Linked Data-Based Social Bookmarking and Recommender System

Vladimir Apostolski1, Ljupco Jovanoski1, Dimitar Trajanov1

1Faculty of Computer Science and Engineering - Karpoš II bb, PO Box 574, 1000 Skopje, Macedonia
{vladimir.apostolski, ljupco.jovanoski, dimitar.trajanov}@gmail.com

Abstract. Social Bookmarking services have spread over the last few years. People often use tagging to organize and share their bookmarks. But this process can also be overloading the users and recommender systems are a popular approach to address this issue. Our goal was to explore the potential of services for semantic annotation and entity extraction to assist tagging and generate recommendations in social bookmarking communities. For that purpose, we have built a prototype of a Linked Data-based recommender and social bookmarking system. The system uses Zemanta to generate semantic tags and later the tags are the basis upon which recommendations are calculated. After that, a set of people used the application, gave feedback and evaluated the recommendations the system generated. In addition, we give a proposal of how tag connections to Linked Data entities expressed with MOAT ontology represent boost reusability and interoperability of gathered information.

Keywords: Social bookmarking, semantic web, semantic annotation, web services, data, information, sharing, entity extraction, Linked Data, collaboration

1 Introduction

Social bookmarking is a concept that has gained in popularity in the last few years. Due to the rapid expansion of the Web, and share web resources they found relevant. But the popular social bookmarking services easily became overflowed with bookmarks, making it difficult to navigate through them. A partial solution was adding meta-data to the stored web resources, known as tagging.

Tags represent a flexible and popular way to describe and share web resources in communities on the Web. They are simple to use and can connect resources across different categories. Despite that, due to tags’ increasing number, users are prone to information overload and counterproductive effects. These characteristics make recommender systems a candidate for filtering and discovering relevant content in social bookmarking environments. But problems in social tagging systems are also yielded to recommender systems that use tags in order to make recommendations.
As denoted in [1][2], collaborative tagging systems have downsides such as defining vocabulary, multiple tags with same meaning, misspelling and ambiguity.

There are efforts to correlate tags between different social online communities, but with limited success because of the lack of meaning of the provided tags [3][4]. Consequently, these tag correlations are difficult to provide personalized recommendations based on the meaning of the tags. Furthermore, in [5] Alag states that the collective set of terms or tags in an application defines the vocabulary for the application. When this same vocabulary is used to describe both the user and the items, we can compute the similarity of items with other items and the similarity of the item to the user’s metadata to find content that’s relevant to the user.

The goal is to propose a way of implementing a semantic tag-based recommender system in social bookmarking communities that will not suffer from the issues of social tagging with reusable vocabularies. This paper covers three aspects: (1) How can services for semantic annotation assist tagging in social bookmarking communities, (2) how can the semantic tags generated by these services be used to make relevant recommendations to the users and (3) how linking semantic tags with Linked Data concepts can improve interoperability and reusability of gathered information.

1.1 Related work

There are related systems that include work on recommender systems like the work of Marinho et al. [6] who find that tags can be used in order to make recommendations for users and web pages. Song [7] shares the same aspect towards recommendation types, but applies machine learning techniques to automate the process. Algorithms like SocialRank [8] are proposed to overcome inefficiencies caused by typical problems in social tagging systems. It ranks the recommendations based on the inferred semantic distance of the query to the tags weighted by the similarity of the querying user and the user who created the tags. In [9], data mining approach is used to merge tags in hierarchical clusters, hence dealing with tag redundancy. But these approaches lack the reusability of tags among different communities.

On the other hand, Semantic Web technologies are also proposed as candidates for recommender systems that could overcome tagging problems and make data reusable. The semantic recommender systems, are described in [10]. Their anatomy is described by Dell’Aglio [11], as well as their major drawback: modeling, building and maintenance of the knowledge base. She also emphasizes the usage of the Linked Open Data cloud to partially solve the problem of maintenance and data interoperability. Authors of ConTag [12], claim that services for semantic annotation and entity extraction can provide relevant tag recommendations for documents, solving the tagging problems. It uses services like Yahoo Term Extraction service to extract entities, WordNet and custom DefTag service to bind them with meanings and align them to an ontology. Other such services that extract Linked Data entities are reported to be applied in a variety of applications. For example, OpenCalais¹, has been used in Tell Me More, an application that given an input text, mines the web for similar stories to

¹ http://opencalais.com
extract entities [13]. Also OpenCalais is reported to make alerts to Semantic MediaWiki administrators when certain entities occur in RSS feeds [14]. Zemanta is reported to be used in a social bookmarking platform with semantic tags, bound to Wikipedia articles, called Faviki [15][16]. It is also used [17] as a basis for recommendation tool when authoring Powerpoint presentations, but connections of entities to Linked Data are not leveraged. NERD is an API that unifies results from various entity extracting services in one ontology and classifies the content according to it [18]. Its authors also make a feature comparison matrix for existing services and also adopt the idea to link extracted entities to Linked Data concepts [19]. Similarly, Halb and team [20] focus on the business potential of linking entities to Linked Data cloud in enriching content for general news articles. Moreover, TagMe! [21] is an application that attempts to assign DbPedia URIs to tags on Flickr images, so that external information could be extracted. All this research outlines the potential and need for connecting entities to Linked Data concepts.

1.2 Our Contributions

Having the related work in mind, we built a prototype of Linked Data-based social bookmarking and recommender application that integrates Zemanta, using the retrieved entities as (semantic) tag recommendations basis. This way we addressed both traditional tagging problems, as well as data reusability. Among the reasons we took this approach of using semantic annotation services for social bookmarking are: automated entity extraction and disambiguation, reduced amount of tedious work from user perspective, reusability and tag data openness in a lightweight manner. We estimated the extent to which these semantic tags can be used to provide relevant personalized recommendations to the users for bookmarking pages, help them discover users with similar bookmarking habits as well. The system is built under two assumptions: (1) services for semantic annotation and entity extraction are treated as black box; (2) Bookmarked pages are articles from different topics or user generated content in English. The recommendations are content-based. We conducted an experiment with real users to evaluate the recommendations by providing positive or negative feedback.

Tagging issues like synonyms, misspellings and defining contextual meaning were addressed by the services for semantic annotation and entity extraction behind the scenes. Tags’ meanings were expressed with resources from popular sources like DbPedia and Freebase. Tags were saved for each bookmarked web page and they were treated as “similarity” factor between them. We built term vectors for each bookmark (vector of tags and their associated weights) and calculated cosine similarity, which is determined by the dot product of term vectors.

We also explored how Zemanta can be used to give relevant recommendations for external resources i.e. web pages that are not bookmarked by any user yet (external recommendations). We chose Zemanta based on researches and conclusions of other authors; for example Faviki, NERD’s and other comparisons [22][23][24].

http://zemanta.com
Also we found that a benefit of using service-generated semantic tags that point to Linked Data entities, not only ease traditional tagging problems, but opens space for other semantic applications to leverage that data. That makes data re-usable and interoperable with other systems. To illustrate this, we used ontology for semantic tags to express the relationships between the tags, users and bookmarked web pages with the Linked Data cloud.

2 System Overview

2.1 Architecture

The architecture of the system is given on Fig. 1. The main stored concepts are tags, bookmarks and users. For each user, it is known which pages are bookmarked, along with the tags used. There are three engines in the system: bookmarking, recommender and semantic engine. The bookmarking engine enables users to create, store and tag bookmarks. The recommender engine provides recommendations. The bookmarking engine retrieves tag recommendations provided by the recommender engine. For that purpose, the recommender engine interacts with Zemanta, the service for entity extraction. For other recommendation types, it uses the stored data in the system. The semantic engine transforms data to machine-readable format. It establishes the connections to the Linked Data cloud.

**Bookmarking Engine.**

Users interact with the bookmarking engine when they want to bookmark a web page via a bookmarklet. When a web page is selected for bookmarking by the user, the bookmarking engine asks the recommender engine to suggest tags to the user. After the user chooses the tags to describe the web page she wants to bookmark, the bookmarking engine stores the bookmarked web page along with the associated tags.

**Recommender Engine.**

In our system, there are three types of recommendations: for tags, bookmarks and users. Tag recommendations are suggested tags when the user wants to bookmark a page. Bookmark recommendations help users discover web pages that might interest them and user recommendations refer to suggesting other users with similar interests.
We will show how each recommendation type is generated in the following subsections:

**Recommendations of Tags.**

When the user wants to bookmark a web page, tag recommendations are automatically generated. A HTTP request is made to the bookmarking web page and its HTML response is passed to the service for semantic annotation and entity extraction. The semantic annotation and entity extraction service (Zemanta) processes the request and the returned results are parsed. Entities returned are treated as tags with description, meaning, relative importance towards the submitted URL (in range \([0,1]\)). We use this weight to calculate term vectors in recommendations for bookmarks and users. In cases where the user inputs a tag, its relative weight is set to 1, because we consider that tag to be specifically relevant for the user.

**Recommendations of Bookmarks.**

Bookmark recommendations represent a feature to offer additional web pages for bookmarking to a given user. They can be internal or external. Internal bookmark recommendations refer to web pages that are already present in the system (meaning another user already bookmarked the same page). Internal bookmark recommendations can be personalized and non-personalized. Personalized internal recommendations are calculated based on the cosine similarity of the term vectors of the bookmarks and the term vector of the logged-in user. The term vector of the bookmark consists of the weight \((W)\) of each tag that it is tagged with. Each tag represents a dimension in the vector. We adapted the method proposed in [25] to our model. The term vectors for bookmarks are represented like in Table 1. Table 1. Calculation of term vectors for bookmarks based on tag weights.

<table>
<thead>
<tr>
<th>Tag</th>
<th>Tag1</th>
<th>Tag2</th>
<th>Tag3</th>
<th>…</th>
<th>Tagn</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bookmarks1</td>
<td>(W_{11}/M_1)</td>
<td>(W_{12}/M_1)</td>
<td>(W_{13}/M_1)</td>
<td>…</td>
<td>(W_{1n}/M_1)</td>
</tr>
<tr>
<td>Bookmarks2</td>
<td>(W_{21}/M_2)</td>
<td>(W_{22}/M_2)</td>
<td>(W_{23}/M_2)</td>
<td>…</td>
<td>(W_{2n}/M_2)</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>Bookmarks(m)</td>
<td>(W_{m1}/M_m)</td>
<td>(W_{m2}/M_m)</td>
<td>(W_{m3}/M_m)</td>
<td>…</td>
<td>(W_{mn}/M_m)</td>
</tr>
</tbody>
</table>

A normalization magnitude is calculated for each vector, in range \(i \in [1, m]\):

\[
M_i = \sqrt{\sum_{j=1}^{n} W_{ij}^2}
\]

Then, all weights are normalized to a vector with magnitude equal to 1. After that, we compute the term vector for the currently logged-in user as in Table 2:
Table 2. Computing Term vector for the user that is logged in

<table>
<thead>
<tr>
<th>User</th>
<th>$W_1/M$</th>
<th>$W_2/M$</th>
<th>$W_3/M$</th>
<th>...</th>
<th>$W_n/M$</th>
</tr>
</thead>
</table>

Each of the weights ( $W_1, ..., W_n$) represent the number of times that user has used that tag when bookmarking a page.

We calculate the normalization magnitude:

$$ M = \sqrt{\sum_{i=1}^{n} W_i^2} $$ (2)

In order to generate the recommendations, we calculate the dot products ( $P_1, ..., P_m$) between each bookmark term vector and the user term vector Table 3:

Table 3. Calculated dot products ( $P_1, ..., P_m$) between term vectors for bookmarks and term vector for the current user

<table>
<thead>
<tr>
<th>User</th>
<th>$P_1$</th>
<th>$P_2$</th>
<th>$P_3$</th>
<th>...</th>
<th>$P_m$</th>
</tr>
</thead>
</table>

Recommendations are sorted in descending order by the dot product $P_i$ where $i \in [1, m]$. Of course, pages that are already bookmarked by the user are filtered out and do not belong to the recommendations.

Non-personalized internal recommendations occur when the user inspects the details of a given bookmarked page. The process is the same except that the dot product is calculated between term vectors for bookmarks.

When the user is inspecting the details of the bookmark, recommendations for similar bookmarks (ordered by the dot product values) appear to the user. Of course, the similarity between the bookmark itself is 1 and is not included in the list of recommendations.

External recommendations refer to web pages that are not already present in the system. They represent a way to introduce new bookmarks in the system. External recommendations leverage Zemanta’s suggest\(^3\) method to get web pages that are not bookmarked yet, but might be of interest to the users. When analyzing content, this method returns related articles which it aggregates from various sources.

Recommendations of People.

To extend the social aspects of the application, it recommends other users who could have similar interests. Like for the bookmarks, the calculation of these recommendations consists of constructing term vectors for all users, normalizing them and then calculating the dot product between them. In this case, the weight for each tag represents the number of times a user has used that tag.

\(^3\)http://developer.zemanta.com/docs/suggest/
Semantic Engine.

As outlined in the introductory part of this paper, one of the problems with tags in online communities is the ambiguity and difficulty of their re-usability and interoperability of tag data with other systems or communities. This piece of the application covers those aspects. In order to make the tags and bookmarks externally accessible, we used the MOAT ontology\(^4\). The MOAT ontology is chosen because it is easily extendable to bridge the gap between tagging and Linked Data, according to Passant and Laublet [26]. According to Hak Lae, Breslin, Scerri, Decker and Hong Gee [27], this is the lastly developed ontology and suitable for modeling tags, meanings and concepts.

The semantic engine of our system exposes data in OWL 2 XML format with the MOAT ontology to enable external interaction with the data. It is available via URL endpoint.

Disambiguated tags have at least one meaning accompanied with URI to Linked Data cloud.

The service for semantic annotation and entity extraction we used includes entity URIs from sources like: DBpedia, Freebase, LinkedIMDB, MusicBrainz etc. We can dereference the URIs of the semantic tags to pull data from other applications (for example extracting information from Wikipedia). From that, one can tell what entities is a web page about, rather than raw unstructured text tags.

Due to openness, external applications can extract tag data that is present in our system. For example, external application can query for web pages tagged with Paris, defined as resource with the following URI: http://dbpedia.org/resource/Paris. Since this is a global URI, both applications can identify the exact resource they refer to, rendering the data interoperable between them. This also opens space for data translation engines like Mosto and LDIF to leverage the data with other vocabularies [28].

3 Evaluation and Results

A set of 41 users was asked to bookmark the sites they find relevant for a period of 7 days. The users were free to bookmark any content that is an article, user-generated content that contains text and that is written in English. The assumption is also that all users behave as instructed. The point is for each user to give to rate the recommendations that are provided. Every time the user logs in, he is asked by the system to rate the recommendations he got with positive or negative mark. That applied to all types of recommendations. We considered each giving of a mark to be one feedback. After that we collected the results and got the following facts:

<table>
<thead>
<tr>
<th>Table 4. Facts gathered from the evaluation of the recommendations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total number of users</td>
</tr>
<tr>
<td>Number of bookmarked pages</td>
</tr>
<tr>
<td>Total number of tags</td>
</tr>
</tbody>
</table>

\(^4\)http://moat-project.org/ontology
<table>
<thead>
<tr>
<th>Tags connected to Linked Data entities</th>
<th>859</th>
</tr>
</thead>
<tbody>
<tr>
<td>User-generated tags</td>
<td>69</td>
</tr>
<tr>
<td>Total Feedbacks</td>
<td>171</td>
</tr>
<tr>
<td>Positive feedback for non-personalized internal bookmark recommendations</td>
<td>82%</td>
</tr>
<tr>
<td>Positive feedback for personalized internal bookmark recommendations</td>
<td>84%</td>
</tr>
<tr>
<td>Positive feedback for external bookmark recommendations</td>
<td>78%</td>
</tr>
<tr>
<td>Positive feedback for recommendations for users</td>
<td>81%</td>
</tr>
</tbody>
</table>

Based on the facts given in Table 4, we can report that users gladly accept the service-generated tags (7% of the tags are user-generated). That indicates that existing tags were either reused or accepted the tags that the service generated. Because of this, common tagging problems like ambiguities are less likely to occur. Regarding the semantics of the tags and their interconnectivity with the Linked Data cloud, we notice that approximately 90% of the tags are accompanied by DbPedia and/or Freebase resources. This clearly shows that the tags and other data can be reused in other applications and vice versa – external information can be added to the existing application. Also, rules for addressing synonyms in the set of tags can be applied. For example, tags linked to the same Linked Data resources but with different labels should be treated as equivalent.

Regarding the feedback provided from the users: the average acceptance rate of 83% shows that semantic tags are a solid base for calculating recommendations for bookmarking pages. Results also show that the web service for entity extraction and semantic annotation helped the users discover relevant content in 78% of the cases, which is significant. Also users consider recommendations for similar users to be relevant in 81% of the cases, so that is a strong indicator of quality.

Based on the conducted evaluation, one can summarize that web services for entity extraction are mature enough to be used in recommender systems for social bookmarking.

4 Conclusion

In this paper, we showed how semantic tags generated by web services for entity extraction can improve tagging in social bookmarking environments. In addition, we proved the potential of those tags for making relevant recommendations to the users. Lastly, we propose means of leveraging connections of tags to popular Linked Data hubs, such as DbPedia.

The results we got lead us to the conclusion that: (1) services for semantic annotation and entity extraction generate relevant tag suggestions. (2) Recommendations based on that tags users also find relevant. (3) Using ontologies like MOAT to represent URIs to popular vocabularies like DbPedia can help overcoming tagging problems like disambiguity and reusability in different systems.
Future work may include leveraging the semantic tags for collaborative-filtering approaches when generating recommendations. Furthermore, we could explore the possible benefits of combining results from multiple web services like in the NERD platform. Also our team is thinking towards addressing common problems in recommender systems like the cold-start problem by using the links to the Linked Data cloud.

5 References