Volume 4, Issue 8, August 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Social Bookmarking Systems for Personalized Social Query Expansion

Pathak Darshita S.
Master of Computer Engineering Student, V.V.P. Engineering College,
Rajkot, India

Abstract— A new approach for social and personalized query expansion using social structures in the Web 2.0. It mainly focuses on social tagging systems , the proposed approach includes (I) the semantic similarity between tags composing a query, (ii) a social proximity between the query and the user profile, and (iii) on the fly, a strategy for expanding user queries

Keywords— information retrieval, image processing, social tagging,query expansion

I. INTRODUCTION

T he advent of online social community platforms (e.g., Flicker, del.icio.us, MySpace, Face book, or YouTube) has changed the way users interact with the Internet. While previously most users were mere information consumers, those platforms are offering an easy and hassle-free way for typical users to also publish their own content, making the users also information producers. As a matter of fact, it has been shown that (in particular teenage) users today spend the majority of their online time on such platforms — as of December 2006, US Internet users spent 11.9% of their total online time on MySpace [16]. On these social platforms, users are encouraged to share photos, videos, opinions, to rate content, but also to explore the online community and to and people with similar interest profiles.

Figure 1

In this sense, online social community platforms not only change the way people interact with the Internet, but also the way users interact with each other. While differing in the type of content that they focus on (e.g., blog entries, photos, videos, bookmarks), almost all online social community platforms work similarly. Initially, users must register in order to join the community. Once registered, they start to produce information, ideally by publishing their own documents1 and by adding tags (or ratings, comments etc) to other content already available in the community. The platforms also offer a way to maintain a list of friends and means to keep friends informed about your latest content items. The size of your friend network is often considered as your reputation in the network; making new "friends" often seems at least as important as publishing new content.

While initially, many users populate the list of friends with people they already know from the offline world or other online communities, as time goes by they typically identify previously unknown users that they share common interests with and also add those users to the friends list.

One particularly interesting feature of these communities is the widely-used opportunity to attach manually generated annotations (so-called tags) to content items. The typically high quality of user-generated tags suggests leveraging this "wisdom of the crowds" for identifying and ranking high-quality and high-authority content. However, the existing, traditional algorithms for searching on the Web fall short of being effective in social networks, as they disregard the social component and focus on the content quality only. This makes a strong case for novel methods that exploit the different entities present in social networks (users, documents, tags) and their mutual relationships.

II. NETWORK GRAPH MODEL

This section casts the entities that occur in social networks into a common unified graph model, representing the different elements of a social network and their mutual relations tips. Such a graph will eventually allow well-founded query execution schemes that go far beyond ad-hoc retrieval models for social networks that often include many hard-to-tune parameters. We identify the following three major types of entities of social networks that are to be represented by nodes in our graph model:

- **User**: people that produce content either by publishing own documents or by tagging existing content
- **Tag**: keyword used to describe a particular content item
- **Document**: content item that is published by a user (e.g., blog entry, bookmark, photo)

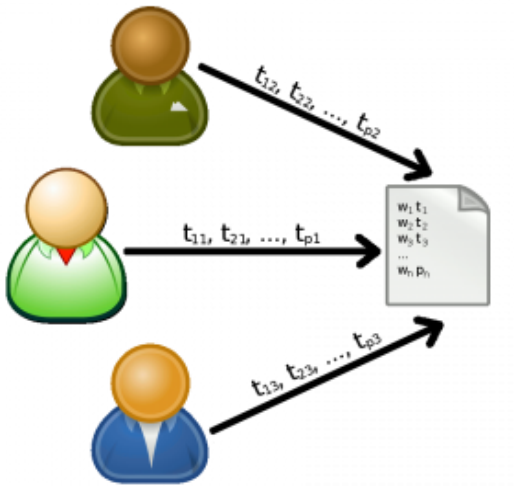

Figure 2

Additionally, social networks exhibit various relationships, both within the nodes of each type (intra-node type) and between nodes of different types (inter-node type) that are represented by edges in our graph. The idea to represent the social network as a set of ternary relations of the form (user,tag,document) is not representable by means of a graph. Therefore, we break this ternary relation into a set of binary relations and additionally extend the tuples to also consider scores, i.e., link weights to indicate the degree of relationship between two nodes. It is important to note that link weights can be defined in many different ways. The examples we provide in the following paragraphs present some of the alternatives, but are not meant to be exhaustive. Our model and the upcoming scoring and ranking models are independent from changes to the weighting functions.

III. QUERY EXPANSION WITH SOCIAL TAGS AS LOGICAL INFERENCE

Let K represent a knowledge system upon which all inference is made. Let d denote a document, and q denote a query. Then, the relevance of d to q with respect to this system can be expressed as $K \quad d \rightarrow q$. If one can prove that $K \quad d \rightarrow q$, then the document is said to be relevant to the query, otherwise the document is said to be irrelevant to the query. Nie [3] applies this representation to model QE, by defining a new query q that constitutes an expanded expression of the original query q. Then, by applying classical logic transitivity, the evaluation of $K \quad d \rightarrow q$ can be done as follows (K is removed henceforth): $d \rightarrow q \;\; V q \;\; \rightarrow q \quad d \rightarrow q$.

This relation means that the new query q is satisfied (implied) by the document, in which case the original query q is also satisfied by the document. Because q can be any query expression, the above deduction can be written as: $q \;\; (d \rightarrow q \;\; V \;\; q \;\; \rightarrow q) \quad d \rightarrow q$. Interpreting this formula in a context that involves uncertainty, the following function P can be defined:

$$P(d \rightarrow q) = P(\forall q \; (d \rightarrow q \; V \; q \rightarrow q)) \ldots\ldots\ldots\ldots\ldots\ldots (1)$$

where $P(d \rightarrow q)$ measures the degree of direct satisfaction of query q to document d, and $P(q \;\; \rightarrow q)$ measures the degree of relatedness of query q to the original query q. Eq. 1 can be interpreted as the probability $P(R|q, d)$ that a document d is relevant to a query q as follows: $P(R|d, q) = \;\; q \;\; P(R, q \;|d, q) =$
$q \;\; P(R|d, q, q \;)P(q \;|d, q)$. Assuming that q is a good approximation of q leads to: $P(R|d, q, q \;) = P(R|d, q \;)$. The derivation of q depends only on q, not on d, hence $P(q \;|d, q) = P(q \;|q)$. Based on this, we get the following expression:

$$P(R|d, q) = qP(R|d, q)P(q|q) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

where $P(R \;|d, q \;)$ denotes the relevance estimation of the document to the derived query, and $P(q \;|q)$ denotes the relationship between the original query q and the derived query q . Eq. 2 can be rewritten in order to express QE on the basis of individual terms, rather than whole queries, as follows (see [3] for the full derivation):

$$P(R|d, q) = tP(R|d, t)P(t|q) \ldots\ldots\ldots (3)$$

where t denotes a term in the expanded query. This formula allows us to consider the uncertainty of the correspondence between the expansion terms and the original query terms as a factor in the estimation of relevance. Eq. 3 has two components. The first component, $P(R|d, t)$, may be interpreted as the term weight within a document, and can be estimated by various different ranking models, for instance with Okapi's BM25 [4], which we use in this work. The second component, $P(t|q)$, may be interpreted as the term importance of a query, and has to be estimated in a way that reflects the probability of finding an expansion term in the query. Applied to our case of QE with tags, $P(t|q)$ denotes the probability of finding a tag (denoted τ) in the query. This probability must be estimated in a way that reflects the salience of the tag. We propose the following IDF-like approximation

$$P(\tau|q) = N/n\tau \ldots\ldots\ldots\ldots\ldots\ldots (4)$$

where N is the number of documents in the collection, and $n\tau$ is the number of documents in the collection that contain the tag τ. The aim of Eq. 4 is to discriminate between tags on the basis of how many documents within a large collection are associated to them (hence 'tagged' by them). Eq. 4 is one suggestion for estimating tag salience, which we evaluate experimentally in Section 3. Further alternative estimations are possible, for instance by relatively straight-forward extensions to IDF, such as RIDF [2], or by more elaborate approximations of tag topicality, such as Zhou et al.'s approach [6] that uses Bayesian Inference.

IV. PERSONALIZED QUERY EXPANSION

The Web has turned from a read-only infrastructure with passive participants into a read-write platform with active players. The content of the Web is no longer generated only by experts but pretty much by everyone (YouTube, Flicker, Last.fm, Delicious, etc). Like any popular revolution, this goes through democratizing the language: instead of subject indexing with a controlled vocabulary, freely chosen keywords are used to tag billions of items, e.g. URL (Delicious). The user generated taxonomy is called folksonomy (folk + taxonomy) and is used to label and share user-generated content (e.g photographs), or to collaboratively label existing content (e.g Web sites, books, or blog entries).

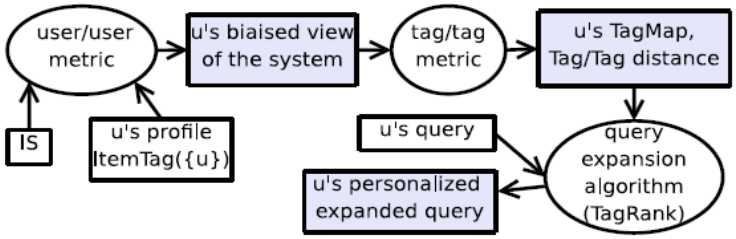

Figure 3

Part of the appeal of a folksonomy is its inherent subversiveness: folksonomies can be seen as a rejection of the traditional search engine status quo in favor of tools that are created by the community. In theory, precisely because folksonomies develop Internet-mediated

personalized environments, one could dynamically discover the tag sets of another user who tends to interpret and tag content in a similar manner. The result could be a rewarding gain in the user's capacity to and related content, a practice known as "pivot browsing". Personalization goes with decentralization. While intriguing, this Web revolution is still in a preliminary stage, and this is at least for two main reasons. First, most collaborative tagging networks are controlled by centralized systems. So as much as users are first class citizens of the system and are free to introduce new items and tag them in their proper language, they are not free to choose where these items are stored and, more importantly, cannot usually freely decide to remove items and tags.

In the long run, this might dissuade users from generating new content and expressing their tagging behavior in an explicit manner. Furthermore, and no matter how powerful servers can be, centralized solutions do not promote the maintenance of personalized relations between users, which might reveal crucial in the search as we will discuss below. These relations grow exponentially with the size of the system and the success of social tagging might simply kill the underlying centralized infrastructure. Second, while the success of collaborative networks is clearly related to the freedom left to the users, this is also a drawback.

The facts that such systems are not governed by specific structures (as opposed to ontologism for instances), and that tags are informally defined, and continually changing, mean there is no insurance that the tagging behavior of a user on some content makes any sense for another one, nor does it prevent junk tagging and synonyms, which introduce significant noise in the process. The reactivity ordered by fully decentralized solutions may solve this issue.

V. THE TAGRANK ALGORITHM: PERSONALISED QUERY EXPANSION

The TagMap represents the personalised relationships between pairs of tags to be used to expand queries. A straightforward solution, used in [3], to exploit the TagMap directly, is to consider only tags close to the tags of the query. This is an issue for the items suffer from a high sparsely: as there is a very large number of items, relationships between tags are sometimes hidden and can be hardly discovered. Consider for example a query on t1, the TagMap provides a link between t1 and t2 (based on a set of items). Consider now that t2 and t3 are also close in the same TagMap (based on a different set of items), this straightforward solution will never discover a link between t1 and t3.

By iterating on the set of added tags, more relevant tags could be added to the query. To this end, we designed an algorithm called TagRank, inspired from Page Rank[4]. The TagMap is represented as a graph in which all the tags in the TagMap are vertices. They are connected by weighted edges so that weight(ti; tj) =TagMap(ti; tj) and weight(ti; ti) = 1 3. In PageRank, a random surfer walks in a graph of Web pages. The importance of each page is the probability of the surfer to be on that page at any time. At each step of the walk, the surfer either follows a link on the page or moves to a page chosen uniformly at random on the whole graph.

In TagRank, the transition probability from one tag to another depends on the edge weight:

Transition Probability(t1; t2) = PTagMap[t1;t2] t2T TagMap[t1;t] The original PageRank algorithm computes a score for each vertex, but that score only depends on the structure of the graph, not on a user query. Like in personalized versions of PageRank, we modify the set of vertices the surfer can move to at random and limit it to the tags of the query. Therefore, the score computed is biased by the query, the query tags being the ones that spread importance into the graph.

Calculating exact TagRank scores in a big graph can be a long process. Since this process is repeated at each query, this might be an issue in the long run. Therefore, we use an algorithm from [5] in order to provide a more efficient approach. For each query, the computation is split in order to compute partial scores for each tag in the query. At the end, all the partial scores are added to get the TagRank score of each tag.

This saves a lot of processing time. The partial scores are approximated through random walks. TagRank(query; TagMap) outputs the list of all the tags in the TagMap associated with a weight. Since each weight is a probability, they sum up to one. The expanded query consists in the original query, plus additional terms chosen by descending weight. The system can either use the top-k extra tags, or add enough tags to \capture" a given amount of the weight.

VI. CREATING THE PERSONALNETWORK

The TagMap of a user is created from the pro file of users which belong to her personalized network. The aim of the personal network is so to connect a user with their k closest users according to the metric presented in the . k represents the trade-of between the amount of available information and the personalization degree of this information (in other words, its quality). We assume that the users are connected through an unstructured overlay implemented by a peer sampling service [1]. Basically, each user is provided with a (changing) random sample of the network (a view of say 20 random users). This protocol ensures that the network is connected and that new relevant users may be discovered when maintaining the personal network. The creation of the personal work is achieved through a clustering gossip protocol. To this end each user maintains a view of k neighbors forming its personalized network. Starting from a random sample (typically provided by the underlying peer sampling service),this network is re fined as follow. Periodically, a user contacts another user from her neighbors to exchange neighbors. When a user receives new neighbors upon a gossip interaction, it keeps from its own neighbors and the discovered one the k closest according to the metric de fined in Section 3.1. This process is iterated and converges in a few cycles [6]. The TagMap of each user is then built from the pro file of those k users, forming the personal network, In order to reduce the message size, users exchange a Bloom filter representing a hash of their items vectors instead of the whole pro file. The Bloom filter provides a reasonably good approximation of the user pro file that can be used to compute the cosine similarity with a small error margin. If the value of the cosine between the user's vector and the one inferred from the Bloom filter, the users are considered closes enough and the entire pro file is then exchanged. Otherwise, there is no further exchange. This avoids the transfers of useless entire profiles.

VII. EVALUATION

To evaluate our approach we run the following experiments on the same trace. Global TagMap, simple query expansion: a global TagMap is built based on the same metric, namely cosine of item vectors. The distance between tags is therefore not personalized as it takes into account the information of all users. The query expansion is the simple one considered before, used in [3] considering only the tags related to the query tags. This is typically representative of a centralized approach, where personalized TagMap are too space intensive to maintain.

Finally, we observe that TagRank also contributes to improving the quality of the results, especially when it comes to producing a long query expansion. The recall is improved by up to 4% with a query expansion size of 50. This experiment demonstrates the limits of the one step distance when using the TagMap. The sparsely of the information in folksonomies limits the number of related tags that can be found. Since TagRank distributes weight in the whole graph, it can find tags that seem not related to the query but are still relevant.

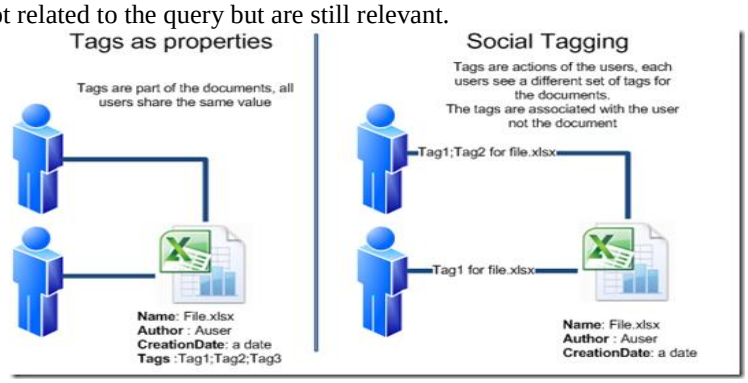

Figure 4

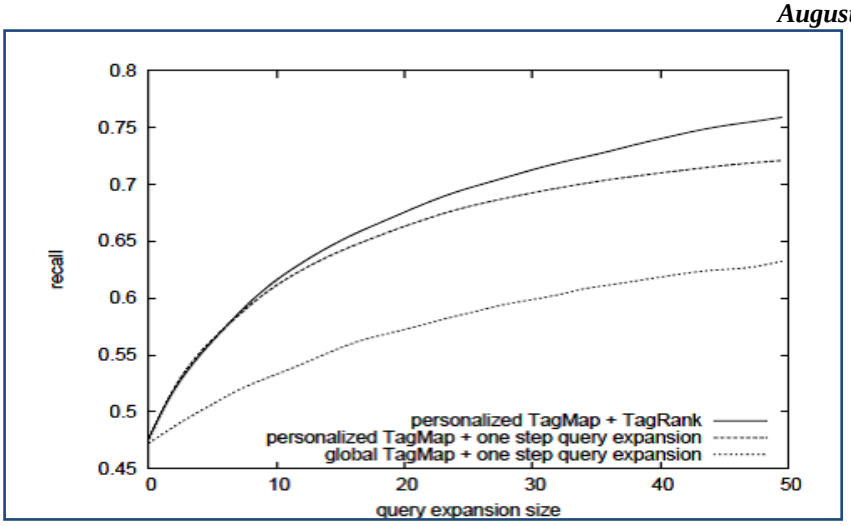

Figure 5

VIII. CONCLUSION & FUTURE WORK

Collaborative social tagging schemes have received agrowing attention, they provide a huge potential for discovering new information through implicit connections.

In this paper, we presented the query expansion feature of Gossple, a user centric system to discover and maintain such acquaintances. The Gossple query expansion mechanism improves the completeness of the search queries over state of the art alternatives without hampering the search accuracy. This is achieved with little information maintained at each peer, in the form of the TagMap. Interestingly, each peer discovers its personal network, locally stores the relevant, to itself, relationships between tags, and its tagging behavior is

only recorded by its personal neighbors. The TagMap

is exploited by an original TagRank algorithm to expand in an electives way queries. We should believe that the way to personalize Internet search, in a world where users are free to express their opinion and interests goes with a fully decentralized system. To the best of our knowledge, we are the forest to propose personalized query expansion in a fully decentralized manner. We foresee many perspectives to that work, such as leveraging the TagMap for recommendation systems for instance and addressing dynamic networks..

REFERENCES

[1] Amati, G.: Probabilistic Models for Information Retrieval based on Divergence from Randomness. PhD thesis, University of Glasgow (2003)

[2] Church, K.W., Gale, W.A.: Poisson mixtures. Natural Language Engineering 1(2), 163–190 (1995)

[3] Nie, J.-Y.: Query expansion and query translation as logical inference. JASIST 54(4), 335–346 (2003)

[4] Robertson, S., Walker, S., Beaulieu, M., Gatford, M., Payne, A.: Okapi at TREC-

[5] In: Harman, D.K. (ed.) NIST Special Publication 500-236: TREC-4, pp. 73–96.Springer, Heidelberg (1995)

[6] Xu, J., Croft, W.B.: Query expansion using local and global document analysis. In:SIGIR, pp. 4–11 (1996)

[7] Zhou, D., Bian, J., Zheng, S., Zha, H., Lee Giles, C.: Exploring social annotations for information retrieval. In: WWW, pp. 715–724 (2008)

[8] David Carmel , Naama Zwerdling , Ido Guy , Shila Ofek-Koifman , Nadav Har'el , Inbal Ronen , Erel Uziel , Sivan Yogev , Sergey Chernov, Personalized social search based on the user's social network, Proceedings of the 18th ACM conference on Information and knowledge management, November 02-06, 2009, Hong Kong, China [doi>10.1145/1645953.1646109]

[9] Matthias Bender et al. Exploiting social relations for query expansion and result ranking. In ICDE Workshops, 2008.

[10] Shenghua Bao , Guirong Xue , Xiaoyuan Wu , Yong Yu , Ben Fei , Zhong Su, Optimizing web search using social annotations, Proceedings of the 16th international conference on World Wide Web, May 08-12, 2007, Banff, Alberta, Canada [doi>10.1145/1242572.1242640]

[11] Peter Mika, Ontologies are us: A unified model of social networks and semantics, Web Semantics: Science, Services and Agents on the World Wide Web, v.5 n.1, p.5-15, March, 2007 [doi>10.1016/j.websem.2006.11.002]

[12] M. Jelasity, R. Guerraoui, A.-M. Kermarrec, and M. vanSteen. The peer sampling service: experimental evaluation of

[13] unstructured gossip-based implementations. In Middleware'04: Proceedings of the 5th ACM/IFIP/USENIX international conference on Middleware, pages 79{98, New York, NY, USA, 2004. Springer-Verlag New York, Inc.

[14] C. Cattuto, D. Benz, A. Hotho, and G. Stumme. Semanticanalysis of tag similarity measures in collaborative taggingsystems. In Proceedings of the 3rd Workshop on OntologyLearning and Population (OLP3), pages 39{43, July 2008.ISBN 978-960-89282-6-8.

[15] V. Zanardi and L. Capra. Social ranking: uncoveringrelevant content using tag-based recommender systems. InProceedings of the 2008 ACM conference on Recommendersystems, pages 51{58, 2008.

[16] S. Brin and L. Page. The anatomy of a large-scalehypertextual web search engine. Computer Networks and ISDN Systems, 30:107{117, 1998.

[17] D. Fogaras, B. Racz, K. Csalogany, and T. Sarlos. Towards scaling fully personalized pagerank: Algorithms, lower bounds, and experiments. Internet Mathematics, pages333{358, 2005.

[18] M. Jelasity and O. Babaoglu. T-Man: Gossip-Based Overlay Topology Management. Engineering Self-OrganisingSystems, 3910:1{15, 2006.

[19] S. Amer-Yahia, M. Benedikt, L. Lakshmanan, and J. Stoyanovich.E

