ABSTRACT

As an increasing number of people get access to the Internet, crowdsourcing is becoming a more viable option for content creation, filtering and distribution. For tasks that rely on cognitive skills and natural language processing, crowdsourcing can be a cheaper, more accurate and easier to implement alternative to automated techniques. The content on the internet isn’t inherently categorized. Tagging has emerged as a popular technique to organize content and enable searching.

In this poster we propose a content sharing platform which uses crowdsourcing and collective intelligence to organize and recommend content to users.

We present a trigger-based system to assign weights to parameters which represent relevance between objects. We also present a modified version of the ECLAT algorithm for association rule mining, better tailored to our scenario, and finally a crowdsourced feedback process in order to validate association rules.

Categories and Subject Descriptors
C 2.0 [Computer-Communication Networks]: Crowdsourcing - http://www.acm.org/about/class/ccs98-html

General Terms
Algorithms, Design, Theory

Keywords
Crowdsourcing, Collective Intelligence, Data Mining, Annotations, Social Media.

1. INTRODUCTION

Our implementation contains two kinds of entities – users and content. Both have a set of weighted tags attached to them, which represent their interests and relevance respectively. Our implementation can be broadly divided into four different stages:

1) Content Creation:
   a. All users can create content and add an initial set of relevant tags.

2) Content Moderation:
   a. Moderators are allowed to edit content. Their edits become permanent after the creator’s approval.
   b. Users are given moderator privileges for specific tags based on their interests and feedback on their previous content.

2. ASSIGNING TAG WEIGHT BASED ON FEEDBACK

All content and users are associated with sets of tags which represent their relevance to topics and interests, respectively. The tag associations are weighted based on how strong the relevance is. These tag weights are used for making recommendations to users and also for searching.

When the content is added initially, the creator assigns some tags. Users that view the content can rate and add tags to the content. Creator tags start off with more weight. Adjustments to the tags are made as follows:

1) As more users enter the same tag for some content, the weight of that tag for the content increases.
2) When a user rates some content, the user’s tags get added to the content and the content’s tags get added to

Figure 1. Feedback between Content Creator, Moderators and Users
the user. Tags common to both increase in weight for both.

3) Tags added automatically in the above way are invisible and are not displayed to the users – they’re only used internally by the system. A moderator or the creator can approve these tags which makes them visible.

4) If a tag isn’t relevant to some content, users can downvote the tag, which decreases the tag’s weight for that content.

5) When a user searches for content, the search keywords are matched with the list of existing tags in the system. If there is a match, the keywords are added to the user (or increase in weight for the user if they’re already associated with them).

3. ASSOCIATION RULE MINING

3.1 ECLAT Algorithm

The ECLAT algorithm is used to analyze a set of “transactions” (content) consisting of “items” (tags) to generate associations of the form A, B, C ⇒ D, E.

The rule suggests that given the fact that {A, B, C} occur in a transaction together, there is a high probability of {D, E} also occurring.

We use the following parameter definitions in the algorithm:

1) Support = Occurrences (A, B, C, D, E)/Number of transactions
2) Confidence = Occurrences (A, B, C, D, E)/Occurrences (A, B, C)
3) Support threshold: Threshold for support below which we discard rules.
4) Confidence threshold: Threshold for confidence below which we discard rules.
5) k-frequent item set(Pk): A set of k items (tags) for which support ≥ threshold

The Algorithm is as follows:

1) Systematically compute Pk for k = 1, 2, 3… until Pk = 0.
2) Add each Pk to S, list of frequent item sets.
3) Generate all possible rules for each element in S.
4) Discard rules with confidence less than the threshold.

3.2 Modified ECLAT

In our system, we use a variation of the ECLAT algorithm which takes into account the fact that the topic tags naturally form a hierarchy. We make the following assumption:

“If a piece of content is about a particular category, it will have more weight in the tag corresponding to it than the tags corresponding to categories that are semantically its children.”

For example, an article talking about Musical instruments might mention flute, which might also be a tag, but the crowd-sourced tags will have more weight in the tag “musical instruments” (which is what the article is mainly about) than the tag “flute”. This allows us to determine the relative position of two items in the hierarchy tree. Higher the weight of a tag, higher up it is in the hierarchy tree. We also present the following argument:

“We can always move up the hierarchy tree while adding tags, but not down.”

When we move down the tree, we specialize, and when we move up, we generalize. So, for example:

Trumpet ⇒ Wind Instrument is correct.

Wind Instrument ⇒ Trumpet is possibly incorrect.

So we discard rules where the left-hand side has a greater average weight than the right-hand side.

We also only consider rules with a single element on the right-hand side. This reduces complexity from 2^k to k. Because rules are matched multiple times with the content, we still get all the possible new tags. We do lose the association between the right-hand side tags, which we do not require.

3.3 Association Rule Validation

We validate our mined rules in various ways.

1) Validation using Twitter:

Given a rule A, B, C ⇒ D, E Twitter is searched for A, B, C. We then look for tweets in the results that also contain D, E. We use the percentage of those tweets as an indication of confidence of the rule.

2) Creator Validation: We automatically determine new probable tags for an article based on its already existing set of tags. We then suggest these new tags to the content creator. We record creator’s response (positive or negative) for each of the suggested tags. For a rule to get validated, it must get more percentage of acceptances than a threshold.

3) User Validation: We also use the learnt rules to introduce new areas of interest to a user’s list of tags. But before doing so, we suggest some content based on the new tag to the user in their recommended article list. We record whether they respond positively (by viewing the new content) or negatively (by ignoring it). This information is used to determine whether the rule has been learnt correctly.

4. CONCLUSIONS

With the growth of the internet, it becomes increasingly important to make sense of the vast amount of data available on the web to allow the users to efficiently find and share information (semantic web). Tagging is a popular method to help a computer make sense of data. But the fact remains that currently humans are better than computers at understanding the semantics of data. Combining human strengths (cognitive tasks) with computer strengths (computation) through crowdsourcing has a lot of potential. Such a system uses data generated by humans to make decisions after further processing. It also needs proper feedback mechanisms to prevent it from accumulating errors. The feedback can be crowdsourced, both directly from the users of the service and from publicly available data like from Twitter.

5. REFERENCES