Pervasive Web Community Structure Summarization: A Machine Learning Approach

Xun Luo, Robert Kenyon
Department of Computer Science
University of Illinois at Chicago
851 S. Morgan (M/C 152)
Chicago, IL 60607-7053, USA
{xluo, rkenyon}@cs.uic.edu

Yun Guan
Department of Mathematics, Statics
and Computer Science
University of Illinois at Chicago
851 S. Morgan (M/C 152)
Chicago, IL 60607-7053, USA
yguan3@uic.edu

Abstract

Although community discovery based on social network has been studied extensively in the Web hyperlink environment, limited research has been done in the case of Web documents. The co-occurrence of Words and entities in sentences and documents usually implies some connections among them. Studying such connections may reveal important relationships. In this paper, we investigate the co-occurrences of named entities in Web pages and blogs, and mine communities among those entities. We show that identifying communities in such an environment can be transformed into a graph clustering problem. A hierarchical clustering algorithm is then proposed, which exploits triangle structures within the graph and the mutual information between vertices. Our empirical study shows that the proposed algorithm is promising in discovering communities from Web documents.

Introduction

Knowledge discovery in social networks has attracted much attention due to its successful application in Web search engines. PageRank (Brin and Page 1998) and HITS (Kleinberg 1998) are two best known Web page ranking algorithms. Both algorithms regard a collection of Web pages as a social network. Each Web page is an entity in the social network, and a hyperlink connecting two Web pages is a relationship between the entities. By exploiting the Web link structure within the social network; these methods can evaluate the importance of individual Web pages.

In addition to its contribution to Web page ranking, HITS discovered that there exist multiple Web communities among relevant Web pages when the query term has several meanings. A Web community can be defined as a group of Web pages that are more closely linked to the peers in the same group than those outside of the group. For instance, for query term "Jaguar", there are three major Web communities, respectively on the Atari video game, the American Football team, and the automobile model. By modeling the social network with a weighted graph, Web communities can be discovered based on the graph topology; the resulting Web communities are usually clusters of Web pages with the same themes.

Going beyond the hyperlinked Web environment, we believe that the concept of community also exists in Web pages and blogs. Blogs are Web pages that consist of journal entries with creation time stamps. Different from Traditional publication process, blogs involve minimal costand censorship. Moreover, many blog sites provide free RSS feeds to their subscribers, so the audience can conveniently access the blog update. These features contribute to blogs' rapidly growing popularity.

Although blogs are different from traditional media in many ways, they have proven their influence in many events. One of those examples is Howard Dean's successful integration of political blog with his campaign during the 2004 presidential primaries. Many blogs are concerned with current events, politics, and responses from bloggers. Therefore, mining knowledge from blogs is of great significance to business marketing, politics, and journalism.

Our interest in this work is community discovery from Web document contents, rather than Web graph, which was used to mine the communities of Web pages. There are many named entity terms, i.e. names of persons, organizations, and locations, in Web documents. When a blogger or Web page author put multiple named entities in the same context, we believe that there are relationships among those named entities. The context, in which the named entities appear, can be considered as implicit links connecting them.

In this paper, we make an attempt to discover communities from Web documents by utilizing named entity terms. Our research is motivated by two major factors.

First, named entity terms are of high interest in Web and blog search. Search engines help average users to retrieve the relevant information from the vast document collection for given string queries. While query strings can vary from a product model to a scientific concept depending on users' interests, named entity terms are among the most frequently searched terms on the Web. This fact could be testified by the search term statistics provided by the main search engines. Among Yahoo's

top 10 search terms in 2004[2], all of the ten search terms are named entity terms: four persons, five sports organizations, and one TV show name. For those frequently searched entities, users' interests are often diverse. For instance, when users search "Tom Cruise" on a search engine, they might be interested in different information, including his acting career, his dating life, or his religious belief. Named entity communities can enhance the retrieval result by targeting the communities of interest.

Secondly, named entities are natural actors according to the definition of social networks. The original concept of social network was proposed by social scientists to quantify the social relationships among people and organizations. In this sense, our focus on named entity communities agrees well with the initial motivation of the social network research. As a consequence, the technique of mining named entity communities can be used to discover social network related knowledge on the Web, when combined with other Natural Language Processing techniques.

The objective of this work is to find communities that an entity is involved in from a collection of Web documents. We believe that an entity is defined largely by its communities. Mining such communities can provide users a simple profile of the given entity. The user can then find further information regarding the entity by navigating other entities in those communities.

Although many community mining algorithms exist (see the next section), we cannot use them for our purpose because they are all partitional algorithms, i.e., they do not allow the same entity to appear in multiple communities. However, that is exactly our purpose, i.e., finding multiple communities that an entity is involved in.

Given a set of Web documents, our technique works as follows: We first convert the Web documents to free text. We then cut each document into sentences, and process them with a named entity parser. Each sentence containing more than one named entity is extracted. To convert the sentences into an undirected graph/network, we map each named entity to a vertex, and each sentence containing more than one entity to an edge. An edge is also weighted by the co-occurrence frequency, which is then changed to mutual information for clustering to mine communities.

To cluster the weighted graph into communities, we use a bottom up (agglomerative) clustering algorithm. Our algorithm combines both graph triangle link structure and mutual information between vertices. Our experiment shows that the algorithm is very effective to identify named entity communities from Web pages and blogs.

Related Work

The work on community structure discovery in social networks on the Web first appeared in the HITS algorithm (Kleinberg 1998; Gibson, Kleinberg, and Raghavan 1998). Since then the issue of community discovery has been studied in a variety of environments. However, we are not aware of any previous work on extracting communities from named entities in Web documents at the time of paper submission.

In the HITS algorithm, an iterative algorithm was proposed to evaluate an authority weight and a hub weight for each page in a collection of related Web pages. After the calculation converges, the Web pages with top authority scores are authority pages, and those with top hub scores are hub pages. The HITS algorithm models each Web page collection with a directed graph, and captures the link weights with a matrix A. The entry (i, j) of A represents the link strength from page i to page j. AA^T may have multiset of eigenvalues. Well-separated eigenvalues often denote the existence of multiple Web communities. For the Web pages retrieved for query "Jaguar", there are three major Web communities as we mentioned in the Introduction section.

Different from HITS, (Eckmann and Moses 2002) approached the Web community issue in a Web graph from a local perspective. The authors introduced the concept of "curvature" for each vertex v to measure how well connected v's neighborhood is. The curvature is a ratio of the actual number of triangles that v participates in over the number of triangles that v could participate in. In a complete graph, the curvature value is 1 for each vertex. Lower curvature indicates higher probability that there are multiple communities at the vertex's neighborhood. Moreover, the authors have made an observation that community expands predominantly by triangles sharing a common side. The same observation was also made in (Toyoda and Kitsuregawa 2001).

(Adamic and Glance 2005) studied the linkage behaviors of liberal and conservative blog communities during 2004 presidential election. The communities are manually labeled according to the blogger's self-identification. Their work is different from ours, as they did not intend to discover communities from blog contents.

In addition to the Web community issue, other works studied the community structure in other cases. (Girvan and Newman 2002) identified research communities from collaboration of researchers. (Dorow and Widdows 2003) borrowed the community concept, and applied it in the Word Sense Disambiguation problem. The "community" in their work consists of words having similar senses. The link analysis on which Web community research is based also has applications in other research fields, such as customer network mining (Domingos and Richardson 2001), influence maximization (Kempe, Kleinberg, and Tardos 2003), anomaly detection in social networks (Noble and Cook 2003), and text summarization (Erkan and Radev 2004). However, these works are not on community finding but on other aspects of social networks, although they also use link analysis.

In addition to community research, another related research focused on extracting binary relations from the Web. In (Brin 1998), the author used several book/author pairs to search on the Web, and found certain format patterns from the Web pages retrieved, which are then

---

[2] http://tools.search.yahoo.com/top2004/

used iteratively to find more book/author pairs from only several seeds.

A phenomenon that we exploit in this paper is the named entity co-occurrence. Term co-occurrence has been used for many purposes (Jing and Tzoukermann 1999). However, they do not find communities.

Framework for Community Discovery

The objective of our work is to discover named entity communities from Web documents. There are two major tasks. The first one is to acquire the weighted graph consisting of named entities. Thereafter, we cluster the named entity graph into communities in the second task.

Named Entity Graph Generation

First, we need a collection of documents regarding a certain named entity. We choose persons in this work. To evaluate our technique, we used two separate document sources: blogs and Web pages. To collect blog documents, we issue query string with person names to the Google Blog Search engine[3]. We choose English as the language, and the blog publication dates range from April through September 2005. To ensure that the returned results are relevant, another query requirement was that the blog titles must include the query string. We manually collected approximately 200 blogs for each month, and merge them together as the test blog documents. For the Web documents, we issue queries of person names to Google search engine[4]. We choose English as the language, and .html and .htm as the desired file formats. We retrieve all top 500 Web page links, and download the HTML files. After stripping off the HTML tags from the files, we obtain the free text documents.

Second, we extract all the sentences containing more than one entity's name, which is a person name in our case. To recognize named entity terms from the documents, we use MINIPAR (Lin 1994) as the named entity parser. Since our focus is in community discovery, we do not deal with coreference issue, which could increase the co-occurrences of person names. From all the sentences parsed by MINIPAR, only those sentences containing more than one name are collected. We then map the selected sentences into a weighted undirected graph. The vertices in this graph are person names, and the edges are co-occurrences of names. For each sentence containing two names, we add an edge between two vertices. If such an edge already exists, we increment the edge weight. Therefore, the weight in this graph is the raw co-occurrence frequency; it will be converted into mutual information later for our clustering purpose.

In the text documents, many of those person names are only first names or last names. Simple heuristics are used to convert single names to full names. They are matched based on information in the document set. If the full name is mentioned somewhere in a document and a first name and/or a last name can match the full name, they are assumed to be the same person. We will keep the single names if they cannot be matched to any full name.

One major problem with named entities is the name resolution or disambiguation, i.e., determining whether the same name refers to different real-world persons. We have not focused on this problem in our current work. The entity disambiguation has not been a major issue since our target applications are mainly political figures and celebrities, who do not interact with a large number of people. Thus, name conflicts are limited. In fact, our proposed method can help solve the name disambiguation problem. Two persons with the same name are not likely to be in the same communities (if they are, there are usually other ways to distinguish them in the text). Our algorithm can separate them due to the fact that they interact with different groups of people.

Clustering Graph into Communities

The object of this step is to generate communities from the named entity graph by graph clustering. So far a variety of techniques on graph clustering have been proposed. Note that PageRank is not suitable for our problem because it is for ranking Web pages according to their importance. In spite of mathematical cleanness, there are few works that use HITS to discover communities. It is computationally expensive. It is also very difficult to set the eigenvalue threshold to select the communities. If the granularity is coarse, all communities will merge into one. If the granularity is too fine, the algorithm is very sensitive to noise and outliers. In (Flake, Tarjan, and Tsioutsiouliklis 2003), Flake et al transformed the graph clustering problem into the "max flow-min cut" problem. By adding a virtual source vertex s and a sink vertex t, the algorithm locates the minimum cut set in the graph. After the cut, the vertices are still connected to the source belong to the same community. Their experiments show that the "maximum flow-min cut" based algorithm delivers good results on Web graph clustering.

(Girvan and Newman 2002) proposed a graph clustering algorithm based on the "edge betweenness" of an edge in a graph. The "betweenness" of an edge is defined as the number of shortest paths between all pairs of vertices that run along the edge. The edges connecting communities have higher "betweenness" because those edges bridge communities together. By removing the edge with the highest "betweenness", the graph will be partitioned.

However, there are two problems with the applicability of "max flow-min cut" and "edge betweenness" algorithms in our named entity graph clustering. The first issue is that both algorithms are "hard clustering" methods; there is no overlapping between resulting clusters. Each vertex can belong to only one community. This assumption may work for Web communities because those communities are more distributed. Obviously, this is not the case in our named entity graph, in which a person normally participates in multiple communities. The second problem is that these algorithms need the user to specify the number of the

---

[3] http://blogsearch.google.com/
[4] http://www.google.com/apis/

clusters, which is hard in practice. Our algorithm does not need this information because it terminates when all the triangles are picked up by cores (this will be clear below).

To remedy these problems, we propose a hierarchical clustering algorithm with *cores* (to be defined later). Vertices are allowed to have membership in more than one community. Let us introduce some definitions below.

Edge weight: In a graph $G = (V, E)$, each edge in the graph connects two vertices, i.e. names in our case. Compared to the absolute co-occurrence frequencies, mutual information (Dumais et al. 1998), which quantifies how strong two vertices depend on each other, is a better option to measure the edge weight.

The following is the equation for calculating the mutual information between vertices a and b. $p(a, b)$ is the co-occurrence probability of (a, b). If the total co-occurrences in the graph is N, and there are n co-occurrences of a and b, then $p(a, b) = n/N$. Similarly, $f(v)$, $v \in V$, is the incidence probability of vertex v. if all the vertex incidences in the graph is M, and vertex v has m incidences, $f(v) = m/M$. From the equation, we can tell that mutual information between two vertices can be negative. It happens when the actual co-occurrence frequency of two vertices is lower than the expected co-occurrence frequency when they are totally independent. A negative mutual information value of two entities indicates they are *negatively* related.

$$I(a,b) = p(a,b) \times \log_2 \frac{p(a,b)}{f(a)f(b)}$$

We give a brief discussion on the concept of mutual information because of its importance in our algorithm. The mutual information concept originates from Shannon Entropy (Shannon 1948). It measures the dependence of two variables. In our community research, the mutual information reflects the closeness of two entities. If several entities are highly related, and share high mutual information, they can create a strong community. Moreover, the entities with higher mutual information should be grouped together earlier than those with lower mutual information. That is the reason that we use mutual information as the measure for our clustering algorithm. Additionally, after the clusters are formed, we can also use mutual information to evaluate the final clustering results. Strong communities should have high mutual information among the vertices.

Triangle: In a graph $G = (V, E)$, V is a set of vertices, and E is a set of edges among V. For vertices $a \in V$, $b \in V$, $c \in V$, if edges $(a, b) \in E$, $(a, c) \in E$, and $(b, c) \in E$, we say vertices a, b, c form a triangle. The *cohesion* of a triangle is measured by the weight of its weakest edge, i.e. the minimum mutual information value among the three edges.

Our interest in triangle components comes from the observation in (Eckmann and Moses 2002): a community expands predominantly by triangles sharing a common edge. Since its focus was on the relationship between "curvature" value and the existence of communities, clustering technique was not proposed in the paper. The

triangle geometry, which consists of three binary transitive relations, indicates a strong connectivity between the vertices. In fact, a triangle is a complete graph by itself, and it can also be viewed as a building block of any complete graphs with more than three vertices. Theoretically, complete subgraphs, i.e. cliques, are of high interest to us if we want to discover the community cores. However, identifying cliques from a graph is a NP-complete problem (Cormen, Leiserson, and Rivest 1990). Instead, we can approach the graph clustering problem starting from its triangle components.

Similarity between Triangles: If two triangles share an edge, we assign the sum of the cohesion value of two triangles as their similarity. If two triangles do not share any edge, we assign their similarity to 0. Two triangles are *fully linked* if merging the two triangles by the common edge can generate a complete graph of four vertices.

There are two assumptions for our algorithm:

1. Each community has a core, which is composed by strongly linked triangles. We do not exclude the possibility that such strongly linked group of triangles do not exist. If so, individual triangles can be community cores by themselves.

2. The community propagates mainly through triangle pairs sharing common edges.Each triangle belongs to one and only one community. On the other hand, each vertex and edge could belong to more than one community.

We believe that our assumptions are reasonable as they reflect real world communities quite well.

The detailed procedures of our clustering algorithm are as follows:

Input: A weighted undirected graph stored in adjacency list.

Output: A set of named entity communities extracted from the graph.

1. Triangle Extraction. We extract all triangles from the graph adjacency list. Triangles with a negative edge weight are filtered out because not all of its vertices are positively related (which are our interest). To reduce the effect of outliers, we set another requirement. For a triangle each edge needs to have two or more incidences.

2. Similarity Calculation. From the triangles extracted from step 1, we calculate the similarity for all the triangle pairs. Sort the triangle pairs according to their similarity in descending order.
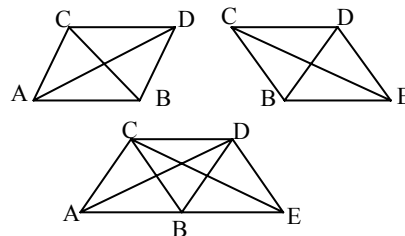
3. Pre-clustering. We first extract all fully linked triangle pairs. In the diagram *ABC/BCD* and *BCD/BDE* are such examples. Then, we process these fully linked

triangle pairs. During this process we have to conserve the validity of assumption 2. If any triangle pairs have a common triangle, we merge them together into a cluster.

　　Again, as *ABC/BCD* and *BCD/BDE* share the same triangle *BCD*, we merge them to create a new cluster. For a triangle pair that does not share any triangle with others, it is still a cluster by itself. All these clusters formed in this step are labeled as *cores*.

4. Triangle Clustering. We traverse the sorted triangle pairs. For a triangle pair *A* and *B*, we try to merge the triangle pairs together. If both *A* and *B* were already pre-clustered in the same core, we do nothing. If only one of them, say, *A* was pre-clustered to a core, we put *B* into the same core. If both of them were not pre-clustered, we merge them together, but the merged cluster is not labeled as core. If both *A* and *B* were already pre-clustered into different cores, we do nothing. The final results of this step are clusters of triangles.

5. Post-Clustering. For those vertices that do not belong to any triangles, we calculate their similarity with all the clusters. The similarity between a vertex and a cluster is calculated by summing up the edge weights between the vertex and all the vertices of the cluster. The vertex is assigned to the cluster having the largest similarity with it.

　　In summary, step 2 uses the triangle pair similarity to set the clustering priority. Step 3 creates the cores for the prospective clusters. Step 4 is the triangle clustering process; in this step we follow our 2 assumptions. Step 5 adds those vertices not in any triangles into the clusters. Clearly, our algorithm uses automatically formed cores as seeds and does not need the number of clusters to be specified by the user.

Empirical Study

Before presenting our experiment results, we comment on the test documents briefly.

　　Our document collection comes from two sources: Google Web search and Google Blog Search. First, we used "Tom Cruise" as the blog query because of his popularity among bloggers. Moreover, he drew much media attention during the period that we carried out this project. The blog documents were collected manually because most of blog sites have yet to provide API for users to access their collections.

　　In addition, another set of documents is retrieved through Google Web search engine. Our selection is based on two reasons. First, we expect Web documents have a broad coverage on a given person entity. The second reason is that we assume that the retrieved top-ranking documents should be of high quality because there are millions of pages relevant to our query entities. In fact, we were somehow surprised by the diversity of retrieved documents. Especially for the political figures, the Web pages include a wide range of topics, from formal ones, such as biography and speech, to informal ones, such as political jokes and harsh critics. From another perspective, the diversity of the documents is also a good test for our algorithm's robustness. Clearly, we should also note that our results are constrained by the pages (our data) found by Google.

　　We applied our algorithm to blogs on one person and Web pages on four persons. Since community is a relatively abstract concept, it is difficult to evaluate the algorithm effectiveness with objective measures. In our experiment, all persons are well-known entities. Readers can inspect the clustering results and make a judgment on the effectiveness of our algorithm.

　　Although there are many existing community mining algorithms, we cannot compare with them because they are all partitional algorithms, i.e., they do not allow the same entity to appear in multiple communities. However, that is exactly the purpose of our work, i.e., finding multiple communities that one entity is involved in.

　　Table 1 shows our experiment results. Column 1 gives the name of each entity that we search. Column 2 shows the ID of each community. Column 3 lists the central entities for each community. Obviously, the entities in a community have different importance. We measure an entity's importance by summing up the mutual information between the entity and its peers within the same community. The entities in column 3 are sorted in descending order of their importance scores. If a community has more than six central entities, we only list top six of them.

　　As we discussed earlier, each Web community usually shares the same theme. This is also the case for entity communities. To summarize the theme of a community, we extracted nouns from sentences that correspond to the links of a community. The nouns are sorted in descending order according to term frequency. The top nouns are listed in column 4. In addition to the automatically extracted nouns, we also manually added some remarks on the discovered communities in column 5. Some events that the communities involved might not be familiar to many people, so we give a list of brief descriptions or events related to the communities. The major news events on "Tom Cruise" and the occurrence dates are manually extracted from news media, and listed in Table 2.

　　Let us look at the communities of "Tom Cruise" first. We can tell T1 and T2 are relatively strong communities. T1's community members contain all the persons involved in the relevant events and other scientology figures. For T2's community, the people who previously had relationship with Tom Cruise and Katie Holmes are also clustered in this group. Both of them contain very relevant persons, and their sizes are relatively large. From another perspective, the summary terms also reflect the community quality. The summary terms for a strong community are quite focused.

　　To our surprise, not many bloggers paid attention to Tom Cruise's movie, "War of Worlds", released in June. As a consequence, T3 was a weak community. Similarly, T4 was also relevant, but a weak community.

Table 1. The Discovered Named Entity Communities

Entities	ID	Entities in Each Community	Summary Terms	Remarks
Tom Cruise	T1	Tom Cruise, Katie Holmes, Brooke Shields, Matt Lauer, Oprah Winfrey, Ron Hubbard.	scientology, love, depression, interview, today, actress, people.	Tom Cruise's on scientology and psychiatry. E2, E3, E4, E8.
	T2	Katie Holmes, Tom Cruise, Nicole Kidman, Penelope Cruz, Chris Klein, Mimi Rogers.	actress, actor, engagement, love, years, news, relationship, scientology.	Tom and Katie's dating life. E1, E7.
	T3	Steven Spielberg, Tom Cruise, Jerry Maguire.	world, director, war, movie, film, spiegel, year.	Tom Cruise's movie. E5.
	T4	Jessica Rodriguez, Katie Holmes, L. Ron Hubbard.	interview, member, question, actress, scientology, arrangement, rumor.	Katie's conversion to Scientology. E6.
Bill Clinton	B1	Bill Clinton, George Bush, John Kerry, Hillary Clinton, John Edwards, R. Reagan.	president, campaign, state, 1992, election, time, senator.	Political community
	B2	Bill Clinton, Hillary Clinton, Paula Jones, Kenneth Starr, Monica Lewinsky.	president, lewinsky, affair, relationship, case, investigation.	The scandal
Hillary Clinton	H1	Hillary Clinton, Bill Clinton, George W. Bush, Al Gore, Monica Lewinsky.	president, time, book, arkansas, senator, husband, woman.	Political community
	H2	Hillary Clinton, John Kerry, George W. Bush, Rudy Giuliani, John Edwards.	president, 2008, democrat, candidate, campaign.	Potential 2008 presidential candidates
	H3	Hugh Rodham, Dorothy Rodham, Hillary Clinton.	illinois, chicago, daughter, mother, family, aggression.	Family
Dick Cheney	D1	Dick Cheney, George W. Bush, John Kerry, Saddam Hussein, John Edwards, Colin Powell.	president, administration, iraq, war, campaign, halliburton, senator, defense.	Political community
	D2	Dick Cheney, Secretary of Defense, Gerald Ford, Donald Rumsfeld, John Ashcroft.	president, secretary, administration, 1975, presidency, defense, military, campaign.	Experience on national security
	D3	Dick Cheney, Lynne Cheney, Mary Cheney, Liz Cheney.	daughter, wife, campaign, child, president, issue, family.	Family
Michael Jordan	M1	Michael Jordan, Nike Jordan, Air Jordan	line, jordan, shoe, product, ad, basketball, company, nba.	Shoe products
	M2	Phil Jackson, Michael Jordan, Scottie Pippen	coach, team, 8, title, years.	Chicago bulls
	M3	Larry Bird, Magic Johnson, Michael Jordan, Bill Russell, Wilt Chamberlain.	player, jordan, basketball, nba, season, star.	Great basketball players

Table 2. The Related News Events with Discovered Communities

ID	Event	Approximate Report Date
E1	Tom Cruise began dating Katie Holmes.	April 26, 2005
E2	Tom Cruise appeared on The Oprah Winfrey Show.	May 23, 2005
E3	Tom Cruise criticized Brooke Shields for using anti-depressant Paxil during an interview with Billy Bush.	May 25, 2005
E4	Tom Cruise talked about Scientology and psychiatry with Matt Lauer.	June 24, 2005
E5	Tom Cruise's movie "War of Worlds" was released	June 29, 2005
E6	Jessica Rodriguez tutored Katie Holmes on scientology.	July 9, 2005
E7	Tom Cruise and Katie Holmes plan to wed.	Sept 12, 2005
E8	Brooke Shields talked about fighting depression on Oprah Show.	Sept 27, 2005

For the communities of "Bill Clinton", we can tell B1 and B2 are relatively strong communities. Both of them also contain very relevant persons, and their sizes are relatively large.

The "Hillary Clinton" communities further confirmed our observation. H1, H2, and H3 have focused summary terms. H3 is a weaker community due to little attention on her childhood family. The "Dick Cheney" and "Michael Jordan" data also have good communities. In the cluster M1, "Nike Jordan" and "Air Jordan" are mislabeled as person names by MINIPAR. In spite of that, the summary terms indicate this is a shoe product community.

After going through all the communities, we can evaluate them fairly quickly because we have more or less knowledge about these topics. Even so, there are some unexpected communities. For example, H2 is one of them. It is a strong community, and it lists many potential candidates for 2008 presidential election (refer to summary terms). The identification of these interesting members indicates that our clustering algorithm is effective to discover communities. In a more broad sense, extracting communities from Web documents can help people achieve knowledge discovery.

Conclusion

This paper studied the problem of mining communities from Web pages and blogs. By exploiting the named entity co-occurrence, we mapped Web documents into a named entity graph. Moreover, we proposed an effective hierarchical clustering algorithm, which utilizes both the triangle geometry inside a graph and the mutual information between vertices. The community quality is evaluated by the summary terms. Our experimental result shows that the clustering algorithm can effectively discover interesting communities. Based on the evaluation, we believe that the technique can enhance our ability to acquire knowledge from Web pages and blogs.

References

Adamic, L. and Glance, N. 2005. The Political Blogosphere and the 2004 U.S. Election: Divided They Blog. In *LinkKDD-2005*. Chicago, IL, USA.

Brin, S. 1998. Extracting Patterns and Relations from the World Wide Web. In *WebDB'98*, 172-183. Valencia, Spain.

Brin, S. and Page, L. 1998. The anatomy of a large-scale hypertext Web search engine. In *WWW7*, 107-117. Brisbane, Australia.

Cormen, T. H., Leiserson, C. E., and Rivest, R. L. 1990. *Introduction to Algorithms*. MIT Press.

Domingos P. and Richardson, M. 2001. Mining the network value of customers. In *KDD-01*, 57-66. San Francisco, CA, USA.

Dorow, B. and Widdows, D. 2003. Discovering corpus-specific word senses. In *EACL*, 79-82. Budapest, Hungary.

Dumais, S., Platt, J., Heckerman, D., and Sahami., M. 1998. Inductive learning algorithms and representations for text categorization. In *CIKM-98,* 148-155. Bethesda, Maryland, USA.

Eckmann, J. and Moses, E. 2002. Curvature of co-links uncovers hidden thematic layers in the World Wide Web. In *Proceedings of the National Academy of Sciences,* 99:5825-5829.

Erkan, G. and Radev, D. 2004. Lexrank: Graph-based centrality as salience in text summarization. In *Journal of Artificial Intelligence Research,* 22:457-479.

Flake, G., Tarjan, R., and Tsioutsiouliklis, K. 2003. Graph Clustering and Minimum Cut Trees. In *Internet Mathematics*.

Girvan, M. and Newman, M. E. J. 2002. Community structure in social and biological network. In *Proceedings of the Nat. Academy of Sciences,* 7821-7826.

Gibson, D., Kleinberg, J., and Raghavan, P. 1998. Inferring Web communities from link topology. In *Hypertext-98,* 225-234. Pittsburgh, PA, USA.

Kempe, D., Kleinberg, J., and Tardos É. 2003. Maximizing the spread of influence through a social network. In *KDD-03*, 137-146. Washington, DC, USA.

Kleinberg, J. 1998. Authoritative sources in a hyperlinked environment. In *Journal of the ACM*, 46(5): 604 - 632.

Jing, H. and Tzoukermann, E. 1999. Information retrieval based on context distance and morphology. In *SIGIR-99*, 90-96. Berkeley, CA, USA.

Lin, D. 1994. PRINCIPAR-An Efficient, broad-coverage, principle-based parser. In *COLING-94*, 482-488. Kyoto, Japan.

Noble, C. and Cook, D. 2003. Graph-based anomaly detection. In *KDD-03*, 631-636. Washington, DC, USA.

Shannon, E. 1948. A mathematical theory of communication, In *Bell System Technical Journal*, 27: 379-423, July, 1948.

Toyoda, M. and Kitsuregawa, M. 2001. Creating a Web community chart for navigating related communities. In *Hypertext-01*, 103-112. Århus, Denmark.