Social Network Model based on Keyword Categorization

Prantik Bhattacharyya Ankush Garg S. Felix Wu
Department of Computer Science
University of California, Davis
{pbhattacharyya, garg, sfwu}@ucdavis.edu

Abstract

A user profile on an online social network is characterized by its profile entries (keywords). In this paper, we study the relationship between semantic similarity of user keywords and the social network topology. First, we present a 'forest' model to categorize keywords and define the notion of distance between keywords across multiple categorization trees (i.e., a forest). Second, we use the keyword distance to define similarity functions between a pair of users and show how social network topology can be modeled accordingly. Third, we validate our social network topology model, using a simulated social graph, against a real life social graph dataset.

1. Introduction

In real life, individuals become friends when they share common interests or passions. Sociologists have termed this tendency of human beings as 'homophily'. Similarly, on online social networks (OSNs), like Facebook or Orkut, users establish friendships when they discover similar profile characteristics. The growth of LinkedIn, a social networking website, demonstrates the impact of profile information very well. Its purpose is to help people build professional networks and find career development opportunities. Using LinkedIn, employers can look into the profile information of users to search for potential employees. Similarly, it helps employees look for potential employers. We feel that categorizing profile information and correlating it with network topology constitutes an important step towards the study of OSNs.

Social networks has been a widely researched area. Milgram [10] tried to ascertain if people in the society are linked by small chains. He asked people to forward letters to their friends who they thought were likely to know the target person. Thus, people implicitly made decisions based on their view of the geographical location or professional links of their friends and the associated likelihood of successful delivery of the letter. Lattice Model [7] uses geographical distance, a user trait, to model social networks. Models based on interest [11] and hierarchy [8] have also been proposed to model the friendship behavior of people. In Davis Social Links (DSL) [3], the social map is defined on the basis of keywords that are set by social peers as their profile attributes. Information transfer takes place only when a social path exists between the end users. Thus, it seems that keywords are going to play an important role in the development of future OSNs.

A typical user profile on an OSN is characterized by its profile entries (keywords) like location, hometown, activities, interests, music, etc. It is important to understand the use of keywords and how they can be used effectively to classify content in OSNs. Consider the scenario, where a newcomer in the city, say Bob, would like to find people interested in soccer. As he doesn't know anyone yet, he tries his OSN profile to search for soccer enthusiasts in the city but uses the word 'football' for the query. Though, both the words 'soccer' and 'football' refer to the same sport, Bob's query returns no successful results because traditional residents use the word 'soccer' for the game. The system fails to understand the underlying semantic relationship between the keyword entered by Bob and profile entries of other users. This shows the importance of extracting relationship(s) from the diverse information provided by users.

Linguists have long been studying such relationships between words. Methods like Latent Semantic Indexing [4] explored semantics based relationship among digital data. Similarity between users as a function of their topological distance was studied in [2]. In this paper, we study the relationship between semantic similarity of user keywords and the social network topology. First, we present a 'forest' model to categorize keywords and define the notion of distance between keywords across multiple categorization trees (i.e., a forest). Second, we use the keyword distance to define similarity functions between a pair of users and show how social network topology can be modeled accordingly. Third, we validate our social network topology model, using a simulated social graph, against Facebook data.

Section 2 presents our findings on keyword usage patterns and discusses the need of categorizing keywords. Section 3 describes the 'forest' model to categorize keywords. Here, we also propose functions to quantify similarity between users and a social network model based on those functions. Section 4 deals with the methods that we used to evaluate and validate our model. Section 5 presents preliminary results of experiments analyzing the proposed model and realistic data. We conclude in section 6 with possible extensions.

2. Why Categorize Keywords?

A typical profile on any OSN consists of numerous sections (e.g. Orkut has Social, Professional and Personal sections; Facebook has Basic, Education & Work and Personal Information sections) that characterize the user. These sections are further sub-divided into various fields, e.g. Personal section on Facebook has Interests, Activities, Favorite Movies, Books, etc. as fields. We call the entries in these fields as Keyword(s) as they represent user attributes. To understand the keyword usage patterns we analyzed 1265 unique Facebook user profiles [12]. Most of the fields contained proper nouns (e.g. movie names, albums, etc.) as entries, hence, for all evaluation purposes, we restricted ourselves to keywords found in the *Interests* (which contained words mostly from an English dictionary) field.

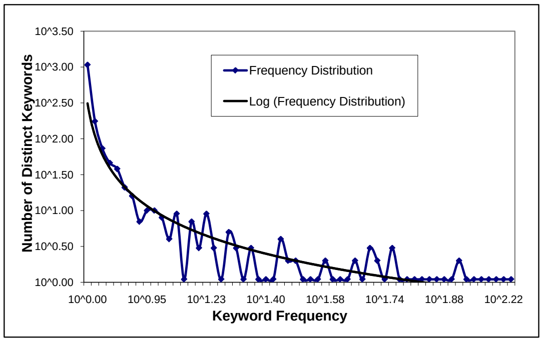

Figure 1. Number of Distinct Keywords vs Keyword Frequency on log-log scale

We looked at the magnitude of information given by users and how keywords can be processed to extract meaningful knowledge. On an average, each user provided 5.8 keywords for the *Interests* field and the keyword set contained 1573 unique keywords. To analyze the distribution of keywords, we plotted the number of distinct keywords for a given keyword frequency on a log-log scale (see figure 1). We divided the keyword frequency in four categories to represent keywords with different frequencies (see figure 2).

The trend line (solid continuous line) for the graph in figure 1 shows an exponential drop in the number of distinct keywords as the keyword frequency increases. The distribution shows consistency with similar results on tag distribution over web applications [13]. It follows the Zipf's Law because the occurrence frequency of a keyword increases as its popularity increases in the frequency list. Thus, we can infer that most of the keywords entered by users are distinct. Figure 2 also substantiates this observation as only 14% of the keywords belong to the high frequency category. This means that only a fraction of keywords are repeatedly used by different users and a large percentage of keywords (44% of the total) occur with very low frequency.

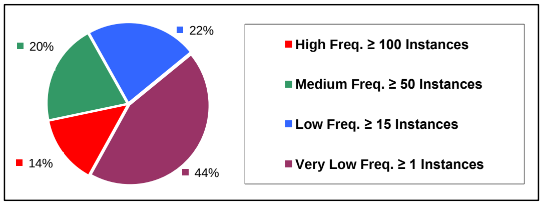

Figure 2. Keyword Frequency Distribution

Two important conclusions can be drawn using the above discussion. First, different topics in which users are interested can be generalized to a small number. This observation uses the fact that there are limited 'community' categories in OSNs. But, the number of unique keywords is large implying that there must be some relationship between these different keywords. Second, it is possible that the very low frequency keywords (which constitute almost half of the total) aren't very dissimilar either with each other or to the other 56% keywords due to the extensive usage scope of English words. Thus, to come with a social network model based on keywords, there is a need to explore the hidden relations among keywords and to categorize them. For instance, in Bob's case, if the OSN could understand the relationship between 'soccer' and 'football' it might give better results for Bob's query.

3. Social Network Modeling

In this section, we first describe a method of categorizing keywords in a data structure to utilize the underlying relationship amongst them. Then, we define functions to quantify the distance between keywords and similarity among social peers based on the distance between their keyword pairs. Finally, we talk about correlating social network topology with keywords using the similarity functions.

3.1. Forest Structure

There is no obvious way of relating words in a dictionary that is based on an alphabetical ordering. A data structure is needed that can help define distance between keywords by capturing hidden relations between them. It must employ methods to clearly distinguish between related and unrelated keywords. A single hierarchical structure (e.g. by a modification of [8]) will be insufficient as it will fail to capture important characteristics of keywords. First, it is not always possible to relate all the words, e.g. 'earthquake' and 'soccer', in a single structure. The distance between such unrelated words must come out to be relatively larger than that between related words. Second, the data structure must capture all meanings of a word as it can be used in different contexts (or in different syntactic categories). E.g., according to WordWeb[1], the word 'stern' could mean 'severe' as an adjective and 'rear part of a ship' as a noun. We propose a forest structure to store keywords where each tree in the forest contains related keywords. As a keyword could have more than one meaning it could occur in different trees. This way, we use multiple hierarchical trees (i.e. a forest) to measure distance between keywords.

Different methods could be used for arranging the keywords in the forest. A method based on etymological relations can be used to construct the forest. For instance, in a language like English, which has been derived from Latin, Greek, etc., most of the words have a root associated with them. Wordinfo[2] lists 61,362 English words which have either Latin or Greek roots. A root word with its derived words can be put in a single tree. E.g. the words 'equine' (horse), 'equestrian' (horse rider), 'equestrienne' (female horse rider) and 'equestrianism' (horsemanship) that come from the Latin root *'equus'* (a horse) can form one tree. All such trees taken together can form the forest.

Another way is to place semantically related keywords in the same tree. E.g. a tree can be made with the keywords related to 'sports'. The next levels could contain various sports like 'football', 'racing', etc., each having its own sub-tree. Similarly, we can have another tree for all the countries under the keyword 'United Nations' (or 'UN') as shown in figure 3. Since, 'soccer' is a hyponym of 'football', Bob's query will get better results as the semantic similarity between the two keywords has been captured by this structure.

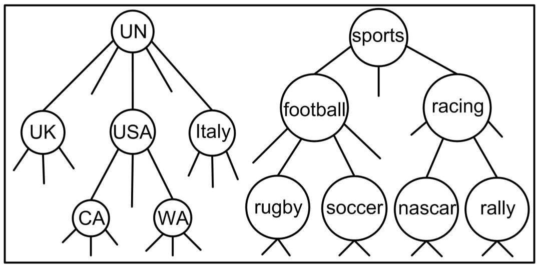

Figure 3. Forest with two component trees

3.2. Similarity Functions

Now we define the notion of distance between keywords based on the forest structure. Let there be t trees $(T_1, T_2, ..., T_t)$ in the forest F. Consider two keywords K_a and K_b such that both of them belong to the same tree. Let LCA be the least common ancestor of K_a and K_b. Also, assume $d(LCA, K_a)$ to be the depth of K_a from the LCA. E.g., in figure 3, if $K_a = soccer$ and $K_b = racing$ then $LCA = sports$ and $d(LCA, K_a) = 2$.

Definition 1. *If K_1 and K_2 are two keywords, then the distance, $D(K_1, K_2)$, between them is given as:*

$$D(K_1,K_2) = \begin{cases} d_{LCA}(K_1,K_2) & \textit{if } K_1, K_2 \in T_i \\ \infty & \textit{if no such } T_i \textit{ exists} \end{cases}$$

where $d_{LCA}(K_1, K_2)$=max$(d(LCA, K_1), d(LCA, K_2))$. If more than one such T_i exists, then the distance is set to the minimum of all the corresponding d_{LCA}'s.

If K_1 and K_2 don't have any relation then $D(K_1, K_2)$ is ∞. Also, minimum of all d_{LCA}'s is used to account for multiple occurrences of keywords in F. These observations justify the benefits of a forest structure (as explained in section 3.1) over a simple hierarchical model to store keywords. A possible metric to define the distance between two keywords could have been the sum of the depths of the keywords from their LCA (i.e. $D(K_1, K_2)$=$d(LCA, K_1) + d(LCA, K_2)$). But, we believe that to capture the distance between keywords from a generic common point (i.e. the LCA), $\max(d(LCA, K_1), d(LCA, K_2))$ is more appropriate as 'max' function gives the farthest distance from the generic point.

Now, we will define the similarity functions between social peers. Assume that a social peer w has N_w keywords and let K_i^w $(1 \leq i \leq N_w)$ be his/her keywords.

[1] WordWeb Software: http://www.wordwebonline.com

[2] Senior Scribe Publications: http://www.wordinfo.info

Consider two peers u and v on the network. Let $k(u,v)$ $(N_u \times N_v)$ be the total number of keyword pairs that they have. Also, let $n(u,v)$ be the number of keyword pairs (K_i^u, K_j^v) such that K_i^u and K_j^v belong to the same tree in F.

Definition 2. *For two social peers u and v on the network, the 'weak similarity', s(u,v), between them is:*

$$s(u,v) \quad = \quad \frac{n(u,v)}{k(u,v)} \tag{1}$$

Definition 3. *For two social peers u and v on the network, the 'strong similarity', S(u,v), between them is:*

$$S(u,v) \quad = \quad \frac{\sum_{1\leq i\leq N_u, 1\leq j\leq N_v} e^{-D(K_i^u,K_j^v)}}{k(u,v)} \tag{2}$$

We call the function s 'weak similarity', as it doesn't take into account the position of keywords in a tree, i.e. keywords with distinct distance values will contribute equally towards the weak similarity. The function S is called 'strong similarity', as it also considers the relative positions of keywords in the forest as keywords with greater distance contribute less towards the similarity value. Exponential function was a natural choice for the definition because it has a finite value at the boundary conditions for $D(K_i^u, K_j^v)$ (as $e^{-0} = 1$ and $e^{-\infty} = 0$). The value of $S(u,v)$ decreases as the distance between the keywords increases implying that u and v share lesser interests or attributes. It may happen that strong similarity is numerically smaller than the weak similarity but still it is a relatively stronger definition as it captures more information. The similarity functions could provide good parameters for performing decentralized search (e.g. [7, 8, 1]) by finding similar friends in OSNs. They may also help in finding potential friends and tackling the link prediction problem (e.g. [9]) in social networks.

3.3. Social Graph

In this section, we formalize steps required to generate the social graph using the similarity functions. Assume that two people 'A' and 'B' share many common interests. It is likely that they know each other as they may have met somewhere because of their common interests. In our model, keywords exactly represent the interests or attributes of a social peer. Hence, if two people share many common keywords (a high value of similarity) then it is likely that they may have a link between them on the social graph. Formally, let the probability that a social peer u is willing to establish a friendship with another social peer v be proportional to $S(u,v)$. We divide $S(u,v)$ by $\sum_v S(u,v)$ (the normalizing constant) to obtain a probability distribution.

Definition 4. *The probability $p(u,v)$ that u is willing to create a friendship with v is given by:*

$$p(u,v) = \frac{S(u,v)}{\sum_v S(u,v)} \tag{3}$$

We used the 'strong similarity' to define $p(u,v)$ as it is a stronger indicator of similarity. Note that $p(u,v) \neq p(v,u)$ even though $S(u,v) = S(v,u)$ because the denominators are different. This captures the fact that both peers might not be equally interested in having a friendship with each other. Let the vertex set V contain a vertex for each peer on the social network. Consider two random independent trials with probability $p(u,v)$ and $p(v,u)$ for the (u,v) peer pair. Join u and v with an edge if both trials yield a positive result i.e. both peers want to establish a friendship. Repeat the above process for all pairs of vertices to get the set of undirected edges E. Then, $G = (V,E)$ is the social network based on keyword similarity among peers. Next, we talk about the techniques that we used to evaluate the effectiveness of this social network model.

4. Evaluation Methodology

Now, we talk about the methods that we used to evaluate the above social network model. We considered two networks and compared the similarity values to observe the effectiveness of the 'forest' structure in correlating profile keywords with network topology (corresponding results are given in section 5). One network represented a realistic scenario (Facebook data as mentioned in section 2) while the other was generated through simulation of our social network model.

4.1. Analyzing the Facebook Network

We used WordNet [5] as the database of English words to build the forest structure. It relates different words by using their sets of cognitive synsets. We used a Java API [6] to look at the meronyms, synonyms, holonyms, hypernyms, hyponyms, derived terms and set of similar words for a keyword. For a user u, we allowed each of its keywords to build its own tree with semantically similar words. The lookup was continued recursively to a depth of three to get trees for all keywords. Thus, we built trees of words which were interlinked by concepts and meanings to form the forest structure (F_u) for a user. Defining a set of general keywords and expanding them to trees would have been a relatively more effective way of capturing the characteristics of keywords. As that would have required help from linguists, we adopted a rather simpler idea to construct the forest and thus, term our results as preliminary.

Once F_u was obtained, we checked if the keywords of other users belonged to the trees of F_u. If any such keyword was found (say for node v) then the value of $n(u, v)$ was incremented. This way we computed $n(u, v)$ values for all possible (u, v) pairs. The number of keyword pairs $k(u, v)$ is given by $N_u \times N_v$. Then, from $n(u, v)$ and $k(u, v)$, we computed the 'weak similarity' $(s(u, v))$ values between all pair of users. The Facebook dataset had more than 1.5 million user pairs of which 7827 pairs where direct friends.

4.2. Analyzing the Simulated Network

In this section, we discuss how we generated a social network graph and determined the 'weak similarity' among its users. The simulated graph had the same number of nodes (1265) as the Facebook data. We assumed a keyword set of 1500 distinct keywords. To mimic a realistic scenario, we used Zipf's distribution to inflate the keyword set to a pool of 6400 keywords such that it contained multiple copies of some randomly chosen keywords. Each user was assigned 6 distinct keywords on an average from the inflated pool.

To generate the forest F, we sub-divided the pool of unique keywords into 185 trees. The trees had different number of keywords and varying depths to simulate real semantic relations of words. Forest F allowed us to compute 'strong similarity' values, $S(u, v)$ for users u and v, across different user pairs. Once $S(u, v)$ values were known for all user pairs, the probability $p(u, v)$ (as defined in Equation 3) was determined. When $p(u, v)$ and $p(v, u)$ both were above a minimum threshold value (θ), an undirected edge (i.e. friendship) was established between nodes u and v. Since $p(u, v) \neq p(v, u)$, an edge was established only when both peers satisfied the minimum interest level, i.e. when $p(u, v) \geq \theta$ and $p(v, u) \geq \theta$. In this way, we got the social network graph, $G = (V, E)$, according to the model presented in section 3.3. Once the social graph G and the forest F were known, we computed the 'weak similarity' values across all pairs of nodes.

5. Results and Discussion

User	Interests
A	wakeboarding, softball, fishing, jesus, god, learning, backpacking
B	running, hiking, hurricanes, tornadoes
C	basketball, dancing, shopping, pictures
Z	running, soccer, tennis, foosball, hiking, knitting, art, tea, lime , pie

Table 1. Sample Users with Keywords

User	k(Z,User)	n(Z,User)	s(Z,User)
A	70	4	0.057
B	40	5	0.125
C	40	8	0.200

Table 2. Weak Similarity with respect to Z

In this section, we present the results obtained from the analysis of the real and simulated networks. First, let us look at four Facebook users with their keywords, for the *Interests* field (table 1). All the users are interested in some type of sporting events. We compare the results of the 'weak similarity' of the first three users with the user Z in table 2. It can be seen that $s(Z, C)$ is maximum even though Z and C have less keyword pairs $(k(Z, C) = 40)$. This is because their profiles match for keywords which can be derived from 'athletic sports' (e.g. pairs formed from basketball, dancing, running, soccer, tennis, etc.). Also both are interested in arts (C has 'pictures' and Z has 'art') implying that Z has more common interests with C than with A or B. A and Z are least similar as A is mostly interested in water sports (and not athletic sports as Z) and doesn't share any other common interest with Z even though they both have a large number of keyword pairs. This shows the effectiveness of characterizing keywords using semantic relationships and that the content of keywords becomes more important than their number for finding similarity values.

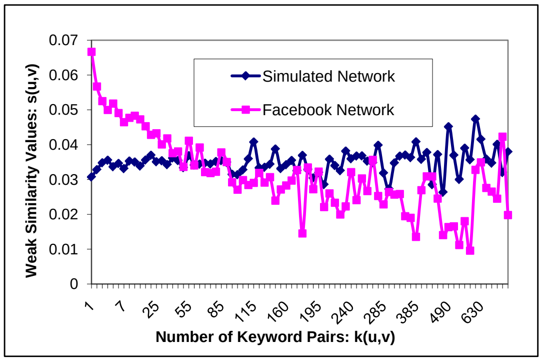

Figure 4. s(u,v) vs k(u,v): All User Pairs

Figure 4 shows the variation of weak similarity, $s(u, v)$, with the number of keyword pairs, $k(u, v)$, for the real and simulated networks across all user pairs. Curves of both the graphs seem to follow a similar pattern, with admissible errors, showing that our social network model is successful in mimicking behavior of the realistic data. Thus, we conclude that our model is an effective way to analyze social network topology.

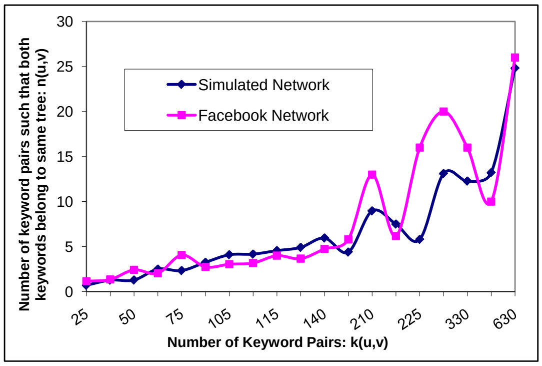

Figure 5. n(u,v) vs k(u,v): Direct Friends

The variation of the number of keyword pairs belonging to the same tree, $n(u, v)$, with different number of keyword pairs, $k(u, v)$, for user pairs that are direct friends is given in figure 5. The curve shows that $n(u, v)$ increases proportionally with $k(u, v)$. This indicates that the weak similarity between user pairs is independent of the number of keyword pairs. From here, we can conclude that no user can successfully alter the similarity value with other users by inflating his profile with unnecessary information. Thus, similarity functions can provide good metrics, which are immune to irrational user activity, to model friendship behavior.

We also analyzed the variation of similarity for friends of friends and the growth of weak similarity with node degree (i.e. number of direct friends) but we omit discussion on these results due to limited space.

6. Conclusion and Future Work

In this paper, we studied the importance of categorizing keywords and defined a 'forest' structure to quantify the similarity between seemingly unrelated user profile information available on OSNs. Based on the similarity functions, we formalized a model of social network topology. We evaluated the effectiveness of our model by simulating and comparing it with a realistic dataset and preliminary results show that our model faithfully emulates the behavior shown by the real OSN. This led us to conclude that the use of keywords is an effective way of modeling and analyzing OSNs. Our model also provides good metrics, immune to irrational user activity, to model friendship behavior.

In future, we would like to explore better methods, based on machine learning techniques, to construct the forest structure. We also intend to gather more real OSN data and analyze that data to form a deeper understanding of the correlation between profile keywords and the social network topology. Also, we would like to study the variation of similarity between user profiles at different topological distances and over different times to address the link prediction problem in OSNs.

Acknowledgments

We are thankful to Matthew Spear who provided us the Facebook data. We also thank the anonymous ASONAM reviewers for their insightful comments. This work was supported by the National Science Foundation grant (award number 0832202).

References

[1] L. Adamic and E. Adar. How to search a social network. *Social Networks*, 27(3):187 – 203, 2005.

[2] L. A. Adamic, O. Buyukkokten, and E. Adar. A social network caught in the web. *First Monday*, 8(6), 2003.

[3] L. Banks, P. Bhattacharyya, M. Spear, and S. F. Wu. Davis social links: Leveraging social networks for future internet communication. In *FIST '09: Workshop on Trust and Security in the Future Internet*, 2009.

[4] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41:391–407, 1990.

[5] C. Fellbaum. *Wordnet: An Electronic Lexical Database*. Bradford Books, 1998.

[6] D. C. Howe. Rita wordnet. Java based API to access Wordnet, available at http://www.rednoise.org/rita.

[7] J. Kleinberg. The small-world phenomenon: An algorithm perspective. In *STOC '00: Proc. of the 32nd annual ACM Symposium on Theory of Computing*, 2000.

[8] J. Kleinberg. Small-world phenomena and the dynamics of information. In *Advances in Neural Information Processing Systems*, pages 431–438. MIT Press, 2001.

[9] D. Liben-Nowell and J. Kleinberg. The link-prediction problem for social networks. *Journal of the American Society for Information Science and Technology*, 58(7):1019–1031, 2007.

[10] S. Milgram. The small world problem. *Psychology Today*, 61:60 – 67, May 1967.

[11] O. Sandberg. *The Structure and Dynamics of Navigable Networks*. PhD thesis, Chalmers University, 2007.

[12] M. Spear, X. Lu, N. S. Matloff, and S. F. Wu. Inter-profile similarity (ips): A method for semantic analysis of online social networks. In *Complex '09: Proc. of the 1st International Conference on Complex Sciences: Theory and Applications*, 2009.

[13] Z. Xu, Y. Fu, J. Mao, and D. Su. Towards the semantic web: Collaborative tag suggestions. In *WWW'06: Proc. of the Collaborative Web Tagging Workshop*, 2006.