Discovering Overlapping Communities of Named Entities

Xin Li[1], Bing Liu[1], and Philip S. Yu[2]

[1] Department of Computer Science, University of Illinois at Chicago, 851 S. Morgan Street, Chicago, IL 60607-7053
{xli3, liub}@cs.uic.edu
2 IBM T.J. Watson Research Center, 19 Skyline Drive, Hawthorne, NY 10532
psyu@us.ibm.com

Abstract. Although community discovery based on social network analysis has been studied extensively in the Web hyperlink environment, limited research has been done in the case of named entities in text documents. The co-occurrence of entities in documents usually implies some connections among them. Investigating such connections can reveal important patterns. In this paper, we mine communities among named entities in Web documents and text corpus. Most existing works on community discovery generate a partition of the entity network, assuming each entity belongs to one community. However, in the scenario of named entities, an entity may participate in several communities. For example, a person is in the communities of his/her family, colleagues, and friends. In this paper, we propose a novel technique to mine overlapping communities of named entities. This technique is based on triangle formation, expansion, and clustering with content similarity. Our experimental results show that the proposed technique is highly effective.

Keywords: Community of named entities, community mining.

1 Introduction

Knowledge discovery in social networks has attracted a great deal of attention due to its successful application in Web search engines. PageRank [2] and HITS [9] are two representative Web page ranking algorithms. Both algorithms regard each Web page as an entity in the social network, and each hyperlink is a relationship between the entities. In addition, HITS discovered that there exist multiple Web communities among relevant pages when the query term has several meanings.

Going beyond the hyperlinked Web environment, we believe that communities also exist among named entities in text documents. In the Web, there are explicit links connecting entities. However, such links do not exist in free text documents. In this work, we consider that named entities are implicitly linked if they co-occur in the same sentence.

Our objective in this work is to discover overlapping communities of named entities, i.e. the names of persons, organizations, from Web contents and text documents. Our research is motivated by two major factors.

1. Named entity terms are among the most frequently searched terms on the Web. Based on a report from Yahoo! in 2005[1], all the top ten search terms are named entities. For those frequently searched entities, users' interests can be diverse. By finding the overlapping communities, we can separate the various facets about the entity of interest.
2. Named entities are natural actors according to the definition of social networks [13]. The original concept of social network was proposed to study social relationships among people and organizations. By automating data analysis from vast volume of texts, we can analyze social network at a grand scale.

Although many community-mining algorithms exist, we are unable to use them for our purpose because they are mainly partitioning algorithms [7][12] that do not allow the same entity to appear in multiple communities. In contrast, an entity belongs to multiple communities in most of realistic social networks.

Given a named entity, our algorithm works as follows. It first collects a set of relevant documents on the named entity. All the entities co-occurring in the sentences are linked together to generate a named entity graph. We also keep the contextual information, which are noun terms in the co-occurrence sentences. The algorithm then identifies community cores, and clusters those fringe members into the cores.

2 Related Work

The work on community structure discovery on the Web first appeared in the HITS algorithm [9]. Since then the issue of community discovery has been studied in a variety of environments. However, we are not aware of any work on extracting communities of named entities from text documents at the time of paper submission.

[7] proposed a Web community mining algorithm based on Max flow-Min cut. In [12], a partitioning algorithm was also proposed, so does [3] but in the email context.

In [5], the authors studied the community issue in a graph from a local perspective. They introduced the concept of "curvature" for each vertex *v* to measure how well connected *v*'s neighborhood is. The authors made an observation that a community expands mainly by triangles sharing a common edge. The same observation was also made in [12].

In addition to the Web community issue, other works studied the community structure from other aspects. [4] applied the concept of community in the Word Sense Disambiguation problem. Link analysis also has other applications, such as group membership detection [10] and text summarization [6].

Another related research focused on extracting binary relations from the Web. In [1], the author designed an algorithm to find a large number of book/author pairs from only several seeds. [8] extracts relations of named entities from a large text corpus. It groups relations of entities according to their text similarity. The work was not concerned with communities because similar relations do not mean that the entities involved are in the same community.

[1] http://tools.search.yahoo.com/top2005/

3 Problem Definition

This section defines communities, and the objective of this work.

Definition (Community): Given a finite set of entities $S = \{s_1, s_2, \ldots, s_n\}$, a community is a pair $C = (T, G)$, where T is a *theme* and $G \subseteq S$ is a subset of S that shares the theme T. If $s_i \in G$, s_i is called a *member* of the community C. If $C = \emptyset$, C is an *empty community*.

A theme defines a community. Given a theme T, the set of members of the community is uniquely determined. Thus, two communities are equal if they have the same theme. A theme can have a variety of forms: it can be an event or a concept.

An element s_i in S can be in any number of communities, i.e. multiple communities may share members. We denote that an entity associates with a set of themes by s_i:$\{T_1, T_2, \ldots, T_m\}$, where T_k is a theme of community C_k, to which entity s_i belongs.

Given a data set, which can be a set of Web pages, emails, or text documents, usually there is no metadata regarding community available. The system needs to discover the hidden community structure from the linkage among entities. The forms that communities manifest themselves may vary.

Web pages. Web page authors sharing common interests often cite others' pages through hyperlinks. Members in a Web community are more likely to be linked with their peers than pages outside the community. The text from those community member pages can be used to extract the community theme.

Emails. Members of a community are more likely to communicate with one another. The email contents of the community provide a good summary of the community theme.

Text documents. Named entities within a community are more likely to appear together in the same sentence. The words in those co-occurrence sentences reflect community themes.

The key form of community manifestation is that its members are "linked" to each other in some sense. Such links indicate that they share a common theme. Given a data set containing named entities, our objective in this work is to discover the hidden communities of the named entities, and identify the community themes.

4 Mining Overlapping Named Entity Communities

There are two main tasks in discovering named entity communities from documents. The first one is to acquire named entity relationships; the second task groups named entities into different communities based on their relationships and the text contents.

4.1 Finding Entity Relationships

Given a named entity, the system first searches the Web, blogs or a document collection to find those relevant documents. It then uses a named entity parser MINIPAR [11] to tag the named entities in sentences. Furthermore, each sentence that contains at least two named entities of same type is extracted. All entities in a

sentence are considered to be connected pair-wise with an edge.

After all documents are processed, a set of distinctive edges is produced. We attach a *strength* to each edge, which is computed using mutual information. In our case, the mutual information reflects the closeness of two entities. Let the entities of an edge be *a* and *b*, and Pr(*a*, *b*) be the co-occurrence probability of (*a*, *b*). If the total co-occurrences of all edges is *N*, and there are *n* co-occurrences of *a* and *b*, then Pr(*a*, *b*) = *n*/*N*. Let *f*(*a*) and *f*(*b*) be the probabilities of occurrences of *a* and *b* respectively in the edges. The mutual information is defined as follows:

$$I(a,b) = \Pr(a,b)\log_2 \frac{\Pr(a,b)}{f(a)f(b)} \qquad (1)$$

4.2 Mining Communities

Our community-mining algorithm is a core-periphery clustering algorithm. First, we find all cohesive community cores based on the graph topology. After the formation of community cores, we exploit the content information of the relationships within each community core, and group the peripheral entities with the community cores to obtain the final communities.

The basic building blocks of community cores in our algorithm are triangles. A triangle is a complete graph itself, and is a component of larger complete graphs. It was observed in [5][12] that a community expands predominantly by triangles sharing a common edge.

Our community-mining algorithm consists of three major steps.

Finding Triangles. A triangle is formed by edges connecting three entities. It is defined as follows:

Triangle: In a graph $G = (V, E)$, V is a set of vertices, and E is a set of edges among V. For vertices $a, b, c \in V$, if edges $(a, b) \in E$, $(a, c) \in E$, and $(b, c) \in E$, we say vertices a, b, c form a triangle if each edge has at least τ instances and a positive mutual information. τ is a parameter.

Finding Community Cores. An ideal community core is a complete subgraph, i.e., a clique, consisting of a set of vertices such that each pair of vertices is directly connected by an edge. However, this definition is too strong for practical use because the data may be incomplete. We thus relax this definition and give an operational definition.

Community core: Two candidate cores c_1 and c_2 are merged to form a larger community core if there are at least one triangle from c_1 and one triangle from c_2 that share a common edge and form a complete graph of 4 vertices. The resulting community core satisfies the criterion that each vertex in the core is adjacent to three or more other vertices within the same core. We illustrate the definition with the following example. Fig. 1 shows two candidate cores, where c_1 is a 4-vectex core and c_2 is a triangle (which is a smallest core). Since the triangle CBD in c_1 and the triangle

BDE in c_2 shares a common edge. If the link CE exists, BCD and BDE form a 4-vectex complete graph BCDE. Therefore, we can join c_1 and c_2 to produce c_3.

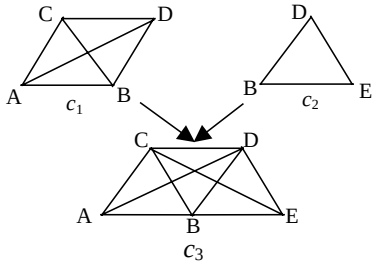

Fig. 1. An example of community core

The core expansion algorithm works as follows. *T* is an array of triangles. For each triangle, a core *c* consisting of only the triangle is created. The algorithm tries to merge these candidate cores by checking every triangle pair to see whether they can join. If the triangle $T[i]$ can join with $T[j]$, then their cores are merged together to form a larger core.

Clustering around Community Cores. Our next task is to group the remaining triangles and edges with the community cores. We exploit the content information of the community cores. If a triangle or an edge has a high content similarity, which is measured by text similarity, with a community core, it indicates that they are likely to share a common theme. Consequently, it will be clustered into that community.

Let the set of *cores* be $C = \{c_1, c_2, \ldots, c_k\}$, and the remaining elements be $S = \{s_1, s_2, \ldots, s_m\}$, which include both triangles and edges. Those elements could not be merged to the cores form their own communities of smaller sizes.

The algorithm first compares the similarity between each element s_i in *S* with each core. It then adds s_i to the core that has the highest content similarity with s_i. If s_i has 0 similarity with every core, s_i forms a small community by itself. The similarity function between triangles is described below.

triangleSimilarity(s_i, c_j) computes the similarity of a triangle s_i and a core c_j. This similarity is the largest similarity between the triangle s_i and triangle members in c_j that share an edge with s_i. If a triangle s_i does not have a common edge with any triangle in the core, the similarity is 0.

The similarity between two triangles is computed as follows: If they do not share any edge, then their similarity is 0. If they share an edge, their similarity is computed like this: Let the two triangles be t_1 and t_2. t_1 has three edges $\{e_a, e_b, e_c\}$, and t_2 has three edges $\{e_d, e_f, e_c\}$. e_c is the common edge. To calculate the triangle similarity between t_1 and t_2, we combine all the keywords in the edges e_d and e_f together to form a vector $v_{d,f}$, and combine all the keywords of edges e_a and e_b to form a vector $v_{a,b}$. The cosine similarity, which is the standard similarity measure in information retrieval, between the two term vectors is the triangle similarity. In the same way, we can compute the similarity between an edge and a community core.

5 Empirical Evaluation

This section evaluates the proposed technique. We first describe the test documents used. They come from different sources, as we want to test if the proposed algorithm is generally applicable. Our first document collection is from top 500 Web pages retrieved through the Google search engine for a given entity. The other two document collections are top 1000 relevant documents from Google blog search, and top 300-500 relevant documents from Financial Times (FT) corpus.

Table 1 shows our experiment results from the Web pages. Column 1 gives the name of each entity. Column 2 shows the community ID. Column 3 lists entities for each community sorted in descending order of their degree centrality scores. Due to the space limit, we used the initials for the first names in Table 1. We automatically extracted the top nouns from community context and listed them in the column 4. We also manually added some remarks on the discovered communities in column 5.

To evaluate community members, we manually checked the co-occurrence sentences extracted from original text documents. If an entity member in the discovered community is related to the community theme, we mark the entity member as correct. In Table 1, we used italic font for incorrect entity members. Among n members extracted for community c, there are m correct members; the community accuracy of c is $A(c) = m/n$.

Let us now look at the communities of "Bill Clinton". We can see that both communities B1 and B2 contain very relevant persons. While B3 is much smaller, it also contains the family topic. In the Cheney's communities, we would like to point out that "Mary Cheney" and "Lynne Cheney" are grouped into both political and family communities. In fact, both entities are legitimate members, and play different

Table 1. The Discovered Named Entity Communities from Web Pages

Entities	ID	Community members	Summary Terms	Remarks
Bill Clinton	B1	Bill Clinton, G. Bush, H. Clinton, J. Kerry, K. Starr, J. Edwards, A. Gore, J. F. Kennedy, R. Reagan, B. Dole, F. Roosevelt, R. Nixon, N. Gingrich, *D. Rather*, D. Cheney, J. Carter, *J. Lehrer*, V. Foster, R. Perot, S. Hussein, B. Laden, D. Morris, M. Beschloss, W. J. Clinton, *T. Jefferson*, *M. Moore*.	president, election, stage, state, senator, campaign	Political community
	B2	Bill Clinton, P. Jones, M. Lewinsky, K. Starr, L. Tripp, J. Reno, K. Starr, W. J. Clinton, G. Flowers, R. Wright, K. Willey, D. Kendall, *L. Johnson.*	case, president, testimony, lawsuit, jury, deposition	The scandal
	B3	Bill Clinton, Hillary Clinton, Chelsea Clinton.	daughter, wife, time,	Family
Dick Cheney	D1	Dick Cheney, G. W. Bush, J. Kerry, S. Hussein, J. Edwards, C. Powell, B. Clinton, L. Cheney, B. Laden, R. Reagan, G. Ford, R. Clarke, R. Cheney, *T. Russert*, M. Cheney, M. Daniels, A. Gore, P. Leahy, D. Rumsfeld, R. Nixon, P. Wolfowitz, J. Lieberman, *H. Chavez*, *J. Nichols*, D. Quayle, P. Goss, *J. Marshall*, J. Wilson, B. Scowcroft, N. Schwarzkopf, A. Williams, R. Perle, Bush Sr, E. Olson, F. Olson, R. Armitage, T. Ridge, N. Mandela, J. Miller.	president, Iraq, war, administration, defense, secretary	Political community
	D2	Dick Cheney, L. Cheney, M. Cheney, L. A. Vincent, L. Cheney.	daughter, wife, child, issue, family	Family

Table 2. The Discovered Named Entity Communities from Blogs

Entities	ID	Community members	Summary Terms	Remarks
Tom Cruise	T1	T. Cruise, B. Shields, K. Holmes, M. Lauer, O. Winfrey, L. R. Hubbard, K. Preston, *J. Fox*, B. bush, *N. Yan*, *M. Jackson*, S. Johansson, D. Miscavige, M.Rogers, P.Kingsley, L. A. Devette, *B. Pitt*, J. Travolta.	scientology, depression, actress, love, show, paxil	Scientology & psychiatry
	T2	Tom Cruise, K. Holmes, N. Kidman, M. Rogers, P. Cruz, C. Klein, S. Vergara, R. Thomas.	actress, relationship, girlfriend, love, marriage, thing	Dating life
	T3	S. Spielberg, T. Cruise, J. Maguire	war, director, film, year, movie, world	Movies
	T4	R. Hubbard, J. Rodriguez, K. Holmes	adviser, interview, member, scientology	Katie& Scientology
Angelina Jolie	A1	A. Jolie, B. Pitt, Maddox, Z. M. Jolie, J. Aniston, B. B. Thornton, J. Voight, G. Clooney, L. Dern, *L. Croft*, *King N. Sihamoni*, J. L. Miller.	child, son, people, divorce, love, marriage	Private Life
	A2	A. Jolie, Good Shepherd, Ro. De Niro, M. Damon	drama, cia, history, thriller, universal	Movie Project

roles in the two communities. These highlight the key feature of our algorithm, mining overlapping communities.

In the Table 2, the communities of "Tom Cruise" were extracted from Weblog data. We can observe that T1 and T2 are strong communities. To our surprise, T3 was a weak community. It indicates that not many blogs paid attention on his movie release. Similarly, T4 was also relevant, but a weak community. The community of "Angelina Jolie" shows the same pattern. Whereas both communities are valid, the private life community is larger than the movie community.

We used a newswire corpus in the last experiment. The results in Table 3 further demonstrate the effectiveness of our algorithm. Taking "Sony" as an example, community S3 lists its peer companies in the entertainment business, and S4 contains its peer Japanese companies. Communities I1 and I2 are also interesting. While there is a considerable overlapping between the workstation and PC makers, the link context reveals two distinct community themes. The accuracy for community extraction from these six entities is 172/193 = 89.1%.

6. Conclusion

This paper studied the problem of mining named entity communities from text documents. So far little work has been done to investigate this issue. By exploiting the named entity co-occurrence, we mapped text documents into a named entity graph. An effective mining algorithm was proposed to mine overlapping communities using triangle expansion and content similarity. We applied our algorithm on a variety of document collections. Our experimental results show that the algorithm is able to

discover interesting communities. This work is potentially useful to enhance the Web search related to named entity queries.

Table 3. The Discovered Named Entity Communities from the FT Newswire Corpus

NE	ID	Community members	Summary Terms	Remarks
Sony	S1	Sony, CBS Records, CBS, MCA, Matsushita, Columbia Pictures, *GE*.	1988, acquisition, purchase, company, year, chairman	Acquisition events
	S2	Motorola, Sony, Apple Computer.	product, media, general, magic, technology, company	Cooperation events
	S3	Sony, Warner Bros., Time Warner, Paramount.	producer, contract, movie, Time, Warner, company	Media companies
	S4	Sony, Toshiba, Panasonic, JVC, Fujitsu, NEC, *IBM*, Hitachi.	Japan, company, electronics, USA, phone, industry	Japanese companies
IBM	I1	IBM, Toshiba, NEC, Microsoft, Fujitsu, Intel, Hitachi, Groupe Bull, HP, Motorola, Apple Computer, NCR, Dell, Sony, Novell, *GM*, *Nasdaq*, TI, *Time Warner*	PC, computer, company, chip, market, software	PC makers
	I2	IBM, Sun, Groupe Bull, HP, MIPS.	workstation, market, RISC, competition, deal, technology	Workstation makers

References

1. Brin, S. Extracting patterns and relations from the World Wide Web. In Selected papers from the International Workshop on the World Wide Web and Databases (1999).
2. Brin, S. and Page, L. The anatomy of a large-scale hypertext Web search engine. In Proceedings of the seventh international conference on World Wide Web 7.
3. Diesner J. and Carley K.M. Exploration of Communication Networks from the Enron Email Corpus. In Workshop on Link Analysis, Counter-terrorism and Security at the SIAM Data Mining Conference (Newport Beach, California, 2005).
4. Dorow, B. and Widdows, D. Discovering corpus-specific word senses. In EACL, (Budapest, Hungary, 2003), 79-82.
5. Eckmann, J., and Moses, E. Curvature of co-links uncovers hidden thematic layers in the World Wide Web. In Proceedings of the National Academy of Sciences.
6. Erkan, G. and Radev, D. Lexrank: Graph-based centrality as salience in text summarization. In Journal of Artificial Intelligence Research (2004), 22:457-479.
7. Flake, G. W., Lawrence, S., and Giles, C. L., and Coetzee, F. Self-Organization and Identification of Web Communities. In IEEE Computer (2002), 35(3): 66-71.
8. Hasegawa, T., and Sekine, S., and Grishman, R. Discovering Relations among Named Entities from Large Corpora. In ACL (2004), 415-422.
9. Kleinberg, J. Authoritative sources in a hyperlinked environment. In Proceedings of ACM-SIAM symposium on discrete algorithms, (1998) 668 - 677.
10. Kubica, J., Moore, A., Schneider, J., and Yang, Y. Stochastic Link and Group Detection. In Proceedings of the Eighteenth National Conference on Artificial Intelligence.
11. Lin, D. PRINCIPAR-An Efficient, broad-coverage, principle-based parser. In Proceedings of the 15th conference on Computational linguistics, (Kyoto, Japan, 1994).
12. Toyoda, M. and Kitsuregawa, M. Creating a Web community chart for navigating related communities. In Proceedings Hypertext-2001 (Århus, none, Denmark), 103-112.
13. Wasserman, S. & Faust, K. (1998) Social Network Analysis: Methods and Applications.