Tag Recommendations in StackOverflow

CS 224W Project Proposal

Logan Short, Christopher Wong, David Zeng

1 Introduction

In many community-based information web sites, such as StackOverflow, users contribute content in the form of questions and answers, allowing others to help and learn through the collaboration of many. These web sites often rely upon tags as metadata that assists in the indexing, categorization, and search for particular content with just a few key words. Almost always, users are given the responsibility to choose tags which identify their own content, and human error suggests, then, that content is not always well-tagged. As such, tag recommendation systems are helpful to guide users into labelling their content with the most appropriate tags.

In this paper, we propose our project to consider the problem of recommending tags in community-based information web sites such as StackOverflow. We begin by reviewing various research papers related to our topic and summarizing relevant key points in Section 2. Then, in Section 3, we critique the collection of papers to brainstorm possible ways we can extend upon current research in this area. Finally, in Section 4, we give our project proposal to explore and implement a better tag recommendation system using underlying network structure.

2 Literature Review

2.1 Tag Recommendation in Software Information Sites (Xia, Lo, Wang, and Zhou, 2013)

In [1], Xia et al. propose an automatic tag recommendation algorithm named *TagCombine* which analyzes objects in software information sites. The algorithm is divided into a model-building phase and a tag-prediction phase. The model-building phase builds the three components of *TagCombine*, each of which tries to assign the best tags to untagged objects: (1) multi-label ranking component, which predicts tags using a multi-label learning algorithm, (2) similarity based ranking component, which uses similar objects to recommend tags, and (3) tag-term based ranking component, which analyzes the historical affinity of tags to certain words in order to suggest tags.

The tag-prediction phase of the algorithm methodically assigns different combinations of weights to the three components, and for each combination, aggregates the components at these specific weights to form the prediction model. A *recall@k* score is then calculated for each prediction model from stratified 10-fold cross validation. Here, k represents the number of tags recommended for each object. (The *recall@k* metric is discussed more in Section 4.) The recall@5 and recall@10 scores of *TagCombine* are compared to those of other existing tag recommendation methods. In both scores, *TagCombine* does significantly better than all other cited models, including a 22.65% recall@5 and 14.95% recall@10 improvement over one algorithm.

2.2 Discovering Social Circles In Ego Networks (McAuley and Leskovec, 2014)

In [2], McAuley and Leskovec propose an algorithm to automatically discover social circles by analyzing the similarities among user profiles in data drawn from the social networking sites Facebook, Google+, and Twitter.

The algorithm begins by building pairwise feature vectors $\phi(x, y)$ to describe the similarities between user x and user y. Each circle C_k then has an associated θ_k that defines which components of $\phi(x, y)$ are important in defining membership of C_k. Using the idea that nodes are well connected within a circle, the algorithm finds the best θ_k, and uses them in conjunction with $\phi(x, y)$ to determine which edges should exist in the ego network.

McAuley and Leskovec examined the alignment of their predicted social circles with ground-truth circles by computing the Balanced Error Rate (BER) between social circles and the F_1 score. The proposed method took better accounted for network and community structure and performed better than other methods such as mixed-membership models when evaluated using BER and F_1 score.

2.3 Group Formation in Large Social Networks: Membership, Growth, and Evolution (Backstrom, Huttenlocher, Kleinberg, and Lan, 2006)

In [3], Backstrom et al. analyze the evolution of communities in social networks over time, using two datasets, LiveJournal and DBLP. Three features characterize the datasets: (1) users are connected to other users via friendship edges, (2) users are labeled with memberships in communities, and (3) communities reflect ideas through activities of the community.

Backstrom et al. open with an analysis of what features influence a user's decision to join a community, and then transition into an analysis of how communities evolve over time. The authors use a decision tree model to classify communities as fast- or slow-growing and show that connectedness of the community within and to those outside of the community plays a huge role in growth rate. Lastly, the authors discuss the relationship between movement of users and movement of ideas in communities, addressing the question: Do ideas follow users, or do users follow ideas? Observations show that most of the time, similar ideas appear in two communities before memberships align.

2.4 Statistical Properties of Community Structure in Large Social and Information Networks (Leskovec, Lang, Dasgupta, and Mahoney, 2008)

In [4], Leskovec et al. characterize the statistical and structural properties of network communities as a function of size using a metric called conductance. With the intuition that a strong community S has densely linked nodes and is sparsely connected to the rest of the network, conductance (ϕ) is defined as

$$\phi(S) = \frac{s}{v} = \frac{s}{s + 2e}$$

where s is the number of edges with one endpoint in S and one endpoint in its complement $\bar{S}$, v is the sum of the degree of all nodes in S, and e is the number of edges with both endpoints in S.

Leskovec et al. measure conductance over 70 sparse real-world networks. For each network, they create a Network Community Profile (NCP) plot, which characterizes the best communities of various sizes. In general, the NCP graphs showed that the communities with highest conductance consistently hover around 100 nodes. At small sizes less than 100 nodes, there are tight communities which can be combined into larger communities, while at large sizes well over 100 nodes, the communities start to “blend in” with the rest of the network and become less connected.

2.5 EnTagRec: An Enhanced Tag Recommendation System for Software Information Sites (Wang, Lo, Vasilescu, Serebrenik, 2014)

In [5], Wang et al. propose a tag recommendation system dubbed *EnTagRec* that aims to leverage both textual data and historical data to automatically recommend accurate tags for posts on software information sites such as StackOverflow. The proposed *EnTagRec* computes tag probability scores using two separate methods, Bayesian Inference and Frequentist Inference, and then obtains a final tag probability by taking a weighted sum of the probability scores.

Bayesian Inference relies on a post’s textual data to compute the probability that a given tag is associated with the post. *EnTagRec* formulates posts into a bag-of-words model and then trains a Labeled Latent Dirichlet Allocation model which is used to compute tag probability scores for a post. The Frequentist Inference used by *EnTagRec* first preprocesses a post to remove all words except for nouns and proper nouns. A set of tags is then inferred using a basic Frequentist Inference approach. Once this set of tags is computed, *EnTagRec* uses these tags to apply spreading activation to a tag network constructed by examining the co-occurrence rate of tags on the site. This step computes additional potentially relevant tags that are then also returned by the Frequentist Inference method.

Experimental results show that *EnTagRec* performs significantly better than *TagCombine* from [1] on Stack Overflow, Ask Ubuntu, and Ask Different datasets, but yields only comparable results on Freecode datasets.

3 Discussion and Brainstorming

We chose to first look at papers directly related to the topic of tag recommendation models for software information sites. In [1], Xia et al. propose a recommendation system that relates the textual features of posts to tags with reasonably good results. However, this paper has a few notable weaknesses. One weakness with [1] is that it fails to look at the network structure of software information sites in *TagCombine*. Posts on sites like Stack Overflow are ultimately connected to each other through an underlying network structure where users and tags that appear on multiple posts represent connections between said posts. In fact, tags exist in order to group similar posts and create an organized structure that allows for more convenient and logical browsing of posts. Thus, it makes sense that knowledge of the network’s structure could be used to enhance a tag recommendation system.

In [5], Wang et al. provide evidence that such an approach could yield significant improvement in tag recommendation results. In [5], the basic *TagCombine* model proposed in [1] is enhanced into a model that uses not only textual analysis of posts, but also network analysis of the tags themselves to obtain better results than *TagCombine*. Still the use of network structure in the model proposed in [5] is very limited, and further incorporation of network structure could potentially lead to more accurate tag recommendations. In fact on the Freecode data, the model in [5] does not perform any better than *TagCombine*, suggesting that room still exists for improvement.

This train of thought leads us to examine papers discussing the clustering of nodes in networks. In [2], McAuley and Leskovec discuss a method for automatically detecting “circles” in networks of users based on similarities in user profiles. A natural extension of this method would be to detect posts associated with common tags based on the similarities in features of the posts or to find circles of tags that could allow for accurate detection of possible associated tags given a tag with high probability. A strength of [2] is that it focuses not only on the cold-start problem, but also on circle maintenance. Both situations are realistic for clustering tags, users, or posts on software information sites. We definitely want to be able to form clusters based solely on the features of tags, posts, or users. We also want to be able to perform maintenance on our circles since new posts or new insights on posts could lead to slight changes in the network.

Another weakness with [1] is that it only addresses tag recommendations during question creation time. That is, tag recommendations need to be made with just the text from the initial post. Discussion generated over time by the post is not factored into the features for the tag recommendation system. However, tagging a post is not an action limited to post creation time. Users may add additional tags to posts later on based on the discussion. In the context of a social network, this is similar to the notion of users joining new communities over time: posts can acquire new tags over time. The work in [3] provides a starting point for studying the evolution of communities in social networks and user memberships in these communities. However, the rigorous analysis in [3] focuses on what features drive users to join communities; the statistical analysis on how communities evolve over time is mostly limited to just studying growth in size. We believe this paper is still important as it presents the idea that, in social networks, movement of ideas seems to precede movement of community memberships. We hope that we can find other literature that studies the flow of ideas in relation to network communities, as it could provide a way for us to analyze how tags on posts will change over time, or even suggest new tags that have not been used before.

4 Project Proposal

4.1 Data and Problem

Our goal is to improve upon existing tag recommendation systems for community-based information web sites like StackOverflow by incorporating features from the network structure of the website. Our primary data source will be data dump of the StackOverflow website, which contains the full texts of posts, including user names, tags, and followup comments and discussions. Given that we are constrained to the time frame of just this quarter, it is likely that we will only be able to focus on the StackOverflow data. If we have additional time, though, we will branch out to datasets from other similar websites like in [1] and [5].

4.2 Network Model

We immediately see a few possible network structures on the StackOverflow website. We can model the posts as nodes of the network and attach edge attributes between posts based on a variety of other components of the data. For example, posts made by and commented on by the same users could have attributes on the edges between them denoting this relationship. Posts with shared tags can also be linked together via edge attributes to further add to the network structure. A graph can also be created with the users as nodes and edges between users denoting shared activity on posts or similarity of tags used. Contrary to the graph created in [5], tags will not serve as the nodes in the network model. Instead we treat tags more as a concept of "communities," where the members of the community will be posts or users.

It will be interesting to see how these different networks fit into the general statistical conclusions from the Network Community Profile (NCP) plots in [4]. With the idea of conductance defined in [4], we can observe which users or posts form tightly knit communities in StackOverflow, and based on tags used by these users or used in these posts, we can infer tag representatives for the communities in this website. In particular, this can be useful for trying to discover specific tags for a post. It is probably very easy to tag posts with broad categories like `java` or `python`, but it is much harder to recognize more specific topics that may also apply. Based on the work in [4], the tightest communities in a social network are actually quite small (~ 100 members), and we hope that we can detect similar community structure on the StackOverflow network and use it to cluster posts into very specific categories.

We can immediately see some exciting possibilities and applications of our tag recommendation system that uses an underlying network structure in which tags do not serve as nodes. In all of our ideas for network models in this section, the network will continuously and significantly change over time as users contribute new material in the form of answer responses and comments. Thus, it is likely that the recommended tags for a particular question may change as the question is linked to new users and many other questions. For example, consider a beginner user who asks a question that is not well-written or is incomplete in terms of content. The initial recommended tags, then, may also not be the most descriptive. Only when experienced users respond – and thus change the network structure for the better – can appropriate tags be suggested. The key point here is that, by analyzing the network, our system can dynamically account for this possibility.

4.3 Evaluation

In [1], Xia et al. introduce the concept of the *recall@k* metric for measuring the success of a tag recommendation model, where k is a tunable parameter that determines how many tags the model recommends for each object. Intuitively, over n objects, the *recall@k* metric measures the average success rate in predicting correct tags for each object, where a "correct" tag is simply a tag that has been used to label that particular object by an actual user. Let R_i be the set of tags recommended for object i (so, $|R_i| = k$), and let T_i be the actual set of tags used to label object i. Then, the formula for *recall@k* is:

$$\textit{recall@k} = \frac{1}{n}\sum_{i=1}^{n}\frac{|R_i \cap T_i|}{|T_i|}.$$

Of course, the most straightforward way to evaluate the success of our tag recommendation system is to compare its *recall@k* scores directly to the scores of systems that have been previously proposed (e.g. [1] and [5]). It will be interesting to see whether incorporating insight about the underlying network structure can improve *recall@k* scores and also if the value of k itself plays a difference. For example, we may find that a recommendation system that uses network structure performs better when k is larger. Since the *recall@k* score is very flexible, we can also apply our recommendation model to different datasets beyond StackOverflow to see if certain networks lend themselves more easily to predicting tags.

By the end of the project, we expect to have designed and implemented a tag recommendation method that utilizes the underlying network structure of a community-based information web site. The dataset from StackOverflow will be used as the initial test case, and the goal is to achieve higher *recall@k* scores than those garnered by previous methods discussed in [1] and [5].

References

[1] X. Xia, D. Lo, X. Wang, B. Zhou. Tag Recommendation in Software Information Sites. MSR, 2013.

[2] J. McAuley, J. Leskovec. Discovering Social Circles In Ego Networks. ACM TKDD, 2014.

[3] L. Backstrom, D. Huttenlocher, J. Kleinberg, X. Lan. Group Formation in Large Social Networks: Membership, Growth, and Evolution. KDD, 2006.

[4] J. Leskovec, K. Lang, A. Dasgupta, M. Mahoney. Statistical Properties of Community Structure in Large Social and Information Networks. WWW, 2008.

[5] S. Wang, D. Lo, B. Vasilescu, A. Serebrenik. EnTagRec: An Enhanced Tag Recommendation System for Software Information Sites. ICSME, 2014.