaf/spec/
9. Li Ding, Lina Zhou, Finin, T., Joshi, A.: How the Semantic Web is Being Used: An Analysis of FOAF Documents. In: System Sciences, 2005. HICSS '05. Proceedings of the 38th Annual Hawaii International Conference on 03-06 Jan. 2005 pages: 113c - 113c
10. Hassan-Montero, Y., Herrero-Solana, V.: Improving Tag Clouds as Visual Information Retrieval Interfaces. In: I International Conference on Multidisciplinary Information Sciences and Technologies, InSciT2006. Mérida, Spain. October 25-28, 2006.
11. Lamantia, J., "Second Generation Tag Clouds", Online posting 23 February 2006, accessed 14 January 2008, <www.joelamantia.com/blog/archives/ideas/second_generation_tag_clouds.html>
12. Flickr page on Wikipedia, accessed 13 February 2008, http://en.wikipedia.org/wiki/Flickr
13. Jefferson Graham, "Flickr of idea on a gaming project led to photo website" Online posting 2 February 2006, accessed 18 February, http:// www.usatoday.com/tech/products/2006-02-27-flickr_x.htm

14. Guy, M., Tonkin, E.: Folksonomies – Tidying Up Tags?. In: D-Lib Magazine, January 2006, Vol. 12 N. 1.
15. Vander Wal, “Floksonomy Research needs cleaning up”, Online posting 17 January 2006, accessed 17 December 2007, <http://www.vanderwal.net/random/entrysel.php?blog=1781>
16. Fredericks, K.A., Durland, M.M.: The Historical Evolution and Basic Concepts of Social Network Analysis. In: New Directions for Evaluation, Special Issue: Social Network Analysis in Program Evaluation, February 2006.
17. Elizabeth F. C., Christine A. H., "Guest Editors' Introduction: Social Networks and Social Networking," IEEE Internet Computing, vol. 9, no. 5, pp. 14-19, Sept/Oct, 2005
18. Efferson, C., Richerson, P.J.: A Prolegomenon to Nonlinear Empiricism in the Human Behavioral Sciences
19. Scott, J. P.: Social Network Analysis: A Handbook. Sage Publications Ltd; Second Edition ed., March 2000
20. Watts, D. J.: Six Degrees: The Science of a Connected Age, Norton, New York, 2003
21. Watts, D. J.: Networks, Dynamics and the Small-World Phenomenon, American Journal of Sociology 1999 105:2, 493-527
22. Mynatt, E. D., et al: Design for Network Communities. In: Proceedings of the SIGCHI conference on Human factors in computing systems, Atlanta, Georgia, United States Pages: 210 – 217, 1997
23. Mynatt, E. D., et al: Network Communities: Something Old, Something New, Something Borrowed... . In: Computer Supported Cooperative Work (CSCW) Journal, Volume 7, Numbers 1-2, Pages 123-156, March, 1998
24. Shadbolt, N., Hall, W., Barners-Lee, T.: The Semantic Web Revisited. In: IEEE Intelligent Systems, 21 (3). pp. 96-101. ISSN 1541-1672, 2006
25. RDF Semantics, 1999-2004, accessed on 2 December 2007, http://www.w3.org/TR/rdf-mt/
26. SPARQL Query Language for RDF, 2006-2008, accessed on 3 January 2008, <http://www.w3.org/TR/rdf-sparql-query/>
27. JENA, Vers. 2.5.5, size 22M, latest version on 18 January 2007, accessed on 10 February 2008, <http://jena.sourceforge.net/>
28. YFiles, Vers. 2.5.0.4, size 28M, accessed on 1 January 2008, <http://www.yworks.com/en/index.html>
29. wxPython, Vers. 2.8.7.1, size 8M, accessed on 1 December 2007, <http://www.wxpython.org/>
30. Rocha, L. M., Bollan, J.: An Adaptive System Approach to the Implementation and Evaluation of Digital Library Recommendation Systems. In: Fourth European Conference on Research and Advanced Technology for Digital Libraries, December 2000
31. Python, Vers. 2.51, size 10M, accessed on 1 December 2007, <http://www.python.org/>

Semantically Enriching Folksonomies with FLOR

Sofia Angeletou, Marta Sabou, and Enrico Motta

Knowledge Media Institute (KMi)
The Open University, Milton Keynes, United Kingdom
{S.Angeletou, R.M.Sabou, E.Motta}@open.ac.uk

Abstract. While the increasing popularity of folksonomies has lead to a vast quantity of tagged data, resource retrieval in these systems is limited by them being agnostic to the meaning (i.e., semantics) of tags. Our goal is to automatically enrich folksonomy tags (and implicitly the related resources) with formal semantics by associating them to relevant concepts defined in online ontologies. We introduce FLOR, a mechanism for automatic folksonomy enrichment by combining knowledge from WordNet and online ontologies. We experimentally tested FLOR on tag sets drawn from 226 Flickr photos and obtained a precision value of 93% and an approximate recall of 49%.

1 Introduction

The popularity of many Web2.0 applications such as Del.icio.us[1], Flickr[2] and YouTube[3] has led to a massive amount of freely accessible, user contributed and tagged content. Despite the presence of tags, the lack of structure and explicit semantics hampers the creation of intelligent user interfaces for annotation, navigation and querying and the integration of content from diverse and heterogeneous data sources. A popular hypothesis, expressed by many web experts ([4, 8, 9, 11, 17]), is that Web2.0 data sources can be used more efficiently by structuring and semantically organising them and that the Semantic Web can provide the needed semantics to achieve that.

This hypothesis motivated two different research approaches to enrich folksonomies. First, some methods rely on the statistical analysis of tagspaces based on tag co-occurrence to identify clusters of related tags. In this cases the meaning of a tag is given by its cluster but it remains implicit, i.e., it is not explicitly stated [3, 15, 16, 20]. Second, recent methods shift from this statistical view to a knowledge-intensive approach where a semantic definition of tags is obtained by aligning them to a knowledge source [13, 10] . The majority of works use WordNet to define the semantics of tags for organizing resources or enhancing their navigation.

Our work is part of the second type of approaches, with the difference that we rely on all online available ontologies as a background knowledge source to

[1] http://del.icio.us
[2] http://www.Flickr.com
[3] http://www.youtube.com

define the meaning of tags. In this paper, we present the **FLOR**, **FoLksonomy Ontology enRichment**, algorithm which takes as input a set of tags (either the tagsets of individual resources or the clusters derived by the statistical analysis of folksonomies) and automatically relates them to relevant semantic entities (classes, relations, instances) defined in online ontologies. An immediate advantage of this correlation between tags and semantic entities is that the tag is automatically associated with the semantic neighborhood provided by the corresponding ontology. For example, for the tag `canine` apart from identifying that *Canine* `SubClassOf` *Carnivore* we also acquire the knowledge that *Canine* `DisjointWith` *Feline*.

In the following we describe the related work (Section 2), our methodology (Section 3) and discuss our experimental results (Section 4). We conclude and elaborate on future work in Section 5.

2 Related Work

Since the term *folksonomy* was coined, research has focused on comprehending the inherent characteristics of folksonomies and exploring their emergent semantics. Two of the primer works exploring and analysing their structure, the types of their tags and the user incentives in tagging are described in [7] and [14]. Additionally, there are two main lines of folksonomy related research.

Early works on folksonomies are based on the assumption that frequent co-occurrence of tags translates to tag association ([3, 15, 16, 20], see [18] for a detailed analysis of the specific methods). They use various statistical methods to identify clusters of related tags without defining the exact relations among them. An exception is the work detailed in [18], where, in addition to clustering the tags, the semantic relations among them are identified.

The second research line focuses on the semantic definition of tags, primarily by using WordNet. For example, [13] try to identify the meaning of tags in order to enrich the relevant resources with RDF descriptions. The authors distinguish six conceptual categories of tags in Flickr. Using WordNet and other knowledge resources for these conceptual categories they organise the tags accordingly. Then they enrich the Flickr photos with RDF triples created for each of the tag categories. These triples are generated either by predefined predicates or from WordNet signatures depending on the categories they belong to.

The authors of [10] describe a method that expands the related tags clusters of Del.icio.us with more related tags based on co-occurrence. The expanded clusters are presented as navigable hierarchical structures or semantic trees. These semantic trees are derived from WordNet. Using a combination of WordNet based metrics they identify the possible WordNet sense for each tag. Then they extract the path of this tag from the WordNet hierarchy and they integrate it into the semantic tree of the tag's cluster.

The TagPlus system described in [12] uses WordNet to disambiguate the senses of Flickr tags by performing a two step query. First a user looks for a tag, then the system returns all the possible WordNet senses that define the tag and

the user selects (disambiguates) which sense he meant. Finally the system looks for all the Flickr photos tagged with this tag and its synonyms.

T-ORG ([1]) performs ontology based organisation of Flickr photos into a set of predefined categories according to the tags describing them. At first the user selects an ontology of interest. Then, the system extracts the concepts and tries to identify semantic relatedness between these concepts and the tags by querying the web with various linguistic patterns between them. Then each tag is categorised under a superclass of the concept to which was more related by the web search.

All the aforementioned works present methods for tag disambiguation, resource organisation and tag cluster enrichment. Our work aims to address the following additional issues. First, the existing works require some initialising from the user's side (e.g., a priori selecting ontology or knowledge resources for the relevant categories of tags) or they require the user contribution to perform the disambiguation of the tags. FLOR is aimed to run entirely *automatically* (i.e., without user contribution). Second, FLOR exploits more than one resources (all the online ontologies and WordNet) aiming to achieve higher coverage of tags compared to the coverage from single resources. Finally, the proposed enrichment links each tag with a relevant semantic entity but also with its semantic neighbourhood as demonstrated in the `canine` example in Section 1.

3 FLOR components and methodology

The goal of FLOR is to transform a flat folksonomy tagspace into a rich semantic representation by assigning relevant Semantic Web Entities (SWEs) to each tag. A SWE is an ontological entity (class, relation, instance) defined in an online available ontology. While in this paper we describe the process of enriching a set of tags with SWEs, the ultimate goal of our system is not just to connect to SWE's but also to bring in other knowledge related to these SWE's. An example of the inputs and expected outcomes to FLOR is demonstrated in Fig. 1. The input consists a set of tags and the output is a set of semantically enriched FlorTags. Note that FLOR is agnostic to the way in which this tagset was obtained. It can either be the set of all tags associated to a resource, or a cluster of related tags obtained through co-occurrence based clustering. The experiments reported in this paper used sets of tags associated with a given resource.

Intuitively, FLOR performs three basic steps (see Fig. 1). First, during the **Lexical Processing** the input tagset is cleaned and all potentially meaningless tags are excluded. We rely on a set of heuristics to decide which tags are likely to be meaningless. Second, during the **Sense Definition and Semantic Expansion** we attempt to assign a WordNet sense to each tag based on its context (i.e., the other tags in its cluster) and to extract all relevant synonyms and hypernyms so that we migrate to a richer representation of the tag. Finally, during the **Semantic Enrichment** step each tag is associated to the appropriate SWE.

Note that there is a strong correlation between the steps of FLOR and the components of the final FlorTag structure. The first step results in the **Lexical**

Representations which is a list of lexical forms for the tag, such as plural and singular forms for nouns, or various delimited types of compound tags (sanFrancisco, san.Francisco, e.t.c). The second step identifies **Synonyms** and **Hypernyms** for each tag. The last step generates the list of **Entities** containing the associated SWE's. Note that a tag can be associated to several relevant SWE's.

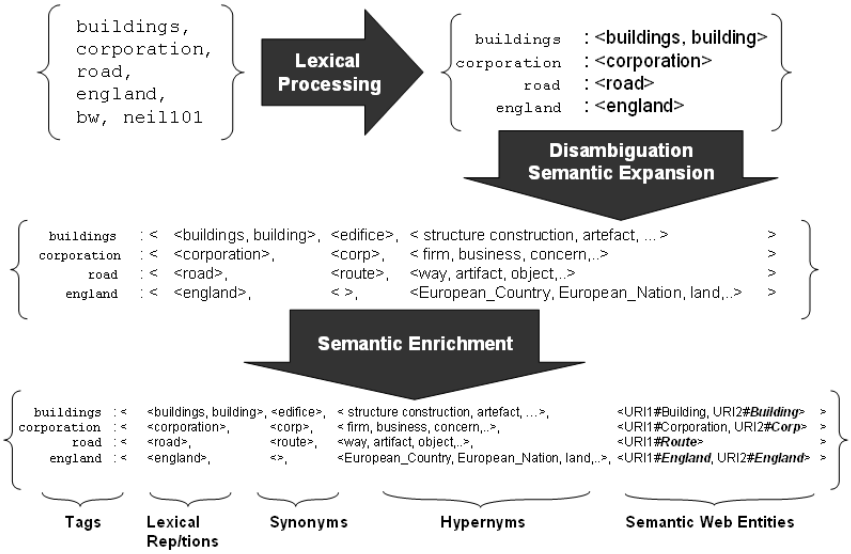

Fig. 1. FLOR Methodology

3.1 PHASE1: Lexical Processing

Due to the freedom of tagging as a basic rule of folksonomies, a wide variety of different tag types are in use. Understanding the types of tags used is the first step in deciding which of them are meaningful and should be taken into account as a basis of a semantic enrichment process. Previous work ([2, 7, 13]) has identified different conceptual categories of tags (event, location, person), as well as tag categories that can be described by syntactic characteristics. For example, there are many tags containing special characters (e.g., `:P`), numbers (e.g., `aug07`), plurals as well as singular forms of the same word (e.g., `building`, `buildings`), concatenated tags (e.g., `littlegirl`) or tags with spaces (e.g., `little girl`) and a big number of non-English tags (e.g., `sillon`). The role of the lexical processing step is to identify these different categories of tags and exclude those that are meaningless and should not be further included in the semantic enrichment process. This is done in two steps.

The Lexical Isolation phase idenfies sets of tags that should be excluded as well as those that can be further processed. Currently we isolate and exclude all tags with numbers, special characters and non English tags. The reason for excluding non-English tags is that our method explores various external knowledge sources (WordNet, Semantic Web ontologies) that are primarily in English. As future work, we will extend FLOR to isolate additional types of tags as well and deal with non-English tags.

The Lexical Normalisation phase aims to solve the incompatibility between different naming conventions used in folksonomies, ontologies and thesauri such as WordNet. This phase produces a list of possible **Lexical Representations** for each tag aiming to maximise the coverage of this tag by different resources. For example, the compound tag `santabarbara` in folksonomies appears as *Santa-Barbara* or *Santa+Barbara* in various ontologies and as ***Santa Barbara*** in WordNet. However, as the lexical anchoring to these resources is a quite complex problem, we try to address it by producing all the possible lexical representations for each tag such as: {santaBarbara, santa.barbara, santa_barbara, santa barbara, santa-barbara, santa+barbara, ...}.

3.2 PHASE2: Sense Definition and Semantic Expansion

Due to polysemy, the same tag can have different meanings in different contexts. For example, the tag `jaguar` can describe either a car or an animal depending on the context in which it appears. Before connecting a tag with a relevant SWE, it is important to determine its intended sense in the given context. This task is performed in the first step of this phase.

Another issue to take into account is that, despite its significant growth, the Semantic Web is still sparse. A direct implication is that while online ontologies might not contain concepts that are syntactically equivalent to a given tag, they might contain concepts that are labeled with one of its synonyms. To overcome this limitation, we perform a semantic expansion for each tag, based on its previously identified sense, in the final step of this phase.

The Sense Definition and Disambiguation phase deals with discovering the intended sense of a tag in the context it appears. As context we consider the set of tags with which the given tag co-occurs when describing a resource. For example, in the tagset: {`panther`, `jaguar`, `jungle`, `wild`} the context of `jaguar` is {`panther`, `jungle`, `wild`}. We use WordNet as a sense repository and rely on its hierarchy of senses to compute the similarities between the senses of all tags in the tagset and thus achieve their disambiguation. WordNet also provides rich sense definitions which facilitate the semantic expansion in the next step.

To define the senses of the tags in a tagset, we identify all the lexical representations for each tag in WordNet. In the cases that a tag has more than one senses in WordNet (synsets) we exploit the contextual information of the tagset to identify the most relevant sense. For this, we calculate the similarity between

all the combinations of tags in the tagset using the Wu and Palmer similarity formula ([21]) on the WordNet graph. The similarity degree between two senses is calculated based on the number of common ancestors between them in the WordNet hierarchy and the length of their connecting path. The result for each calculation is a couple of senses and a similarity degree for these senses. We select the two senses of the tags that return the highest similarity degree provided that this is higher than a specified threshold. If a tag has low similarities when compared to all the other tags in its cluster, then it is assigned to the most popular WordNet sense.

We currently use a threshold value of 0.8 which we observed to correctly indicate relatedness in most of the cases. Indeed, as high values as 0.7 are often assigned to unrelated tags. For example, in the tagset: {`girl`, `eating`, `red`, `apple`} the similarity between `red` and `girl` is 0.7 for the senses:

Bolshevik, Marxist, Pinko, Red, Bolshie (emotionally charged terms used to refer to extreme radicals or revolutionaries)

Girlfriend, Girl, Lady_friend (a girl or young woman with whom a man is romantically involved)

These two senses are connected through the concept ***Person*** in the WordNet hierarchy, however the two tags are unrelated in the context of this tag cluster. While this empirically established 0.8 value lead to reasonable results and was sufficient for this proof of concept prototype, we plan to establish an optimal value through systematic experiments.

Thanks to the modular architecture of FLOR, the disambiguation and sense selection method can be replaced by other methods (e.g., such as those used in [19] and [22]). Or our current method could be modified to exploit a different similarity measure between two concepts such as the Google Similarity Distance [5]. Another possible improvement could be achieved by further expanding the resource tagset with more related tags. These can be discovered with statistical measures based on tag co-occurrence as described in [18]. For example, the expanded tagset of {`apple`, `mac`} could be {`apple`, `mac`, `computer`, `macOs`}. So instead of trying to disambiguate with two tags we increase the possibilities of finding the correct sense by disambiguating with a more specific context.

The Semantic Expansion includes the synonyms and hypernyms of a tag in the FlorTag (see Fig. 1). For the purpose of this work we used WordNet to extract the synonyms of the correct sense and the synonyms of this sense's hypernym in WordNet. For example, if in the specific context the tag `jaguar` refers to an animal then the semantic expansion would include a list of synonyms: {***Panther***, ***Panthera onca***, ***Felis onca***} and a list of hypernyms: {***Big cat***, ***Feline***, ***Carnivore***}.

3.3 PHASE3: Semantic Enrichment

This phase of FLOR identifies the SWEs that are relevant for each tag by leveraging the results of lexical cleaning and semantic expansion performed in the

previous two phases. The final output of FLOR is produced by this phase (see Fig. 1) and it is a set of FlorTags enriched with relevant SWEs and their semantic neighbourhood (e.g., parents, children, relations).

The relevant SWEs are selected by querying the WATSON semantic web gateway[6], which gives access to all online ontologies. We search for all ontological entities (Classes, Properties, Individuals) that contain in their local name or in their label(s) one of the lexical representations or the synonyms of a tag.

Such queries often result in several SWEs some of which are very similar (or the same when they appear in ontologies that are versions of each other). To reduce the number of SWEs, we perform an entity integration process similar to the one described in [19]. The goal of this process is to "collapse" entities that have a high similarity into a single semantic object, thus reducing redundancy. To compute similarity between two entities we compare their semantic neighbourhoods (superclasses, subclasses, disjoint classes for classes; domain, range, superproperties, subproperties for properties) and their localnames and labels. The similarity $simDgr$ for two SWEs e_1 and e_2 is calculated as:

$$simDgr = W_l * simLexical(e_1, e_2) + W_g * simGraph(e_1, e_2)$$

$simLexical(e_1, e_2)$ is the similarity between the lexical information of two entities, i.e., their labels and localnames, computed with the Levenshtein distance metric. $simGraph(e_1, e_2)$ is the similarity of the entities' neighbourhoods, where the similarity of each neighbourhood element is computed based on string similarity. Because we consider the similarity of the semantic neighbourhoods more important than the similarity of the labels, we set the weights as $W_l = 0.3$ and $W_g = 0.7$. Note that these weights will be fine-tuned through systematic experiments. If the similarity between two entities is higher than a threshold we merge them in one entity by integrating their neighbourhoods into one. Then we repeat the process until all entities are sufficiently different from each other, i.e., their similarity falls under a chosen threshold.

Consider for example Fig. 2 where five SWEs $e_{1,5}$ are compared against a threshold value of 0.5. We start by performing their pair-wise comparison and observe that the pairs (e_1, e_4), (e_1, e_5), (e_2, e_3) and (e_2, e_5) have a similarity equal or above the set threshold. We proceed by merging the first two entities with the highest similarity, e_1 and e_5, to one entity e_1+e_5 and compute the similarities between the new entity and the remaining ones. This process continues until all similarities are lower than the set threshold, which implies that the obtained entities are sufficiently different.

Once the merged entities are created we enrich the tag with the relevant entities. This is done by comparing the ontological parents of the merged entity with the hypernyms retrieved from WordNet. The ontological parents are the superclasses of classes, the superproperties of properties and the classes of individuals. For example, as shown in Fig. 3, the tag `moon` is enriched with two entities. The superclasses of both the entities have as localname one of the hypernyms extracted from the WordNet sense of `moon`. Also, apart from the semantic definition of the tag with the respective entity, we further enrich the tag with the information carried by the entity, *EarthsMoon* `TypeOf` *Moon*.

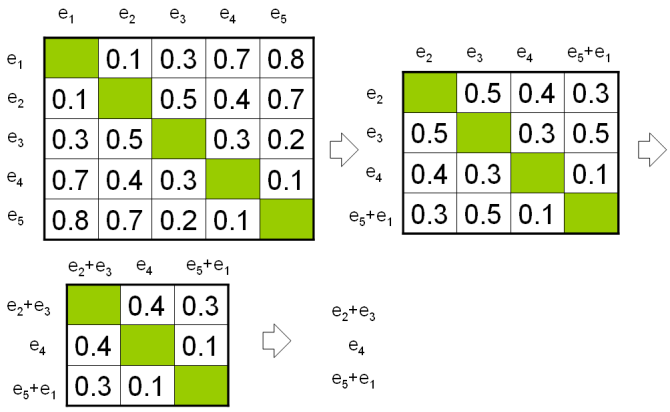

Fig. 2. Merging Strategy with threshold 0.5

3.4 An Enrichment Example

In this section we present a full cycle of the FLOR semantic enrichment method for the tag `lake`, which was found in the following five tagsets: {`rush`, `lake`, `pakistan`, `rakaposhi`, `mountain`, `asia`, `kashmir`, `snow`, `glacier`, `green`, `white`, `sky`, `blue`, `clouds`, `water`}, {`moraine`, `alberta`, `banff`, `canada`, `lake`, `lac`, `rockies`, `scan`}, {`rising`, `sunlight`, `lake`, `quality`, `bravo`}, {`lake`, `nature`, `landscape`, `sunset`, `water`, `organisms`} and {`lake`, `finland`, `suomi`, `beach`, `bubbles`, `blue`, `sunlight`, `kids`, `natural`}. Note that these tagsets contain the tags that remain after the lexical processing performed in the first phase. Fig. 4 shows the information contained in the automatically obtained FlorTag.

moon

Lexical Representations	Synonyms	Hypernyms	Entities
moon		satellite celestial_body heavenly_body natural_object object physical_object entity	http://www.ida.liu.se/~adrpo/modelica/rdf/inheritance.owl#moon type (of) http://www.ida.liu.se/~adrpo/modelica/rdf/inheritance.owl#*CelestialBody* http://www.cyc.com/2003/04/01/cyc#moon subClassOf http://www.cyc.com/2003/04/01/cyc#*NaturalStaellite* type http://www.cyc.com/2003/04/01/cyc#*EarthsMoon*

Fig. 3. Enriched FlorTag `moon`

For the second phase of FLOR, Sense Definition and Semantic Expansion using WordNet, the available WordNet senses for ***Lake*** are considered. These are the following:

WordNet 1: ***Lake→Body of water***, ***Water→Thing→Entity***
(a body of (usually fresh) water surrounded by land)
WordNet 2: ***Lake→Pigment→Coloring material→Material → Substance→Entity***
(a purplish red pigment prepared from lac or cochineal)
WordNet 3: ***Lake→Pigment→Coloring material→Material →Substance→Entity***
(any of numerous bright translucent organic pigments)

lake

Lexical Representations	Synonyms	Hypernyms	Entities
lake		lake body_of_water water thing entity	**http://lonely.org/russia#lake** subClassOf http://lonely.org/russia#***waterway*** http://lonely.org/russia#***Lake_Baikal*** – type **http://lsdis.cs.uga.edu/proj/semdis/testbed/#lake** subClassOf http://lsdis.cs.uga..../testbed/#***Water_Feature*** subClassOf http://lsdis.cs.uga.edu/proj/semdis/testbed/#***Thing***

Fig. 4. Enriched FlorTag `lake`

Applying the Wu and Palmer formula for the senses of `lake` and the senses of the rest of the tags in these tagsets we obtained variable similarities from 0 to 0.86. The zero similarities were obtained for location names such as `banf`, `pakistan`, `suomi` and for generally unrelated tags such as `quality`, `scan`, `sunlight`, `sunset`. Interestingly, `lake` returned zero similarity for the tags `glacier` and `mountain` while they should be related. This is due to the fact that, in WordNet, ***Glacier*** and ***Mountain*** are hyponyms of ***Geological formation*** which is a hyponym of ***Natural object*** while ***Lake*** is a hyponym of ***Body of water*** which is a direct hyponym of ***Thing***. Furthermore ***Glacier*** is a hyponym of ***Ice mass*** but there is no subsumption relation between ***Ice mass*** and ***Ice*** or ***Water*** that would allow for a connecting path between ***Lake*** and ***Glacier***. This fact motivates further research on how to identify similarities between tags of a tagset beyond the subsumption relations provided by WordNet.

The highest similarity, 0.86, for `lake` was obtained with the tag `water`, because Sense 1 of ***Lake*** is related to ***Body of water*** (Sense 2 of ***Water***) with a

direct hyponymy relation. Note that, in most of tagsets the first sense of ***Water***, ***Liquid***, is selected as this is the most common sense in which the tag is used. Therefore, this is a nice example of phase 2 identifying a non-trivial correlation.

Sense 1. ***Water***, ***H2O***: (binary compound that occurs at room temperature as a clear colorless odorless tasteless liquid) → ***Binary Compound*** AND → ***Liquid***

Sense 2. ***Body of water***, ***Water***: (the part of the earth's surface covered with water) → ***Thing***

Once the correct sense is selected and the tag is semantically expanded with hypernyms (there are no synonyms for this sense of ***Lake***) then the third phase of FLOR queries the online ontologies through WATSON and selects the SWEs that correspond to this sense. As shown in Fig. 4 both selected entities have the term *Lake* in their localname and their superclass in the ontology contains one or more of the hypernyms returned by WordNet, *Water* and *Thing*, as a whole or as a compound. This example shows that our anchoring to ontologies is strict for the tags to be defined (their lexical representations and synonyms) and the localnames and labels of the entities and flexible for the ontological parents and hypernyms. Note also that the selected SWEs carry additional information about two superclasses of *Lake* (*Waterway*, *Waterfeature*) and an instance of *Lake* (*Lake Baikal*) thus further enriching the tag.

4 Experiments and Results

To assess the correctness of FLOR enrichment (i.e., whether tags were linked to relevant SWEs) we applied FLOR on a Flickr data set comprised of 250 randomly selected photos with a total of 2819 individual tags. During the Lexical Isolation we removed 59% of the initial tags resulting to 1146 tags in total. We isolated 45 tags with two characters (e.g., `pb`, `ak`), 333 tags with numbers (e.g., `356days`, `tag1`), 86 tags with special characters (e.g., `:P`, `(raw → jpg)`), and 818 non English tags (e.g., `turdus`, `arbol`). Then we filtered out the photos that exclusively contained the isolated tags (24 photos) and obtained a dataset of 226 photos with a total of 1146 tags. After running the FLOR enrichment algorithm for these 226 photos, one of the authors manually checked all the assignments between tags and SWE's.

The assignment of a SWE to a tag is considered correct if the concept described by the SWE is the same as the concept of the tag in the context of its tagset. To decide that the evaluator was given a tagset and the SWEs linked to its tags. She evaluated each tag enrichment as CORRECT if the tag was linked to the appropriate SWE and INCORRECT otherwise. In cases when she was not sure about the intended meaning of the tag, she rated the enrichment as UNDETERMINED. Finally, a NON ENRICHED value was assigned to tags that were not associated to any SWE. The results are displayed in in Table 1.

Out of the individual 1146 lexically processed tags, FLOR correctly enriched 281 tags and incorrectly enriched 20 tags thus leading to precision results of 93%.

Enrichment Result	# of Tags	Percentage
CORRECT	281	24.5%
INCORRECT	20	1.7%
UNDETERMINED	4	0.3%
NON ENRICHED	841	73.4%
Total	1146	100%

Table 1. Evaluation of semantic enrichment for individual tags.

An example of incorrect enrichment is that of `square` in the context {`street`, `square`, `film`, `color`, `documentary`}. While its intended meaning is ***Geographical area***, because during the disambiguation phase `square` did not return high similarity with any of the rest of the tags, the WordNet sense assigned to it was the most popular one, ***Geometrical shape***. This lead to the assignment of non-relevant SWE's namely, *Square* `SubClassOf` *Rectangle* and *Square* `SubClassOf` *RegularPolygonShaped*. Despite this error, the rest of the tags in this tagset were correctly enriched.

FLOR failed to enrich 841 tags, i.e., 73.4% of the tags (see Table 1). Because this is a significant amount of tags, we wished to understand whether the enrichment failed because of FLOR's recall or because most of the tags have no equivalent coverage in online ontologies. To that end we selected a random 10% of the 841 tags (85 tags) and manually identified appropriate SWE(s) using WATSON and taking into account the context(s) of the tags in the tagset(s) they appear. Out of the 85 tags we manually enriched 29. We therefore estimate that the number of tags that could have been enriched by FLOR (i.e., those for which an appropriate SWE exists) is approximately 287. Thus, taking into account that the overall number of tags that should be correctly enriched was 568 (281+287) but only 281 were enriched by FLOR this leads to an approximate recall rate of 49%. While this is quite a low recall, these results are highly superior to the ones we have obtained in previous experiments where phase 2 was not part of FLOR, i.e., we directly searched for SWEs for the tags without relying on WordNet as an intermediary step. Indeed, the WordNet sense definition and expansion of the tags with synonyms and hypernyms (FLOR phase 2) increased the tag discovery in the Semantic Web thus having a positive effect on recall.

FLOR failed to enrich the above 29 tags due to the following reasons. The majority of the failures (55%) was due to **different definition** in terms of superclasses in WordNet and in online ontologies For example, the definition of `love` in WordNet and the relevant entity found in the Semantic Web are:

WordNet: ***Love→Emotion→Feeling→Psychological feature***
(a strong positive emotion of regard and affection)
Semantic Web: *Love* `SubClassOf` *Affection*

Although both these definitions refer to the same sense, and additionally the superclass *Affection* belongs to the gloss of ***Love*** in WordNet, they were not

matched because *Affection* does not appear as a hypernym of *Love*. Current work investigates alternative ways of Semantic Expansion.

A further 24% of the tags not connected to any SWE were assigned to the **wrong sense** during phase 2. For example, `bulb` referring to `light bulb` in its tagset is assigned the incorrect sense ***Bulb*** → ***Stalk*** → ***Stem*** → ***Plant organ***. The rest of the unenriched tags are due to failures in anchoring them into appropriate SWE's. For example, the sense of `butterfly` was correctly identified, but non of its lexical forms matched the label of the appropriate SWE (*Butterfly_Insect*):

WordNet: ***Butterfly***→***Lepidopterous insect*** → ***Lepidopteron*** → ***Lepidopteran*** → ***Insect***
Semantic Web: Identified entity with localname *Butterfly_Insect*

In the case of 4 tags the evaluator could not determine whether the enrichment was correct or incorrect (Table 1). This is because the meaning of the tag was unclear even when considering its context and the actual photo. For example, in the photo of Fig. 5 the meaning of the tag `volume` is unclear. In the second phase of FLOR the tag was expanded with the hypernyms ***Measure*** and ***Abstraction***. Then, it was related to the SWE *Volume* `SubClassOf` *Measure*. As the meaning of the tag was not clear for the evaluator, she evaluated it as {UNDETERMINED}. More generally, there are several cases when tags only make sense to their author (and maybe to his social group) and thus will be difficult to enrich by FLOR.

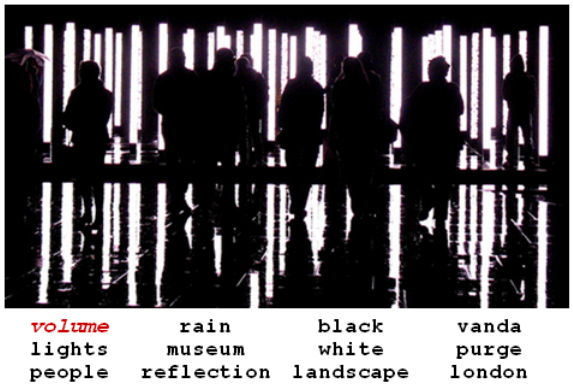

Fig. 5. UNDETERMINED Enrichment

After evaluating the individual tag enrichments the evaluator was able to draw conclusions on the overall enrichment of the tagset i.e., by photo. The evaluation output is displayed in Table 2. This would result to {CORRECT, INCORRECT, MIXED, UNDETERMINED, NON ENRICHED}. According to

this table, 179 enrichments (about 80%) were {CORRECT}, i.e., all the enriched tags of the photo are enriched correctly. Note that the {CORRECT} enrichment results are much higher from a photo-centric perspective as many tags may appear in many photos. For the total of 20 {INCORRECT} and {MIXED} enrichments, 3 of the photos had all enriched tags incorrect and 17 had at least one tag incorrectly enriched. Finally the above 4 {UNDETERMINED} tags resulted to 4 {UNDETERMINED} enrichments one of which is displayed in Fig. 5. Finally if no enriched tag appears in the photo then the result for the photo is {NON ENRICHED}.

Enrichment Result	**# of Photos**	**Percentage**
CORRECT	179	79.2%
INCORRECT	3	1.3%
MIXED	17	7.5%
UNDETERMINED	4	1.8%
NON ENRICHED	23	10.2%
Total	226	100%

Table 2. Evaluation of SWE assignment to photos.

5 Conclusions and Future Work

We presented the methodology and the experiments we performed to test the hypothesis that **enrichment of folksonomy tagsets with ontological entities can be performed automatically**. We selected a subset of Flickr photos and after performing lexical processing and semantic expansion we correctly enriched the 72% (179 of 250) of them with at least one Semantic Web Entity. We enriched approximately the 49% of the tags with a precision of 93%. Compared to our previous efforts to define the tags with Semantic Web Entities without previously expanding them with synonyms and hypernyms, this is a significant improvement. Analysing the results we identified a number of issues to be resolved to enhance the performance of FLOR.

The **Lexical Processing** phase requires supplementary methods to identify and isolate additional special cases of tags (e.g., photography jargon, dates). Furthermore, the understanding of the impact of excluding these tags from the overall process, the implementation of strategies to deal with them and their integration in FLOR will be addressed by our future work.

As indicated by the results in Section 4, the cases of incorrect enrichment and lack of enrichment were mainly caused due to the failure of the **Sense Definition and Semantic Expansion** phase. The following issues are currently investigated in order to correct the errors and enhance the performance of this phase. First, it is essential to extend the tag similarity measure to also identify

generic relations rather than only subsumption relations. This flaw was exemplified in the case of `lake` and `glacier` which were considered unrelated based the hierarchical structure of WordNet (Section 3.4). Also, in the example of `square` co-occurring with `street`, the incorrect sense definition for `square` caused further incorrect enrichment (Section 4) . One of the possible solutions to this is the context expansion based on tag co-occurrence. For example, expanding the {`square`, `street`} tagset with their frequently co-occurring tags e.g., {`building`, `park`} can increase the semantic relatedness between the tags and potentially lead to mapping the tags to the correct sense. Finally, to solve cases where the WordNet sense and the SWE are the same but with different hypernyms (see the example of `love`) the goal is to identify more relevant words as hypernyms or synonyms in order to achieve higher coverage in the Semantic Web.

The quality of the results returned from the **Semantic Enrichment** phase depends on (1) the input provided to this phase by the Semantic Expansion step and (2) on the anchoring of the tags' lexical representations and synonyms into online ontologies (see the case of `butterfly`). Alternative strategies for flexible anchoring to increase the number of successful enrichments and the same time keep the number of irrelevant matches low, are investigated by our current work. Also, we aim to experimentally identify optimal values for the thresholds and weight used in the second and third phases.

Finally, we aim to evaluate FLOR in large scale experiments and to assess the usefulness of the semantic enrichment in a real content retrieval application. This is to identify the possible implications of the overall process that are not apparent in a small scale study like the current one.

To conclude, we demonstrated that the **automatic enrichment of folksonomy tagsets using a combination of WordNet and online ontologies is possible** without user intervention in any step of the methodology and by using straightforward methods for lexical isolation, disambiguation, semantic expansion and semantic enrichment. The goal is to create a semantic layer on top of the flat folksonomy tagspaces, that allows intelligent annotation, search and navigation as well as the integration of resources from distinct, heterogeneous systems.

Acknowledgements

This work was funded by the IST-FF6-027595 NeOn project.

References

1. R. Abbasi, S. Staab, and P. Cimiano. Organizing resources on tagging systems using t-org. In *4th European Semantic Web Conference*, pages 97–110, Innsbruck, Austria, 2007.
2. S. Angeletou, M. Sabou, L. Specia, and E. Motta. Bridging the gap between folksonomies and the semantic web: An experience report. In *4th European Semantic Web Conference*, pages 30–43, Innsbruck, Austria, 2007.

3. G. Begelman, P. Keller, and F. Smadja. Automated tag clustering: Improving search and exploration in the tag space. In *15th International World Wide Web Conference*, Edinburgh, Scotland, 2006.
4. R. Benjamins, J. Davies, R. Baeza-Yates, P. Mika, H. Zaragoza, M. Greaves, J. Gomez-Perez, J. Contreras, J. Domingue, and D. Fensel. Near-term prospects for semantic technologies. *Intelligent Systems, IEEE*, 23:76–88, 2008.
5. R. Cilibrasi and P. Vitanyi. The google similarity distance. *Transactions on Knowledge and Data Engineering, IEEE*, 19(3):370–383, 2007.
6. M. dAquin, M. Sabou, M. Dzbor, C. Baldassarre, L. Gridinoc, S. Angeletou, and E. Motta. Watson: A gateway for the semantic web. In *4th European Semantic Web Conference*, Innsbruck, Austria, 2007.
7. S. Golder and B. Huberman. Usage patterns of collaborative tagging systems. *Journal of Information Science*, 32(2):198–208, 2006.
8. M. Greaves. Semantic web 2.0. *Intelligent Systems, IEEE*, 22(2):94–96, 2007.
9. J. Hendler. The dark side of the semantic web. *Intelligent Systems, IEEE*, 22(1):2–4, 2007.
10. D. Laniado, D. Eynard, and M. Colombetti. Using wordnet to turn a folksonomy into a hierarchy of concepts. In *Semantic Web Application and Perspectives - Fourth Italian Semantic Web Workshop*, pages 192–201, Bari, Italy, Dec 2007.
11. O. Lassila and J. Hendler. Embracing "Web 3.0". *Internet Computing, IEEE*, 11(3):90–93, 2007.
12. S. Lee and H. Yong. Tagplus: A retrieval system using synonym tag in folksonomy. In *International Conference on Multimedia and Ubiquitous Engineering*, pages 294–298, Seoul, Korea, 2007.
13. M. Zied Maala, A. Delteil, and A. Azough. A conversion process from flickr tags to rdf descriptions. In *10th International Conference on Business Information Systems*, Poznan, Poland, 2007.
14. C. Marlow, M. Naaman, D. Boyd, and M. Davis. Position paper, tagging, taxonomy, flickr, article, toread. In *15th International World Wide Web Conference*, Edinburgh, Scotland, 2006.
15. P. Mika. Ontologies are us: A unified model of social networks and semantics. In *4th International Semantic Web Conference*, pages 522–536, Galway, Ireland, 2005.
16. P. Schmitz. Inducing ontology from flickr tags. In *15th International World Wide Web Conference*, Edinburgh, Scotland, 2006.
17. N. Shadbolt, T. Berners-Lee, and W. Hall. The semantic web revisited. *Intelligent Systems, IEEE*, 21(3):96–101, 2006.
18. L. Specia and E. Motta. Integrating folksonomies with the semantic web. In *4th European Semantic Web Conference*, pages 624–639, Innsbruck, Austria, 2007.
19. R. Trillo, J.Gracia, M. Espinoza, and E. Mena. Discovering the semantics of user keywords. *Journal of Universal Computer Science*, 13(12):1908–1935, 2007.
20. X. Wu, L. Zhang, and Y. Yu. Exploring social annotations for the semantic web. In *15th International World Wide Web Conference*, pages 417–426, Edinburgh, Scotland, 2006. ACM.
21. Z. Wu and M. Palmer. Verb semantics and lexical selection. In *32nd Annual Meeting of the Association for Computational Linguistics*, pages 133 –138, New Mexico, USA, 1994.
22. C. Yeung, N. Gibbins, and N. Shadbolt. Understanding the semantics of ambiguous tags in folksonomies. In *International Semantic Web Conference*, Busan, South Korea, 2007.

Disambiguating Identity through Social Circles and Social Data

Matthew Rowe, Fabio Ciravegna

Web Intelligence Technologies Lab
Department of Computer Science
University of Sheffield, UK
{m.rowe, f.ciravegna}@dcs.shef.ac.uk

Abstract: This paper presents an approach to disambiguate extracted identity information relating to different individuals through the use of social circles. Social circles are generated through the extraction and pruning of social networks using the analysis of existing social data. Social data encompasses information such as images, videos and blogs shared within a social network. Identity information is extracted by involving the user in both selecting their key identity features for disambiguation, and validating the retrieved information. Our approach provides a methodology to monitor existing identity information, applicable to addressing such issues as identity theft, online fraud and lateral surveillance.

Keywords: communities, disambiguation, identity, semantic web, social networks, social web

1 Introduction

The social web has seen enormous growth over the past 2 years. For example, in the UK alone, there are now more than 9 million unique Facebook users, 5 million unique MySpace users and 4.1 million unique Bebo users [16]. Commonly, users of such services use these sites to create an on-line social environment very similar to the one they experience in their everyday life, but extended and complemented by the on-line features of these services. Common tasks include organising events, and interacting with friends through messaging, blogging and sharing photos.

In parallel with the growth of the Social Web, the number of cases of malicious use of personal information has also grown [18]. The problems of identity theft and online fraud are of great significance in many countries, where personal details of individuals are stolen daily and used for malicious purposes. Users of online social networking sites commonly share information intended for social interaction. However, such personal information (e.g. the date of birth provided to help people remind someone's birthday) can be misused with malicious intention (e.g. the date of birth is partly used to check a person's identity when accessing phone services in the UK). The number of reported cases of cyber stalking and online harassment has increased, and the practice of lateral surveillance has also risen, including reports of cases where potential employees are vetted based on their online presence.

Such issues present several challenges and opportunities for research. The development of technologies able to monitor personal information of a given individual provide a stepping stone to assessing the risk of an individual becoming a victim of identity fraud. This requires first and foremost the ability to identify and integrate personal identity information from heterogeneous web resources. This paper focuses on the first part of this task, i.e. the challenge of discovering personal identity information from semi-structured Web resources, and the required disambiguation of individuals contained within this information. Semantic technologies provide a useful means for carrying out disambiguation, by formalising identity and using semantic information to assert facts about an individual.

The approach presented in this paper utilises social circles derived from social data to disambiguate individuals. This is split into several stages:

Firstly a user's social network is extracted from social networking sites and integrated. The resulting network is then pruned into a social circle containing identifiable relationships with other individuals by analysing existing social information. A social circle is denoted as a group of people linked to a central individual by some identifiable common relation. Social information describes content related to a given person that contains social characteristics, and provides a useful source of socially annotated data due to the rise in sharing facilities in social Web sites. This can be of any type including blogs, images tagged with person names, and instant messaging conversations.

The user decides which identity features are best suited to minimally distinguish their identity from others. The process of discovering identity data begins by extracting information from the set of resources using these identity features: We use a social approach to allow users to rate and critique resources based on the accuracy and volume of information available. Each user has the ability to contribute their opinion about an identity resource based on the information present relating to them. Other identity features are used that have been repeatedly chosen by other users to extract data. Other resources are discovered and analysed automatically via the Web, e.g. FOAF-web.

Finally all resources found to contain the user's details are analysed to disambiguate between potential individuals sharing similar identity properties . The disambiguation procedure uses the user's generated social circle by analysing resources found to contain user details. Comparisons are made between the social circle and the resource to derive the probability of the information present belonging to the user. Upon completion of the disambiguation process, all known identity

information attributed to a person is displayed for validation. A user in the loop can then validate it, correct it and re-start the process of extraction.

In order to implement the approach we have created a Facebook application capable of extracting a users social network from the social networking site. Our application is able to access all the images and conversation data from Facebook relating to each member of a user's social network and prune the social network to produce a circle consisting of the user's closest friends. Images are chosen due to the social tagging application commonly found in social networking sites. Our implementation then uses this social circle for the disambiguation process. The Facebook application is available for downloading and testing[1].

This paper is structured as follows: Section 2 presents related work to disambiguating identity through social circles. Section 3 outlines the semantics of personal identity, and the selection of prevalent features. Section 4 details the presented approach, explaining the methods used and technical details. Section 5 explains the proposed methodologies of evaluation and explains the reasons for this. And finally, section 6 discusses the primary conclusions from the investigation so far, and outlines the proposed future work in this area.

2 Related Work

The related work to our approach covers several fields of research: Name disambiguation literature is included describing similar approaches to disambiguating persons using related contextual information. Social network analysis literature covers formal definitions of social circles and groupings. Social network mining literature discusses differing methods for extracting social network information from various sources. Object identification literature presents approaches that could be adapted to identifying individuals, and commercial systems are included detailing work towards identity disambiguation through the provision of identity theft risk assessments.

The problem of disambiguating individuals, also known as instance unification, is addressed in [1]. Citations are used to discover additional information about a given individual author; this information is then used to mine the web. Social networks are constructed surrounding the author based on the co-authorship of papers. Similarly work by [7] investigates the challenge of identifying misspelt and abbreviated names by using clustering together with Naïve Bayes to compute the probability of a given name belonging to a name cluster. Our methodology is similar to [1] by using existing information to derive the initial social network, however we utilise further social information such as image content and conversation data to prune the network.

Work to identify social circles is presented in social network analysis literature such as [14] where cliques are initially discovered from an individual's social network, categorised as a sub-graph where a relation connects all pairs of points within the graph. A social circle is the aggregation of overlapping cliques within the

[1] http://apps.facebook.com/socialcircular

[3] http://www.garlik.com

larger social network graph, and the key group of friends related to a given central individual. Social grouping enables socially linked individuals to be clustered based on a common relation where the relation can simply be a binary classifier used to prune an individuals social network to only contain those individuals positively classified. Our approach uses this definition to generate the social circle from the initial social network, we use classifiers over image and conversation data to derive social links.

A technique for mining social networks is presented in [9] and [6] utilising a three-step approach: Firstly, mining the web for social network information identifying links between two individuals, secondly monitoring real world interactions between individuals to confirm relationships between them, and thirdly, monitoring interactions between users on the web by capturing online communications between individuals. Further work will mine social network information from the wider web. At present our approach only generates social networks from social networking sites.

Work described in [10] uses a two-part methodology to gather social network information by mining information from the web and crawling for semantic documents containing information described using the FOAF [2] ontology. Social network information is mined from the web by querying a search engine, with pairs of names of individuals considered to be friends. The number of pages returned containing both names co-occurring is the count for that pair; this gives the strength of the relationship between the two individuals. Work by [4] presents an approach to social network extraction using FOAF files by crawling FOAF-web, extracting information from each FOAF file and aggregating with information from other FOAF files. Assertions are made about discovered individuals using the supplied semantic information. Our approach also uses FOAF-web to extract information, however we are only concerned with identity information during the mining phase. The later disambiguation phase checks the FOAF content for relationships corresponding to the social circle.

Work by [8] presents a methodology to identify labels for relations between two socially linked entities; the label word along with the two entities then form the query to be entered into a search engine to retrieve additional information. This methodology allows ontologies to be generated using the entities and relations that link them together. Threshold tuning is used to refine the importance between two entities using objective and subjective criteria. An approach described in [11] identifies relations between two socially linked entities and uses labels derived from the collective context of the entities to define the relation in a similar manner to work in [8].

Literature relating to object identification presents a similar approach to identity extraction by using features to recognise objects. An interesting methodology presented in [17] details a general method for learning rules to map data for object identification using domain independent attributes for identification. Work by [12] describes a heavily cited framework for object identification presenting a straightforward modulated approach using clustering for the pre-selection of similar object pairs. Literature such as [15] utilises the features of objects to predict the likelihood probability of a match occurring for object pairs. Work by [3] describes an automatic approach to identifying and disambiguating relevant information using a library of string metrics to deduce term record similarity. Our approach similarly uses

string metrics to compare the properties of identity at a low-level, mining the web for identity information, and disambiguating extracted identity information.

Several systems utilising social networks and social information offer users the ability to monitor their personal information. Garlik[3] offers services to enable the monitoring of personal information, but fails to correctly disambiguate between individuals in several cases. Garlik only uses the presence of social networking accounts when detecting personal information; our approach differs by using the information within the accounts to extract relationship information. Maltego[4] tracks social networks relating to a given individual through mining the web for information, and identifying real world links between people, and groups. Spock[5] is another similar application that crawls the social web and the wider web for occurrences of names that have been searched for, information is then aggregated together using the similarities in the derived content. Our work is similar to Maltego by mining the web for person names, however we place a greater emphasis on an individual's social circle rather than their wider social network. The presented approach also differs by allowing the user to select their most prevalent identity feature, and prioritising identity features with the greatest cumulative prevalence.

[4] http://www.maltego.com
[5] http://www.spock.com

3 The Semantics of Identity

The semantics of identity are extremely important when extracting identity information and disambiguating between individuals. The notion of personal identity described in [19] splits identity into 3 tiers: My identity containing persistent identity features; shared identity containing attributes assigned to an individual by others; and abstracted identity containing identity denoted by grouping. The first tier, 'My identity', contains identity features of the greatest significance when disambiguating one individual from the next; name, date of birth, etc. However, the prevalence of these identity features differs between individuals. Consider a scenario involving a man named John Smith, John must decide on what features of his identity are the most prevalent. He knows that his name is very common, but he knows that he is the only 'John Smith' on his street, so he selects his postcode. As certain identity features build a cumulative level of prevalence, these features become inherently used by the approach.

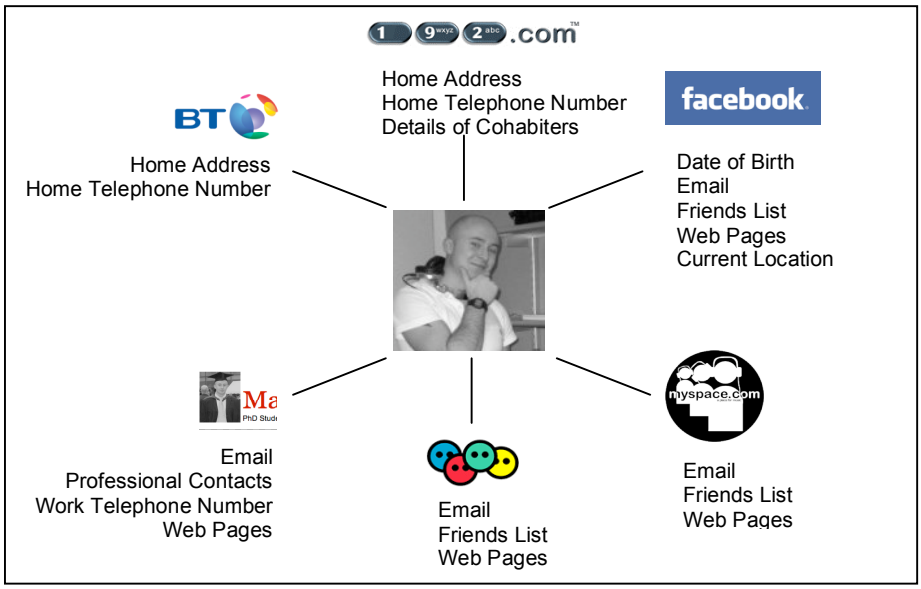

Figure 1. Distribution of features of my identity throughout the Web

As figure 1 shows the distribution of identity features contain enough information about a given individual to compile a fairly complete identity profile. The social network feature of identity is related to the 'Shared identity' tier presented in [19]; this tier contains temporary relationships prone to either becoming stronger or breaking down. The creation of an ontology to encapsulate the properties of an individuals identity and their social network based on the FOAF [2] specification is

required. FOAF provides sufficient properties to define certain identity information, however it fails to provide other features that would differ between domains e.g. Social security number, and national insurance number. Such a formalisation would aid with integration of identity information and reasoning.

4 An Approach to Disambiguating Identity using a Social Circle

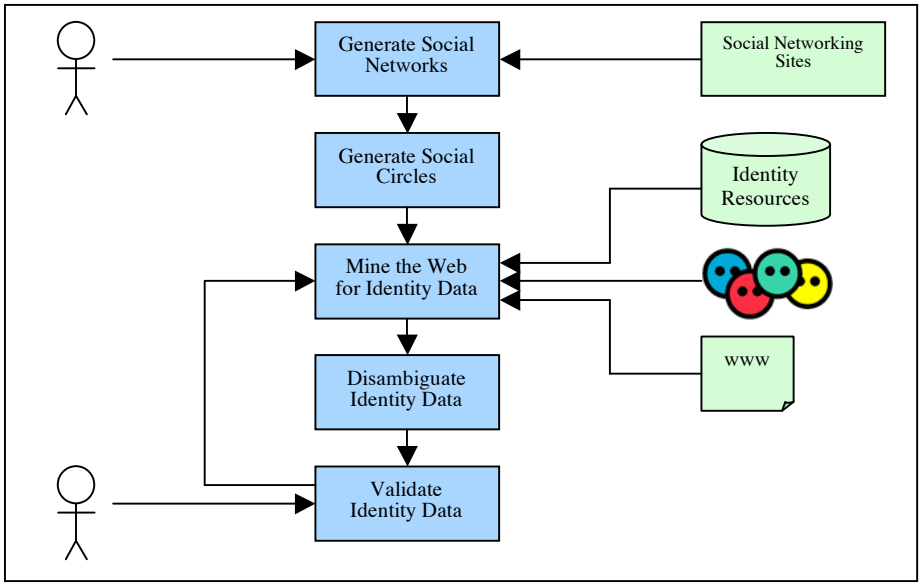

Figure 2. Approach Overview

This section explains the details of our approach to disambiguate individuals using social circles, and present disambiguated information for validation. This section is composed of sub-sections describing the processes performed in each stage of the approach as displayed in figure 2. To illustrate these processes we use the example of John Smith, a man who is concerned about his identity being stolen or misused, and wishes to know what information exists about him. We explain our approach in terms of a completed system.

4.1 Generating Social Networks

In order to generate social networks, existing accounts with social networking sites belonging to the user are needed to seed the disambiguation process. At present only Facebook data is being used to seed the approach. However, data from other accounts can be used. John logs into a specially created Facebook application. The application works by accessing Facebook's Developer API and extracting the social network

information into FOAF [2]. This compiles a comprehensive list of all of John's friends he is acquainted with. In the future, other social networking sites and alternative services can be used to seed the process providing John has an account with them. Existing approaches could also be used to generate the social network such as entity co-occurrence and mining FOAF-web for social networks [7,8].

4.2 Generating Social Circles from Multimedia Data

Using the algorithms described in figure 3 and figure 4, strengths are derived describing the relationships the user has with each member of their social circle. Several users of social networking sites are prone to adding friends who are more likely to be acquaintances and not necessarily people within their social circle or clique. Figure 3 details the derivation of relationship strengths from image data; images are extracted from a web resource annotated with unique identifiers of individuals. Should the unique identifier match that of the user, the social bond is strengthened. The same principle is used in the algorithm to derive the social bond from conversation data by analysing all received and submitted messages to identify the user.

```
-----------------------------------------------------------------------

   Extract all friends from friend list
   For each friend from friend list
       Extract all photos the friend appears in
             For each photo from photolist
               If you appear in the photo with your friend
               Increment friend strength by one
       Divide the friend strength by the total number of
   photos
        Store the social bond strength
-----------------------------------------------------------------------
```

Figure 3 – Algorithm deriving relationship strengths from images

```
-----------------------------------------------------------------------

   Extract all friends from friend list
   Extract all messages you have received
   For each friend from friend list
       Extract all the messages sent to their profile
       For each message they have received
           If you received the message
               Increment message count by 1
       Divide message count by the total number of
   messages
       For each message you have received
           If the message was sent by the friend
               Increment received message count by 1
```

```
      Divide received message count by the total number
  of received messages
      Add the message count to the received message count
      Divide combined message count by 2
      Store the combined message count
-------------------------------------------------------------------------
```

Figure 4 – Algorithm deriving relationship strengths from conversational information

Using these algorithms, John's social network is pruned to fewer components: For each of John's friends the Facebook application extracts all the photos they appear in. Each photo is then verified to see if John also appears in the photo. A score is kept to count how many times John appears in his friend's photos, this score is then used to derive the weighting of their relationship. The same process is carried out using conversational information. Both of the derived weightings are used to compute an average social bond, the <*social-bond*> property is added to the existing FOAF specification to contain the strength of the social bond for each friend.

4.3 Mining the Web for Identity Data

As discussed in section 3 the prevalence of identity features are purely based on the user. This approach incorporates both identity features specifically selected by the current user, and the social dynamic of cumulatively prevalent identity features following their frequent selection. The mining for identity data begins by using the features of identity that have been chosen by the user. John Smith knows that his name is very common, he must therefore select accompanying features of his identity. He knows that he is the only John Smith on his street so he chooses his postcode, and he also knows that his email address is unique so he selects that.

Matching identity information compares instances of identity to derive a similarity measure corresponding to the properties of each instance. At a meta-level this allows the comparison of objects regardless of the differences in the format of their properties. In our approach we only compare textual components, so one application of instance matching would use matching names and any other identity features through simple string matching using the SimMetrics package [3]. As resources are parsed, the information is analysed for any possible occurrences of the identity features being searched for. If any matches take place, then the URI of the resource is stored.

The mining process begins by firstly mining FOAF-web to extract semantic information about the user using the prevalent identity features selected by the user. Due to their semantically rich format, parsing FOAF files is a simple process allowing basic comparison of the identity features specified by the user and the found information. Using the <foaf:seeAlso> property, linked FOAF files are also mined to gather more Semantic information.

Secondly existing identity resources are accessed to extract identity information using the prevalent identity features selected by the user. Identity resources can be

shard by users of the implementation that contain identity information. The community of users are able to provide feedback regarding the accuracy and volume of the information present in each resource. Resources are marked by the community as useful sources for identity information, and become prioritised when the process of mining identity information begins. When mining both FOAF-web and identity resources for information, any found social content is flagged for later use. This includes any names observed from the individual's social circle.

Finally the wider web is mined for identity information by submitting structured queries to a search engine. Queries are structured to detail the most important and prevalent features of the user's identity selected at the start of the process, together with the cumulatively most prevalent identity features. In John's case three queries would be used: His name and his postcode. His name and his street name (derived from the postcode). His name and his email address.

4.4 Disambiguating Identity Data

Following the collection of all possible identity instances attributed to the user it is important to disambiguate between information relating to different individuals. Social circles are used to perform the disambiguation process. Each resource is parsed to derive information about any of the user's friends from their social circle. In our approach entity extraction is used to find any names within the identity resources and compile the names into a list using the rule based document annotator; Saxon [5]. Each name in the list is then compared with the names from the user's social circle by computing the Smith-Waterman-Gotoh distance using the SimMetrics [3] package, and comparing this distance to a predefined threshold. If a match occurs then the resource is marked for re-extraction, and the social bond between the matched friend name and the user is strengthened. The members of the user's social circle with the strongest bond are prioritised to force them to be compared to the name list first. Should no match be found, then the confidence level that the resource contains information relating to the individual is minimal, and as a consequence no further extraction is performed.

Identity information that was found relating to John Smith produces two pieces of information from different resources. After parsing the first resource three names were found and compiled into a list: John Smith, Bobby Moore, and Geoff Hurst. Bobby Moore is one of John's friends from his social circle. Therefore the information contained within the resource is valid and belongs to John. The second resource is parsed, and the names are added to a list: John Smith, and Boris Becker. Boris Becker is not in John's social circle; therefore we cannot tell if this resource contains information belonging to John.

Upon completion of the disambiguation process, the resources found to contain information correlating with the user's social circle are passed on for validation. Resources found to contain more people from the user's social circle are given a higher confidence rating and are therefore presented as being the strongest candidates for containing information relating to the user.

4.5 Validating Identity Data

Once the information has been aggregated and linked together, it is displayed to the user. The information is presented as a mapping showing the resource where the information occurs and within what context the recognition took place, describing the friends from the social circle that were matched around the user. The user is able to validate the results by confirming or rejecting the extracted information. If the returned information does not belong to the user then the extraction process should be performed again, but with less prevalence towards the friends who were responsible for the misidentification.

John is presented with an interactive diagram containing occurrences of his identity on the web. Each occurrence is labelled with the location, and his friends and identity properties that were also found there. Upon inspection of the information, he doesn't agree that one piece of information is about him. He clicks on the resource link to the web page containing the information and realises that he was correct; the information is about another John Smith. He returns to the diagram and informs the system of an incorrect find.

5 Evaluating Identity Disambiguation

The evaluation of the described approach uses a user based study consisting of 60 users each with a variable level of presence on the web, both within the wider web including personal web sites, and the social web including online accounts with social networking sites. This many users yield enough results to perform statistical evaluation. Each user tests the approach in two stages:

The first stage evaluates the pruning of each user's social network into a more compact and relational social circle. The evaluating user is required to analyse whether the derived social circle contains links that they deem to be appropriate with their peers, and mark any errors that exist.

The second stage of the evaluation process involves the evaluation of the disambiguated identity information. The evaluating user verifies all the extracted occurrences of their identity information prior to the disambiguation process, marking whether each resource does contain information describing their identity or not. Once the disambiguation process has completed, the user is then presented with disambiguated information for validation. This second evaluation step is already included in the previously described approach to provide a feedback mechanism for the mining of extracted information. Both steps of evaluation use information metrics to derive the precision, and error rate produced by the system, evaluating the efficiency and effectiveness. User satisfaction also is also evaluated using questionnaires completed by users of the implementation.

6 Conclusions and Future Work

The presented approach is currently being implemented and will be ready for evaluation by prospective users. Through implementation several interesting issues have arisen that will be investigated further. One of the main issues concerns the construction of search queries to the wider web. At present this last stage of the mining process can yield low levels of precision even when the user has declaratively specified their most prevalent identity features. This can be related to the lack of online presence in the search engine domain should a given individual not have any personal web sites, therefore a technique to tune the approach must be considered.

We believe that the described approach presents a novel technique both to the pruning of social networks to form social circles, and also for the disambiguation of individuals using social circles. The former challenge has been addressed in an abstract manner to allow the inclusion of alternative data sources such as blogs, and the sharing of bookmarks, or emails. The algorithm for deriving the social bonds can be applied to analyse any similar social interactions. Images were chosen due to the increase in the social tagging phenomena evident in several social networking sites. Such techniques could be easily adapted to provide risk assessments for users concerned with identity theft and online fraud. By providing a model that specifies what identity features must become accessible for identity theft to be possible, a user could be informed simply by submitting the identity information provided by our approach. This model must be an adaptive model to permit transfer between domains where the criteria for assessment may alter (i.e. Different countries require different identity features).

At present our approach uses two binary classifiers to derive the existence of a social bond between two individuals. We analyse conversations and images to classify individuals as being friends. The usage of such classifiers is assumed to generate a satisfactory social circle, however a further advancement would allow the comparison of classifiers. This would allow further evaluation of the proposed methodology for social circle generation, at present there is little indication of the quality of the classifiers being used.

The methodology that we present for performing the disambiguation process is fairly limited. It is composed of a straightforward comparison technique to derive the string distance between two words. An alternative suitable approach could utilise decision models compiled from the string distances of each name extracted from a single resource, a decision is then reached for a single resource from the analysis of all decision models [13]. The current approach is limited by only requiring a single name match from a resource to denote relation to the individual in question.

Upon completion of the implementation evaluation will take place to derive the effectiveness of generating social circles, and disambiguating between items of identity information. The methodology presented in chapter 6 sufficiently covers the two main objectives of this approach, although both evaluation steps do require an exhaustive process of auditing the generated information to produce a gold standard, particularly in the second stage.

Other future work is to develop a visualisation technique for user information similar to figure 1 detailing the distribution of personal information, and allowing the individual to analyse the information.

References

1. Aswani. N., Bontcheva. K., Cunningham. H.: Mining Information for Instance Unification. In the proceedings of 5th International Semantic Web Conference, Athens, GA, USA (2006).
2. Brickley. D., Miller. L.: FOAF Vocabulary Specification. (2004).
3. Chapman. S., Norton. B., Ciravegna. F.: Armadillo: Integrating Knowledge for the Semantic Web. Proceedings of the Dagstuhl Seminar in Machine Learning for the Semantic Web , 13-18 February (2005).
4. Finin. T., Ding. L., Zhou. L., Joshi. A.: Social Networking on the Semantic Web. The Learning Organisation, vol. 1 , no. 5, pp. 418-435 (2005).
5. Greenwood. M., Iria. J.: Saxon: An Extensible Multimedia Annotator. To Appear in the Proceedings of the International Conference on Language Resources and Evaluation, Marrakech, Morocco (2008).
6. Hamasaki. M., Matsuo. Y., Ishida. K., Nakamura. Y., Nishimura. Y., Takeda. H.: Community Focused Social Network Extraction. Proceedings of 2006 Asian Semantic Web Conference (2006).
7. Han. H., Zha. H., Giles. L. C.: A Model-based K-means Algorithm for Name Disambiguation. Semantic Web Technologies for Searching and Retrieving Scientific Data Workshop. International Semantic Web Conference (2003).
8. Jin. Y., Matsuo. Y., Ishizuka. M.: Extracting Social Networks among Various Entities on Web. The Semantic Web. International Semantic Web Conference 2006. pp. 487-500 (2006).
9. Matsuo. Y., Hamasaki. M., Nakamura. Y.: Spinning Multiple Social Networks for the Semantic Web. Proceedings of the 2006 Asian Artificial Intelligence Conference (2006).
10. Mika. P.: Bootstrapping the FOAF-Web: An Experiment in Social Network Mining. 1st Workshop on Friend of a Friend, Social Networking and the Semantic Web, Galway, Ireland (2004).
11. Mori. J., Tsujishita. T., Matsuo. Y., Ishizuka. M.: Extracting Relations in Social Networks from Web using Similarity between Collective Contexts. International Semantic Web Conference (2006).
12. Neiling. M., Jurk. S.: The Object Identification Framework. In KDD03 Workshop on Data Cleaning, Record Linkage, and Object Consolidation, Washington DC (2003).
13. Rendle. S., Schmidt-Thieme. L.: Object Identification with Constraints. In Proceedings of the Sixth international Conference on Data Mining (December 18 - 22, 2006). ICDM. IEEE Computer Society, Washington, DC. pp. 1026-1031 (2006).
14. Scott. J.: Social network analysis : a handbook. London, Sage (2000).
15. Singla. P., Domingos. P.: Object identification with Attribute-Mediated Dependences. In Conference on Principals and Practice of Knowledge Discovery in Databases (PKDD), pages 297--308, Porto, Portugal (2005).
16. Sweeney. M.: Facebook sees first dip in UK users. http://www.guardian.co.uk/media/2008/feb/21/facebook.digitalmedia (2008)
17. Tejada. S., Knoblock. C. A., Minton. S.: Learning Object Identification Rules for Information Integration. Special Issue on Data Extraction, Cleaning, and Reconciliation, Information Systems Journal. vol. 26. (2001).

18. Wallop. H.: Fear over Facebook identity fraud.
http://www.telegraph.co.uk/news/main.jhtml?xml=/news/2007/07/03/nface103.xml
19. Windley. P. J.: Digital Identity. O'Reilly Media (2005).

OntoPair: Towards a Collaborative Game for Building OWL-Based Ontologies

Peyman Nasirifard, Slawomir Grzonkowski and Vassilios Peristeras

Digital Enterprise Research Institute
National University of Ireland, Galway
IDA Business Park, Lower Dangan, Galway, Ireland
firstname.lastname@deri.org

Abstract. Collective Intelligence takes advantage of collaboration, competition and integration. It often uses mixed groups of humans and computers to research in new unexplored ways. Ontologies, which are the main building block of the Semantic Web, are usually prepared by domain experts. We introduce a novel approach, which employs Collective Intelligence, towards building simple domain ontologies through a game called *OntoPair*, an entertaining web-based game that is able to build simple OWL-based ontologies based on collected information from players. The game collects properties and common-sense facts regarding an object by means of some fixed templates and translates them into OWL representation by aid of a mediator/mapper and builds simple domain ontologies after refinement in several iterations. We define the game and preform a small experiment that proves our idea.

1 Introduction

Ontologies are the main building blocks of the Semantic Web technologies. They try to define a specific domain in a systematic way. The can be expressed using different standards and languages like RDFS [3] and OWL [10]. One of the main concerns of Semantic Web researchers is building domain ontologies and collect sufficient instances for them. Because building domain ontologies is not an entertaining task, they are usually build by domain experts

In computer science, human-based computation is a technique in which a computational process performs its function via outsourcing certain steps to humans [16]. In other words, there are some tasks that most humans can do easily, but current computers can not perform them in a logical time (e.g. CAPTCHA [13]).

Collective Intelligence is a form of intelligence that emerges from the collaboration and competition of many individuals [17]. This phenomenon has been observed by many researchers for years. Among them, Pierre Levy [6] described its potential for the internet technologies. He pointed out that rapid and open data exchange would coordinate the intelligence in new unexpected ways.

In this paper we present a game called OntoPair which aims at harnessing the benefits of the Collective Intelligence phenomenon to create ontologies. We show

how to create them by a number of human-human competitions. We describe how computers should proceed and integrate the obtained results in a way that leads us to obtain well-defined ontologies.

2 OntoPair Structure

OntoPair is a two- or one-player game which provides an interactive environment between anonymous players to play and build simple ontologies. The game is based on traditional word guessing games [15] with some fixed templates which have been carefully chosen to be translated into OWL by means of a mediator/mapper. The game is composed of two different phases which are separated from each other, but the result of first phase is the input of next phase. Roughly speaking, these two main phases can be called *Collecting properties* and *Collecting common sense facts* about an object by means of some fixed templates. The game is mainly for two players, but it can be also played in single mode. The players do not know each other, they can not communicate, and they are randomly paired. The game can be played in two main modes: *graphic-based* and *text-based*. In the graphic mode, the players look at a same image and play, whereas in the text mode, the players look at a same text-based word or keyword and they play; e.g. in the graphic mode, the players may look at an image of a *car*, a *house* or a *bicycle*, but in text mode, they will see the explicit words of a *car*, a *house*, or a *bicycle*. In next sections, we describe each phase in a more detailed manner.

2.1 Collecting Properties

In first phase of *OntoPair*, collecting properties, the main goal is collecting properties, components, and characteristics of a specified object. This phase is very similar to ESP Game [7] and Google Image Labeler [5], but there exist several cruicial differences. The main difference is that in ESP game or Google Image Labeler, the players try to annotate an image and catch the objects that are located in images, whereas in this phase of OntoPair, players play to catch the properties and characteristics of a specific object in text-based or graphic-based mode. The other main difference is that ESP game and Google Image Labeler work only in graphic mode and text-based ESP game does not make sense; whereas in OntoPair, as we mentioned earlier, the game can be played in both graphic and text mode.

The graphic mode of this phase should be based on ESP game or Google Image Labeler, as we need the explicit name of objects that are located in the image. In other words, the result of ESP game or Google Image Labeler can be used in this phase of graphic mode of OntoPair. In graphic mode of OntoPair, the players will look at the same image and they have a *hint* which is actually the name of one of the objects that is located in image and the players should mention properties of that object and agree upon a property. In the text-based

mode, again both players will look at the same word which is fetched from a database of objects.

The result of this phase is actually a collection of properties of different objects. We store these properties in a data store and link them to the object. To clarify what we are looking for, we give some hints to the players. These hints are two general questions: *What does an object X contain/have?*, and *Which parts/components/characteristics does object X have?*. in these templates, *X* is replaced with the name of the object; e.g. *What does a car contain?* and *Which parts/components/characteristics does a car have?*

One of key concepts in games is *points*. Games without points do not make sense and players will lose their motivation to play after a while. In this phase of the game, we also give points to the players. After agreement of the players upon a property, both players get points and the game continues and shows another image or text, depending on game's mode. The game continues until one player quits or time is up, as the game is played in time intervals. If one player quits during the game, we try to find another player randomly. If it takes a long time, as the number of players is not always even, the game can go through single player version. The single player version is actually playing with a log file from previous games with the same object. Actually we store all properties of an object during each game. This will also help to evaluate the data source of the properties. It is obvious that at startup of the game with an empty knowledge base, the game can not be played in single player version and there should be always an even number of players. To avoid the game being boring, the players can skip current image/text and continue to see next random image/text, if the players find an object boring and they can not agree upon a property.

It is obvious that there exist some properties in an object that most players often mention, e.g. most players will say that a book has title or author, but probably a few of them will say about ISBN, price, or publishing date. To fight with these issues, we detect these often-used words and we prevent users to mention such words again and again by indicating them as *prohibited words*. The prohibited words which are assigned to the objects are calculated based on the number of each property which has been mentioned in previous plays. There is no doubt that a single object is played as long as the players are not able to detect a new property in it and they always skip the object. The prohibited words make the game more difficult, but more fun. The other advantage of the prohibited words is that these words can be used as hints for players to guess what information we are looking for as a property, part, characteristic, or component.

As a concrete example, suppose that both players are looking at an image of a *book*. They should play and mention the properties and different parts of a book, one after the other. First player says *title*, the second player says *author*; game continues: the first player says *chapter*, the second player says *ISBN*; game continues: the first player says *publisher*, the second player says *title*. At this point, both players agreed upon *title*, so both players get points and the game continues by showing another random image or word. It is obvious that

after several times of playing and putting common words to prohibited list, we catch, say, a complete collection of book's properties like author, chapter, ISBN, publisher, etc.

2.2 Collecting Triples and Common Sense Facts

The second phase of OntoPair, collecting triples and common sense facts, is also totally separated from the previous phase. Note that this phase is played by different players which are not necessarily the same as players in the first phase. The result of the first stage is used in this phase. Like the previous phase, there are also two players in this phase, who do not know each other and they can not communicate. The players are randomly paired. In this phase, we collect some pieces of information which are called common sense facts about an object by means of some fixed templates. Informally, a common-sense fact is a true statement about the world that is known to most humans [8]: "a book has one title", "a human has two legs", etc. As we mentioned earlier, collecting these common sense facts is done through fixed templates and based partially on properties that we have collected in the previous phase. This phase of OntoPair is a word guessing game that one player (narrator) should guide the other player (guesser) to guess a word which is actually the object that we are trying to find some common sense facts about it. In other words, in this stage, an image or a word is assigned to one player and he/she should complete pre-defined templates to guide the next player to guess the word. As soon as one template is completed, it will be sent to next player and as soon as the next player could come up with the right word, both players get points, the role of players will switch and game continues by showing another randomly-chosen image or word. These pre-defined templates have been chosen for a purpose: To translate them simply into OWL using a mediator/mapper. The explicit templates that we present to the players are listed below:

- It has at least ___ Y: The Y will be replaced by a list of real properties of the item that comes actually from the knowledge base of properties that we have collected in previous phase and the player can choose arbitrary property from a combo box. Here we catch the minimum cardinality of the property, if and only if it makes sense.
- It has at most ___ Y: The Y will be replaced by a list of real properties of the item that comes actually from the knowledge base of properties that we have collected in previous phase and the player can choose arbitrary property from a combo box. Here we catch the maximum cardinality of the property, if and only if it makes sense.
- It is kind of ___: With this template, we catch hierarchical information of the item.
- It could be either ___ or ___ (or more): From one perspective (see also next template), this template provides different types of the item. The player can extend this template by adding more items.

- It could be union of ___ and ___ (and more): From the other perspective (see also previous template), this template provides different types of the item. The player can extend this template by adding more items.
- It is complement of the ___: This template provides complement objects/-concepts of the item.
- It is disjoint with (opposite of) ___: This template provides the objects/concepts that are disjoint with the item.
- It is equivalent to the ___: This template provides equivalent objects/concepts to the item.

Note that in this phase we have also the notion of prohibited statements. Prohibited statements are actually those statements that most players decide to choose first. we are not interested to collect these statements all time, so we do not give the opportunity to the player (narrator) to use them.

However, the players should use these templates, as we build OWL ontologies by aid of these templates, but we give also the option to the narrator to build arbitrary sentences as well, if the templates can not be useful. These arbitrary sentences will build comments for the generated ontology.

3 Generating OWL-based Ontology

In this section, we introduce the translation mechanism that we use to generate OWL-based ontologies. The Ontology will be created for the object that the players are playing, e.g. book, computer, car. After every play using an object (item), we collect some common sense facts about that item and we can build an ontology for that. The first iteration of generating ontologies is draft and can not be considered as a complete ontology. In other words, the ontology is created during several iterations and not at the first time.

3.1 Concepts

For the approved properties, i.e. the properties that their frequencies are more than a threshold, a *owl:class* is generated. These classes are actually the transformation of properties into OWL representation using a mediator/mapper which is simply able to generate classes and their properties. Suppose a domain like a *book*: For every approved property or concept, a class and a link will be generated to associate this class to main concept which in our example is a book. Figure 1 illustrates the mapping between some selected properties and their OWL representations. As we mentioned earlier, the properties will be stored in a knowledge base (KB) and as soon as they are *mature* enough to be linked, the mapper will translate them into OWL and link them to the main concept. In the following sections, we provide a more detailed description.

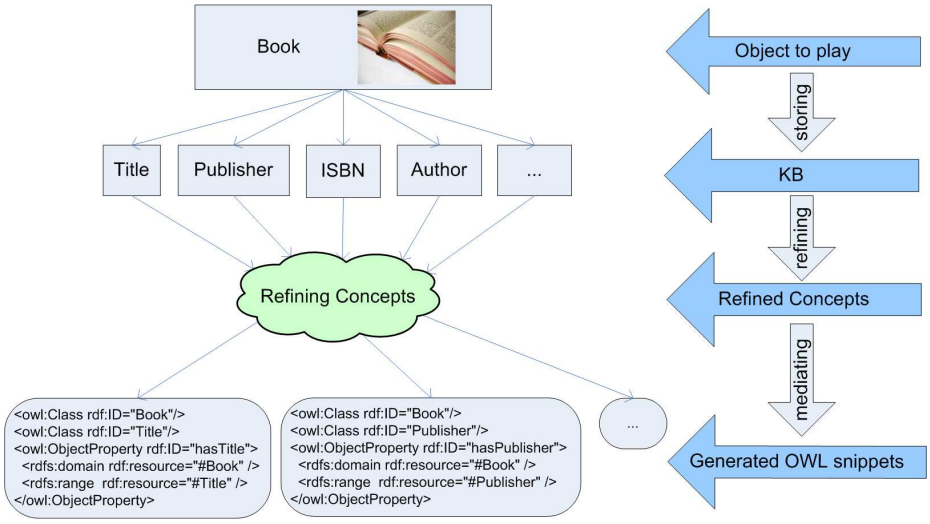

Fig. 1. Generating OWL for Properties Using a Mapper/Mediator

Pre-Refinement of Concepts (Refining Before Mediation). As we mentioned earlier, the concepts need to be refined. The refinement process is as follows: Because a specific object can be played more than once, we assign a counter to every object and the counter increases if the players are playing that object. We call this counter *objectCounter* in which the word *object* will be replaced with the explicit name of the object. A counter is also assigned to every property that the players agree upon that during the game and after further agreement by other players, the counter increases. We call this counter *objectPropertyCounter* which *object* will be replaced with the explicit name of the object and *property* will be replaced with the explicit name of the property of the object. The *variance* is defined for each property and is calculated by *objectCounter* minus *objectPropertyCounter*. If the result is greater than *threshold1*, the property will be moved to prohibited list, as many pairs agreed upon that property and if it is less than *threshold2*, the property will be deleted, as only very few pairs agreed upon that property. Note that, we do not care about uppercase and lowercase of alphabetic letters. Listing 1.1 demonstrates the pseudocode of this refinement.

Listing 1.1. Pseudocode of Refining Concepts

```
1  if (object is selected) then
2          objectCounter++;
3
4  if (objectProperty is selected) then
5          objectPropertyCounter++;
6
7  variance(objectProperty) = objectCounter - objectPropertyCounter;
8
9  if (normalize(variance(objectProperty)) > threshold1) then
10         move objectProperty to prohibited list;
11
12 if (normalize(variance(objectProperty)) < threshold2) then
13         delete objectProperty;
```

Concept Mediator/Mapper. Concept mediator/mapper is simply a mapper that gets the property or concept as input and generates OWL statements as output. The OWL statement contains also the link that associates the property to the main object. Figure 1 demonstrates some sample inputs and outputs of the mediator/mapper. However, in this step, we do not have our ontology and we have just gathered only properties and built their links. The ontology will be created after gathering sufficient facts about the object.

Post-Refining (Refinement After Mediation). After generating OWL representations of properties, they need also to be purified. Refining statements is an iterative task and tries to build a summarized version of statements based on resource URIs. Figure 2 demonstrates a sample of this post-refinement.

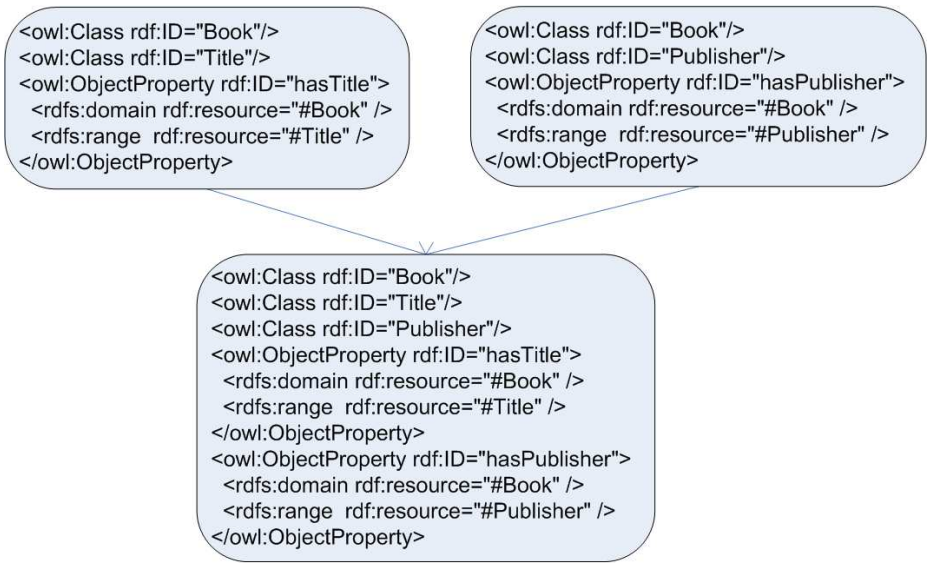

Fig. 2. Properties Refinement Sample

3.2 Statements

In the previous section, we presented the fixed templates that we use to gather common sense facts about objects. As we mentioned, those templates were carefully chosen for two main purposes: First, to be able to be translated into OWL using a mediator/mapper and second, to avoid the game being boring, as we need to entertain players, instead of assigning tasks to them. Table 1 demonstrates the general translation of templates. Note that *&xsd;* refers to XSD namespace which is actually *xmlns:xsd = "http://www.w3.org/2001/XMLSchema#"*. To avoid a huge messy table, we decided to use acronyms.

Table 1: Templates and Their OWL Representations

Template	**Generated OWL**
X has at least ___ *Y*	<owl:Restriction> <owl:onProperty rdf:resource = "#hasY" /> <owl:minCardinality rdf:datatype = "&xsd;nonNegativeInteger"> *some value* </owl:minCardinality> </owl:Restriction>
X has at most ___ *Y*	<owl:Restriction> <owl:onProperty rdf:resource = "#hasY" /> <owl:maxCardinality rdf:datatype = "&xsd;nonNegativeInteger"> *some value* </owl:maxCardinality> </owl:Restriction>
X is kind of ___	<owl:Class rdf:ID = "*some value*"/> <rdfs:Class rdf:resource = "#X"> <rdfs:subClassOf rdf:resource = "#*some value*"/> </rdfs:Class>
X could be either ___ or ___ *(or more)*	<owl:Class rdf:ID = "*some concept*"/> <owl:Class rdf:ID = "*other concept*"/> <owl:Class rdf:ID = "*more concept*"/> <owl:Class rdf:ID = "X"> <owl:intersectionOf rdf:parseType = "Collection"> <owl:Class rdf:about = "#*some concept*"/> <owl:Class rdf:about = "#*other concept*"/> <owl:Class rdf:about = "#*more concept*"/> </owl:intersectionOf> </owl:Class>
X could be union of ___ and ___ *(and more)*	<owl:Class rdf:ID = "*some concept*"/> <owl:Class rdf:ID = "*other concept*"/> <owl:Class rdf:ID = "*more concept*"/> <owl:Class rdf:ID = "X"> <owl:unionOf rdf:parseType = "Collection"> <owl:Class rdf:about = "#*some concept*"/> <owl:Class rdf:about = "#*other concept*"/> <owl:Class rdf:about = "#*more concept*"/> </owl:unionOf> </owl:Class>
X is complement of ___	<owl:Class rdf:ID = "*some concept*"/> <owl:Class rdf:ID = "X"> <owl:complementOf> <owl:Class rdf:about = "#some concept"/> </owl:complementOf>

Continued on next page

Table 1 – continued from previous page

Template	Generated OWL
	</owl:Class>
X is disjoint with (opposite of) ___	<owl:Class rdf:ID = "*some concept*"/> <owl:Class rdf:ID = "X"> <owl:disjointWith> <owl:Class rdf:about = "#some concept"/> </owl:disjointWith> </owl:Class>
X is equivalent to ___	<owl:Class rdf:ID = "*some concept*"/> <owl:Class rdf:ID = "X"> <owl:equivalentClass> <owl:Class rdf:about = "#some concept"/> </owl:equivalentClass> </owl:Class>

Pre-Refinement of Statements (Refinement Before Mediation). The main goal of *Pre-Refinement* is to select the statements that can be translated into correct OWLs. The process is as follows: Like previous refinement, we assign a counter to an object. we call this counter *objectCounter2*. We assign also a counter to every instance of a template related to object. We call this counter *objectTInstanceCounter*. We log all instances that will be sent to guesser. If the instance was helpful and the guesser could guess the word correctly, we increase the *objectTInstanceCounter*, but if the instance was not useful and the guesser was not able to guess the word, we decrease the *objectTInstanceCounter*. We compare the *objectTInstanceCounter* with some thresholds and then we decide whether to keep, delete or move it into the prohibited list. Note that in this refinement, we do not care about uppercase and lowercase of alphabetic letters. Listing 1.2 demonstrates the pseudocode of this refinement.

Listing 1.2. Pseudocode of Refining Instances

```
1  if (object is selected) then
2          objectCounter2++;
3
4  if (objectTInstance was helpful) then
5          objectTInstanceCounter++;
6  else
7          objectTInstanceCounter--;
8
9  variance(objectTInstance) = objectCounter2 - objectTInstanceCounter;
10
11 if (normalize(variance(objectTInstance)) > threshold3) then
12         move objectTInstance to prohibited list;
13 if (normalize(variance(objectTInstance)) < threshold4) then
14         delete objectTInstance;
```

Statement Mediator/Mapper. Statement mediator/mapper is simply a mapper that gets the template instance as input and generates OWL statements as output. The OWL statements also contain all necessary links to the main object.

Table 1 demonstrates the OWL translation of some fixed templates. Note that the italic words are those variable words that are used by the narrator.

Post-Refinement (Refinement After Mediation) and Ontology Assembler. After generating OWL representations, they need to be purified. Refining statements is an iterative task that tries to build a summarized version of statements based on resource URIs. Figure 3 demonstrates a sample of statement refinement.

As we mentioned earlier, the fixed templates are just highly-recommended proposals to be used. If they are not helpful for the narrator to help the guesser, he/she may simply use English sentences. As these sentences have no structure, we keep them as comments for the ontology, if they were helpful for guesser.

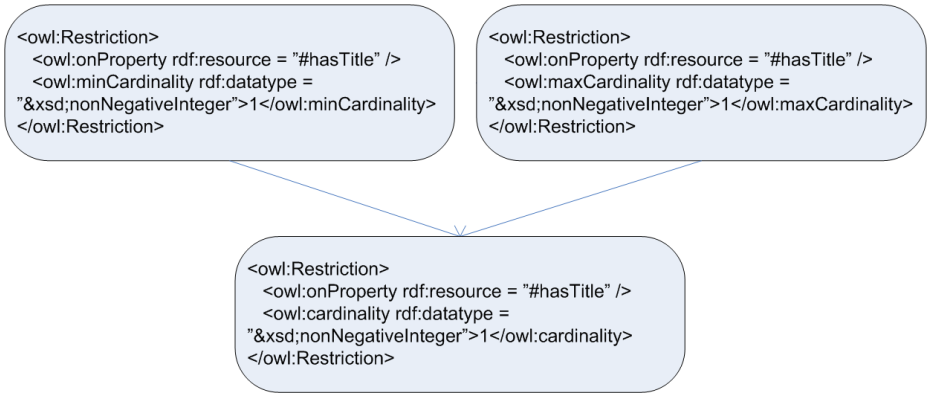

Fig. 3. Statement Refinement Sample

After all these processes, the general assembler is able to merge these statements and build the first version of the ontology. This is an iterative task and the ontology will be completed after several plays. Every Ontology has a version track using *owl:versionInfo* that enables us to keep the history of generated ontologies. Figure 4 demonstrates the iterative life cycle of generating ontologies.

4 Experimental Results

To evaluate the quality of the generated ontologies, we have checked how they change with an increasing number of rounds. To make our presentation feasible, we have reduced the number of rounds to ten and the number of concepts to two (tree and book).

In the first round (see Section 2.1), the properties *color*, *height*, and *age* were collected for the word *tree*. After ten rounds, we additionally collected *leaves* and *species*. The same test performed for the word *book* resulted in five properties:

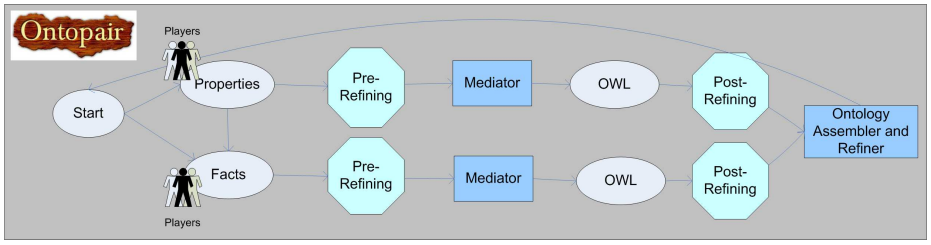

Fig. 4. Iterative Life Cycle of Generating Ontologies

author, *language*, *publisher*, *title*, and *year of publishing*. Five more rounds gave us additionally three more properties: *number of pages*, *language* and *index*. Tables 2 and 3 present the results that we have collected; we show both the words that affected the created ontology and the words that were rejected. However, the rejected words can become properties of the ontology, if we perform more rounds.

By analyzing more and more examples, we noticed that the number of properties does not grow linearly with the number of rounds. Additionally, some of the players were using plural versions of the words. This problem can be solved, however, by using dictionaries. Moreover, the results provided by the native speakers were much more accurate and they responded faster. We suggest using the lists of forbidden words; such lists impose users, specially non-English-spoken players, to use more and more sophisticated vocabularies, otherwise they stop getting points at some time. Hence, they have to learn new vocabularies.

The next part of our experiment was to evaluate the second phase (see Section 2.2), in which each person was asked a set of questions related to the common sense facts. Again we used the same words: *tree* and *book*. For the word *tree*, there were just three questions that let the players to successfully complete a round: *it is a kind of a plant*; *it has at least 1 height*; *it could be either oak or larch*. Five more rounds introduced additionally two more facts to our knowledge base: *it is disjoint with animals*; and *it has at least 1 root*. The same example for the word *book* resulted in three common sense facts in five rounds: *it has at least 1 edition*; *it has at least 1 language*; *it could be either hard-copy or electronic*. Five more rounds resulted in two new statements: *it has at least 1 author*; and *it has at least 1 title*. Again we note that more and more rounds are necessary to improve the quality of the ontologies.

Table 2: Results of Phase 1: Tree

Rounds	**Accepted Words**	**Rejected Words**
5	Color, Height, Age	Bark, Animals, Location, Kind, Fruit, Root, Branches, Green, Flower, Species, Width, Status, Leaves Falling, Seeds,

Continued on next page

Table 2 – continued from previous page

Rounds	Accepted Words	Rejected Words
		Kind
10	Color, Height, Age, Leaves, Species	Bark, Animals, Location, Kind, Fruit, Root, Branches, Green, Flower, Width, Status, Type, Name, Leaves Falling, Seeds, Kind

Table 3: Results of Phase 1: Book

Rounds	Accepted Words	Rejected Words
5	Author, Language, Publisher, title, year of publishing	Pages, Chapters, Words, Paragraph, Index, Foreword, Thickness, audience age, ISBN, Wtext, abstract, color
10	Author, Language, Publisher, title, year of publishing, number of pages, publishing, Language	Pages, Chapters, Words, Paragraph, Index, Foreword, Thickness, audience age, ISBN, text, abstract, color, cover type, domain

5 Discussions

The aim of the *OntoPair* game is to build simple ontologies for different objects that are located in images or even text-based objects in a short time. Our main concern is that the game should be entertaining to encourage people to play it. For this reason, we should avoid complex domains to be *played*. Some complicated concepts like business categorizations can be out of scope of this game, as these complicated domains may make the game boring and players will not come back again. The other point is that the generated ontologies may not contain all information regarding a domain, as the players are very ordinary people and not from Semantic Web domain. This is the main advantage of the game, as it cleverly uses people from different domains to help the Semantic Web domain experts and scientists. However, we believe that ontologies will be complicated after each play.

Even though we proposed that the players should be randomly paired, there exist some cheating potentials; players could agree to login at the same time to be paired together and maliciously annotate the objects. To avoid this case, based on previous plays, at some random times, we propose presenting specific images or texts that we know exactly the properties of objects in them and if we notice that the players are not playing honestly, we let them play as long

as they want. The same solution is foreseen for second phase of the game. As we mentioned, to increase certainty, we only assign properties and statements to objects, if and only if a certain amount of players agreed upon that. As an example, if only two players agreed upon *a car has wing* among other players, we give a low ranking to *wing* and after filtering the properties using a threshold, we omit the *wing*.

Statistics and our experiences show that word guessing games are played by many people as these games are entertaining. Many people from non-English speaking countries play these game to improve their English.

For evaluating the generated ontologies, the game can be played in single mode and the single player will play against already-generated ontologies. If generated ontologies contain sufficient knowledge, the guesser should be able to guess the correct words, otherwise a low ranking will be assigned to the generated ontology. The other approach towards evaluating OntoPair is comparing the generated ontologies with ontologies that have been created by domain experts; e.g. we can compare two ontologies for a domain like *book*, one from OntoPair repository and the other which has been generated by hand.

6 Related Works

In [11], the authors present an approach for building ontologies using a game called OntoGame. They use Wikipedia articles as conceptual entities, present them to the players, and have the users judge the ontological nature and find a common abstractions for a given entry [11]. Our approach is different, as we do not build a tree structure for objects. In two phases, we gather properties and cardinalities plus different instances of an object.

There exist also some efforts towards building a knowledge base by means of computer-based games. These games have been designed mostly for two players. The ESP game [7] tries to annotate images by enforcing players to come up with the exact objects located in images. Peekaboom [9] is another game which tries to come up with approximate location of objects in an image. Verbosity [8] is a word guessing game which composes of two players: narrator and guesser; The former should guide the latter to come up with the word that he is looking for by using some fixed templates for this purpose. Common Consensus [4] is very similar to Verbosity [8], but it has its own templates which begin mostly with *Wh** questions. Phetch [12] is another game which is composed of two players: narrator and guesser; the narrator should give guesser some keywords to help him/her to select the right image from a list of images. In other words, Phetch's main goal is finding a specific image in a bunch of similar images.

There exist also some other efforts in this general direction mostly for designing single player games. *Labelme* [2] is one example which assigns you an image for annotation. *Cyc* [1] is an artificial intelligence project that attempts to assemble a comprehensive ontology and database of everyday common sense

[1] http://www.cyc.com/

knowledge, with the goal of enabling AI applications to perform human-like reasoning [14]. Cyc offers a web-based game called *FACTory* [2] which gives the single player several sophisticated common sense facts regarding different domains and the player should mark them as true or false statements in a short time period.

At the beginning of 1980s Wille [18] initiated his work on a theory known as Formal Concept Analysis. The aim of the theory is to analysis data and identify conceptual structures among data sets. This work rapidly expanded several years later and has been successfully applied for some specific domains, e.g. bio-medicine [1]. However, such an approach often requires domain experts to approve the results.

7 Conclusion and Future Works

We have presented our work towards OntoPair, a game that uses Collective Intelligence for building OWL-based ontologies. OntoPair collects properties and common sense facts about an object in an entertaining environment and builds simple domain ontologies. We described how players should compete and how computers should process and integrate results. We also performed a simple experiment showing now our knowledge base grows. Our prototype implementation is still being implemented[3] and it needs some work in the data and user management areas. Moreover, the future work will include a reputation model that will give more impact to users who are given high esteem. Linking different ontologies together can be also considered as next phase. As an example, if we build an ontology for a *wheel*, and we have a common sense fact indicating that *a car has wheel*, we may link the car and wheel ontologies. Furthermore, we would like to perform more experiments to research how long would it take for a domain expert and ontology engineer to build an equivalent ontology. We also would like to test OntoPair in more specific domains.

Acknowledgments. The authors would like to thank Dr. Axel Polleres for his valuable comments. This work is partially supported by Ecospace (Integrated Project on eProfessional Collaboration Space) project: FP6-IST-5-35208, Lion project supported by Science Foundation Ireland under Grant No. SFI/02/CE1/I-131, and Enterprise Ireland under Grant No. *ILP/05/203*.

References

1. Franz Baader, Bernhard Ganter, Baris Sertkaya, and Ulrike Sattler. Completing description logic knowledge bases using formal concept analysis. In Manuela M. Veloso, editor, *IJCAI*, pages 230–235, 2007.
2. Bryan C. Russell, Antonio Torralba, Kevin P. Murphy, and William T. Freeman. LabelMe: a database and web-based tool for image annotation. In *MIT AI Lab Memo AIM-2005-025*, 2005.

[2] http://207.207.9.186/

[3] `http://sourceforge.net/projects/OntoPair`

3. Dan Brickley and R.V. Guha. Resource Description Framework (RDF) Schema Specification. http://www.w3.org/TR/rdf-schema/, February 2004.
4. Henry Lieberman, Dustin Smith, and Alea Teeters. Common Consensus: A Web-based Game for Collecting Commonsense Goals. In *Workshop on Common Sense for Intelligent Interfaces, ACM International Conference on Intelligent User Interfaces (IUI-07)*, Honolulu, Hawaii, USA, 2007. ACM Press.
5. Google Inc. Google Image Labeler. http://images.google.com/imagelabeler/, 2007. Online; accessed 3-May-2007.
6. Pierre Levy. *Collective Intelligence*. Plenum Publishing Corporation, January 1997.
7. Luis von Ahn, and Laura Dabbish. Labeling images with a computer game. In *CHI '04: Proceedings of the 2004 conference on Human factors in computing systems*, pages 319–326. ACM Press, 2004.
8. Luis von Ahn, Mihir Kedia, and Manuel Blum. Verbosity: a game for collecting common-sense facts. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 75–78, New York, NY, USA, 2006. ACM Press.
9. Luis von Ahn, Ruoran Liu, and Manuel Blum. Peekaboom: a game for locating objects in images. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 55–64, New York, NY, USA, 2006. ACM Press.
10. Sean Bechhofer, and Frank van Harmelen, and Jim Hendler, and Ian Horrocks, and Deborah L. McGuinness, and Peter F. Patel-Schneider, and Lynn Andrea Stein. OWL Web Ontology Language Reference. http://www.w3.org/TR/owl-ref/, February 2004. Online; accessed 2-May-2007.
11. Siorpaes Katharina, and Martin Hepp. OntoGame: Towards Overcoming the Incentive Bottleneck in Ontology Building. In *3rd International IFIP Workshop On Semantic Web and Web Semantics (SWWS '07), co-located with OTM Federated Conferences, Vilamoura, Portugal*, pages 1222–1232, 2007.
12. Luis von Ahn, Shiry Ginosar, Mihir Kedia, Ruoran Liu, and Manuel Blum. Improving accessibility of the web with a computer game. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 79–82, New York, NY, USA, 2006. ACM Press.
13. Wikipedia. Captcha — wikipedia, the free encyclopedia, 2007. [Online; accessed 14-December-2007].
14. Wikipedia. Cyc — wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=Cyc&oldid=125786119, 2007. [Online; accessed 7-May-2007].
15. Wikipedia. Guessing game — wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=Guessing_game&oldid=116214370, 2007. Online; accessed 6-May-2007.
16. Wikipedia. Human-based computation — wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=Human-based_computation&oldid=122965665, 2007. [Online; accessed 7-May-2007].
17. Wikipedia. Collective intelligence — wikipedia, the free encyclopedia, 2008. [Online; accessed 17-March-2008].
18. R. Wille. Restructuring lattice theory: An approach based on hierarchies of concepts. In *Ordered Setsand in I. Rivals (Ed.)*, volume 23, 1982.

Semantically Enhanced Webspace for Scientific Collaboration

Daniel Harezlak[1], Piotr Nowakowski[1], and Marian Bubak[1,2]

[1] Academic Computer Center CYFRONET AGH
ul. Nawojki 11, 30-950 Krakow, Poland
[2] Institute of Computer Science AGH
al. Mickiewicza 30, 30-059 Krakow, Poland
d.harezlak@cyf-kr.edu.pl

Abstract. The paper presents an approach to constructing a collective Web-based system for knowledge management. The work refers to the concepts and ideas widely promoted by modern web communities, such as user-created and user-annotated content or reliable search mechanisms. Also, formal ways such as ontology-to-model dependencies within collective knowledge are used to build the proposed system. The main focus of this effort is directed towards scientific communities in which large amounts of experimental data need to be classified and verified. For this purpose an enhanced set of available Web tools needs to be assembled and made available as a unified system.

Key words: semantic models, web management, application plan, collaborative research

1 Introduction

The need to represent knowledge by a language that both people and computers can comprehend is obvious and has been proven almost a decade ago [1]. Since then significant effort was invested in combining the formalisms of descriptions that can be parsed by computers with free-text content published by people all over the world, creating the new notion of the Semantic Web. According to the survey [2] the Semantic Web is increasing its momentum by expanding in the areas of Internet computing such as trade, business and travel, not to mention the science domain. Currently we observe that the technologies and tools used for knowledge representation and management are becoming more stable and thus models and services are being proposed [3, 4] to realize the vision of large-scale knowledge integration.

This paper focuses on scientific aspects of the Semantic Web, especially on knowledge- and data-intensive applications, which need to better benefit from the possibilities that become available through the manifestation of the Semantic Web and its extensions. The basic challenge is to combine the collaborative and global methods of using Web resources with individual and geographically-scattered research activities. Many modern approaches try to exploit the techniques available in social Web management such as tagging, ranking or editing

Web content by all users. However, more formal mechanisms are required for scientific purposes. This goal can be supported by applying a strict semantic framework to the way in which Web research is conducted. That is why we propose a solution that incorporates a semantic layer into the available Web management routines to facilitate scientific research.

A need for such environment was observed in the ViroLab project [5] which develops a virtual laboratory [6] to facilitate medical knowledge discovery and provide decision support for HIV drug resistance [7]. Three groups of users have been identified: clinicians using decision support systems for drug ranking, experiment developers who plan complex biomedical simulations, and experiment users who apply prepared experiments (scripts) [8]. An experiment is a kind of processing which may involve acquiring input data from distributed resources, running remote operations on this data, and storing results in a dedicated space, which should not only limit its functionality to the medical disciplines but extend into other areas of science.

The following section contains current achievements in the Semantic Web area. Subsequently, a list of requirements for the proposed solution is presented. The following two sections contain the architecture and proposals of semantic enhancements, followed by current implementation status and a summary with a future workplan.

This work tries to go beyond the present state in building scientific web communities by proposing a system which covers traditional computation infrastructures with lightweight yet reliable and oriented on research web interfaces supporting knowledge management. In principle, it builds upon existing achievements of Semantic Web, however, a novel approach of managing semantic descriptions by web community members is introduced. This requires new combinations of tools for managing semantic metadata and social techniques of editing web content.

2 Related Work

Modern systems in which semantic descriptions are used to represent knowledge generally apply tested and reliable languages, such as OWL [9], which is based on an older RDF specification [10]. Another standard used by a significant group of people is WSMO [11], which provides methods to semantically describe Web services. A problem, however, arises when different groups of researchers try to create descriptions of the same phenomena or elements of reality, resulting in inconsistencies when such descriptions are merged. This requires manual alignment, which can be very time-consuming and inefficient. In order to efficiently build ontologies, a semiautomatic tool is required to provide feedback on preexisting descriptions and enable scientists to further build upon them, thus ensuring coherency.

It is easy to observe that the social Web has evolved into a global collaboration space where people from all over the world exchange experience using systems such as Facebook [12] or Flickr [13]. This way of collaboration has made

the Web an interesting tool for scientific communities, with which to exchange research results and knowledge. Several attempts were undertaken to benefit from those ideas, resulting in applications like [14] and new trends in semantic computing [15]. These attempts, however, still lack general acceptance and stability. Nevertheless, several environments are already available and are being used by minor groups. For example, myExperiment [16], currently in its beta testing phase, is a successor to well-accepted workflow management systems such as Taverna [17] or BIOSteer [18]. The project delivers a Web-based system for sharing workflows among community members; however, the infrastructure does not provide features that allow workflow execution and result management.

3 Requirements

In order to satisfy potential researchers, any new system should ease their work. Therefore, basic requirements should be identified first. Below we present a list which attempts to formalize the process in which research is conducted. In particular, it is assumed that each type of supported scientific research can be aided either by applying a computer system to conduct a virtual experiment (such as a simulation) or by presenting the results in a digital format. Following is a list of basic requirements for a knowledge Web management system.

- *application plan storage* - The notion of an application plan exists in various domains of science and can be described as a list of steps necessary to achieve a certain result. There are many ways to represent such a list. It can be accomplished either by building a workflow (e.g. with the BPEL [19] notation) or by using a script (with any available scripting language). The requirement is to provide a facility for application storage that can be accessed by authorized users. In this way published applications can be discovered, reused, assessed and improved by other scientists.
- *managing application execution* - For the application plan execution to be possible, an underlying infrastructure has to be deployed and a proper application plan execution engine needs to be set up. The whole process of application execution has to be visualized to the user and, if necessary, intermediate results should be delivered.
- *managing scientific results* - The outcome of a research activity should be represented by a result stored in a dedicated database. The results should be properly annotated and classified, available for other scientists for verification purposes.
- *collaborating with other scientists* - The system should provide collaboration tools enabling scientists to discover their work, properly restricted by security and copyright agreements. It also should be convenient to exchange experience and validate other's work within one system.

The presented list of requirements should be supported by a semantic model that facilitates all the functionalities which are to be provided by the proposed system.

Another non-functional requirement is to separate the processes of application development and conducting research. On the one hand developers do not want to be laden with the semantics of a certain research area but only restrict to e.g. data format, computation optimization, etc. On the other hand scientists want to focus only on the research without knowing the specifics of the actual implementation. This requires a certain separation layer provided by the experiment plan. the common parts between the mentioned groups are only notions of experiment plan, input data and experiment result. Developers write experiments together with underlying services, components, etc., wchich require input data and produce results (of course the format of the data is to be agreed between those two groups). The researchers execute the experiments, validate and classify the data being able to manage the semantic layer.

One last requirement that was identified is the cross-disciplinary cooperation of researchers. Creating a global and ultimate ontology seems to be an impossible challange. However, it might be possible to find intersections between them and benefit from what others work on. The approach in the proposed system is to make all the semantic metadata available to all participants. In order to do that an advanced editor is required to assist the researchers in the process of managing the metadata.

4 System Architecture

4.1 General Overview

In Fig. 1 the basic architecture is presented. The system is divided into four layers. At the bottom, the resource layer consists of services and data sources which are used to build application plans using workflow or script notations that provide some level of abstraction. In the same layer the *Metadata Store* and the *Application Repository* are deployed and used to archive semantic data and application plans respectively. The last two components are accessed by the Web application layer (shown in green) directly. The next, yellow layer is the middleware which provides an abstraction over the low-level resources and ensures unified access to the variety of technologies that implement data sources and computational services. In this way access to data and services is seamlessly woven into the notation. The *Application Execution Engine* also maintains the state of the applications during execution.

The third layer, representing Web applications, contains two modules, namely the *Metadata Engine* module and the *Execution Client* module. The first module is the one responsible for managing semantic descriptions available in the system. It also constitutes a filter and a tool that helps users manage the semantic content they provide or browse. Based on the semantic model presented in the next section users are able to:

– import their own semantic descriptions by semi-automatically aligning and mapping them against existing ones,

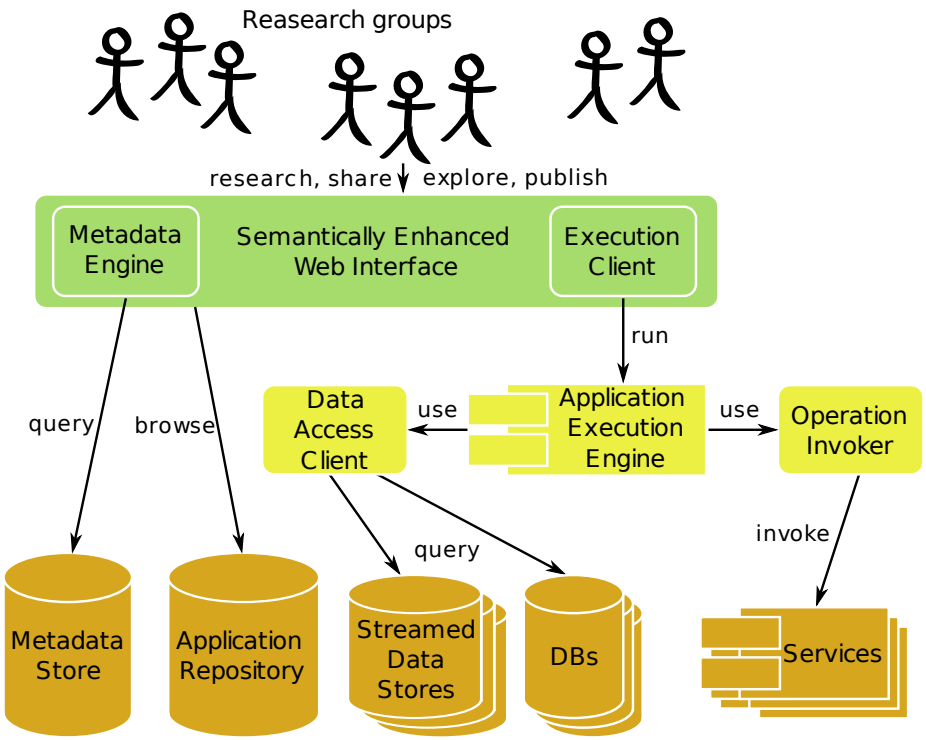

Fig. 1. Basic components of the proposed system.

– browse the existing knowledge by conveniently searching through existing ontology triples,
– quickly obtain application plans, results or publications of interest by providing key words (the whole knowledge space is tagged and annotated),
– tag and annotate the existing objects in the knowledge space.

The second module - *Execution Client* - is responsible for communication with the application execution engine and keeping the users updated with the current execution status using AJAX-oriented techniques (e.g. implemented with the GWT toolkit [20]).

4.2 Metadata Engine

The *Metadata Engine* is the main component which provides the reasoning functionality over the ontologies built within the system. It covers the low-level *Metadata Store* and exposes convenient methods to manage the knowledge structure.

In Fig. 2 a detailed architecture of the *Metadata Engine* is presented. It contains a client that enables it to access the underlying metadata store and facilitates the use of the query language used by the store. The deduction module

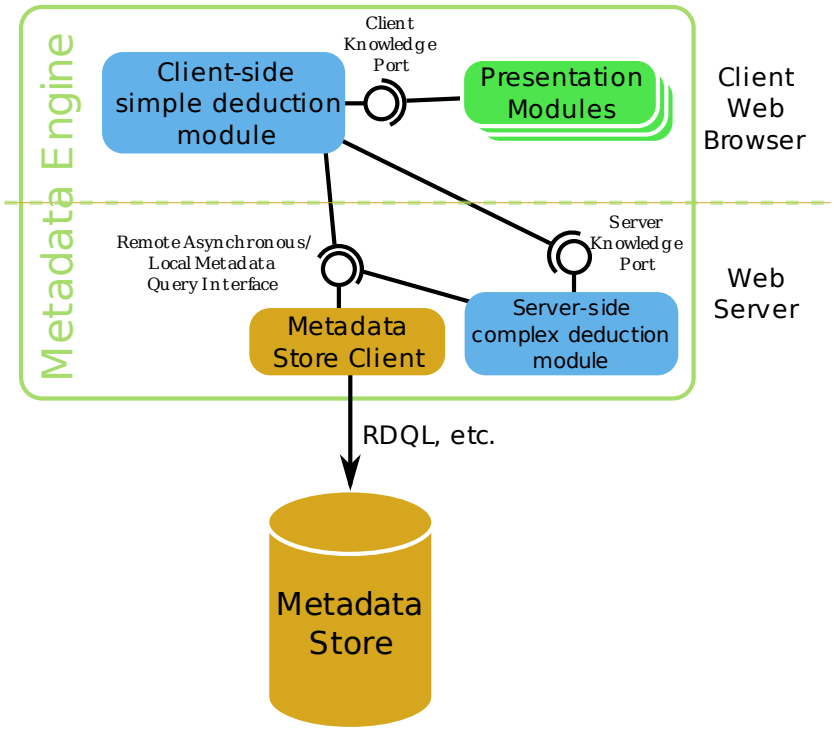

Fig. 2. Internal architecture of Metadata Engine.

is divided into two parts. For simple queries for which response times should be short the part on the client-side is used. It communicates with the client through an asynchronous channel according to the techniques used in web client-server communication models (built over standard request-response model). The calls are made directly by the visual components which concludes with their visual state update. If the queries are more complex then the deduction module on the server-side is used. To the visual components this however is transparent with only longer repsonse times.

5 Semantic Enhancements

5.1 Basic Approach

In Fig. 3 a sample of the ontology model is presented. This model is used as the basis for the *Metadata Engine* module to manage the collaboration space.

The model consists of three parts:

– *Science Domain* - (blue) - This part of the semantic description is extendable by users. This ensures that the model remains dynamic and, when required,

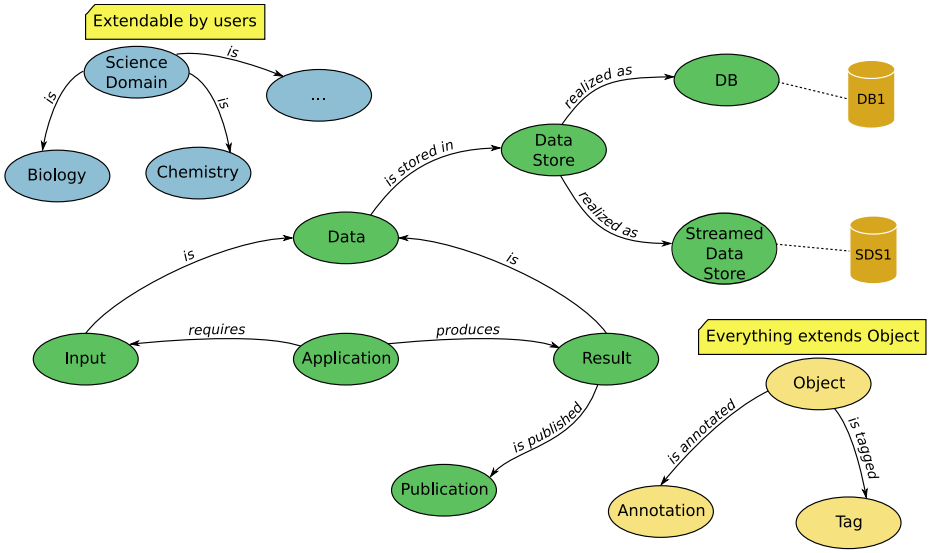

Fig. 3. Samples of semantic descriptions used in the proposed system.

users may add custom ontological descriptions to existing ones. The process is semi-supervised by the system in order to maintain coherency.

– *Basic Model* - (orange) - This model is the core of the application and its basic models. It assumes (in accordance with social Web content management) that every item within the collaboration space may be tagged or annotated. This enables the space to be enhanced by a quick search mechanism or by building a tag cloud (used for space browsing).
– *Application model* - (green) - This ontology model allows the *Metadata Engine* to keep track of the content managed by users. In particular, users are able to submit specific queries that navigate to accurate pieces of data stored in the collaboration space (e.g. list all publications that describe the outcomes of a particular application plan, etc.)

The presented model is just a proposition, showing how the final implementation could look and it remains a subject of ongoing research. It is also possible to test several different models in different research contexts.

5.2 Role and Ontology Management

In order to ensure hierarchy in the process of managing and building the ontologies proper groups need to be modelled with certain permissions. Also, a way of assessing the quality of the ontologies is required to introduce formal models of the management process.

Figure 4 depicts a sample structure of such ontology. the main *Object* node is assigned the *is editable by* relation which specifies what roles are permitted to

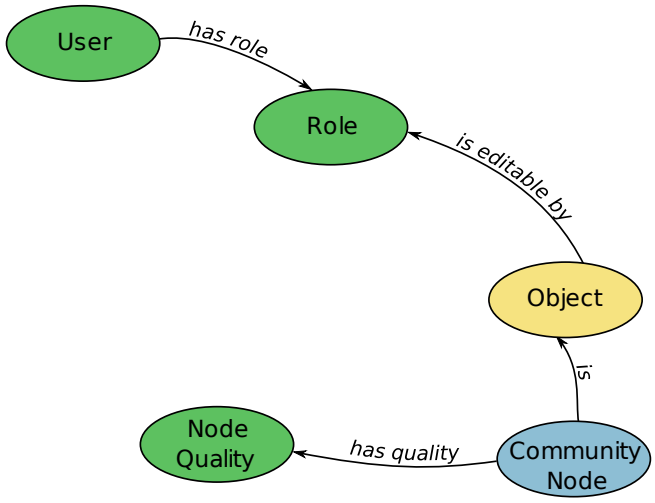

Fig. 4. Role management dependency semantics.

edit a given node. All *Role* nodes are referred by *User* nodes which creates the authorization net in the proposed model.

To enable users with the possibility of extending the current ontology graph a *Community Node* is introduced. This node is inherited by all the nodes created by community members and in the process of collaborative cooperation of scientific communities it is assessed and the quality information is stored in the individuals of the *Node Quality*. The quality will be measured by analyzing statistics of use of such knowledge node(e.g. the more users use and cite a given ontology node the higher rank it has). Further improvements of such approach will categorize the semantic descriptions into approved and validated and those still being unassessed. Hopefully, this will lay ground for building community ontologies across different science domains. The model itself may be changed while the system is working.

6 Implementation Status

Currently the presented model is being implemented within the virtual laboratory supporting the scripting approach to representing application plans [6]. The application execution engine is already [21] operational and capable of running test application plans. Simple ontology models have been built; however, they still require user assessment in order to be improved.

With respect to the web application layer a prototype of the user interface was built and a screenshot is depicted in Fig. 5.

The interface is divided into three parts:

– *application management* - In this widget the user is able to browse the collaboration space in search for application plans of interest. The search is

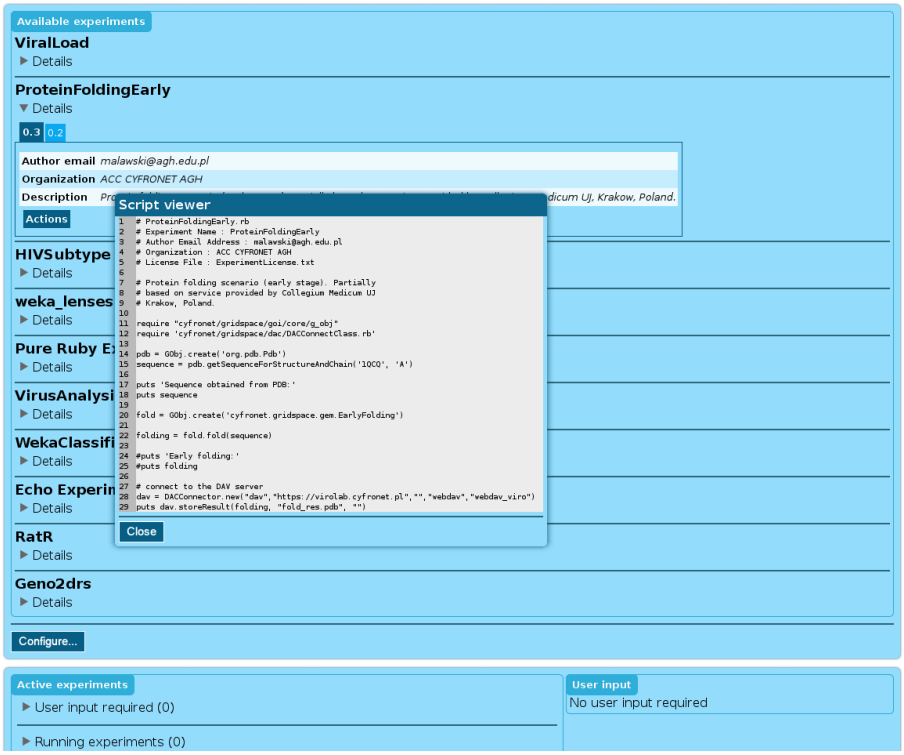

Fig. 5. Screenshot of a semantic collaboration space interface prototype.

supported by the *Metadata Engine*, so the application plans can be found according to the history of previous executions, produced results, owners or publications.

- *result management* - Results are managed by this view. Annotations and tags can be provided to assess particular results.
- *application execution status* - With this tab users may follow the execution status of their application plans and input intermediary data. The input is also supported by the *Metadata Engine* and previous results may be used as the inputs. When result type model is provided the engine suggests suitable inputs.

The overlapping window in the middle is displayed as popup and in this case is used to show the application plan script. Each application plan may be supplied with a license regarding its usage restrictions.

7 Conclusions and Future Work

This paper presents a semantic Web-based approach to constructing a scientific collaboration space. The solution combines social Web routines with the

formalisms of semantic content descriptions to facilitate the process of on-line research. Main improvements of the approach include integration of the application runtime system with result management and adoption of widely-used Web content management techniques in the area of scientific research.

At present the ViroLab virtual laboratory already integrates biomedical information related to viruses (proteins and mutations), patients (viral load) and literature (drug resistance); it enables to plan and run experiments transparently on distributed resources. Different experiments from the virology domain are executable, such as: from virus genotype to drug resistance interpretation, querying historical and provenance information about experiments, assisting a virologist with the Drug Resistance System, a simple data mining with classification. Further work will extend the list and explore re-usability in different science disciplines.

Future plans include the extension of the semantic model used for building the prototype and extending the user community to test and assess the approach. The aim is to benefit from the ideas brought by the Semantic Web trends and extend the present solutions in the area of community-driven research to make the process more reliable and efficient.

Acknowledgments. This work is partly funded by the European Commission under the ViroLab IST-027446 and the IST-2002-004265 Network of Excellence CoreGRID projects.

References

1. Chandrasekaran, B., Josephson, J.R., Benjamins, V.R.: What are ontologies, and why do we need them? IEEE Intelligent Systems **14**(1) (January/February 1999) 20–26
2. Cardoso, J.: The semantic web vision: Where are we? IEEE Intelligent Systems **22**(5) (September/October 2007) 84–88
3. Missier, P., Alper, P., Corcho, O., Dunlop, I., Goble, C.: Requirements and services for metadata management. IEEE Internet Computing **11**(5) (September/October 2007) 17–25
4. Carroll, J.J., Dickinson, I., Dollin, C., Reynolds, D., Seaborne, A., Wilkinson, K.: Jena: Implementing the semantic web recommendations. Technical report, HP Labs (2003)
5. Virolab Consortium: ViroLab - EU IST STREP Project 027446 (2008), `http://www.virolab.org`
6. ACC CYFRONET AGH: Virolab virtual laboratory (2008), `http://virolab.cyfronet.pl`
7. Sloot, P.M., Tirado-Ramos, A., Altintas, I., Bubak, M., Boucher, C.: From molecule to man: Decision support in individualized e-health (2006)
8. Gubala, T., Bubak, M.: Gridspace - semantic programming environment for the grid. LNCS 3911 (2006) 172–179
9. W3C: Owl web ontology language (2004), `http://www.w3.org/TR/owl-features`
10. W3C: Rdf: Resource description framework (2001), `http://www.w3.org/RDF`

11. Roman, D., Keller, U., Lausen, H., de Bruijn, J., Lara, R., Stollberg, M., Polleres, A., Feier, C., Bussler, C., Fensel, D.: Web service modeling ontology. Applied Ontology **1**(1) (January 2005) 77–106
12. Facebook Team: A social utility that connects people with friends and others who work, study and live around them (2008), `http://www.facebook.com`
13. Yahoo! Inc: Photo sharing web space (2008), `http://www.flickr.com`
14. Fox, G.C., Guha, R., McMullen, D.F., Mustacoglu, A.F., Pierce, M.E., Topcu, A.E., Wild, D.J.: Web 2.0 for grids and e-science. In: INGRID 2007 - Instrumenting the Grid, 2nd International Workshop on Distributed Cooperative Laboratories - S.Margherita Ligure Portofino. (2007)
15. Goble, C., Roure, D.D.: Grid 3.0: Services, semantics and society. In: Proceedings of Cracow Grid Workshop 2007, ACC CYFRONET AGH (2008) 10–11
16. The University of Manchester and University of Southampton: myexperiment home page (2008), `http://www.myexperiment.org`
17. Oinn, T., Addis, M., Ferris, J., Marvin, D., Senger, M., Greenwood, M., Carver, T., Glover, K., Pocock6, M.R., Wipat, A., Li, P.: Taverna: a tool for the composition and enactment of bioinformatics workflows (2004)
18. Lee, S., Wang, T.D., Hashmi, N., Cummings, M.P.: Bio-steer: A semantic web workflow tool for grid computing in the life sciences (2007)
19. OASIS: Web services business process execution language (2007), `http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel`
20. Google: Google web toolkit (2008) `http://code.google.com/webtoolkit`
21. Ciepiela, E., Kocot, J., Gubala, T., Malawski, M., Kasztelnik, M., Bubak, M.: Gridspace engine of the virolab virtual laboratory. In: Proceedings of Cracow Grid Workshop 2007, ACC CYFRONET AGH (2008) 53–58