Link Prediction in a Bipartite Network using Wikipedia Revision Information

Yang-Jui Chang and Hung-Yu Kao
Department of Computer Science and Information Engineering,
National Cheng Kung University
Tainan, Taiwan, R.O.C.
cjchang@ikmlab.csie.ncku.edu.tw, hykao@mail.ncku.edu.tw

***Abstract*—We consider the problem of link prediction in the bipartite network of Wikipedia. Bipartite stands for an important class in social networks, and many unipartite networks can be reinterpreted as bipartite networks when edges are modeled as vertices, such as co-authorship networks. While bipartite is the special case of general graphs, common link prediction function cannot predict the edge occurrence in bipartite graph without any specialization. In this paper, we formulate an undirected bipartite graph using the history revision information in Wikipedia. We adapt the topological features to the bipartite of Wikipedia, and apply a supervised learning approach to our link prediction formulation of the problem. We also compare the performance of link prediction model with different features.**

***Keywords*-Wikipedia; link prediction; bipartite graph**

I. INTRODUCTION

Given a snapshot of a social network, inferring which new interactions among its members are likely to occur in the near future is formalized as the link prediction problem. The goal of link prediction is to understand which measures of "proximity" in a network lead to most accurate prediction. Common approaches often consider the network as a general graph but not the bipartite.

In this paper we focus on the link prediction for bipartite networks for Wikipedia. Bipartite stands for an important class in networks, which contains edges between two types of entities, for instance item rating graphs, authorship graphs and document-feature networks. While bipartite graphs are the special case of general graphs, the local link prediction methods cannot be generalized to these graphs. Link prediction problem is usually defined on unipartite graphs, and local link prediction functions only depend on the immediate neighborhood of two nodes. In the bipartite network, two nodes belong to different clusters will not have any common neighbors, and therefore the local methods cannot predict the edge occurrence without any specialization.

Many unipartite networks can be reinterpreted as bipartite networks when edges are modeled as vertices, such as co-authorship networks. For these bipartite cases, special link prediction algorithms are necessary. In earlier works, Benchettara et al. [4] have obtained the unimodal graph from bipartite by projecting the graph over one type of its entities. Through this way, there come up two variations of link prediction problem: predicting links in a bipartite graph and predicting links in a unimodal graph. With experiments on the original graph and its unimodal graph, the analysis shows that taking into account the bipartite nature of the graph can enhance substantially the performance of prediction model.

The studies of link prediction problem play an important role as a basic question in social network evolution. Lots of area can benefit from promising interactions or collaborations that have not yet been utilized within its social network. In this paper, we are interested in the domain of edit-network in Wikipedia. Three of the main reasons are considered here.

First, Kittur and Kraut [11] have examined the development of and interactions between coordination and conflict in sample of several wiki production groups. The coordination mechanisms of Wikipedia such as article talk, user talk may be a social benefit to communication between editors in reducing the likelihood of conflict between them. Thus, among the editors of Wikipedia, there might contain some social information (such as co-authorship) which is meaningful and can help improve the performance of prediction models.

Second, for the member-maintained online communities, the social science theory suggests that reducing the cost of contribution will increase members' motivation to participate [3]. With this suggestion, the link prediction approach particularly for Wikipedia is similar to task recommendation. Thus, it can reducing the cost of finding articles that align with editors' interests, we can therefore improve the quality and quantity of the articles on Wikipedia.

Third, over the past decade, Wikipedia has become the largest online collaborative encyclopedia which can be edited by anyone on the Internet, and its entire editing history has been made publicly available as well. Besides, Wikipedia contains various kinds of concepts such as general terms, domain-specific lexicons, named entities which belong to different fields. Among these concepts, there are many kinds of relation, such as redirect, category, disambiguation or internal links for words that are semantically relevant to the context. Such huge scalability and complete link structure has made Wikipedia a valuable resource of research. Thus, we chose it for our edit-network construction.

Based on the above reasons, we constructed the edit graph using the history revisions of Wikipedia, and reformulate it as a link prediction problem, which is expressed as a two-class discrimination problem. We then extracted several features from the edit graph and fed them into machine learning algorithm. The aim of this work is to build a supervised machine learning approach for bipartite link prediction.

978-0-7695-4919-4/12 $26.00 © 2012 IEEE
DOI 10.1109/TAAI.2012.49

The rest of this paper is organized as follows. In Section II, we offer a brief overview on the related research. Then in Section III, we describe the detail of our approach, the system flow chart, and other experimental setup. In Section IV, we will give the description of the dataset and how we obtain the samples, while in Section V we show the results we obtained. Finally in Section VI we discuss conclusions and future work.

II. Related Work

Various link prediction approaches have been proposed in scientific literature. The brief overview on the criteria of link prediction approaches is given as follows:

A. Neighborhood based methods

For a node x, let $\Gamma(x)$ denote the set of neighbors of x in graph G. A number of approaches are based on the idea that two nodes x and y are more likely to form a link in the future if their sets of neighbors $\Gamma(x)$ and $\Gamma(y)$ have large overlap; this follows the natural intuition that if nodes x and y represent authors with many colleagues in common, they are more likely to come into contact themselves. These methods are also classified as triangle-closing models in [15]. The most frequently used neighborhood based attributes are the following:

Common Neighbors. This is the number of neighbors that x and y have in common. Newman [18] has computed this quantity in the context of collaboration networks, verifying a correlation between the number of common neighbors of x and y at time t, and the probability that they will collaborate in the future. This measure is defined as:

$$\text{score}(x, y) = |\Gamma(x) \cap \Gamma(y)| \quad (1)$$

Jaccard's Coefficient. This is a commonly used similarity metrics in information retrieval [20], which measures the probability that both x and y have a feature f, for a randomly selected feature f that either x or y has. If we take features here to be neighbors in G, the measure would be defined as:

$$\text{score}(x, y) = |\Gamma(x) \cap \Gamma(y)| / |\Gamma(x) \cup \Gamma(y)| \quad (2)$$

Adamic/Adar. Adamic and Adar [1] proposed that friendship between two persons can be predicted by measuring their similarity to each other. The simple similarity can be measured by shared items, which are weighted towards the uniqueness, i.e. the item shared only by these two people (and not by other) are more valuable than the item which is shared among many people. Therefore the measure is defined as follows:

$$\text{score}(x, y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{\log|\Gamma(z)|} \quad (3)$$

Preferential Attachment. The preferential attachment has received considerable attention as a model of the growth model of networks [17]. The basic idea is that new edge has a probability to be incident on a node is proportional to the current neighborhood's size of that node. Barabasi et al. [2] have verified this idea through the empirical analysis. This measure is defined as:

$$\text{score}(x, y) = |\Gamma(x)| \cdot |\Gamma(y)| \quad (4)$$

All the above neighborhood based methods, except the preferential attachment model, are not applicable to bipartite graphs because they make some assumptions based on triangle-closing model: *triangle closing* and c*lustering*. The former indicates that new edges tend to form triangles and the latter indicates that nodes tend to form well-connected clusters in the graph.

In bipartite graphs these assumptions are not true, since triangles and larger cliques cannot appear. Because of the fact that two vertices would be connected (from different clusters) do not have any common neighbor, methods based on common neighbors will not perform well.

B. Distance based methods

A number of methods refine the notion of shortest-path distance by implicitly considering the ensemble of all paths between two nodes.

Shortest Distance. This is the shortest distance between two nodes. It is a basic approach that ranks node-pairs <x, y> by the length of their shortest path in G. Such a measure follows the idea that collaboration networks are "small worlds," in which members are related through short chain [19].

Katz. Another distance based measure frequently used in affiliation analyses between nodes in a graph, is the measure proposed in [10]: it consist in computing a weighted sum of all paths between x and y.

In bipartite, although the distance based measure of two vertices from different clusters can be computed with edges that cross the cluster, this still cannot hold all the cases especially the unreachable side in the node-pair.

C. Methods based on Random Walk

These approaches often predict edge occurrence using random walk. Random walk is a Markov chain describing the sequence of nodes visited by a random walker [21]. This process can be described by a transition probability matrix P.

D. Methods based on Similarity

Commonly, two nodes are more likely to be connected if they are more similar, where a latent assumption is that the link itself indicates a similarity between the two endpoints and this similarity can be transferred through the links.

E. Methods based on temporal information

The number of approaches based on temporal information is relatively small. In other words, the link established at different time point is considered having the same effect by most approaches. Recently, the temporal issue is taken into consideration by several works. Examples are works proposed in [22].

F. Higher-level approaches

A number of methods can be used in conjunction with any of the methods discussed above.

Unseen Bigrams. Link prediction is similar to the problem of estimating frequencies for unseen bigrams in language modeling: pairs of words that co-occur in a test corpus, but not in the corresponding training corpus (e.g. [5]). Following the ideas proposed in [14].

Clustering. The performance of a predictor might improve through a clustering procedure, which means running the predictor on a "cleaned-up" graph after deleting some "tenuous" edges in G. In the clustering procedure, one can first compute the score(u, v) for all edges in G, and then delete the ρ fraction of these edges with the lowest score. Once the graph has been cleaned, the score(x, y) for the remaining edges can be recomputed; in this way the similarity of node-pair is determined using only the edges that give more confidence through the considered measure.

III. Approach

For predicting whether an edge is likely to form between the particular node-pair in the future, we build a classifier and train it with the features extracted from the past snapshot of the edit-network, and evaluate the trained models on the testing samples in the time period following the training period. Let G_{obs} be the observed graph that summarizes in some way the temporal sequence $G = \langle G_{t1}, \ldots, G_{tn} \rangle$, and the G_{tn+1} is referred as the labeling graph. As in many other works, G_{obs} is computed as the union of all snapshots in the sequence G. Two examples will be generated for each couple of nodes $<x, y>$ such that x and y belong to both G_{obs} and G_{tn+1}. Let the time sequence be $\{t_1, t_2, \ldots, t_n, t_{n+1}\}$, we partition the data into two sub-ranges. The training samples, which are the observed snapshots of the graph, are obtained from the first sub-range, $[t_1, t_n]$, and the testing samples are from the second sub-range $[t_n, t_{n+1}]$. The positive examples are the editor-article pairs that did not have an edge between them in t_1, but had an edge by t_n, meaning that the editor edited that particular article during this time frame. The negative examples are those that did not have an edge between the pair both in t_1 and t_n, representing that the editor did not edit that article. We train our models with samples from $[t_1, t_n]$. Then, we make predictions with our trained models on editor-article pairs in t_n. Finally we evaluate our predictions by examining t_{n+1}. We choose SVM to be our classifier. Our supervised learning approach is summarized in Figure 1.

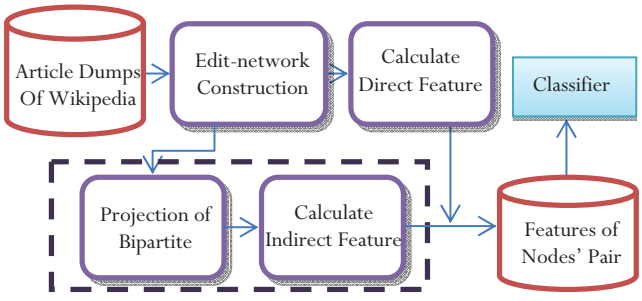


Figure 1. Flowchart of our approach

A. Edit-network construction

For the edit-network construction, we used the dump for some articles on Wikipedia as of 2010-01-01. The crawling of the titles of articles started from the category: Religious_objects, and around 27000 titles were sampled. We considered the Wikipedia as an undirected bipartite graph, where articles and editors are nodes in the graph, and an edge between a particular editor-article pair represents that an editor had edited that article at some time point in the past. To predict the edge occurrence between an editor-article pair is similar to decide whether a particular article is a good candidate to recommend to some editors.

The formal definition of the edit-graph is as follows, G is the bipartite graph, with two sets of nodes, E and A, and a set of edges L, where an edge exists between some $e \in E$ and $a \in A$ if editor e has edited article a at some time point in the past.

B. Features extraction

The main component of our approach is to come up with a list of features that we believe are informative to feed into the machine learning algorithms. Since the edit-graph is a bipartite, we adapted the measures mentioned above, which are commonly used in unipartite graphs. First we adopted the following notation. For a node x, we define $\Gamma(x)$ to be the set of x's neighbors, and $\Gamma'(x)=\bigcap_{y\in\Gamma(x)}\Gamma(y)$, which is the set of x's neighbors' neighbors.

a) Direct feature

We define the following measures as direct features based on the equations mentioned above, since they can be derived from the original bipartite graph directly without any transformation to the graph:

SN (*Sum of Neighbors*). For an article, this is the number of editors that edited it. For an editor, it is the number of articles he/she has edited. Therefore each pair will have two features, one for editor, one for article. The sum of neighbors might be meaningful because the more articles an editor edited, the more likely he will edit more articles because it suggests that he is more active. On the other hand, if an article is edited by many editors, it might indicate that it is a popular, controversial, or a complicated topic. We denote both as SN.

CN (*Common Neighbors*). For an editor-article pair $<e, a>$, the common neighbors is defined to be $|\Gamma(e) \cap \Gamma'(a)|$. This feature is adapted to be the intersection of the articles that editor e edited, and the articles that edited by who edited the article a. This basically captures the idea of "people who edited this article also edited …" We denote this feature as CN.

JC (*Jaccard's Coefficient*). Similar to the common neighbors, this feature measures the similarity between two sets. In the bipartite graph, this is defined as $|\Gamma(e) \cap \Gamma'(a)|/|\Gamma(e) \cup \Gamma'(a)|$. It is the normalized version of common neighbors, which should be more informative. We denote this feature as JC.

AA (*Adamic/Adar*). This feature uses the frequency of common features to compute the similarity between two nodes. In the bipartite graph, the feature is the neighbors, and this feature is defined as $\sum_{z\in\Gamma(e)\cap\Gamma'(a)} 1/\log|\Gamma(z)|$. This feature is used to capture the notion: in the intersection of articles, if the number of editors that edited the particular article is smaller than it of other article, it means to some extent that this article is much more unique to this editor than others. We denote this feature as AA.

PA (*Preferential Attachment*). This is similar to the sum of neighbors measure above and suggests how active the editor is and how popular the article is. Taking the scalar multiplication of the two features can quantify the neighborhood's size. It is defined as $|\Gamma(e)|\times|\Gamma(a)|$ and we denote it as PA.

SD (*Shortest Distance*). This is the minimum hop count between an editor and an article. We hypothesize that the shorter the distance between an editor and an article, the more likely the editor will edit the article. We denote this feature as SD.

All the above features, except Sum of Neighbors, can be categorized into the set of topological features, which characterizes the roles of nodes in the unipartite network. For the bipartite graph, the meaning of topological features would be slightly different from it for the unipartite. The link in a unipartite graph is between the nodes of the same type, while the link in a bipartite graph is between the nodes from different clusters.

b) Indirect feature

The following approach we have applied follows the same general idea of works presented in [4]. We formulate the link prediction as a supervised learning problem. The goal is to discriminate between linked class (positive examples) against not-linked class (negative examples.) A bipartite graph is defined as follows: $G = <X, Y, E>$ where X and Y are two mutually exclusive sets of nodes. E is a set of edges of G and is a subset of $X \times Y$. A unimodal graph can be obtained from a bipartite graph by projecting the graph over one of its nodes' sets. For example, the projection over the X set is defined by a unimodal graph where nodes from X are tied if they are linked to at least n common nodes in the initial bipartite graph G. In a more formal way, let $\Gamma_g(x)$ bet the set of neighbors of node x in a graph g. Projections of a bipartite graph G are then defined as follows:

$$G_X^n = \langle V_X \subseteq X,\ E = \{(a,b): a,b \in X, |\Gamma_G(a) \cap \Gamma_G(b)| \geq n\}\rangle \quad (5)$$

$$G_Y^m = \langle V_Y \subseteq Y,\ E = \{(a,b): a,b \in Y, |\Gamma_G(a) \cap \Gamma_G(b)| \geq m\}\rangle \quad (6)$$

With these unimodal graphs, we can define the indirect features. Let $f_G(e\in E, a\in A)$ be a direct feature. In comparison with the direct features, the indirect features cannot be computed from the original bipartite graph. With the unimodal graph, we can introduce the following two indirect features:

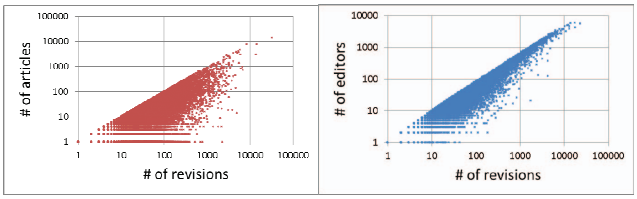
$$\varphi_{u\in\Gamma_G(a)} f^i_{G_E^n}(e,u) \quad (7)$$

$$\varphi_{v\in\Gamma_G(e)} f^i_{G_A^m}(v,a) \quad (8)$$

These two features are built from f and computed in the projected graph G_X and G_Y respectively. Without loss of generality, let us consider the first indirect feature. It computes f between e and all neighbors of a in G_E. An aggregate function φ selects the most appropriate value from the set {min, max}, depending on the feature. For example, if the feature is the number of shared neighbors, φ will be the max function. If the feature is the shortest distance, φ will be the min. function. Notice that both projection parameter n and m can take different values.

IV. Network Analysis

Before we proceed on the link prediction task, we want to have an understanding of the general characteristics of the graph. In these target categories, 27,125 articles were used, and we kept all the revisions for all the articles in these categories. For each revision, there would be a corresponding editor who contributed to that article. There are 791,759 distinct editors in these categories. By the time of year 2010, there are total 1,789,227 links in the target categories. The degree distribution of the dataset follows a power law, as show in Figure 2.

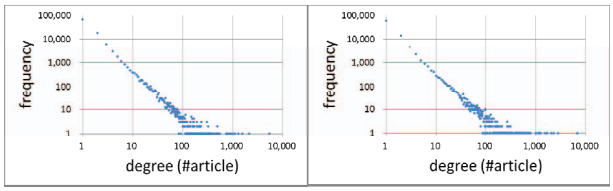


(a) x-y plot of editors (b) x-y plot of articles

Figure 2. Number of revisions of editors and articles

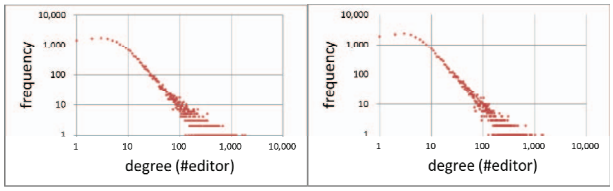
As we would expect only a few number of articles would have many edits and a few number of editors would be the power-users. The distribution of this dataset is similar to the analysis by Voss [23]. In this graph, the editors that only edited few revisions of articles are usually the vandalizers or the anonymous users identified by IP-address. On the other hand, the editors with the highest number of contributions and articles are usually the bots in Wikipedia, which are the automated or semi-automated tool that carry out repetitive and mundane tasks in order to maintain the large amount of works in Wikipedia. Among these "active" editors, few editors are some really aggressive editors that wrote the article, some of them are even the administrators. The editors with the higher number of revisions but less articles are the users that with narrow focus, who might be some subject matter experts or just the incidental users who are interested in some specified topics only. In the previous research, Iba et al. [8] have analyzing the edit behavior of these Wikipedians, and showed that only a few editors who make most of the edits amid the millions of active Wikipedia editors. Through the first look, we know that there are lots of editors that are "noise" in the set of editors, though the number of editors is much higher than the number of articles. If we examine the set of articles, we will get a similar classification of articles.

Since we are performing link prediction problem, we are interested in what the distribution of the number of newly added edges to graph. Figure 3 shows the distribution of newly edges between the year of 2006 and 2007.



(a) Link distribution of editors before 2007

(b) Newly-added link distribution of editors from 2007 to 2008



(c) Link distribution of articles before 2007

(d) Newly-added link distribution of articles from 2007 to 2008

Figure 3. Degree distribution of editors and articles

To obtain the samples for training and testing, we chose the time sequence to be $\{t_1, \ldots, t_n, t_{n+1}\} = \{t_1, t_2, t_3\} = \{2006, 2007, 2008\}$. The training period is [2006, 2007] and testing period is [2007, 2008].We chose these times because the numbers of edge additions in both periods are comparable, and we chose the range of one year as it reduces possible variance if the range is too short.

For the selection of articles, we kept the articles which were created before year 2006, and the last modified time of it was after year 2007. Each revision of articles is accompanied with editors; hence we kept the editors who have edited the selected articles. As addition, since there are many editors who are incidental users that only contributed one revision to one or few articles, a threshold is needed to filter out these noises. We define the active editors to be the editors that contributed to at least k articles and kept these active editors in our edit-graph. The value k will filter out editors with lower number of revisions. We set k to 2.

After the selection of nodes, we chose edges based on the following conditions. For positive training samples, we randomly select an edge in 2007 snapshot of graph and make sure both editor and article exist in the 2006 graph. If both exist in the 2006 graph, this pair will be added to the set of positive training samples. Then we lock on the editor node, and randomly select an article in the 2006 graph. If there is no edge between that pair in 2007, we add that pair to the set of negative samples. We decided to lock on the editor node for choosing samples for the following reasons:

Since our goal is link prediction, we would like to know whether our approach can actually distinguish good candidate articles from worse candidate articles for the same editor. Thus, we keep a positive example as well as a negative example for each chosen editor.

The testing set was also obtained by the same method. Using this method for choosing editor-article pairs, we computed the aforementioned features on all the pairs for the training and testing sets. We obtained 460569 samples in training set, and 227722 samples in the testing set. Both sets are balanced with 50% positive and 50% negative samples.

V. Experiment Evaluation

We use three metrics that are commonly used: precision, recall and F1-value. F1-value is the aggregation of precision and recall, which considers both. For the link prediction of Wikipedia, since the number of articles in Wikipedia is quite large, we can pay more attention on the precision measure. We randomly divide all the training samples into five groups, each are balanced with half positive and half negative. Then we evaluate each group with testing set. Since the results of five groups are similar and we focus on the pros and cons of the features chosen, we take the average to be the precision score of prediction model. The results are showed in Table 1.

Table 1. Results (Neighborhood = {CN+JC+AA+PA})

Features used	Accuracy	Class	Precision	Recall	F-value
SN only	70.60%	0	0.68	0.75	0.71
		1	**0.73**	0.65	0.69
SN+SD+ Neighborhood	60.77%	0	0.57	0.81	0.67
		1	0.68	0.40	0.50
SN+ Neighborhood	66.91%	0	0.65	0.72	0.68
		1	0.69	0.61	0.65
SN+ Indirect CN (n=5, m=6)	68.55%	0	0.69	0.66	0.67
		1	0.67	0.71	0.69
SN+ Indirect CN (n=5, m=3)	65.43%	0	0.71	0.51	0.59
		1	0.62	**0.79**	0.69

The results give us a sense of, with different features' combination, how well the models can predict whether links will form in the Wikipedia edit graph. Since we are more interested in recommending articles to authors than in filtering unwanted articles for one specific editor, we will concentrate on the prediction of the positive samples. Therefore, the more suitable measures to compare are precision and recall for positive classes. The F-value is a harmonic mean of precision and recall, so it takes both values into account. Note that the Neighborhood is the set of feature: Common Neighbor (CN), Jaccard's Coefficient (JC), Adamic/Adar (AA) and Preferential Attachment (PA).

We can see that with the Sum of Neighbors (SN), the model is able to achieve the highest precision of 0.73 among other combinations of features. However, when the Shortest Distance (SD) and Neighborhood are taken into account, the precision is reduced to 0.68. If we excluded the SD, the precision will recover to 0.69. When SD is taken into account, the recall of positive class is reduced to 0.40, which is quite low. The reason that SD did not perform well might be the category of articles dumps is not broad enough. Since all the articles are in the same category, it is easy that one can link an editor to any article that he/she has never written before. Thus, using the SD can result in lots of false positive samples. However, if we take the weighted distance, the results might be different.

It is interesting that if we combine the indirect Common Neighbor with SN, the recall of positive class can achieve the highest 0.79. In the projection of authors, the indirect

common neighbor stands for the number of co-authors. The reason why other features do not work well might be the characteristics of different classes of editors. We divide the result into three classes, Bot (BOT), IP Address (IP) and Registered User (REG). The results are showed in Table 2. We can see that the F-value of other features can achieve a better performance for REG class. BOT are programmed and the anonymous users seldom contribute to articles. Therefore, we can focus on the editors of REG class and it shows that the more informative features such as JC and AA can still lead us to a better results.

Table 2. F1-measure of different editor-classes

	BOT	IP	REG
SN	0.6540	0.7878	0.6921
SN+JC	0.6593	0.7876	0.6927
SN+AA	0.6442	0.7768	0.6940

VI. DISCUSSION

Our observation shows that node with high degree can still be an important factor to the link prediction model. It is important to note that we only evaluated these algorithms on a small period of time and on a specific category. The supervised link prediction model is based on the amount of history information for some editors or articles. Thus, we would like to know what kind of information on the bipartite network can improve the performance significantly in the future. In that way, we can predict the edge occurrence even for nodes with low or average degree. Furthermore, we want to examine whether the different communities exist in Wikipedia and what is the scale of these communities. Since the community can be a core part of social network, studying the characteristics of different communities might improve the prediction model.

ACKNOWLEDGEMENT

This research was supported by the "Advanced Sensing Platform and Green Energy Application Technology Project" of the Institute for Information Industry which is subsidized by the Ministry of Economy Affairs of the Republic of China.

REFERENCES

[1] L. A. Adamic and E. Adar, "Friends and neighbors on the Web," *Social Networks,* vol. 25, pp. 211 - 230, 2003.

[2] A. L. Barabasi, H. Jeong, Z. Neda, E. Ravasz, A. Schubert, and T. Vicsek, "Evolution of the social network of scientific collaborations," *Physica A: Statistical Mechanics and its Applications,* vol. 311, pp. 590-614, 2002.

[3] G. Beenen, K. Ling, X. Wang, K. Chang, D. Frankowski, P. Resnick, and R. E. Kraut, "Using social psychology to motivate contributions to online communities," in *Proceedings of the 2004 ACM conference on Computer supported cooperative work* Chicago, Illinois, USA: ACM, 2004, pp. 212-221.

[4] N. Benchettara, R. Kanawati, and C. Rouveirol, "Supervised Machine Learning Applied to Link Prediction in Bipartite Social Networks," in *Proceedings of the 2010 International Conference on Advances in Social Networks Analysis and Mining*: IEEE Computer Society, 2010, pp. 326-330.

[5] U. Essen and V. Steinbiss, "Cooccurrence smoothing for stochastic language modeling," in *Proceedings of the 1992 IEEE international conference on Acoustics, speech and signal processing - Volume 1* San Francisco, California: IEEE Computer Society, 1992, pp. 161-164.

[6] M. A. Hasan, V. Chaoji, S. Salem, and M. Zaki, "Link Prediction Using Supervised Learning," in *In Proc. of SDM 06 workshop on Link Analysis, Counterterrorism and Security*, 2006.

[7] Z. Huang, X. Li, and H. Chen, "Link prediction approach to collaborative filtering," in *Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries* Denver, CO, USA: ACM, 2005, pp. 141-142.

[8] T. Iba, K. Nemoto, B. Peters, and P. A. Gloor, "Analyzing the Creative Editing Behavior of Wikipedia Editors: Through Dynamic Social Network Analysis," *Procedia - Social and Behavioral Sciences,* vol. 2, pp. 6441 - 6456, 2010.

[9] J. Kamps and M. Koolen, "Is Wikipedia link structure different?," in *Proceedings of the Second ACM International Conference on Web Search and Data Mining* Barcelona, Spain: ACM, 2009, pp. 232-241.

[10] L. Katz, "A new status index derived from sociometric analysis," *Psychometrika,* vol. 18, pp. 39-43, March 1953.

[11] A. Kittur and R. E. Kraut, "Beyond Wikipedia: coordination and conflict in online production groups," in *Proceedings of the 2010 ACM conference on Computer supported cooperative work* Savannah, Georgia, USA: ACM, 2010, pp. 215-224.

[12] J. M. Kleinberg, "The small-world phenomenon: an algorithm perspective.," in *STOC*: ACM, 2000, pp. 163-170.

[13] J. Kunegis, E. W. D. Luca, and S. Albayrak, "The link prediction problem in bipartite networks," in Proceedings of the Computational intelligence for knowledge-based systems design, and 13th international conference on Information processing and management of uncertainty Dortmund, Germany: Springer-Verlag, 2010, pp. 380-389.

[14] L. Lee, "Measures of distributional similarity," in *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics* College Park, Maryland: Association for Computational Linguistics, 1999, pp. 25-32.

[15] J. Leskovec, L. Backstrom, R. Kumar, and A. Tomkins, "Microscopic evolution of social networks," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining* Las Vegas, Nevada, USA: ACM, 2008, pp. 462-470.

[16] D. Liben-Nowell and J. Kleinberg, "The link prediction problem for social networks," in *Proceedings of the twelfth international conference on Information and knowledge management* New Orleans, LA, USA: ACM, 2003, pp. 556-559.

[17] M. Mitzenmacher, "A Brief History of Generative Models for Power Law and Lognormal Distributions " *Internet Mathematics,* vol. 1, pp. 226-251, 2004.

[18] M. E. J. Newman, "Clustering and preferential attachment in growing networks," *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics),* vol. 64, p. 025102, August 2001.

[19] M. E. J. Newman, "The structure of scientific collaboration networks," *Proceedings of the National Academy of Sciences of the United States of America,* vol. 98, pp. 404--409, 2001.

[20] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*: McGraw-Hill, Inc., 1986.

[21] J. G. K. J. L. Snell, "Finite Markov Chains," Springer-Verlag, 1976.

[22] T. Tylenda, R. Angelova, and S. Bedathur, "Towards time-aware link prediction in evolving social networks," in *Proceedings of the 3rd Workshop on Social Network Mining and Analysis* Paris, France: ACM, 2009, pp. 1-10.

[23] J. Voss, "Measuring Wikipedia," 2005.