A Weighted Tag Similarity Measure Based on a Collaborative Weight Model

G.R.J.Srinivas
Search and Information Extraction Lab,
IIIT Hyderabad,* India
srinivasg@research.iiit.ac.in

Niket Tandon†
Max Planck Institute, Germany
ntandon@mpi-inf.mpg.de

Vasudeva Varma
Search and Information Extraction Lab,
IIIT Hyderabad, India
vv@iiit.ac.in

ABSTRACT

The problem of measuring semantic relatedness between social tags remains largely open. Given the structure of social bookmarking systems, similarity measures need to be addressed from a social bookmarking systems perspective. We address the fundamental problem of weight model for tags over which every similarity measure is based. We propose a weight model for tagging systems that considers the user dimension unlike existing measures based on tag frequency. Visual analysis of tag clouds depicts that the proposed model provides intuitively better scores for weights than tag frequency. We also propose weighted similarity model that is conceptually different from the contemporary frequency based similarity measures. Based on the weighted similarity model, we present weighted variations of several existing measures like Dice and Cosine similarity measures. We evaluate the proposed similarity model using Spearman's correlation coefficient, with WordNet as the gold standard. Our method achieves 20% improvement over the traditional similarity measures like dice and cosine similarity and also over the most recent tag similarity measures like mutual information with distributional aggregation. Finally, we show the practical effectiveness of the proposed weighted similarity measures by performing search over tagged documents using Social SimRank over a large real world dataset.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: Miscellaneous

General Terms

Experimentation, Measurement

*International Institute of Information Technology, Hyderabad

†Most of the work has been done when the author was working in IIIT Hyderabad

SMUC'10, October 30, 2010, Toronto, Ontario, Canada.

Keywords

Tagging, Vector Space Model, Tag weighting, Similarity Measures, Tag Similarity

1. INTRODUCTION

Social bookmarking systems like Delicious, Bibsonomy, CiteULike etc. have become extremely popular in recent years [10]. Users share resources by adding keywords in the form of tags, leading to the creation of an aggregated tag-index called folksonomy[1]. This large amount of user-generated content has created significant interest in the research communities to exploit the hidden semantics.

Social bookmarking systems are built upon three dimensions: Resource, User and Tags. Existing models consider two out of the three dimensions i.e. resource and tags, and ignore the user dimension. Some rich information is lost due to the loss of user dimension. For example, when considering the relevance (rank) of a tag with respect to a document considering only overall frequencies(ignoring user dimension), results in assigning exceedingly high weights to some generic and uninformative tags like *web2.0, Internet* and during normalizing weights these highly generic tags push down important yet less frequent tags. This is clearly a drawback of existing weight models. We address the problem of loss of user dimension, first by, proposing a weight model for tags that does not account only tag frequency to provide weightage(importance) to a tag. The weight model is built upon vector space model with some variations. We observed that simple weighting approaches like TF-IDF do not work well for social bookmarking systems' weight model, thereby making it a challenging task.

Another topic of active research is, computing tag similarity that finds application in a wide range of applications like tag clustering, tag recommendation, query expansion, and semantic web amongst other applications. Several methods of computing similarities using ontological resources like WordNet have been proposed [3, 12, 13, 17]. However, these approaches cannot be applied for folksonomies. When users are free to choose tags, the resulting metadata can include homonyms, synonyms about the subject. The terms in a folksonomy may have inherent ambiguity as different users apply terms to documents in different ways. Folksonomies

[1]A folksonomy is a system of classification derived from the method of collaboratively tagging resources with descriptive strings, called tags to annotate and categorize content.

provide for no synonym control; for example, the tags *mac, macintosh,* and *apple* are used to describe Apple Macintosh Computers. Similarly, both singular and plural forms of terms appear (e.g., *flower* and *flowers*), thus creating a number of redundant tags. In addition, as most of the tagging systems do not allow word separators, many users use compound tags (combinations of words) for tagging resources. Such uncontrolled vocabularies lead to ambiguity, polysemy and basic level variation [8, 18]. Hence, these ontology based similarity measures cannot be applied directly to folksonomies.

There are also some existing approaches for extracting tag similarities from folksonomies in the literature. The distribution of tag co-occurrence frequencies has been investigated by Cattuto et.al. in [6]. In [19], Zhang et.al. infer some global semantics from a folksonomy by applying some statistical methods. In [8, 9], Golder, Halpin et.al. have performed extensive analysis to infer global semantics from folksonomies. In [15], Mohammad and Hirst have concluded that distributional measures can easily provide domain specific similarity measures for a large number of domains. In [4, 14], Hotho, Stumme et.al. extended some of the traditional similarity methods of finding semantic relatedness to folksonomy.

Majority of the approaches mentioned above consider the frequency of co-occurrence of tags for computing similarity. These approaches suffer from assigning high relatedness values to extremely generic terms and low relatedness values to relevant specialized terms. Consider a document about *The future of videos*, some of the tags assigned are $video, future, model, toread \ldots$. Note here that, some users tend to give self-organisation tags like *toread*. Now, consider a query expansion task, where the query is "programming model". For the expansion task, the similarity of 'video' is computed with the remaining tags including *similarity* $(video, toread)$. Existing similarity measures are directly proportional to the number of co-occurrences of the two tags. Here, the similarity value accumulates 1's in the numerator, thereby giving a higher value to sim(video, toread). This is clearly not intended. The apparent problem occurs because the existing similarity measures consider co-occurrence whereas the two tags have different relevance to the document. We address these problems using our approach by proposing a concept of weighted similarity. The weighted model considers the weights of tags in calculating similarities instead of frequency of co-occurrence. Consider the previous example, assume we have the weights to the tags: video:1.0, toread:0.0, a weighted co-occurrence value is proportional $\sum[weight(video) * weight(toread)]$ i.e. over all co-occurrences of 'video' and 'toread'. This approach gives a very low co-occurence weights to these tags, hence the similarity measure is less, which is desirable.

We find that the proposed weighted similarity measures perform better than the existing measures. We use extensive evaluation to show the effectiveness of the proposed weighted similarity, based on the weight model we present. Further, we demonstrate practical advantages of our weight model in tag visualization and show effectiveness over existing frequency based tag clouds. The weighted similarity measures proposed find its use in several applications like tag clustering, query expansion, tag recommendation, semantic search. Over a large real world dataset, we demonstrate more than two folds improvement in precision while searching tagged documents using Social SimRank that uses the weighted co-occurrence.

The remainder of this paper is organized as follows. Section 2 discusses the formal folksonomy model. Section 3 explains the proposed weight model for tagging systems. Section 4 presents our weighted similarity measure concept. Section 5 presents the experimental setup and results over different benchmarks and applications like social search. Finally section 6 provides concluding remarks followed by future work.

2. FOLKSONOMY MODEL

We use the formal definition of a folksonomy provided by Hotho et.al. in [11].

Formal Definition:

A folksonomy is formally defined as a tuple F := (U, T, R, Y) where U, T, and R are finite sets, whose elements are users, tags, resources and Y is a ternary relation between them i.e. $Y \subseteq U \times T \times R$. A post is a triple (u, T_{ur}, r) where u ∈ U, r ∈ R and a non empty set $T_{ur} := \{t \in T \mid (u, t, r) \in Y\}$.

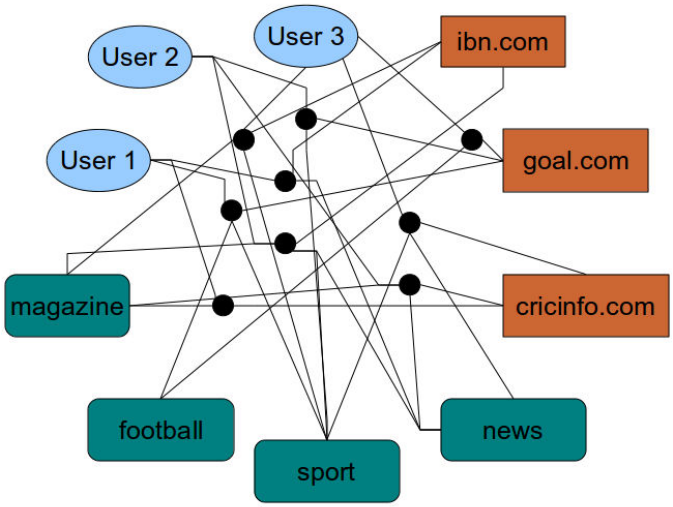

Figure 1: Example of a folksonomy

A folksonomy can be represented as a network shown in the figure 1. In this example there are 3 users, 3 resources and 4 tags. Each dot in the figure represents an annotation (tag posting). In this example user1 annotated goal.com with the tag football.

3. WEIGHT MODEL FOR TAGS IN A FOLKSONOMY

Our weight model for tags is based on Vector Space Model (VSM). In VSM, individual documents are represented as vectors in term space. Terms are words, phrases, or any other indexing units used to identify the content of a text. In case of folksonomy, we consider tags as terms. We represent resources as vectors in tag space. Since different terms have varying importance in a text, an importance indicator, term weight is associated with every term. The term weighting scheme plays an important role for the similarity measures.

According to [16] the weight of a term is calculated using the following formula.

$$a_{ij} = g_i * t_{ij} * d_j \quad (1)$$

Where g_i is the global weight of the i^{th} term, t_{ij} is the local weight of the i^{th} term in the j^{th} document; d_j is the normalization factor for the j^{th} document.

There are three components in a tag weighting model:

$$w_{td} = g_t * l_{td} * n_d \quad (2)$$

Where $t \in T_r$, w_{td} is the weight of tag t with respect to a document, g_t is the global weight of the tag, l_{td} is the local weight of the tag in the document d, n_d is the normalization factor of the document d. Let us visit the three components one by one, with their context into tagging systems:

3.0.1 Local weight

Local weight depends only on the frequencies within the document and not on inter-document frequencies. In case of tagging, a single user will not repeat exactly the same tag to a resource. In many cases, we cannot see duplicates in the tags given by a single user to a particular resource. However, we have observed that users tend to give some morphological variations of the words as tags. For example, consider the following set of tags given by a user for a resource. Mathematics, algorithms, math, matrix, multiplication, parallel, maths, optimization. Here both math and maths are same. It indicates the importance of the tag math to that resource. Hence, we have performed stemming during preprocessing. We have experimented with two variants of term frequency for local weighting. Simple term frequency (tf) and normalized term frequency (TF') which is calculated as shown in formula 3 .

$$tf' = \frac{tf}{document\ length(dl)} \quad (3)$$

We chose simple term frequency based weighting as it will not be large in case of tagging. We chose normalized term frequency as some users tend to give too many tags to a resource.

3.0.2 Global weight

Global weighting tries to give a discriminative value to each term in the corpus. It is used to place emphasis on terms that are discriminating based on the dispersion of a particular term in the corpus. Many schemes are based on the idea that the less frequently a term appears in the whole collection, the more discriminating it is. This is particularly true in the case of tags, because tags are usually generic in nature. But there are tags which are more into detail, and hence more discriminating, and are generally less frequent.

3.0.3 Normalization

The third component of the weighting scheme is the normalization factor, which is used to correct discrepancies in document lengths. E.g. In case of tagging systems, a resource that has been given more tags, will be favoured if weights are not normalized. Since it is not always true that a resource that has been given more number of tags is more relevant than a resource with lesser number of tags. Hence, it is useful to normalize the document vectors so that documents are not favoured based on their lengths. We have used cosine normalization which is computed using the formula.

$$n_d = \frac{1}{\sqrt{\Sigma_{i=1}^{n}(g_{t_i} * l_{t_i d})^2}} \quad (4)$$

where n is the number of terms in the document d
t_i is the i^{th} term in the document.
$l_{t_i d}$ is the local weight of the term t_i in the document.
g_{t_i} is the global weight of the term t_i in the corpus.

Weighting method	Formula
TF - IDF	$(tf\ in\ d) * g_t$
TF' - IDF	$(tf\ in\ d)/dl * g_t$
TF'	$(tf\ in\ d)/dl$

Table 1: Weighting methods used in TRU

3.1 Proposed Tag Weighting Model

We have computed weights of tags in folksonomies using two different models. These two models differ in the perspective of a document. We name these two models as Tag-Resource-User(TRU) model and Tag-Resource(TR) model.

3.1.1 Tag-Resource-User(TRU) model

In this model, we consider tags at user level. So we named it Tag-Resource-User (TRU) model. We consider the set of tags (T_{ur}) given by a user to a resource as a document. Each tag given by the user is considered as a term in the document. We consider all the posts associated with a resource as a collection of documents (corpus) for computing global weight. In this model, we calculate global weight using formula 5.

$$g_{tr} = \log \frac{|U_r|}{|\{u \in U_r | (u,t,r) \in Y\}|} \quad (5)$$

where U_r is the set of users who have annotated resource r. We compute the weight of a tag w.r.t each document using the weighting formulae listed in table 1. Then to obtain the weight of a tag w.r.t a resource w_{tr}, we add the weights of the tag obtained across all the users who tagged the resource as shown in equation 6.

$$w_{tr} = \Sigma_{u \in U_r} w_{tur} \quad (6)$$

Then we normalize the weights of tags using cosine normalization.

3.1.2 Tag-Resource(TR) model

In this model, we consider tags at the resource level. So we named it as Tag-Resource (TR) model. We consider the set of tags (T_r) given by all the users to a resource as a document. The tags given by the users are considered as terms of the document. In this model, we calculate global weight using the following formula.

$$g_t = \log \frac{|R|}{|\{r \in R | (u,t,r) \in Y\}|} \quad (7)$$

We compute the weight of a tag in each document using the weighting formulae listed in table 2.

We compute w_{rt} which gives the weight of tag t for the resource r for all the resources and tags. This will be used in

Weighting method	Formula
TF - IDF	$(tf\ in\ d) * g_t * n_d$
TF' - IDF	$(tf\ in\ d)/dl * g_t * n_d$
TF'	$(tf\ in\ d)/dl$

Table 2: Weighting methods used in TR

	magazine	football	sport	news
cricinfo.com	0.41	0	0	0
goal.com	0	0.41	0.41	0
cnn.com	0	0	0	0.41

Table 3: Weights using TRU(TF-IDF) model according to user1's annotations

computing similarity between pairs of tags. We have experimented with different variations of weighting in TRU and TR assumptions of a document as listed in tables 1 and 2.

We show the weights obtained using the above weight models for the tags in the folksonomy depicted in the figure 1. Using TRU(TF-IDF) model, we get the weights shown in table 3 for the tags given by user1 to the three URLs.

Similarly, we compute the weights of the tags given by user2 and 3. Then we get the final weights as shown in table 4 after aggregating across all users.

Using TR(TF-IDF) model, we have obtained the weights shown in table 5 for all the tags w.r.t the three URLs. In this case sport got a weight of 0 because global weight becomes zero. In this example, sport is related to all the URLs. However in real world scenarios, a single tag cannot be used for all the resources because the information content of the tag is low. Such tags are not useful for practical applications like query suggestion, ranking and other applications. Thus global weight penalizes very frequent tags, but as a downside, it also penalizes some of the important terms that occur in every document.

Along with these variations of tag weighting models, we have also experimented with a machine learning (ML) approach to obtain weights.

3.1.3 Tag weighting using ML Approaches

In addition to the weighting models mentioned in sections 3.1.1 and 3.1.2, we have also experimented with machine learning approaches for tag weighting. We view the problem of weighting tag-resource annotation as a one class classification problem. The probability of the annotation being relevant is considered as the weight of the tag w.r.t the resource. We have classified the tags as relevant/non relevant using different classification algorithms like Adaboost, LibSVM, RandomForest etc. We have trained the classifiers using frequency(tf) and the weights obtained using the formulae given in table 1,2 in the TRU and TR models as

	magazine	football	sport	news
cricinfo.com	0.51	0	0.69	0.51
goal.com	0	0.71	0.71	0
cnn.com	0.42	0.57	0.57	0.42

Table 4: Weights using TRU(TF-IDF) model

	magazine	football	sport	news
cricinfo.com	0.5	0	0	0.5
goal.com	0	1	0	0
cnn.com	0.4	0.2	0	0.4

Table 5: Weights using TR(TF-IDF) model

	magazine	football	sport	news
magazine	-	0.5	0.8	1.00
football	0.5	-	0.8	0.5
sport	0.8	0.8	-	0.8
news	1.00	0.5	0.8	-

Table 6: Similarities using Dice similarity

features. These different variations of weighting techniques get the importance of a tag for the resource. Then we have used the probability of a tag belonging to the relevant class as weight of the tag w.r.t resource.

4. SIMILARITY MEASURES

In this section we define some of the existing similarity measures and also our weighted similarity model. We have compared the similarities obtained using our model with dice, cosine and mutual information with distributional aggregation. According to [14] mutual information with distributional aggregation is the best performing method. Dice and cosine are some of the best corpus based measures. Hence, we have considered these measures as baselines to compare our model. In this section, we first define the baselines we have considered and then we define our weighted similarity measures.

We will use the following notations throughout the paper.
$\sigma(t_1, t_2)$ is used to denote the similarity of pair of tags t_1 and t_2.
t_i is used to denote a tag.
T_i is the set of resources tagged with t_i.
$|T_i|$ is the cardinality of the set of resources T_i.

4.1 Dice Similarity

Dice similarity for two sets X and Y is defined as

$$sim = \frac{2|X \cap Y|}{|X| + |Y|} \tag{8}$$

Similarly, in case of folksonomies we have computed dice similarity of pair of tags using the following formula.

$$\sigma(t_1, t_2) = \frac{2 * |T_1 \cap T_2|}{|T_1| + |T_2|} \tag{9}$$

For the tags the folksonomy shown in figure 1 we obtain the similarities shown in table 6 using dice.

The simple example in table 6 explains the similarity measure using a dice similarity measure. The value of Sim_dice (Football, sport) is relatively higher than Sim_dice (Football, news).

	magazine	football	sport	news
magazine	-	0.5	0.82	1.00
football	0.5	-	0.82	0.5
sport	0.82	0.82	-	0.82
news	1.00	0.5	0.82	-

Table 7: Similarities using Cosine similarity

	magazine	football	sport	news
magazine	-	0.68	1.35	1.34
football	0.68	-	1.06	0.68
sport	1.35	1.06	-	1.35
news	1.34	0.68	1.35	-

Table 8: Similarities using MI

	magazine	football	sport	news
magazine	-	0.09	0.28	0.63
football	0.09	-	0.51	0.14
sport	0.28	0.51	-	0.29
news	0.63	0.14	0.29	-

Table 9: Similarities using Weighted Dice(TF')

	magazine	football	sport	news
magazine	-	0.72	1.3	1.16
football	0.72	-	1.2	0.72
sport	1.3	1.2	-	1.3
news	1.16	0.72	1.3	-

Table 10: Similarities using Weighted MI(TF')

It indicates that football and sport are more related compared to football and news. In this example, (football, sport) and (magazine,sport) are given the same similarity values. However (football, sport) are more related when compared to (magazine,sport).

4.2 Cosine Similarity

Cosine similarity for two tags t_1,t_2 is defined as

$$\sigma(t_1, t_2) = \frac{|T_1 \cap T_2|}{\sqrt{|T_1|.|T_2|}} \tag{10}$$

For the tags the folksonomy shown in figure 1 we obtain the similarities shown in table 7 using cosine similarity. This measure also faces the same problems mentioned in dice similarity (section 4.1).

4.3 Distributional Mutual Information

According to [14] mutual information using distributional aggregation for a folksonomy is computed as

$$\sigma(t_1, t_2) = \Sigma_{r_1 \in T_1} \Sigma_{r_2 \in T_2} p(r_1, r_2) \log \frac{p(r_1, r_2)}{p(r_1)p(r_2)} \tag{11}$$

where

$$p(r) = \frac{\Sigma_t w_{tr}}{\Sigma_{t,r} w_{tr}}, p(r_1, r_2) = \frac{\Sigma_t min(w_{tr_1}, w_{tr_2})}{\Sigma_{t,r} w_{tr}} \tag{12}$$

For the tags in figure 1 we obtain the similarities shown in table 8 using Mutual Information. In this example, sim (sport, news) is the same as sim (magazine, news) which shouldnt be.

4.4 Proposed Model - Weighted Similarity Measures

The similarity measures discussed i.e. dice, cosine and mutual information give higher similarity values to tag pairs proportional to the co-occurrence count. This is not desirable as depicted in the example on $sim('video','toread')$ in Section 1. We compute the upper and lower bound values of the weighted co-occurrence. Consider two tags a, b whose similarity we want. If a, b are both completely relevant to a document then weighted co-occurrence for that document is upper bounded by a weighted co-occurrence of 1. Whereas, if one of the tags is irrelevant to the document, then the weighted co-occurrence for that document becomes zero, this is the lower bound.

4.4.1 Weighted Dice Similarity

We propose a modified version of Dice Similarity which uses the weight of a tag w.r.t a resource in computing the similarities of tag pairs. We consider the association of a tag to a resource as fuzzy relation where the value of association is the weight of the tag w.r.t resource. We define the weighted dice similarity as

$$sim(t_1, t_2) = \frac{\Sigma_{r \in T_1 \cap T_2} w_{t_1 r} * w_{t_2 r}}{\Sigma_{r_1 \in T_1} w_{t_1 r_1} + \Sigma_{r_2 \in T_2} w_{t_2 r_2}} \tag{13}$$

Table 9 gives similarity measures of the tag pairs computed using weighted dice similarity with normalized term frequency weighting. In this case, (football, sport) is given more similarity value when compared to (magazine,sport).

4.4.2 Weighted Mutual Information

We have also evaluated the impact of our weights in case of distributional mutual information. In case of weighted mutual information(weighted MI) we use the weights w_{tr} obtained using our weight model in the formula 11.

Table 10 shows the similarities of the pairs obtained using weighted MI with weights obtained using TR(TF-IDF) model.

5. EVALUATION

In this section, we first describe the data used for our experiments.

5.1 Data collection

We have used a publicly available crawl of Delicious[2] provided by DAI-Labor[3]. This dataset contains all public bookmarks of about 950,000 users retrieved from del.icio.us between December 2007 and April 2008. The retrieval process resulted in about 132 million bookmarks or 420 million tag assignments that were posted between September 2003 and December 2007. For reasons of tractability, we randomly chose a smaller subset of 100 URLs from this dataset for our experiments. The subset contains 39,632 users, 100 urls, 7,495 tags and 190,724 tag assignments.

[2]http://www.delicious.com
[3]http://www.dai-labor.de

5.1.1 *Labelled Data for ML Approaches*

For training and testing, we randomly chose url-tag tuples among the 420 million tuples. The tags of these tuples were then labelled manually as either relevant or irrelevant. The number of labelled examples is 717. We experimented with several learning algorithms, including RBF-kernel support vector machines as implemented in LIBSVM [7], Random Forest and Adaboost amongst others. In Section 5.2.1, we report and evaluate results obtained with alternative algorithms. For evaluation, we rely on 10-fold leave-one-out cross-validation, where the set of labelled examples is randomly partitioned into 10 equal-size parts, and then an average score is computed over 10 runs. In each run, a different part is reserved for testing, and the remaining 9 parts are used as the training set.

5.2 Evaluation of Tag Weighting approaches

5.2.1 *Weight Model Accuracy*

Table 11 gives the cross validation results for the weights obtained using different machine learning approaches like SVM, Random Forest, Adaboost etc. These results indicate that the Adaboost is the best performing approach in case of tag weighting.

Classifier	Precision	Recall	F_1-Measure	ROC
SVM	0.63	0.455	0.528	0.655
J48	0.643	0.507	0.567	0.655
BF-Tree	0.69	0.425	0.526	0.669
Random-Forest	0.624	0.54	0.579	0.653
Adaboost	0.685	0.466	0.555	0.684

Table 11: Cross validation results for weights learned.

5.2.2 *Visual Analysis*

Popular tag visualization techniques like Tag Clouds, weigh tags in the visualization (e.g. cloud) based on the frequency of the tags. We use a modified tag cloud based on weights from our weight model thereby assigning relevance weightage instead of frequency. We compare the tag clouds for the url [4] in Figure 2,3. Tags in existing techniques assign higher weights to generic tags like web2.0 , Internet. These tags being uninformative, and not supportive during tag based search. Our weight model is able to penalize the high frequency terms that are not relevant to the url post.

5.3 Evaluation of Similarity Measures

There are two ways of evaluating tag similarity measures. One way of evaluating is having a two ranked lists of word pairs with two different similarity measures and obtaining the correlation between them using standard correlation coefficients. Another way is doing an indirect form of evaluation by the performance of these similarity measures in tasks like automatic spelling correction, word sense disambiguation etc. We used the first way of evaluation. WordNet [5] is a semantic lexicon of the English language. There are a number of semantic relatedness measures based on WordNet. We have used the evaluation method proposed in [5]

[4] `http://37signals.com/svn/archives2/dont_scale_99999_uptime_is_for_walmart.php`

[5] `http://wordnetweb.princeton.edu/`

Figure 2: Tag cloud based on Frequency

Figure 3: Tag cloud based on the proposed Model

using WordNet. According to [3] the method proposed by Jiang and Conrath[12] performs the best amongst the wordnet based measures.

We obtained the semantic relatedness measure of pairs using Jiang-Conrath distance and considered it as a gold standard. Then, we have obtained a ranked list of 2000 tag pairs according to the Jiang-Conrath distance. We have also obtained the similarities of those pairs of tags using dice, MI and weighted MI and ranked them.

We have used Spearman's rank correlation coefficient and Kendall tau rank correlation coefficient for calculating the correlation between the ranked pairs. For computing Kendall tau we have used the efficient implementation of Knight's $O(NlogN)$ algorithm by [2].

Figure 4 depicts the Kendall's τ correlation and Spearman's ρ correlation coefficient between each measure and the WordNet reference. We have also compared our weighted dice similarity measure with dice, cosine and mutual information similarity measures. The correlation coefficients give a measure of the association between the rankings given by any similarity measure and the gold standard i.e. WordNet. From the results shown in the figure 4, weighted dice similarity with normalized term frequency is correlating well with the gold standard better than other measures. It is the best performing method among the existing similarity measures.

We have also evaluated our weighted similarity measures by on their performance in tag search.

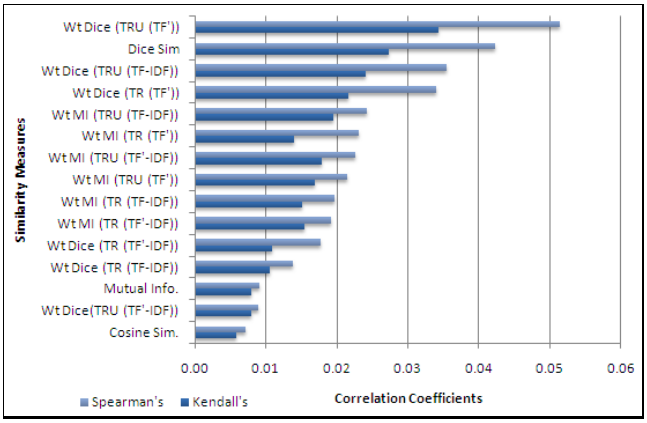

Figure 4: Kendall's and Spearman correlation coefficient comparison for all Sim measures. TRU sim measure with TF' achieves the highest co-efficients, signifying the best performance.

5.3.1 *Evaluation of similarity measures in Search performance*

Let $q = q_1, q_2, \ldots, q_n$ be a query that consists of n query terms and $A(p) = a_1, a_2, \ldots, a_m$ be the annotation set of web page p, Equation 14 shows the similarity calculation based on the Social SimRank proposed by Shenghua Bao et.al. in [1].

$$sim_{SSR}(q,p) = \sum_{i=1}^{n}\sum_{j=1}^{m} S_A(q_i, a_j) \qquad (14)$$

Finding a good set of queries and relevant results for them is not an easy task. We used the approach by Shenghua Bao in [1] to use DMOZ categories as global ground truth.

We had 10 queries and relevant documents related to these queries. But, this is not sufficient to compute precision, so we solve this problem by injecting irrelevant documents to this set. Consider a query q_1, we find two queries q_2 and q_3 that are most unrelated to q_1 through manual inspection. The set of documents related to q_2 and q_3 are irrelevant to q_1.

Next, we compute ranking scores based on Social SimRank, Eq:14. We compute this score using a Dice Similarity Measure and using our Weighted Dice Similarity. In order to compare the results, at different settings of threshold of SimRank score, we compute the Precision values. Figure 5 clearly shows the high precision obtained by Weighted Dice Similarity, outperforming the Dice Similarity.

Figure 6 depicts the high $F_1 - Measure$ obtained by Weighted Dice Similarity, outperforming the Dice Similarity.

Further, three different users manually rank the top 10 results of 10 queries. In order to check the correlation of the ranking order of the Dice and Weighted Dice Similarity measure, we compute the average Kendall's value over all the ten queries. Figure 7 clearly shows that for majority of the queries, Weighted Dice Similarity outperforms Dice Similarity and comes closer to manual rankings.

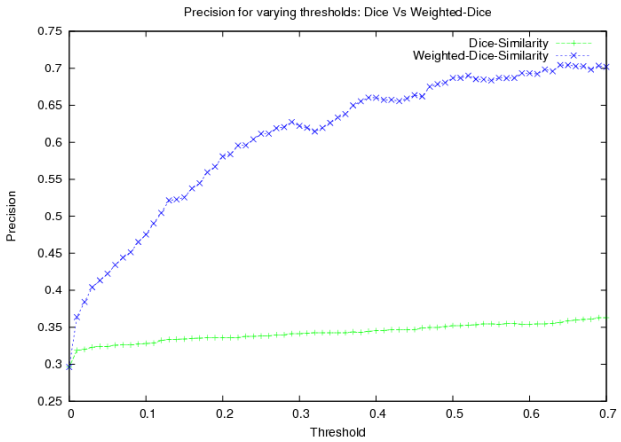

Figure 5: Precision values at varying thresholds, for Dice and Weighted Dice

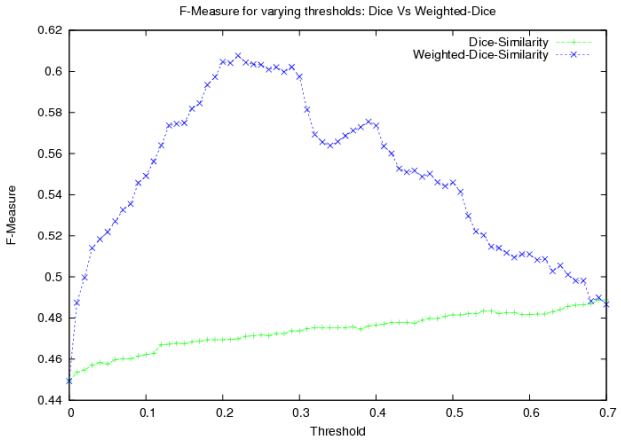

Figure 6: F-Measure values at varying thresholds, for Dice and Weighted Dice

6. CONCLUSION AND FUTURE WORK

We have proposed a weight model for tags in a folksonomy. Among the variants of weight models, Normalized Term Frequency with Tag Resource User(TRU) model is the best performing model. We showed the application of our weight model in tag visualization achieving more intuitive tag cloud than frequency based tag clouds. We introduced the concept of weighted similarity, and proposed similarity measures extending the traditional similarity measures using weighted similarity concept. The proposed similarity measure outperforms the existing similarity measures on metrics like Kendall correlation, Spearman correlation coefficient and gives impressive results on search using Social SimRank over a large real world dataset. As a further extension of our work, we plan to explore the effectiveness of our weighted similarity model in applications like tag clustering, tag recommendation, resource similarity etc..

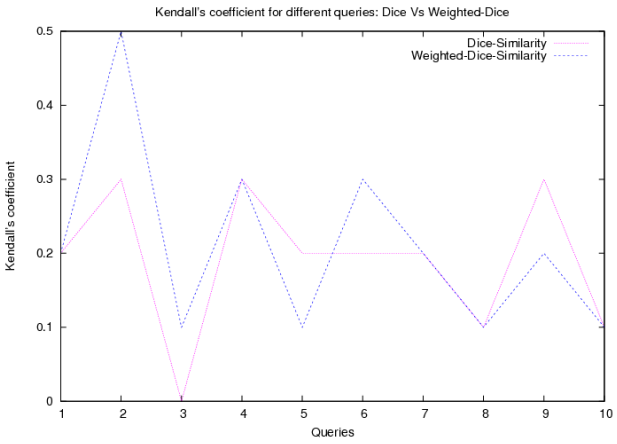

Figure 7: Kendall Coefficient values for Dice and Weighted Dice for search rankings when compared to human evaluated ranking.

7. REFERENCES

[1] S. Bao, G. Xue, X. Wu, Y. Yu, B. Fei, and Z. Su. Optimizing web search using social annotations. In *Proceedings of the 16th international conference on World Wide Web*, pages 501–510. ACM, 2007.

[2] P. Boldi, M. Santini, and S. Vigna. Do your worst to make the best: Paradoxical effects in pagerank incremental computations. In *Proceedings of the third Workshop on Web Graphs (WAW), volume 3243 of Lecture Notes in Computer Science*, pages 168–180. Springer, 2004.

[3] E. Budanitsky and G. Hirst. Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32:13–47, 2006.

[4] C. Cattuto, D. Benz, A. Hotho, and G. Stumme. Semantic analysis of tag similarity measures in collaborative tagging systems. In *Proceedings of the 3rd Workshop on Ontology Learning and Population (OLP3)*, pages 39–43, Patras, Greece, July 2008.

[5] C. Cattuto, D. Benz, A. Hotho, and G. Stumme. Semantic grounding of tag relatedness in social bookmarking systems. In *ISWC '08: Proceedings of the 7th International Conference on The Semantic Web*, pages 615–631, Berlin, Heidelberg, 2008. Springer-Verlag.

[6] C. Cattuto, V. Loreto, and L. Pietronero. Semiotic dynamics and collaborative tagging. *Proceedings of the National Academy of Sciences (PNAS)*, 104(5):1461–1464, January 2007.

[7] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001.

[8] S. Golder and B. A. Huberman. The structure of collaborative tagging systems. *Journal of Information Science*, 32(2):198–208, Aug. 2005.

[9] H. Halpin, V. Robu, and H. Shepard. The dynamics and semantics of collaborative tagging. In *Proceedings of the 1st Semantic Authoring and Annotation Workshop (SAAW'06)*, 2006.

[10] T. Hammond, T. Hannay, B. Lund, and J. Scott. Social bookmarking tools (i). *D-Lib Magazine*, 2005.

[11] A. Hotho, R. Jaschke, C. Schmitz, and G. Stumme. Information retrieval in folksonomies: Search and ranking. In Y. Sure and J. Domingue, editors, *The Semantic Web: Research and Applications*, volume 4011 of *Lecture Notes in Computer Science*, pages 411–426, Heidelberg, June 2006. Springer.

[12] J. Jiang and D. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proc. of the Int'l. Conf. on Research in Computational Linguistics*, pages 19–33, 1997.

[13] D. Lin. An information-theoretic definition of similarity. In *In Proceedings of the 15th International Conference on Machine Learning*, pages 296–304. Morgan Kaufmann, 1998.

[14] B. Markines, C. Cattuto, F. Menczer, D. Benz, A. Hotho, and G. Stumme. Evaluating similarity measures for emergent semantics of social tagging. In *WWW '09: Proceedings of the 18th international conference on World wide web*, pages 641–650, New York, NY, USA, 2009. ACM.

[15] S. Mohammad and G. Hirst. Distributional measures as proxies for semantic relatedness. *Submitted for publication*, 2005.

[16] N. Polettini. The vector space model in information retrieval- term weighting problem, 2004.

[17] P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *In Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 448–453, 1995.

[18] Spiteri and F. Louise. *Structure and form of folksonomy tags: The road to the public library catalogue*, volume 4, chapter 2. Webology, 2007.

[19] L. Zhang, X. Wu, and Y. Yu. Emergent semantics from folksonomies: A quantitative study. *Journal on Data Semantics*, VI:168–186, 2006.