Effectiveness of Learning to Rank for finding User Similarity in Social Media

Van Duc Thong Hoang, Thomas Demeester, Chris Develder
Dept. of Information Technology
Ghent University, Ghent, Belgium
{thoang, tdmeeste, cdvelder}@intec.ugent.be

Hyoseop Shin
Advanced Technology Fusion Dept
Konkuk University, Seoul, Korea
hsshin@konkuk.ac.kr

ABSTRACT
This paper focuses on an automatic and accurate approach for finding similar users in social networks. Many types of social networks could benefit from such techniques, but the focus in this paper is on online photo services. The similarity between users needs to be considered on two different levels, i.e., the semantic similarity (or correspondence in tagging behavior), and the similarity in terms of social relations. In recent work, heuristic formulas were introduced for the *tag commonness* (TC) and the *link strength* (LS), with an adaptive combination scheme to describe how relevant each of these similarity aspects are for particular users, in order to define the user similarity. This paper presents an experiment, where a Learning-to-Rank approach is used to find suitable combinations of TC and LS related parameter values, hence taking into account the *proficiency* of users to tag their photos, and their *noticeability* in the online community, in order to obtain an overall user similarity. The user experiments show that the results with this learning-to-rank approach are significantly better than with a former, heuristic, approach.

Categories and Subject Descriptors
D.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *retrieval models, information filtering, search process*.

General Terms
Algorithms, Measurement, Experimentation, Human Factors.

Keywords
User similarity, Tag commonness, Link strength.

1. INTRODUCTION
Recently, the online photo services such as Flickr [3], Picasa, etc… have become one of the major types of social media on the Web. These applications allow users to share their photos with friends, family and other members in the social interaction.

In this paper, we address the problem of finding similar users in photo sharing services, by using a Learning-to-Rank approach to evaluate a number of user characteristics. Finding similar users is a popular application in social media. When a user finds someone's photos interesting, he may want to find more unknown photo owners whose photos are *similar* to those of the given user [2,14]. The application is not necessarily meant for the purpose of finding similar photos, but more generally, to allow for social networking. The similarity between users can be measured on two different levels: the *semantic similarity* and the *link information similarity* of social relations. The semantic similarity can be captured in a textual or image content-based approach. Although the state of the art in content-based image retrieval is progressing, textual annotations such as tags are still considered more effective for capturing the semantics of a photo. Flickr users provide manual annotations to describe their photos, for search purposes. We assume that the user-issued image tags enable users to find images related to a particulars topic or context, and should hence allow finding similar users by comparing their main topics with those of the given user. The other considered aspect in the similarity between users relates to their social relations in the social network. The notion here is that if a visiting user has expressed an interest (by establishing a link) with both user u_a's photos and those from user u_b, then u_a and u_b are probably similar. These links can be conveniently captured in Flickr because each photo is linked to a set of users who pick the photo as a favorite. The link structure can provide additional insight about the relationships among users (e.g., even within the photos of a same topic, a user can express his interest in a particular photo).

The total similarity can be written as a weighted sum of the link-based and tag-based similarity:

$$sim(u_a, u_b) = \omega_{tag} \cdot sim_{tag}(u_a, u_b) + \omega_{link} \cdot sim_{link}(u_a, u_b) \quad (1)$$

with $\omega_{tag} + \omega_{link} = 1$.

In the field of Information Retrieval, a popular technique to estimate the semantic similarity between documents is Cosine similarity [4] in a vector space representation of the documents, based on term frequencies. Alternatively, a number of dimension-reduction techniques are available, such as Latent Semantic Analysis (LSA) [5]. Link-based similarity measures have been used as well for finding related documents. Co-citation was first proposed by Small [6], as a similarity measure between scientific documents, and based on the assumption that authors will only cite documents that are related to their own work. Also with the goal of determining the similarity between documents, David Cohn [7] proposed a probabilistic estimate based on an *aspect model* to evaluate the probability of each pair of documents. However, semantic similarity or link-based similarity on their own, are too limited for comparing between generic web pages as well as between documents of a specific type such as blogs. Therefore, the two approaches have often been combined with different weight factors that could be determined by heuristics or machine learning techniques, in order to improve the performance in many kinds of applications. Cohn and Hoffman [8] combine between PLSA (for semantic) and PHITS (for link) to find the relationship between documents and topics. Instead of applying each model separately, the authors think that it is reasonable to merge the two models into a joint probabilistic model. In a different approach, Fillippo Mencezed [9] combined the relationship of content and link based similarity for a large number of web pages to estimate the semantic similarity. The mentioned contributions suggest that content and link information can indeed be combined so as to obtain a better similarity accuracy.

However, the usual implementation of (1) with fixed weights has a critical drawback. It is based on the assumption that each user uses *common tags* which other users may often use as well, and that each user has *sufficient links* so that the links of the user can be compared with those of other users. In case a user is using only

rare tags or has *insufficient* links, his similarity with other users is no longer fairly evaluated. Therefore, the conventional combination schemes based on the equation above will produce sub-optimal results for the situations described above. It is only effective if both users to be compared use common tags and have *sufficient links* with the other users.

To address this problem, we already proposed an adaptive combination scheme of tag-based similarity and link-based similarity in which the weight factors, ω_{tag} and ω_{link} are dynamically determined for each user separately by evaluating their characteristics such as *tag commonness* (TC) and *link strength* (LS), in order to optimize the precision of the similarity between the users [1]. However, the work from [1] is based on heuristics and hence does not always accurately reflect the users' characteristics. This is the reason why we employ a machine learning technique to measure the users characteristics in social media.

This paper is organized as follows. In section 2, we apply a Learning to Rank method RankNet [2] for the TC and LS for finding the similar users. In section 3, we present the adaptive combination scheme of tag-based and link-based similarity with RankNet. In section 4, we show the experimental results with a collection using Flickr data. Section 5 concludes our work.

2. APPLICATION OF RANKNET FOR SIMILARITY PARAMETERS

This paper is an extension of [1], where the proficiency in tagging behavior of the users (shortly called their *proficiency*), and their noticeability within the social network community (the *noticeability*), are introduced. However, the heuristic approach of [1] displays a number of shortcomings in its ability to describe the actual *proficiency* and *noticeability* as a real test user would. One of the reasons is that the users' characteristics do not fully comply with regulations of human consciousness; different people will have different reactions in the same situation. Moreover, the heuristic function is unable to discover the background knowledge within a subconscious mind of users in the social media. Another reason is that the previous work did not give us an obvious way to optimize the formula for TC and LS. We could not straight away figure out a method to compound the properties in the *proficiency* and *noticeability* in order to match the evaluators' most obvious choice. In [1], we defined the users' characteristics based on their properties. However, it is still unknown which of these properties are more important and how they affect each other.

To address the mentioned problems, we employ the Learning to Rank method RankNet [10] to evaluate the discussed characteristics for real users of the social network, based on a training set from human annotators. The starting point for RankNet is a natural probabilistic cost function for a pair-wise comparison. This cost function is minimized using the commonly known backpropagation algorithm [11], by adjusting the weights of a neural network. Note that other learning techniques could be used as well.

2.1 Tag Commonness

To calculate the RankNet values of TC, representing a user's *proficiency*, we consider a pair of users $[u_i, u_j]$, with respective TC values $\left[TC_{u_i}, TC_{u_j}\right]$, together with the following (trivial) target probability $\overline{P_{TC_{u_i},TC_{u_j}}}$. If user u_i has more *proficiency* than user u_j, we take $\overline{P_{TC_{u_i},TC_{u_j}}} = 1$, whereas it is set to zero if user u_i has less *proficiency* than user u_j. For each user u_i, we now use two TC-related properties $\{p_{TC_1}(u_i),\ p_{TC_2}(u_i)\}$, defined by:

$$p_{TC_1}(u_i) = \log(numCT_{u_i} + 1)$$

$$p_{TC_2}(u_i) = numCT_{u_i} \,/\, \left(numCT_{u_i} + numRT_{u_i}\right),$$

in which $numCT_{u_i}$ and $numRT_{u_i}$ are the number of *common* tags, respectively, *rare* tags of user u_i.

We now construct the RankNet function $F_{TC}^{u_i}$, mapping the parameters $p_{TC_1}(u_i)$ and $p_{TC_2}(u_i)$ onto a real value that directly determines the rank order of the considered user. In other words, $F_{TC}^{u_i} > F_{TC}^{u_j}$ means that user u_i has more *proficiency* in tags than user u_j. The difference $\Delta F_{ij} := \left(F_{TC}^{u_i} - F_{TC}^{u_j}\right)$ can hence be mapped to the probability $P\left(F_{TC}^{u_i} > F_{TC}^{u_j}\right)$, written shortly as P_{ij}, using a logistic function:

$$P_{ij} = e^{\Delta F_{ij}} / \left(1 + e^{\Delta F_{ij}}\right)$$

We can now write the cross entropy C_{ij} between the target probability $\overline{P_{TC_{u_i},TC_{u_j}}}$, shortly written as $\overline{P_{ij}}$, and the modeled posterior probability P_{ij}, as

$$C_{ij} = -\overline{P_{ij}} \log P_{ij} - \left(1 - \overline{P_{ij}}\right) \log(1 - P_{ij})$$

The cross entropy is now applied as the cost function for training the neural network depicted in Figure 1.

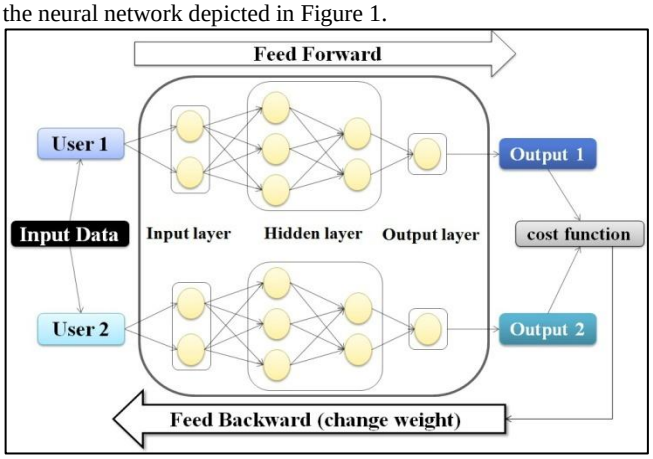


Figure 1. RankNet of tag *commonness* (TC)

Running RankNet in the neural network is based on the principle of backpropagation. This requires two coupled steps: *feed forward* and *feed backward*. The *feed forward* step will provide the output for each user pair. When we run the *feed forward*, each node in the network will be activated and the gradient values are stored by following the backpropagation algorithm. The TC *proficiency* depends on the two parameters introduced above, hence the two input nodes per user in the neural network (see Figure 1). In the second step, the *feed backward* is called to readjust the weight of each connection by calculating the cross entropy between the desired and the calculated probability. Note that we are only considering two hidden layers, both containing two or three nodes, and one output. More hidden layers and/or nodes could however be applied. After training on the human-annotated data, the weights are set so as to calculate the RankNet function, and hence correctly predict the TC *proficiency*, of an unseen user u_k as $F_{TC}^{u_k}$.

2.2 Link Strength

We again apply RankNet, now to calculate the LS-related *noticeability* in a similar way. It is characterized by three properties: the number of links, the weight of each link and the variation of the considered user. For each user u_i, we define the corresponding parameters, respectively, $\{p_{LS_1},\ p_{LS_2}, p_{LS_3}\}$. These are calculated as:

$$p_{LS_1} = n$$

$$p_{LS_2} = n/\left(n + \sum_{i=1}^{n}\left|fb_{u_{i,k}} - \left(\frac{\sum_{i=1}^{n} fb_{u_{i,k}}}{n}\right)\right|\right)$$

$$p_{LS_3} = n/\left(\sum_{i=1}^{n}\frac{1}{fb_{u_{i,k}}}\right)$$

where n is the number of links of user u_i, $fb_{u_{i,k}}$ is the weight of his k-th link of user u_i.

The used neural network is similar to the one shown in Figure 1, except for the input layers, that contain three nodes per user, corresponding to the three considered input parameters.

In a similar manner as in Section 2.1, the output of the neural network's output represents the LS-related noticeability, written for user u_k as $F_{LS}^{u_k}$.

3. ADAPTIVE COMBINATION OF USER SIMILARITY BY USING RANKNET

The RankNet values of TC and LS are now used for computing the weights of the tag-based similarity and the link-based similarity, respectively.

Suppose we have a query user u_p that wants to find similar users. His RankNet values for TC and LS are calculated, and combined as follows to find his overall similarity with respect to a user u_q:

$$sim(u_p, u_q) = \frac{F_{TC,n}^{u_p}}{F_{TC,n}^{u_p} + F_{LS,n}^{u_p}} \cdot sim_{tag}(u_p, u_q) + \frac{F_{LS}^{u_p}}{F_{TC,n}^{u_p} + F_{LS,n}^{u_p}} \cdot sim_{link}(u_p, u_q). \quad (2)$$

The subscript n denotes the fact that the RankNet values are normalized by their maximum value, in order to give a balanced importance to both the TC and LS parameters. Furthermore, $sim_{tag}(u_p, u_q)$ is the tag-based similarity between both users, which is evaluated by the Cosine similarity [4] of the *tf * iuf* (tag frequency, respectively, inverse user frequency of a tag) vector of tags for both users, and $sim_{link}(u_p, u_q)$ is the link-based similarity between both users which is calculated by the Jaccard similarity [15] between the link weight vectors. Further information on the calculation of the separate similarities can be found in [1].

4. EXPERIMENTAL RESULTS

4.1 Data Set

The data set was gathered as follows. We took one day's worth of the "interesting photos from the last 7 days" from Flickr (around 500 photos in total). These were used as the seed photos. For all users that selected one or more of those photos as a favorite, all of their tags, links, and favorite users were recorded, for posts between January and March, 2009. An overview of the crawled data quantities is given in the table below.

Table 1. Data Set Description

no.posts	no.favorites	no.users	no.tags
51,742,309	24,991,762	1,454,042	9,857,175

4.2 TC and LS user studies

Before applying RankNet to measure the users' characteristics, we performed two user studies to collect data for the Ranknet learning process. Tag and link information for each user pair were shown to the evaluators. They then indicated the user with the higher *proficiency* in TC and *noticeability* in LS. In order to decide upon the *proficiency*, the evaluators were given the tag name, total number of tags and *iuf*. The shown link information, for the decision on *noticeability*, included the number of links, the weight of each link and the variation of the concerned user. We performed user studies in two steps: (1) *Generic user study*: we chose 50 random users and made C_2^{50} user pairs for the user study. (2) *Narrow user study*: in order to more delicately evaluate the efficiency of the proposed methods, we tried to choose more competing users. We first sorted all the users in descending order of the TC score and the LS score, as computed by the heuristic function (shortly written HF) from [1]. Then, we chose the 50th-ranked user as the first user and picked every 100th user to make a selection of 50 users.

We invited seven persons as the evaluators, most of them experienced with Flickr data. Table 2 shows the TC and LS user studies with HF results. We mixed the *generic* and the *narrow user study* to accumulate general (combined) results for TC and LS. *Threshold* indicates the minimum number of evaluators with the same answers. *Number of valid pairs* expresses the number of pairs that were selected from those C_2^{50} user pairs, based on each *threshold*. *HF accuracy* indicates an accuracy percentage of the HF method, with respect to the *number of valid pairs*. Table 3 presents the Kappa statistic [12], a common measure for agreement between judges or evaluators. The Kappa statistic appears to be fair to good in the *combined* case for the TC and LS.

Table 2. TC and LS user studies of HF results

Combined	*Threshold*	**4**	**5**	**6**	**7**
TC	Number of valid pairs	2379	2208	1952	1327
	HF accuracy (%)	64.82	65.90	66.65	70.69
LS	Number of valid pairs	2219	2057	1949	1659
	HF accuracy (%)	75.35	75.94	75.68	75.05

Table 3. Kappa statistics of TC and LS in HF

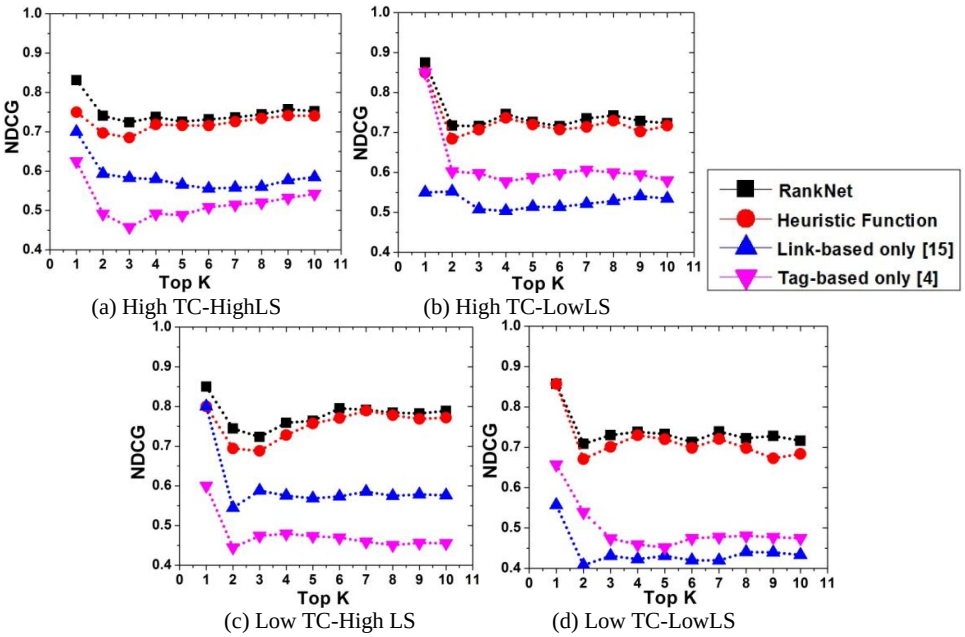
Combined	
TC	***LS***
73.45%	76.2%

4.3 TC and LS with RankNet

The user pairs for the RankNet experiments, were chosen based on threshold 5 (selected arbitrarily). We implemented the TC and LS of RankNet by following the schemes from Section 2. The user pairs were separated into two groups. We randomly selected 60% of the user pairs for training and 40% for evaluation. The weights for the TC and LS neural networks were determined with the training data. After that, the RankNet score for each test user pair was calculated, allowing to judge who had the higher TC, respectively, LS. This procedure was repeated ten times with a new random separation between training and testing data, and the reported precision is the average over these experiments of the faction of correctly predicted pairs. The HF precision is the fraction of correctly predicted pairs, over the user-annotated data set. All experiments were performed 10 times with a new random separation between training and test data, and the results reported in this paper are the average of these experiments. Table 4 summarizes the accuracy of TC and LS in RankNet, as compared to the heuristic function. As expected, the RankNet method yields better results both for TC and LS.

Table 4. Accuracy of TC and LS in RankNet

Threshold 5		*Accuracy*
TC	*RankNet*	74.63%
	Heuristic Function	65.90%
LS	*RankNet*	78.98%
	Heuristic Function	75.94%



(a) High TC-HighLS (b) High TC-LowLS

(c) Low TC-High LS (d) Low TC-LowLS

Figure 2. NDCG results of different user similarity schemes

4.4 User Similarity

In the second user study, we compared the performance of different user similarity schemes: adaptive combination of HF and RankNet, tag-only similarity, and link-only similarity. The last two schemes correspond to the classical cosine similarity and Jaccard similarity, respectively, and can be seen as a baseline to the adaptive combination scheme from [1] and the new adaptive combination scheme with the RankNet coefficients.

To estimate the performance of user similarity, we performed the following user experiments with the new RankNet values. First, we selected query users with different characteristics in TC and LS: 10 users having *high* TC and *high* LS values, 10 users having *high* TC but *low* LS values, 10 users with *low* TC and *high* LS values, and finally 10 users having both *low* TC and *low* LS values. The different schemes were used to generate the top 10 similar users for each query user, which were merged and presented to the evaluators. For evaluation, we use the NDCG measure [13], in order to consider the ranked position, as well as the ratio of the relevant answers among top k-answers recommended by a ranking scheme. Figure 2 shows the NDCG results for the different schemes. Our adaptive combination of RankNet outperforms the other schemes, including HF. One can notice a strong performance shift for the link-based and for the tag-based similarity schemes, according to the user group. For example, the link-based only scheme gives a better performance than the tag-based only scheme in the LowTC – HighLS case, and vice versa for the HighTC – LowLS case. However, the adaptive combination schemes yield a consistent performance in all cases.

5. CONCLUSIONS

This paper employs a Learning to Rank method, RankNet, to evaluate the *proficiency* and *noticeability* of users' characteristics in social media. The similarity between users can be calculated by computing the weight factors in a neural network, both for the *tag commonness*, and the *link strength*. The experimental results show that the new method with the parameters calculated by RankNet and adaptively combined into an overall similarity, outperforms all other schemes, including a recent heuristics-based method.

6. REFERENCES

[1] N.H. Phan, V.D.T Hoang, H. Shin. Adaptive Combination of Tag and Link-based User Similarity in Flickr. ACMM 2010.

[2] H. Kwak, H. Shin, J. Yoon, and S. Moon. Connecting Users with Similar Interests Across Multiple Web Services. ICWSM 2009.

[3] http://www.flickr.com/

[4] http://en.wikipedia.org/wiki/Cosine_similarity

[5] S. Deerwester, S. T. Dumais, G. W. Furnas, Landauer. T. K, and R. Harshman. Indexing by latent semantic analysis. Journal of the American Society for the Information Science, pages 41, 1990.

[6] H. G. Small. Co-citation in the scientific literature: A new measure of relationships between two documents. Jounal of the American Society for Information Science, July 1973.

[7] D. Cohn and H. Chang. Learning to probabilistically identify authoritative documents. In Proceedings of the 17th International Conference on Machine Learning, 2000.

[8] D. Cohn and T. Hofmann. The missing link--a probabilistic model of document content and hypertext connectivity. In Proceedings of Neural Information Processing Systems 13, pages 430-436, Vancouver, British Columbia, 2001.

[9] F. Menczer. Combining Link and Content Analysis to Estimate Semantic Similarity. WWW 2004.

[10] Christopher J. C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, G. N. Hullender. Learning to rank using gradient descent. ICML 2005.

[11] R. Rojas. "Neural Networks – A Systemmatic Introduction". Springer-Verlag, Berlin, New York, 1996.

[12] J. Carletta, Assessing agreement on classification tasks: kappa statistic, Computationals Linguistics 22:249-254, 1996.

[13] K. Jarvelin and J.Kekalainen. Cumulated gain-based evaluation of ir techniques. ACM Trans. Inf.Sust., 20(4):422-466, 2002.

[14] J. Sun, Z. Zhu, Y. Mei. Study on Similar Case Determination of Personalized Recommendation System. Applied Mechanices and Materials 2011.

[15] http://en.wikipedia.org/wiki/Jaccard_index