Finding similar items by leveraging social tag clouds

Chu-Cheng Hsieh
University of California, Los Angeles
4732 Bolter Hall
Los Angeles, CA 90095, USA
(+1)310-773-0733
chucheng@ucla.edu

Junghoo Cho
University of California, Los Angeles
4732 Bolter Hall
Los Angeles, CA 90095, USA
(+1)310-773-0733
cho@cs.ucla.edu

ABSTRACT
Recently social collaboration projects such as Wikipedia and Flickr have been gaining popularity, and more and more social tag information is being accumulated. In this study, we demonstrate how to effectively use social tags created by humans to find similar items. We create a query-by-example interface for finding similar items through offering examples as a query. Our work aims to measure the similarity between a query, expressed as a group of items, and another item through utilizing the tag information. We show that using human-generated tags to find similar items has at least two major challenges: *popularity bias* and *the missing tag effect*. We propose several approaches to overcome the challenges. We build a prototype website allowing users to search over all entries in Wikipedia based on tag information, and then collect 600 valid questionnaires from 69 students to create a benchmark for evaluating our algorithms based on user satisfaction. Our results show that the presented techniques are promising and surpass the leading commercial product, Google Sets, in terms of user satisfaction.

Categories and Subject Descriptors
H.3.3 [**INFORMATION STORAGE AND RETRIEVAL**]: Information Search and Retrieval – *clustering, information filtering, retrieval models, selection process.*

General Terms
Algorithms, Experimentation

Keywords
World Wide Web (WWW), Social tags, Information retrieval, Entity resolution, Folksonomy

1. INTRODUCTION
The dominant interface of search engines today requires users to pinpoint their information needs with a few keywords. However, users sometimes find it difficult to identify the keywords that best describe their needs. For example, a user who plans to apply for a graduate school in California may issue the query "outstanding universities in California". Many outstanding schools, such as Stanford University, are missing in the top results of all major search engines, because the keywords "outstanding" and "California" are not presented in the web pages of those schools.

As a potential solution to this problem, we study methodologies for providing a "query-by-example" interface. In this interface, users provide a few representative examples of the ultimate information they seek. The system then returns search results most similar to the examples provided; for instance, to find outstanding graduate schools, a user may issue a query like "Caltech, UC Berkeley" and expects that the system will return similar outstanding schools in California such as UCLA and Stanford University.

The major challenge in building such a system is to identify similar items based on the user-provided set of examples. In this study, we leverage the tag clouds that are collaboratively created by web users in defining and measuring the similarity between multiple items. To verify the effectiveness of our solutions, we conduct experiments on one of the largest social collaboration projects, Wikipedia. In the Wikipedia dataset, we consider a wiki page or entry as an entity and a category label of a page as a tag. We aim to identify and rank entities that are similar to the user-provided examples based on tag information.

As other researchers [9] have noted, the uncontrolled nature of user-generated metadata, such as free-form tagging in Wikipedia, often causes problems of imprecision and ambiguity when these tags are used as a foundation of designing algorithms. In our study, we identify and deal with two challenges associated with free-form uncontrolled tag clouds: *popularity bias* and *the missing tag effect* (Section 3.2).

We propose several approaches to overcome the challenges and subsequently build a prototype website allowing users to issue a query by examples. Our results show that our techniques are able to return a sizable number of high-quality similar items even when the user provides only a few examples in the query. The proposed approaches are evaluated against a benchmark dataset that is built based on 600 valid questionnaire responses from 69 students. In terms of user satisfaction, the questionnaire responses show that our techniques outperform Google Sets.

In summary, we highlight our contributions as follows:

- We investigate how to extract a set of similar items through analyzing noisy social tags created by human beings, and show that the tag information is effective in identifying relevant similar items.
- We identify and solve two challenges in tag-based search frameworks: popularity bias and the missing tag effect.
- We propose and compare several models based on tag information. We build algorithms on top of these models, and study their advantages and disadvantages.

SAC'12, March 25-29, 2012, Riva del Garda, Italy.

- We perform an extensive evaluation based on user surveys and show that in terms of user satisfaction, our tag-based approaches outperform Google Sets in most testing cases.

2. PROBLEM OVERVIEW

The essential problem can be phrased as the following: users want to retrieve relevant items sharing some characteristics, and their queries are composed of representative examples with desired characteristics. We consider the "*query-by-example*" task as a process of finding similar items in a dataset, where the query is composed of a number of items from the same dataset.

Assuming a user issues a query X_Q: $\{x_{q1}, x_{q2}, \dots\}$, the input to the framework is the query itself, where X_Q should be composed of entities, like {Caltech, UC Berkeley}. Our goal is to create a function R to measure the similarity between an entity x_i in the dataset and the query X_Q, where a higher score measured by $R(x_i, X_Q)$ implies a higher similarity between the entity x_i and the query X_Q.

2.1 Tag generation

We start the discussion of our tag-entity model with an artificial example, as shown in Fig. 1, to explain how tags are generated. In this example, we say a user reads the content of a Wikipedia page about Washington D.C.; then, the user labels the page with the tags *City*, *Capital*, *North America*, etc. Note that a tag does not have to be the same word used in the content; for example, the concept of a metropolis is matched by the tag *City*. In most social collaboration projects, a user can select any phrase as a tag, i.e. free-form tagging. The phrase may be an existing tag, a modified phrasing of an existing tag, or even a newly created tag.

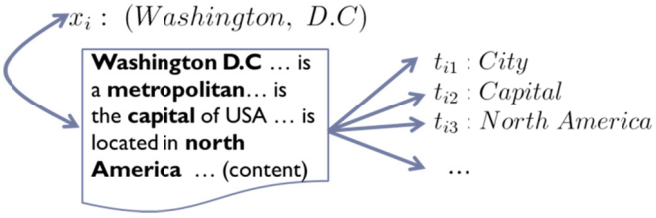

Fig. 1 The data model

Although we illustrate our tag-entity model by considering "wiki pages" as entities and their titles as identifiers, the content of entities are not limited to plain texts. Our proposed solutions utilize only tag information, ignoring the content of entity. This simplification allows our techniques to be broadly applicable to non-textual dataset as well.

We use X_U to represent all entities in a dataset, that is, the universe of entities. We use $T_i = \{t_{i1}, t_{i2}, \dots\}$ to represent the tags associated with x_i. The universe of tags is denoted by T_U, referring to all tags in a dataset.

We illustrate our notations with the following example:

Example 1 Consider a dataset that contains four entities:

x_1: $Beijing$ →T_1: {$City, Capital, Asia, Summer\ Olympic$}
x_2: $Washington\ D.C.$ →T_2: {$City, Capital, North\ America$}
x_3: $London$ →T_3: {$City, Capital, Europe, Summer\ Olympic$}
x_4: $Los\ Angeles$ → T_4: {$City, North\ America$}

In this example, the dataset contains a total of six tags:

t_1: $City$, t_2: $Capital$, t_3: $Asia$, t_4: $Summer\ Olympic$,
t_5: $North\ America$, t_6: $Europe$ □

We define the function $\mathcal{E}(t_n)$ as an operation for acquiring all entities associated with the tag t_n. For example, $\mathcal{E}(t_2: Capital) = \{x_1: Beijin,\ x_2: Washington\ D.C.,\ x_3: London\}$. In addition, we define $\mathcal{E}(t_m, t_n, \dots)$ as $\mathcal{E}(t_m) \cup \mathcal{E}(t_n) \cup \dots$ Namely, in Example 1, $\mathcal{E}(t_3: Asia,\ t_6: Europe) = \mathcal{E}(t_3) \cup \mathcal{E}(t_6) = \{x_1: Beijin,\ x_3: London\}$.

Without referring to the content of an entity, we limit our similarity measurement to tag information. One might argue that tag-entity relations are unreliable and sometimes a reasonable relation is missing from an entity. We will discuss these concerns regarding the imperfect nature later in Section 3.2.

3. THE INTERSECTION-DRIVEN APPROACH

The core challenge in providing a query-by-example service is to figure out what types of entities a user is looking for based on the input entities provided by the user. To convey how we approach this problem, we start with an artificial scenario.

In Example 1, when the user-provided input is the query {$x_1: Beijing,\ x_2: Washington\ D.C.$}, what will be a reasonable interpretation of the user's intention? Both entities are associated with the tags $t_1: City$ and $t_2: Capital$ as we can see from $T_1 \cap T_2 = \{t_1: City, t_2: Capital\}$. That is, both entities are cities and capitals. Given this result, we may have two interpretations: the user is looking for cities that are also capitals, or the user is simply looking for cities, but the input examples happen to be capitals as well. Since the second interpretation is possible, not only $t_1: City$ but also $t_2: Capital$ should be weighed when we identify other similar entities.

In the intersection-driven approach, if a tag is associated with only a subset of input examples, we claim that the user is unlikely to only look for entities associated with such a tag. For instance, although the input entity Beijing is also associated with the tag $t_3: Asia$, it is unlikely that the user is only looking for cities in Asia because this tag is not associated with $x_2: Washington\ D.C.$

3.1 Similarity Measurement

Here, we define the degree of similarity between an entity x_i and the query X_Q as follows:

$$R(x_i, X_Q) = |T_i \cap T_Q^{\cap}| \quad (1)$$

The symbol $T_Q^{\cap}$ stands for tags associated with all entities in a query, i.e. $T_Q^{\cap} = T_1 \cap T_2 \cap \dots \cap T_n,\ \forall x_i \in X_Q$. In Example 1, if a query consists of entities {$x_1: Beijing,\ x_2: Washington\ D.C.$}, the set $T_Q^{\cap} = \{t_1: City,\ t_2: Capital\}$.

According to our definition in Equation (1), the more tags in $T_Q^{\cap}$ an entity is associated with, the more similar (to the query X_Q) the entity is. For instance, if an entity x_j is a city but not a capital, such as the entity $x_4: Los\ Angeles$, it is associated with $t_1: City$ but not with $t_2: Capital$. According to the similarity definition in Equation (1), the ranking score is $R(x_4: Los\ Angeles, X_Q) = 1$, meaning that the entity $x_4: Los\ Angeles$ has only one tag in common with $T_Q^{\cap}$; Likewise, the entity $x_3: London$ has the ranking score of $R(x_3: London, X_Q) = |T_3 \cap T_Q^{\cap}| = 2$, meaning that the entity $x_3: London$ has two tags in common with $T_Q^{\cap}$.

In the above example, our approach ranks $x_3: London$ higher than the entity $x_4: Los\ Angeles$. This result seems intuitive because no matter whether the user's intent is to find cities or to find cities that are also capitals, the entity $x_3: London$ is always an appropriate answer. The entity $x_4: Los\ Angeles$ is inferior to $x_3: London$ because $x_4: Los\ Angeles$ is an inappropriate answer if the user's intent is to find cities that are also capitals. Since we cannot deny such a possibility, ranking $x_3: London$ higher than $x_4: Los\ Angeles$ is reasonable.

3.2 Challenges

We create Example 2 based on our observations on the Wikipedia dataset. The example illustrates and highlights the challenges we expect to encounter in a real dataset. In Example 2, an underlined notation signifies that a tag is very popular. Also, we strike-through a tag, representing that a tag-entity relation should exist but does not appear in a real dataset.

Example 2 Consider a dataset that contains six entities.

$x_1: Beijing \rightarrow T_1$: {*City, Capital, Asia, Summer Olympic, China, ~~Object~~* }
$x_2: Washington\ D.C. \rightarrow T_2$: {*~~City~~, Capital, North America, Object* }
$x_3: London \rightarrow T_3$: {*City, ~~Capital~~, Europe, Summer Olympic, Object* }
$x_4: Los\ Angeles \rightarrow T_4$: {*City, North America, Object* }
$x_5: Michael\ Phelps \rightarrow T_5$: {*Summer Olympic, Object* }
$x_6: Lyon \rightarrow T_6$: {City, Europe, Object }

In this example, the dataset contains a total of eight tags:
$t_1: City, t_2: Capital, t_3: Asia, t_4: Summer\ Olympic,$
$t_5: North\ America, t_6: Europe, t_7: China, t_8: Object$ □

3.2.1 Missing Tag Effect

Not only in Example 2, but also in practice a tag-entity relation can be missing. There are several possible reasons of why this may happen. In any social collaboration project, a newly created entity might not be well tagged until its editors finish revising all the content of the entity. At the same time, the community may not be aware of a newly created tag or may decide not to use the newly created tag for other entities.

Missing tag-entity relations could cause the system to misinterpret user intent. For example, suppose that a user's query $X_Q = \{x_2: Washington\ D.C., x_3: London\}$ has been issued against Example 2 dataset; since the tag $t_1: City$ is missing in $x_2: Washington\ D.C.$, and the tag $t_2: Capital$ is missing in $x_3: London$, the intersection–driven approach interprets the user intention as finding entities associated with the tag $\{t_8: Object\}$. Given these two input entities, intuitively, we feel that such an interpretation "Object" is problematic because the interpretation is too general. The intersection–driven approach considers the entity $x_1: Beijing$ as an irrelevant one, and suggests that $x_4: Los\ Angeles$, $x_5: Michael\ Phelps$, and $x_6: Lyon$ are equally similar to the query X_Q.

The impact of the missing tag effect is more pronounced as more entities are included in a query. Suppose that a user is looking for a set of entities associated with a tag t_k; ideally, the tag t_k is expected to be associated with all the entities in the query. If we use the notation α to represent the probability of missing the tag t_k, the probability of t_k being not associated with all input examples becomes

$$P(missing\ \ t_k\ in\ T_Q^{\cap}) = 1 - (1-\alpha)^{|X_Q|} \qquad (2)$$

, where $|X_Q|$ is the number of entities in the query. If the value of α is 20% and the number of input examples is 3, the probability of missing the tag t_k in $T_Q^{\cap}$ is close to a half (48.8%). When the number of input examples increases to 10, the chance of missing the desired tag increases to 89.26%.

3.2.2 Partial Weighting Generalization

To generalize the intersection-driven approach for addressing the missing tag effect, one solution is to assign scores in real number, instead of either zero or one, to tags that are associated with only some entities in a query. In the previous example ($X_Q = \{x_2: Washington\ D.C., x_3: London\}$), we now assign 0.5 to these tags: $t_1: City$, $t_2: Capital$, $t_4: Summer\ Olympic$, $t_5: North\ America$, and $t_6: Europe$, because we have two entities in the query and each tag associates with only one entity. These tags, although not in $T_Q^{\cap}(\{t_8: Object\})$, are now assigned partial weights in real numbers, and thus contribute to the similarity function $R(x_i, X_Q)$. As a result, $R(x_4: Los\ Angeles, X_Q) = 2$ because not only the tag $t_8: Object$ contributes 1 point, but also $t_1: City$ and $t_5: North\ America$ contribute 0.5 points separately. Similarly, $R(x_5: Michael\ Phelps, X_Q) = 1.5$ and $R(x_6: Lyon, X_Q) = 2$. The system returns a better ranking result {1st position: "$x_4: Los\ Angeles$ and $x_6: Lyon$", 2nd position: "$x_1: Beijing$" and "$x_5: Michael\ Phelps$" }.

The strategy weights tags that appear in $T_Q^{\cap}$ the most heavily, and thus ensures that we follow the same intuition: the more tags in $T_Q^{\cap}$ an entity is associated with, the more similar (to the query X_Q) the entity is. Moreover, in case the intent cannot be captured in $T_Q^{\cap}$, the strategy helps the system return some related results as long as some tags are associated with one or more entities in the query.

Such a generalization brings more entities to results. For example, in Example 2, assigning partial weight to the tag $t_4: Summer\ Olympic$ brings $x_5: Michael\ Phelps$ to the result. Nevertheless, introducing this suspicious result may not be a good idea. Therefore, if the system already returns satisfied results, we tend to not adopt generalization unless the user asks for more results.

3.2.3 Popularity Bias

Both results in the previous research [5] and results in our experiments show that the number of tags associated with an entity follows a power law distribution. That is, only a few entities are associated with a large number of tags, and most entities are associated with a small number of tags.

We define the popularity of an entity x_i based on the number of tags associated with x_i, denoted by $|T_i|$. Then, an entity x_i is more popular than another entity x_j if $|T_i| > |T_j|$. Similarly, we say that a tag t_k is more popular than another tag t_k' if $|\mathcal{E}(t_k)| > |\mathcal{E}(t_k')|$, where $\mathcal{E}(t_k)$ represents the set of all entities associated with t_k.

We define the *popularity entity-bias* as follows: if we measure the similarity between an entity and a query based on tag information in the query, we tend to favor entities that are more popular. For example, a query X_Q consists of $\{x_1: Beijing, x_6: Lyon\}$. To process the query, we firstly identify all cities. Then, to further rank all cities, we assign full weight (1.0) to the tag *City, and*

partial weight (0.5) to these tags: *Capital*, *Asia*, *Summer Olympic*, *China*, *Europe* and *Object*. Since most of the tags originate from $x_1{:}\,Beijing$, entities similar to $x_1{:}\,Beijing$ are more likely to be returned in the result and ranked in higher positions.

The popularity bias happens in tags as well, and we name this kind of bias as *popularity tag-bias*. We argue that although a popular tag like *Object* in Example 2 is associated with some input example(s), this popular tag is probably not the concept the user intends to search for. That is, the tag $t_8{:}\,Object$ exists in $T_Q^{\cap}$ probably only because it is popular, i.e. $|\mathcal{E}(t_8{:}\,Object)|$ is a large number. In Example 2, if we randomly select two entities to create a query, $t_8{:}\,Object$ is the most likely tag shown in $T_Q^{\cap}$.

4. BALANCED VOTING MODEL

We now further refine the intersection-driven approach to include the following properties: (1) a popular entity in a query should not unfairly influence the results such that the results are similar only to the popular entity, but not to others, and (2) even a few tag-entity relations are missing in input examples, the system should be able to identify relevant entities based on tags associated with a subset of input examples.

To achieve the above desired properties, in this model, we define the similarity between an entity x_i and the query X_Q as follows:

$$R(t_n, X_Q) = \begin{cases} \sum_{x_i \in X_Q} \frac{1}{|T_i|} & \text{, if } t_n \in T_i \\ 0 \quad , & otherwise \end{cases} \tag{3}$$

$$R(x_i, X_Q) = \sum_{t_n \in T_i \cap T_Q^{\cup}} R(t_n, X_Q) \tag{4}$$

Equation (3) can mitigate both the missing tag effect and the popularity entity-bias. In a nutshell, we consider every entity in the query as having a total score of one in Equation (3). Then, each tag associated with the entity is assigned an equal portion of the entity's score. The non-zero assignment alleviates the missing tag effect in that we now value significance of tags that are missing in some input examples. Furthermore, the assigned score of a tag is inversely proportional to $|T_i|$, i.e., the number of tags associated with x_i. As a result, a tag originating from a popular entity will get a lower score than a tag originating from a less popular entity, alleviating the popularity entity bias.

In Equation (4), we introduce the symbol $T_Q^{\cup}$ to represent tags associated with one or more entities in a query X_Q, where the symbol ∪ stands for the notion "union". We calculate the relevance score for every entity in the dataset with Equation (4) by summing up the relevance scores of all tags associated with the entity. Here, $R(t_n, X_Q)$ represents the relevance score of the tag t_n to the query X_Q. Note that in this equation we assign a non-zero score to each tag in $T_Q^{\cup}$; therefore, a tag associated with only a subset of the input entities will still get a non-zero score. The following example shows how the scores are computed under this definition.

In Example 2, each tag associated with the entity $x_1{:}\,Beijing$ gains 0.2 points because five tags are associated with it. Similarly, each tag associated with the entity $x_6{:}\,Lyon$ gets 0.33 points. Namely, if a query X_Q consists of $\{x_1{:}\,Beijing, x_6{:}\,Lyon\}$, the $R(t_1{:}\,City, X_Q) = 0.2 + 0.33 = 0.53$. Although $x_1{:}\,Beijing$ is associated with more tags than $x_6{:}\,Lyon$, each tag in $x_6{:}\,Lyon$ contributes a higher ranking score than a tag in $x_1{:}\,Beijing$.

We calculate the relevance score for every entity in the dataset with Equation (4). For instance, the entity $x_3{:}\,London$ gets a relevance score of 1.39 points by summing up the score of the tags associated with $x_3{:}\,London$, including $t_1{:}\,City$ (0.53), $t_4{:}\,Summer\ Olympic$ (0.2), $t_6{:}\,Europe$ (0.33), and $t_8{:}\,Object$ (0.33).

Tags belonging to the set $T_Q^{\cap}$, such as $t_1{:}\,City$, still contribute the highest ranking scores. Therefore, the balanced voting approach clings to our belief that tags shared by all entities in a query likely capture a user's intention; meanwhile, it compensates biases caused by a popular entity in a query. Compared to our intersection-driven approach, each tag in a popular entity like $x_1{:}\,Beijing$ contributes less influence in terms of ranking scores now. This difference makes our balanced voting approach less sensitive to the *popularity entity-bias*.

5. ONE-CLASS PROBABILISTIC MODEL

So far, we view the similarity measurement as determining a good weighting scheme for tags associated with input examples. Alternatively, we can consider the problem as computing the probability that an entity is the target a user is looking for. Given some entities as possible results, we can then rank the entities based on the calculated probabilities.

We start by thinking about how people create a query for finding similar items. We think that a user firstly has a desired property in mind, and then the user tries to recall some entities with the desired property based on his/her knowledge. If the intent can be captured by a tag (or tags) in a dataset, entities associated with the tag(s) in the dataset are considered as similar entities.

We propose the one-class probabilistic model based on the following assumptions. At first, we assume that a user's intent corresponds to one tag t_k in a dataset. Since the intent is unpredictable, we claim that the desired tag t_k is randomly selected from all tags. Then, once the desired tag is selected, the user randomly selects $|X_Q|$ entities from $\mathcal{E}(t_k)$ to synthesize the query, where $\mathcal{E}(t_k)$ are all entities that are associated with t_k.

5.1 Measuring Similarity Using Probability

In order to better understand how our one-class probabilistic model works, in this section we discuss the model under the premise that there is no missing tags in the dataset. We will reconsider the problem of missing tags later in the next section.

When a query X_Q is given, based on our one-class probabilistic model, the probability that x_i is what a user is looking for can be computed as

$$P(x_i|X_Q) \triangleq \sum_{t_k} P(x_i|t_k) * P(t_k|X_Q) \tag{5}$$

In Equation (5), the $P(t_k|X_Q)$ stands for the probability that a tag t_k is the desired tag when the system is given the user query X_Q; $P(x_i|t_k)$ stands for the probability that the system should return an entity x_i if the desired tag is t_k. Here, we assume that all tags

are independent from each other, so we can sum up all the probability scores when more than one tag is considered.

Using Bayes' theorem, we can find an equivalent form to Equation (5), as shown in Proposition 1.

Proposition 1 *Ranking entities based on* $P(x_i|X_Q)$ *is equivalent to ranking entities through the formula* $\sum_{t_k \in T_i} P(X_Q|t_k)$. *That is,*

$$P(x_i|X_Q) \propto \sum_{t_k \in T_i} P(X_Q|t_k) \quad (6)$$

(Proof) Because of space constraints, we defer all derivations and proofs in this section to the appendix*.

Given a query X_Q, $|X_Q|$ is the number of input examples. In this model, if the tag t_k is the desired tag, representing a user's intention, our premise says that the query is created by randomly selecting $|X_Q|$ entities from the set $\mathcal{E}(t_k)$. $|\mathcal{E}(t_k)|$ represents the number of entities associated with a tag t_k, so we have $\binom{|\mathcal{E}(t_k)|}{|X_Q|}$ distinct choices to create a query because the user can randomly select $|X_Q|$ entities from $\mathcal{E}(t_k)$, where $\binom{|\mathcal{E}(t_k)|}{|X_Q|}$ stands for the number of combinations in a set with $|X_Q|$ distinct elements. Since the query X_Q is one of the $\binom{|\mathcal{E}(t_k)|}{|X_Q|}$ outcomes, the probability of having X_Q being the query and t_k being the desired tag is

$$P(t_k|X_Q) = \frac{1}{\binom{|\mathcal{E}(t_k)|}{|X_Q|}} \quad (7)$$

In this model, entities in the query are selected from entities associated with the desired tag t_k, representing the desired property in the user's mind. If all tag-entity relations in a dataset are established, the desired tag must be one of the tags that are shared by all entities in the query, i.e., $t_k \in T_Q^{\cap}$. In addition, we know an entity x_i can be a similar entity to the query only if the desired tag t_k is associated with the entity, i.e., $t_k \in T_i$. Thus, for any entity x_i, only tags in $(T_i \cap T_Q^{\cap})$ are considered.

If an entity x_i is associated with two or more tags in $T_Q^{\cap}$, we sum up all probability values to get the probability of the entity x_i being a similar entity. Then, we rank every entity in a dataset based on this probability.

We summarize the above discussions as follows:

Proposition 2 *If every tag is correctly associated with all entities to which it is related (no missing tag), we show that*

$$\sum_{t_k \in T_i} P(t_k|X_Q) = \sum_{t_k \in (T_i \cap T_Q^{\cap})} \frac{1}{\binom{|\mathcal{E}(t_k)|}{|X_Q|}} \quad (8)$$

(Proof) See the appendix.

* The appendix can be downloaded from http://oak.cs.ucla.edu/~chucheng/publication/SAC_2012_Appendix_Chucheng.pdf

Equation (8) has an advantage of alleviating the popularity tag-bias because the system will assign a low value to $P(X_Q|t_k)$ for a popular tag. The equation conveys the notion: when a rarely seen tag ($|\mathcal{E}(t_k)|$: a small number) is shared by all entities of the query, the probability that the rarely seen tag is what the user desires is high because it is unlikely that input examples are associated with the tag by chance. As a result, Equation (8) places more value on it than on a popular common tag ($|\mathcal{E}(t_k)|$: a large number).

In Example 2, suppose that we see a tag $t_8: Object$ in $T_Q^{\cap}$; we could argue that $t_8: Object$ exists because a user is looking for objects, or because $t_8: Object$ is associated with almost every entity in the dataset. If $|\mathcal{E}(t_8: Object)|$ is a large number, the chance of the latter is high, and thus we could reason the tag $t_8: Object$ might not be the desired tag. In other words, the model alleviates *popularity tag-bias* through assigning a low value $P(X_Q|t_k)$ to a popular tag.

5.2 Refinement of the Approach

In this section, we deal with the missing tag effect. We start with introducing some new notations for extending our one-class probabilistic model.

The function $\mathcal{E}^C(t_k)$ returns entities that are relevant to the tag t_k but the tag-entity relation is missing, and $|\mathcal{E}^C(t_k)|$ is the number of entities missing the tag t_k. For instance, in Example 2, $\mathcal{E}^C(t_8: Object)$=$\{x_1: Beijing\}$.

The symbol m_k denotes the number of entities missing a tag t_k in a query. For example, in Example 2, a query $\{x_1: Beijing, x_2: Washington\ D.C.\ , x_3: London\}$ has values $|X_Q| = 3$ because the query contains three input examples, and for the tag $t_1: City$, $m_1 = 1$ because only one entity ($x_2: Washington\ D.C$) is missing the tag $t_1: City$.

The symbol u stands for the number of all entities in our dataset; the symbol $T_Q^{\cup}$ represent all tags associated with "at least one" input example. Then, we generalize Proposition 2 as follows:

Proposition 3 *When considering the missing tags effect , we show that* $\sum_{t_k \in T_i} P(t_k|X_Q)$ *in Proposition 1 can be approximated by the following formula:*

$$\sum_{t_k \in T_i} P(t_k|X_Q) = \sum_{t_k \in (T_i \cap T_Q^{\cup})} \left[\left(\frac{|\mathcal{E}^C(t_k)|}{u} \right)^{m_k} * \frac{1}{\binom{|\mathcal{E}(t_k)| + |\mathcal{E}^C(t_k)|}{|X_Q|}} \right] \quad (9)$$

,where we claim that the dataset has the following properties: (1) $u \gg |\mathcal{E}(t_k)|$ *and* $u \gg |\mathcal{E}^C(t_k)|$ *and* (2) $(|\mathcal{E}(t_k)| + |\mathcal{E}^C(t_k)|) \gg |X_Q|$.

(Proof) See the appendix.

The two properties in Proposition 3 are easily satisfied in practice. The first property, $u \gg |\mathcal{E}(t_k)|$ and $u \gg |\mathcal{E}^C(t_k)|$, is satisfied if

the number of all entities in a dataset is larger than the number of entities associated with any individual tag. In most social collaboration projects, the number of entities is a very large number, for example, 3,459,565 entities in our experiment dataset (Wikipedia), so the property is satisfied. The second property, $(|\mathcal{E}(t_k)| + |\mathcal{E}^C(t_k)|) \gg |X_Q|$, is also satisfied because we believe that in practice a user provides a small portion of desired results as input examples and expects a sizable number of results [1].

The part of the equation $(|\mathcal{E}^C(t_k)|/u)^{m_k}$ can be interpreted as an adjustment for missing a tag t_k in some input examples. When a tag is shared by all the input entities, the number m_k is zero and this part of the equation becomes one, meaning that no adjustment are required. As the number of input entities missing a tag t_k increases (m_k increases), the value decreases exponentially. Thus, this part adheres to the notion we learned in naïve intersection model that tags in $T_Q^{\cap}$ are important.

When we consider the missing tag effect, the part $1/\binom{|\mathcal{E}(t_k)| + |\mathcal{E}^C(t_k)|}{|X_Q|}$ can be interpreted as a refinement for addressing the popularity of a tag. The concept is identical to the explanation of $1/\binom{|\mathcal{E}(t_k)|}{|X_Q|}$ in Proposition 2, where we tend to place more value on a rarely-seen tag in the query X_Q than a popular tag, so that the model alleviates the popularity tag-bias.

In our experiments, since the ratio of missing tags is unknown, we simply make the following assumption: 50% of tag-entity relations being missing. Thus, if half tag-entity relations are missing, we could conclude $|\mathcal{E}^C(t_k)| \sim |\mathcal{E}(t_k)|$. In practical, to make a reasonable approximation of tag-missing ratio requires the understanding of target datasets and some heuristic trials.

6. EXPERIMENT

Evaluating effectiveness of different approaches is a challenging task because the quality of results is subjective and no standard corpus exists. Thus, we think the best way to evaluate a ranking algorithm is to build a search engine and see how well users perceive our new ranking results.

Some IR communities, such as TREC (Text retrieval conference; http://trec.nist.gov), provide datasets for evaluating keyword search. Unfortunately, those datasets are not collected for testing query-by-example search. As a result, we download the dataset of Wikipedia and create a search interface on top of the dataset for collecting user surveys[+].

In this section, we are going to provide the overview of our dataset, explain the design of the surveys, and then analyze the users' satisfaction scores for each proposed algorithm.

6.1 A study of Dataset – Wikipedia

We implement our system based on a Wikipedia snapshot dumped on November 3rd, 2009. Wikipedia is the largest collaborative encyclopedia, and it contains more than 3 million articles in English. The collaborative nature means that all tags in Wikipedia are generated by human, and some tag-entity relations might be missing.

We consider every wiki page as an entity, so the name (page title) of the wiki page becomes a unique identifier of the entity. For tag information, every wiki page contains "category" entries, and we consider each category as a tag. When a category is labeled with an article, we consider the category (the tag) is associated with the article (the entity). The dataset contains 471,443 unique tags, 3,459,565 entities, and 13,196,971 associations.

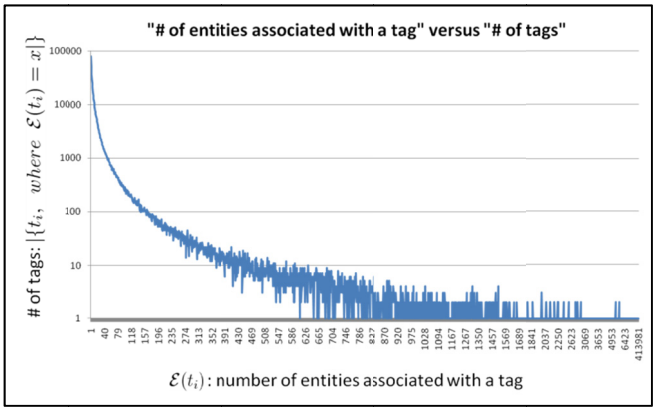

Fig. 2 Size of category distribution

In Fig. 2, the x-axis refers to $|\mathcal{E}(t_i)|$, the number of entities associated with the tag t_i, and the y-axis refers to the number of tags with the size $|\mathcal{E}(t_i)|$. For example, the value of y-axis of "x=1" is 79,870, which means that 79,870 tags (about 16% of all tags) appear only once. These tags are possibly newly created, or they have not yet been accepted by other users. On the other hand, the right-most point of the line in Fig. 2 is the *Living People*, and its y value, one, means that only this tag is associated with 413,981 entities (value of x-axis).

The chart in Fig. 2 shows that some tags are extremely popular and many tags are unpopular. Our statistics show that about 16% of tags are singleton, 74% of tags are associated with 2 to 50 entities, and less than 10% of tags are associated with more than fifty entities ($|\mathcal{E}(t_i)|$>50). Golder et al. [5] reported a similar distribution on Del.icio.us data where a power law distribution was also observed.

6.2 Effectiveness Evaluation

Evaluating the similarity between an entity and a query is difficult because whether the entity is similar to another entity in the query is subjective. For example, someone might argue that the entity Nikon (a manufacturer of cameras) is not similar to the entity Toyota (a manufacturer of cars), but another person may feel they are similar because both of them are Japanese companies. Thus, we decide to create a benchmark through conducting user surveys.

6.2.1 Experiment design

We collected 600 valid questionnaires from 69 students in UCLA to create a benchmark for evaluating user satisfaction. In each questionnaire, the system randomly selects one of 10 pre-set scenarios (listed in the appendix), where every scenario contains two to three entities to form a query.

[+] The user surveys can be downloaded from http://oak.cs.ucla.edu/~chucheng/publication/qbe_survey_results_public.zip

Once the scenario is selected, the query is issued to our system and we collect the top 200 results from each algorithm. We mix all the top 200 results, and the questionnaire is then generated through randomly selecting entities from the mixed set. Note that we run a customized questionnaire generation process so that high-rank results are reviewed by more persons. Entities in the first few pages, high-ranked results, are often considered important because users tend to read results according to the order of entities. With limited resource (questionnaire takers), we are more interested in knowing how each approach performs in these high rank results. Therefore, in the process of questionnaire generation, 30 questions are drawn from high-rank entities, particularly the entity in the top 40 results of any model, and 10 questions are drawn from the other entities.

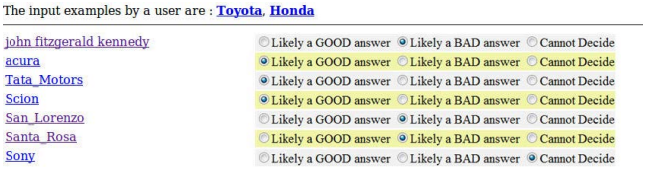

Fig. 3 A screen clip of a questionnaire

To avoid bias, questionnaire takers are unaware of how their questionnaires are generated. They are told that entities in a query belong to a group where all entities share some properties (the intent of the query). They are asked to examine whether an entity in a questionnaire belongs to the intent of the query. The intent of the query, such as Car Manufacturers, is not revealed to the questionnaire taker. They are asked to read the Wikipedia pages and make an evaluation of possible intentions to the query. Fig. 3 is a (partial) snapshot of a questionnaire.

Every questionnaire contains 50 entities; however, 10 of 50 entities are known to be clearly unrelated to the query for filtering out invalid surveys. If any pre-set unrelated entity is marked as a positive example, we rule out the entire questionnaire. Thus, only 600 of 714 questionnaires are considered valid after the screening process, and a total of 24,000 feedbacks are collected.

6.2.2 *Benchmarks and Results*

We use Equation (10) to calculate the satisfaction score of an entity to a query. Whenever a questionnaire taker reports a good answer, the corresponding entity earns the full credit of one. While he cannot make a clear judgment, we still assign 0.5 point to the entity. After averaging scores of an entity, a high satisfaction score implies that most users believe that the entity and entities in the query are similar. And a score close to 0.5 could imply a situation that users hold their opinions. We then consider the satisfaction scores as benchmarks for evaluating different algorithms.

$$S(x_i|X_Q) = \frac{\#\ of\ Good + 0.5 * \#\ of\ Cannot\ Decide}{\#\ of\ total\ response} \quad (10)$$

Fig. 4 contains the comparison results of all models based on the ten scenarios we use in experiments. In addition to the four algorithms proposed in the paper, we add Google Sets [6] and the term frequency–inverse document frequency (TFIDF) algorithm into the comparison. The x-axis represents for top N results, and the y-axis indicates the running average of satisfaction scores for top N results.

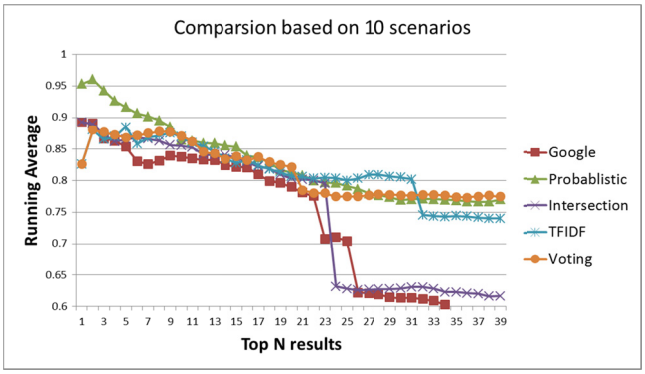

Fig. 4 Comparison of different models (top 40)

In Fig. 4, our one class probabilistic model outperforms other models in the top 10 results; our balanced voting model and our one class probabilistic model have high user satisfactions while comparing top 40 results. We notice that the intersection model and Google Sets return approximately only 25 entities on average. As a result, the average user satisfaction promptly drops in that a user cannot find any results. The probabilistic model reports a promising result: 5.26% more satisfaction than the Google Sets based on 2400 questions in 600 questionnaires.

6.2.3 *Discussion*

Although the algorithms of Google Sets are not disclosed, we compare our results against its result because Google is currently the leading search engine and Google Sets aims to solve the same problem: finding similar items through quires consisting of examples.

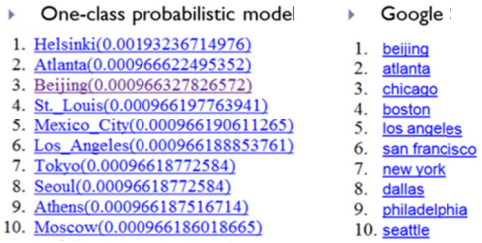

Fig. 5 One class probabilistic model vs. Google Sets

Fig. 5 demonstrates a query in which most survey responders feel the results provided by Google Sets are inferior or questionable. We create a query {Beijing, Atlanta}, where the intent behind the query is to find cities that hosted the Olympic Games. Google Sets seems to interpret our intent as finding cities in America. In contrast, our one-class probabilistic model identifies cities that hosted the Olympic Games.

Though which interpretation is better is controversial, during interviews after the survey, many responders said that their first impressions after seeing {Beijing, Atlanta} shown in the query were about the event that Beijing hosted the 2008 Summer Olympics, or about the notion that both of them are metropolises. Even for responders preferring the notion "metropolises", they were still unhappy when seeing a result that is biased towards Atlanta and neglects properties contributed by Beijing.

7. RELATED RESEARCH

The study of social collaboration tagging system has been attracting attention from researchers. Mathes [9] investigated the

social tag data and pointed out that it was a fundamentally chaotic. Shirkly [12] also argued that using tag information is difficult because a user has the freedom of choosing any word he or she wants as a tag. Later, Gloder et al. [5] analyzed the structure of collaborative tagging systems on top of Del.icio.us data and concluded that tag data follow a power law distribution. Their studies back up our argument that some tags are extremely popular while others are rarely used.

Despite the challenges in using tag-entity information, many researchers continue to work in the field and have shown the potential of social tag clouds. Tso-Sutter et al. [14] used relationships among tags, users, and entities to recommend possible interesting entities to users. Penev et al. [10] used WordNet to acquire terms relating to a tag and applied TFIDF [11] similarity on both the tags and their related terms for finding similar entities (pages in their research). They used WordNet to interpret the meaning of tags, trying to measure the similarity between entities based on keywords through expanding tags with WordNet. Although we aim to solve similar problems, our approaches focus on using only tag information, because we believe, as Strohmaier et al. [13] suggested, that users use social tags for categorizing and describing resources.

We believe that our study can benefit many applications. For example, Givon et al. [4] showed that social tags can be used in dealing with recommendations in large-scale book datasets. Moreover, the keyword generation task can be considered as a similar problem, for example, Fuxman et al [3] draw an analogy from a keyword to a url and an entity to a tag. Finding similar entities based on shared tags is similar to finding related keywords based on shared urls that users click on.

Many researchers also work on tag ranking or tag aggregation. Recently, Wetzker et al. [15] focused on creating a mapping between personal tags to aggregate tags with the same meaning, Heymann et al. [7] used tag information to organize library data, Wu et al. [16] explained how to avoid noise and compensate for semantic loss, Liu et al. [8] studied how to rank tags based on importance, and Dattolo et al. [2] studied how to identify similar tags through detecting relationships between them.

8. CONCLUSION

In this paper, we investigated the problem of finding similar items with query-by-example interface on top of social tag clouds. We introduced three approaches, and built a search engine on top of them, creating a benchmark for evaluating the users' satisfaction through collecting 600 questionnaires. The experiment results suggest that social tag data, even though they are uncontrolled and noisy, are sufficient for finding similar items. Finally, we show that both the voting model and the one-class probabilistic model reach high user satisfaction.

We explain two important challenges of utilizing tag information: popularity bias and the missing tag effect, and explain how to overcome these difficulties through partial weight strategies and probability utilization. We show that, in terms of users' satisfaction, our algorithms are superior or at least compatible to Google Sets and TFIDF model. Our approaches return hundreds of relevant entities without sacrificing the quality in the top results. Moreover, our models rely on only social tag information.

Our proposed framework not only provides an ability to find similar items, but also shows the application potential of social tag information. We demonstrate that the task can be accomplished through providing a query consisting of entities and using only tag information, even though the tag information is uncontrolled and noisy. Through this study, queries for finding similar items, such as "Honda or Toyota or similar", are handled properly. Our research also highlights the value of using social collaboration data, tag clouds, to refine existing search technologies.

9. REFERENCE

[1] Arampatzis, A., and Kamps, J. A study of query length. In Proceedings of the Annual ACM Conference on Research and Development in Information Retrieval, 2008, 811-812.

[2] Dattolo, A., Eynard, D., and Mazzola, L. An integrated approach to discover tag semantics. In *Proceedings of the ACM Symposium on Applied Computing*, 2011, 814–820.

[3] Fuxman, A., Tsaparas, P., Achan, K., and Agrawal, R. Using the wisdom of the crowds for keyword generation. In *Proceddings of the International World Wide Web Conference*, 2008, 61-70.

[4] Givon, S., and Lavrenko, V. Large Scale Book Annotation with Social Tags. In *Proceedings of International AAAI Conference on Weblogs and Social Media,* 2009, 210-213.

[5] Golder, S. and Huberman, B. A. Usage Patterns of Collaborative Tagging Systems. *Journal of Information Science*, 32(2), 2006, 198-208

[6] Google Sets. http://labs.google.com/sets

[7] Heymann, P., Paepcke, A., and Garcia-Molina, H. Tagging human knowledge. In *Proceedings of the Web Search and Web Data Mining*, 2010, 51-60.

[8] Liu, D., Hua, X., Yang, L., Wang, M., and Zhang, H. Tag ranking. In P*roceedings of the International World Wide Web Conference*, 2009, 351-360.

[9] Mathes, *A. Folksonomies - cooperative classification and communication through shared metadata.* Computer Mediated Communication, LIS590CMC (Doctoral Seminar), University of Illinois Urbana-Champaign, Dec. 2004.

[10] Penev, A., and Wong, R. K. Finding similar pages in a social tagging repository. In *Proceeding of the 17th international conference on World Wide Web* , 2008, 1091–1092.

[11] Robertson, S. E., and Sparck-Jones, K. Relevance weighting of search terms. *Journal of the American Society for Information Science, 27(3)*, 1975, 129–146.

[12] Shirky, C. Ontology is overrated: categories, links and tags. http://www.shirky.com/writings/ontology-overrated.html, 2005.

[13] Strohmaier, M, Körner C., and Kern, R. Why do Users Tag? Detecting Users' Motivation for Tagging in Social Tagging Systems, In *Proceedings of International AAAI Conference on Weblogs and Social Media,* 2010, 339-342.

[14] Tso-Sutter, K.H.L., Marinho, L.B., and Schmidt-Thieme, L. Tag-aware recommender systems by fusion of collaborative filtering algorithms. In *Proceedings of the ACM Symposium on Applied Computing*, 2008, 1995–1999.

[15] Wetzker, R., Zimmermann, C., Bauckhage, C., and Albayrak, S. I tag, you tag: translating tags for advanced user models. In *Proceedings of the Web Search and Web Data Mining*, 2010, 71-80.

[16] Wu, L., Yang, L., Yu, N., and Hua, X. Learning to tag, In *Proceedings of the International World Wide Web Conference*, 2009, 361-370.