Discovering Important Nodes through Graph Entropy The Case of Enron Email Database

Jitesh Shetty
University of Southern California
University Park
Los Angeles, CA 90089
jshetty@usc.edu

Jafar Adibi
USC Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292
adibi@isi.edu

ABSTRACT

A major problem in social network analysis and link discovery is the discovery of hidden organizational structure and selection of interesting influential members based on low-level, incomplete and noisy evidence data. To address such a challenge, we exploit an information theoretic model that combines information theory with statistical techniques from area of text mining and natural language processing. The Entropy model identifies the most interesting and important nodes in a graph. We show how entropy models on graphs are relevant to study of information flow in an organization. We review the results of two different experiments which are based on entropy models. The first version of this model has been successfully tested and evaluated on the Enron email dataset.

Categories and Subject Descriptors

H.4 [**Link Discovery, Data Mining, Social Network Analysis**]: Miscellaneous; D.2.8 [**Graph Theory**]: Social Networks

General Terms

Graph theory

Keywords

Entropy, Link Discovery

1. INTRODUCTION

A new challenge in the area of Link Discovery (LD) [18]. and social network analysis (SNA) is to exploit communication pattern information and text information within knowledge discovery processes such as discovery of hidden organizational structure and selection of interesting prominent members. An interesting example of such a challenge is to discover hidden groups and prominent people by analyzing their email logs.

Email logs have been considered as a useful resource for research in such areas. Email logs are of prime importance and relevance in the study of information flow in an organization. Email has become the vital means of communication in the information community. Inherent advantages like ease of sending an electronic mail, archiving communications and the ability to reference past communications have made email the most acceptable and widely used means of communication. Though it is highly used in the business and professional domain its scope is not confined to it. Email is the most archived evidence data on interpersonal communication in electronic form. It can also act as an evidence database for law enforcement and intelligence organizations in their effort to detect hidden groups in an organization which are engaged in illegal activities. All these advantages make email a perfect test bed for relevant research like the study of information flow in an organization.

The study of information flow in an organization is germane to issues of productivity, efficiency and drawing some useful conclusion about the business processes of the organization. It can lead to insights on interaction patterns of employees within an organization at different levels of the organization hierarchy. Most of the experiments in this domain are performed on synthetic data due to lack of an adequate or real life benchmark. The recent availability of large datasets of human interaction like the Enron email dataset can be a touchstone for such research. This dataset shows intercommunication between employees of an organization hence it is perfect to study flow of information in an organization. This dataset is also similar to the kind of data collected for fraud detection or counter terrorism and hence it is a perfect test bed for testing effectives of techniques used for fraud detection and counter terrorism.

In this paper we adopt event based graph entropy (we refer to this as both ”event based graph entropy” or ”graph entropy”) to determine the most prominent yet interesting people in the Enron email dataset.

The rest of this paper is organized as follows. We begin with the problem of order in networks. Next, we describe our novel event based graph entropy model. At the end, we report our results of exploitation of such techniques on Enron dataset followed by related work and conclusion remarks.

2. ORDER IN NETWORKS

Most of the work in SNA nd LD represent their environment with a graph or network. We use both terms in this paper frequently. The question is what sort of mathematical model would work best. One way to describe a threat organization, or a social network is in terms of a graph. In this model, each node would represent an individual member and an edge linking two nodes would indicate direct communication between those two members. Mathematically, we may ask how many nodes must we remove from a given graph before it splits into two or more separate sub-graphs? For graphs of various sorts, it’s possible to estimate the probability that the removal of a certain number of nodes would split the graph into two or more separate units based on a set of policies and criterias. However, a

KDD ’2005 Chicago, Illinois

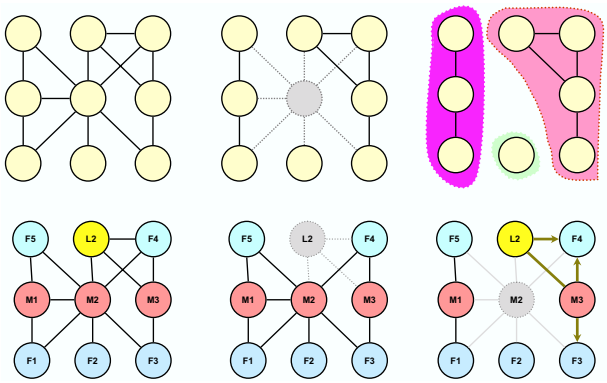


Figure 1: *Leaders* and *Followers* Example. L2 is *leader*, M1, M2 and M3 are *middlemen*; F1, F2, F3, F4 and F5 are *followers*. Up: example of a network. As it shows removing M2 splits the graph to three disconnected subgraphs Down: the same network after information about *leaders*,*middlemen* and *followers*. As it shows even though M2 splits the graph to three disconnected subgraphs, there are at least 2 pathes form *Leaders* to *followers* while removing L1 destroy such path.

graph model might not be the best representation of organizations such as drug dealers, terrorist organization and threat groups. In his recent work, Jonathan Farley explains clearly [6] that modelling terrorist networks as graphs does not give us enough information to deal with the threat. Modelling these networks as graphs ignores an important aspect of their structure, their *hierarchy*, and the fact that they are composed of *leaders* and *followers*. Hence, it is not enough to split the network since the remnant may contain a leader and enough followers to pursue their plans. [6] assume the network structure is known and authors try to find the optimum way to disrupt communication between leaders and followers. However, in our work we try to identify those important nodes as much as possible.

Figure 1 illustrates an example of such a phenomenon. The graph in the left shows a network consisting of three *leaders*: $L1$, $L2$ and $L3$; three *middlemen*: $M1$, $M2$ and $M3$; and three *followers* $F1$, $F2$ and $F3$. The graph in the right illustrates the same network without $M2$. As it is clearly seen that though such a removal splits the graph into two separate remnants, each sub-graph has leaders, middlemen and followers to carry orders and and execute the plan. Hence in this type of networks the relationship of one individual to another in a network becomes important. *Leaders* are represented by the topmost nodes in a diagram of the ordered set representing a network and *followers* are nodes at the bottom. Disrupting the organization would be equivalent to disrupting the chain of command, which allows orders to pass from *leaders* to *followers*.

Hence, the interesting problem here is to determine important nodes or leaders in a network. In other word, we are looking for those nodes whose removal has the maximum effect on the command chain.

3. GRAPH ENTROPY

We assume we have an evidence database (EDB) full of transactions among individual such as *email, Phone Call* etc. After exploiting the various explicit and implicit evidence fragments given in the EDB, we try to identify prominent members in a graph by looking at their transactions with others. To find prominent people in a network, we need to aggregate links between them and discover which node has the most effect on such a network. The entropy model can identify an entity or a set of entities which has the most effect on the graph entropy and thus provide a ranked list based on such effect. To do this we need to exploit facts such as individuals sharing the same property (e.g., having the same address) or transactions like being involved in the same action (e.g., sending email). Since such information is usually recorded by an observer we refer to it as *evidence*. Without loss of generality we only focus on individuals' actions in this paper, but not on their properties.

We transform the problem space into a multigraph $G =< V, E >$ in which each node represents an entity (such as a person or organization) and each link (edge) between two entities represents an action they are involved in. The term multigraph refers to a graph in which multiple edges between nodes are either permitted. For abstraction we summarize the set of actions (e.g., emails, phone calls etc in each edge and refer them as *link*) . Hence each *link* represent a set of actions in a vector. For instance an edge e_7 could be a set of two actions as $e_7 = [a_2, a_5]$. Also please note that it's possible to distinguish between email sender and receiver.

$V = \{v_1, \ldots, v_{|V|}\}$ Number of vertices
$E = \{e_1, \ldots, e_{|E|}\}$ Number of edges
$A = \{a_1, \ldots, a_{|A|}\}$ Type of actions

The EDB consist of tables representing individuals and actions among them at a given time. The table in Figure 2 shows an example of such data.

Assume we have a small society of 4 people who have been in contact with each other through actions. Figure 2 shows an example of such a database. There are four people and three possible actions: sending *Email ,making* a*Phone Call* and participating in a*Meeting*. When a person is not involved in any of the abovementioned actions at a particular time we show with action φ.

Hence $V = \{v_1, v_2, v_3, v_4\}$, $E = \{e_1, e_2, e_3 e_4\}$ and $A = \{Email, phoneCall, Meeting, \varphi\}$. For the matter of representation we show A as $A = \{E, C, M, \varphi\}$. The table in 2 illustrates actions among these individuals along with the action time.

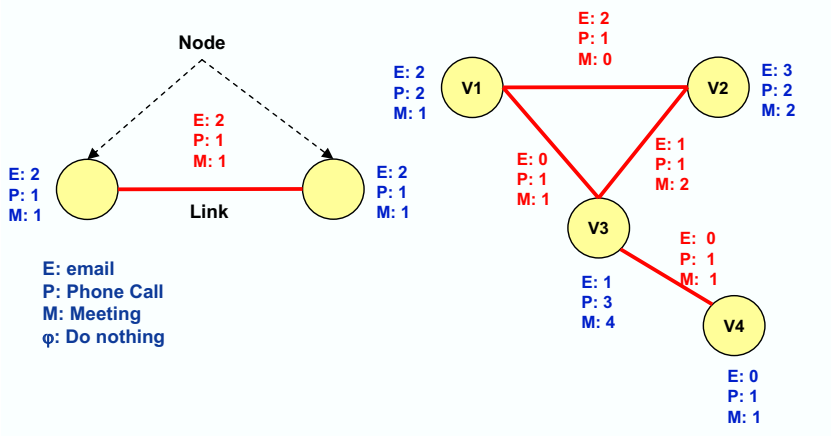
This graph has a major conceptual difference with well-known Bayesian and other similar graphical representations. Unlike such conventional techniques in which nodes are variables and links are statistical relation among variables (causal relations), here nodes represent entities, and links are relations among entities.

3.1 Graph Entropy

There is no commonly used definition of graph entropy. Indeed, one can define the graph entropy as the Kolmogorov complexity of its adjacency matrix and one can even use this definition to obtain interesting theoretical bounds for several important graph characteristics , but the Kolmogorov complexity is incomputable [4].

In the following we adopt the notion of graph entropy which is equal to Korner definition [13]of graph entropy when the graph is complete. Korner definition of graph entropy also has this limitation that all elements of the graph are being emitted by a discrete memoryless and stationary information source according to the probability distribution P. We show how we add memory to graph entropy definition by looking at sequences with length greater than 1.

Korner gave several descriptions of graph entropy $H(G, p)$ including the following.



Time	Action	V1	V2	V3	V4
1	V1 Email V2	E	E	Φ	Φ
2	V3 Phone Call V4	Φ	Φ	C	C
3	V2 Meeting V3	Φ	M	M	Φ
4	V1 Email V2	E	E	Φ	Φ
5	V2 Email V3	Φ	E	E	Φ
6	V1 Phone Call V3	C	Φ	C	Φ
7	V1 Phone Call V2	C	C	Φ	Φ
8	V2 Meeting V3	Φ	M	M	Φ
9	V3 Meeting V4	Φ	Φ	M	M
10	V1 Meeting V3	M	Φ	M	Φ
11	V2 Phone Call V3	Φ	C	C	Φ

Figure 2: MI Example. V1, V2, V3 and V4 represent people. *E, C, M*and φ stand for *Email, phoneCall, Meeting* and *doing nothing* respectively. The table on the right shows activities among people and the graph on the left illustrates such network.

$$H(G,P) = min_{x \in StableSet(G)} \sum_{v \in V(G)} p_v log(p_v) \qquad (1)$$

Where $StableSet(G)$ denotes the family of stable sets in vertices of G. A subset of vertex set is called stable set if it does not contain any edge. Stable sets in graphs form one of the important models in integer programming and have various applications. However, the stable set problem is NP-hard and also not easy to treat in practice. Even though there are some approximate ways to calculate such a set our definition of graph entropy is a special case of such a definition. However we extend such a definition to cover dependencies in the graph.

Let $G =< V, E >$ be a graph. Let P be the probability distribution on the vertex set $V(G)$. We will think of $V(G)$ as a finite alphabet. How we define such a alphabet depends on the nature of the problem. This definition has similarities with [16].

$$H(G,P) = \sum_{i=1}^{|V|} p(v_i) \log(1/p(v_i)) \qquad (2)$$

In general if we plot $H(x)$ in terms of $p(x)$ there are two sides of the curve that play an important roles. Those x with high probability and x with lower probabilities. We believe our model finds those instances. Figure 3 illustrates such phenomenon.

A great concern in LD domain is that elements of the data are not independent. For instance if the link $AsendemailtoB$ and link $BsendsemailtoC$ are dependent to each other, this means B may forward $A's$ email to C. Hence, we can change the probability space from $length = 1$ to $length = 2$ and more. This means our space consists of sequence of emails if the second one is dependent to the first one and so on.

Since discovering such dependency is not easy we provide three approaches to address such an issue. In the following we describe these cases.

- For every single transaction (for example *email*) we examine if it is similar to other received emails by a given individual. i.e. if she forwards an email, or copy and pastes a major part of an email.
- If a transaction happens immediately right after a given transaction (for instance if

  $$Time(transaction_i) - Time(transaction_j) < window$$

  we consider that a dependent transaction.
- Another alternative is exploitation of Markov Blanket type of model. In this model we assume an event (link) between two nodes is only dependent to those node's events (links connected to those nodes). For instance in Figure 3.1 we assume *red (dark)* event is only dependent to rest of the *black* links. In a more advanced model for any event e we can drive a set of dependent events such as $D_e = \{d_e^1, \cdots, d_e^{|D|}\}$ each with the probability of $P_e = \{p_e^1, \cdots, p_e^{|D|}\}$which shows the probability of dependencies to e. This probability could be derived from domain knowledge.

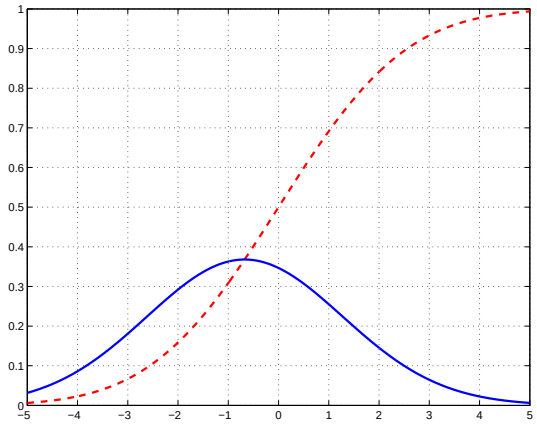


Figure 3: ($P(x)$ is a normal distribution, and the bell curve is the distribution of $H(x)$. We are mostly interested in right and left part of the $H(x)$.

We extend this notion to cover deeper levels of dependencies. For example, consider the domain of emails. A first level measure of graph entropy would be the predictability of an arbitrary email within that graph. In this approach, X would be the set

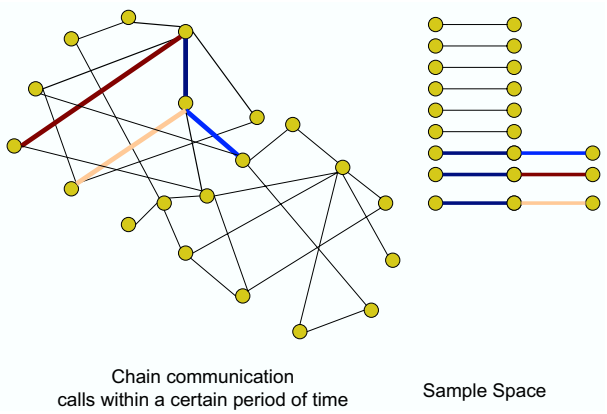


Figure 4: Dependent links

of all emails such as $A_{email}B$ contained in the graph. Furthermore, $P(A_{email}B)$ as the number of occurrences of $A_{email}B$ in the graph, divided by the size of the graph.

A more sophisticated approach would be to let X be all substrings of a certain length n. $P(X)$ would then be the number of occurrences of sequence X, divided by the total number of possible sequences with length n in the graph. As an example, let us choose length of $n = 2$. Hence we are counting sequences such as $A_{email}B_{email}C$ and $B_{email}D_{email}E$ and $p(A_{email}B_{email}C)$ would be the number of occurrences of such sequence over all possible sequences with length equal to 2 in the graph.

There are couple of issues associated with this definition. First of all it is obvious from entropy definition that more the regularity in sequence of events, the lower its graph entropy will be. As certain sequences occur more frequently, the probabilities of these sequences will increase, and probabilities for other sequences will decrease subsequently. As we mentioned earlier entropy is highest when the probabilities are uniform, and it decreases as the probability distribution becomes less uniform. Second, based on our definition there is no single entropy measure for a given graph; the value is dependent on the selected alphabet size, n. The value of n depends on the nature of the database and comes form intuition and domain knowledge. $n = 1$ only measure the entropy of nodes labels, without considering relationships between individuals. On the other hand very large number of n make the whole calculation very expensive and the interpretation will be very difficult.Finally, we consider the time of an event when we make our alphabet. Hence if $Time(A_{email}B) > Time(B_{email}C)$ we do not consider $A_{email}B_{email}C$ as a sequence.

3.2 Important Nodes

Our interpretation of important nodes are those who have the most effect of the graph entropy when they are removed from the graph. The intuition for this idea is that those who send more commands through the network and their messages are forwarded are important. In addition those who send unusual messages through the network also might be important people. To do this we execute the following procedure. First we calculate the entropy of the whole graph. Next for all nodes in the graph we remove them one by one and recalculate the graph entropy for the remnant graph. Following table illustrates such procedure.

Pseudo Code for Discovering Important Nodes

1. Compute the graph entropy using 2 as $Entropy_all$
2. For all nodes $N(i)$ in the do the following
3.
   - Compute the entropy of one node $N(i)$ by calculating the entropy of all of its edges as $E(i)$
   - Drop $N(i)$ from the graph
   - Calculate the entropy or remnant graph as $EN(i)$
   - Calculate the cross entropy of $EN(i)$ and $E(i)$
   - $Effect(i) = EN(i)/log(EN(i)/E(i))$
4. Rank nodes based on $Effect(i)$

4. EXPERIMENTAL RESULT

Below we report the results of applying the graph entropy model to the Enron Email Dataset [1]. There are many reason for using Enron dataset to evaluate our techniques. First of all, it is probably the only actual corporate email dataset available to public. Second, email logs are of prime importance and relevance in the study of information flow in an organization. Third, the study of information flow in an organization is germane to issues of productivity, efficiency and drawing some useful conclusion about the business processes of the organization. Finally this dataset is also similar to the kind of data collected for fraud detection or counter terrorism and hence it is a perfect test bed for testing effective of techniques used for fraud detection and counter terrorism.

The Enron email dataset was made public by the Federal Energy Regulatory Commission during its investigation. Database was later collected and prepared by Melinda Gervasio at SRI for the CALO (A Cognitive Assistant that Learns and Organizes) project; most of the integrity problems in the dataset had been resolved. It contains all kind of emails personal and official. Some of the emails have been deleted as part of the redaction effort due to requests from affected employees. William Cohen from CMU has put up the dataset on the web for researchers [2]. This version of the dataset contains around 517,431 emails from 151 users distributed in 3500 folders. The dataset contains the folder information for each of the 151 employees. Each message present in the folders contains the senders and the receiver email address, date and time, subject, body, text and some other email specific technical details.

We created a *MySql* database [3] for the dataset to catalyze the statistical analysis of the data and cleaned the dataset by removing a large number of duplicate emails. Folders such as *discussion threads* and *all documents* were generated by the computer and were not user created. We cleaned up the whole data to make it ready for our purposes. For detail of the data cleaning please refer to [19]. Our cleaned Enron email dataset contains 252,759 messages from 151 employees distributed in around 3000 user defined folders. Our prototype is written in Java and visualization made either by in-house developed Java Applet or using NetDraw [2]. The prototype is applicable to apply to any similar dataset. The database

[1]This database contains private emails, while reading this paper please be considerate about the privacy of the people who were not involved in any of the actions which precipitated the investigation. Authors do not attach any label to anyone in this dataset by no means. The main purpose of this study is to evaluate some novel techniques on actual real world dataset.

[2]http://www-2.cs.cmu.edu/ enron

[3]http://www.isi.edu/ adibi/Enron/Enron.htm

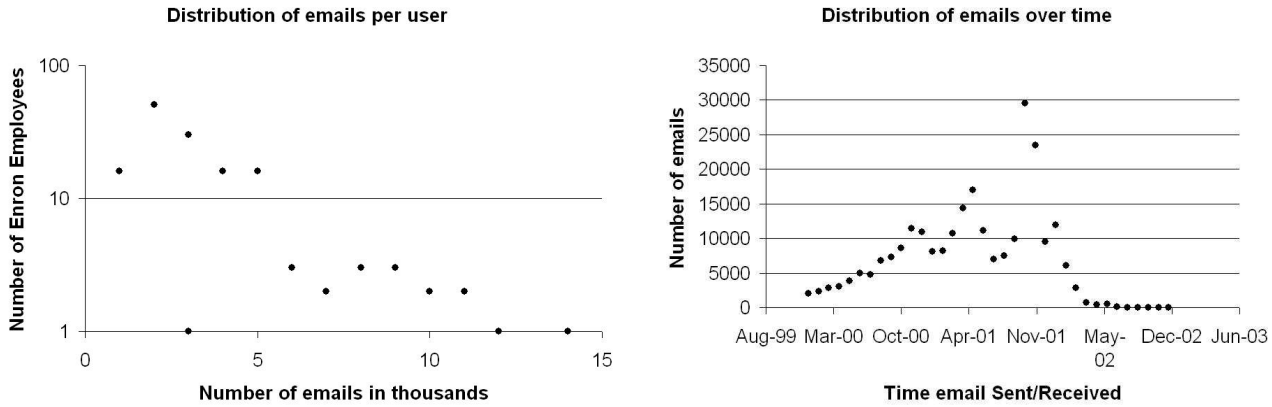


Figure 5: Enron Database distribution. Left: distribution of the messages per user. Right: distribution of the emails over time

scheme is very intuitive and general which make it easy to map to any other email dataset. A report on Enron database schema and dataset characteristics is available at [19].

Figure 5 (left) shows the distribution of the messages per user. The x-axis represents the number of email messages in log scale. The y-axis represents the number of Enron employees in log scale. The graph clearly shows that the messages are not evenly distributed between the users. A small number of users have a large number of messages. However, there are employees distributed through out the y-axis which reflects that the dataset contains employees with all amount of email messages. Figure 5 (right) shows the distribution of the emails over time. The figure clearly reflects that most of the emails have been sent and received in the year 2001. The x-axis represents the year in which the email has been sent or received and the y-axis shows the number of emails.

To illustrate the Enron network, we transform the Enron database into a graph as we discussed; each vertex of this graph represents an Enron employee. An edge exists between two employees if the two employees have exchanged emails. This graph constitutes of 151 employees of Enron. The graph is shown in Figure 6. We found out the position of every employee in the ex organization hierarchy. The color of the nodes stands for the position of the employee in the ex organization. The major type of communication are "TO" and "CC".

The Enron email database has more than 70K emails which are referenced emails; these are emails which refer some other emails. But another scenario is where a particular email doesn't technically refer some other email but has relevant information. This bring up a very interesting phenomenon with the original Enron graph, the edges which represent exchange of information don't end at the receiver node but the information flows much deeper into the Enron graph involving a lot of other nodes. Here we expand the scope of influence of nodes to every such node which share a particular information. There are certain intricate issues involved in detecting referenced emails and in particular detecting a pattern of how some particular information was conveyed to other nodes in the graph. There is no evidence in the database about the generator node or transient nodes of the forwarded emails. Also when some information is conveyed further there is some more information added or the original information changed. We detect the referenced emails based on the percentage similarity with the original email.

4.1 Enron Important Nodes

We compute the entropy of the entire Enron graph. We then drop a node and also drop the edges fanning in and out that particular node and recalculate the entropy. We measure the change in graph entropy. We do this for each node present in the graph. We generate a ranked list based on the change in graph entropy. We conclude that the node whose absence brings maximum change in graph entropy is the most influential node in the graph.

We repeated such procedure for the following two experiments.

1. Sequence of length = 1. Here we only consider emails among individuals as our space. This is the procedure for detecting influential nodes in the graph using the entropy model at *length = 1*. The model at *length = 1* limits the scope of influence of every node in the graph to directly connected nodes. But past work using epidemic models [1] on social networks show that information is passed by hosts in a social network to other interested people in the network. This shows that when certain people are engaged in some activity in a network they pass information amongst each other and might not be in direct contact with each other.

2. Sequence of length = 2. In a network like the Enron graph there is a possibility that information might be hopped through nodes deliberately. This expands the scope of influence of nodes over other nodes. In the next step we calculate the graph entropy at *length = 2*. If a node in the graph is not directly in contact with some other node, but receives information from it through a third node, then its presence in the graph has influence over though they are not directly in contact with each other. This influence is taken into consideration in the *length = 2* computation of entropy.

To measure if an email is dependent to one of the the previous emails in a mail box we conducted the following procedure.

We created an Enron dictionary which contains all the words in the organization vocabulary, there are certain words which are not there in traditional dictionaries like organization jargons, some proper nouns etc. These words in the dictionary don't contain stop words and are stemmed words. Stop words are those words like conjunctions, prepositions and articles which do appear often in the document yet alone carry little meaning. We used the porter stemming algorithm for stemming the words. The porter stemming algorithm is a process of removing the commoner morphological and in flexional endings from the words in English. Its main purpose is as part of the term normalization process that is used when stepping up information retrieval systems. We normalized all emails using this. We generate a vector representation for each email. Then we compare the vectors using the Jaccards Algorithm.

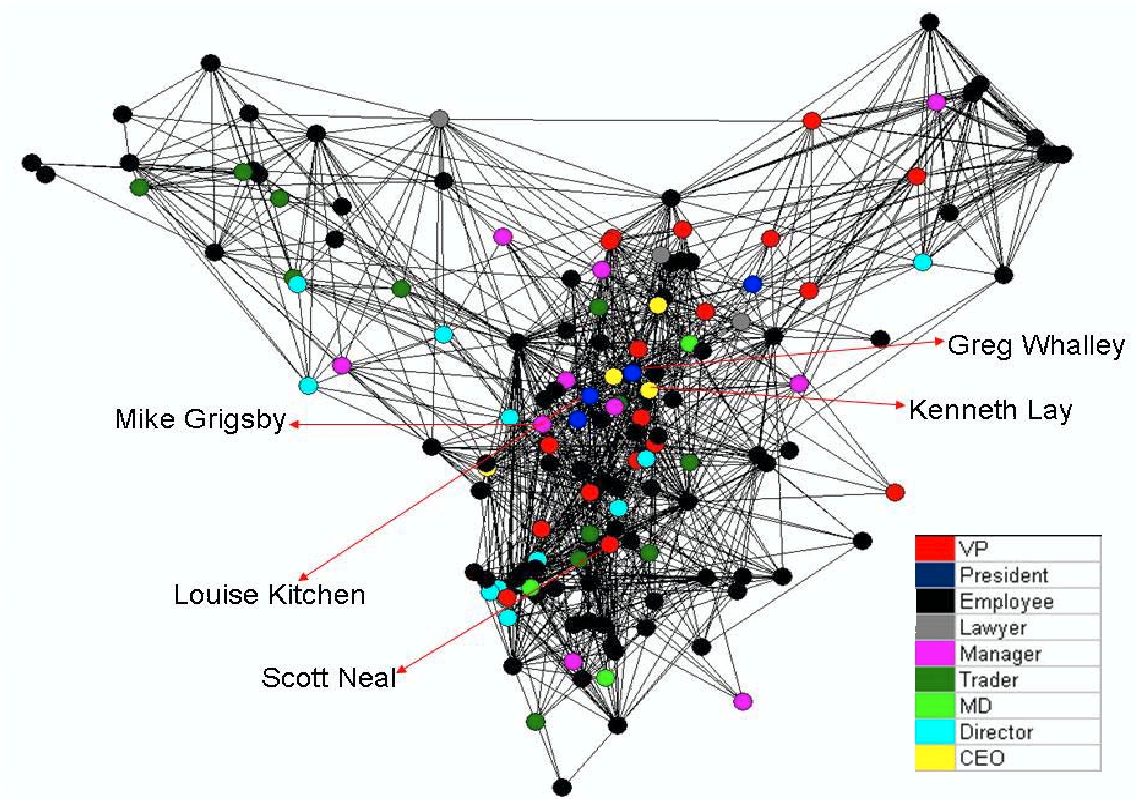


Figure 6: Enron Network

$$Similarity(X,Y) = \frac{X \bigcap Y}{X \bigcup Y} \quad (3)$$

The percentage similarity of the Jaccards algorithm is ratio of intersection of two vectors and their union. Thus the emails referenced by the original emails are all those emails which have more than 60% Jaccard score. We take the threshold as 60% based on emperical results. We performed experiments on the referenced emails in the Enron database and calculated their percentage text similarity. The average percentage text similarity between referenced emails in the Enron database is 63.71%. So we conclude that if any two emails are more than 60% similar the context of talk is same, and is thus linked.

The emails are further ordered based on the time stamp. This gives a hierarchy for each email if it has been referenced and the nodes at each hierarchy can be obtained from the database. Thus we can relate the influence of nodes to those edges which are not in direct contact. This is used in computing the entropy at *level 2*. In *level 2* computation we calculate the entropy of the entire graph in the same way as we did for *level 1*. But then when we drop each node, we not only drop the edges spanning in and out from this node but we also drop those edges which have used this node in its path of information flow. So we drop edges which are not directly in contact with this node but they have either been originally generated from this node or used this node as a transient, and calculate the change in entropy. Then we generate a ranked list/seed group based on the change in graph entropy. The results of this for the Enron graph are shown in next section.

The results for *length = 1* are shown in Table 1. *Louise Kitchen* the ex president of Enron online is the most influential node in the Enron graph based on the entropy model at *length = 1*. The second most influential node in the Enron graph is *Mike Grigsby* who is an ex manager at Enron, followed by *Greg Whalley* ex president, *Scott Neal* ex employee and *Kenneth Lay* ex CEO. We now generate the ranked list/seed group based on the entropy model at *length = 2* as discussed earlier. At *length = 2* we take into consideration the information which has not been received from a direct contact but the information has been forwarded from some other node in the graph. Here again we show only the first five members in our ranked list/seed group. The results for *length = 2* is illustrated in 2.

Table 1: Most Important Nodes $length = 1$

Rank	Name	Designation at Enron
1	Louise Kitchen	President
2	Mike Grigsby	Manager
3	Greg Whalley	President
4	Scott Neal	Employee
5	Kenneth Lay	CEO

In the ranked list group generated based on the entropy model for *level 2* members like *Greg Whalley* and *Kenneth Lay* have a higher rank. *Louise Kitchen* and *Mike Grigsby* get lower ranks. The *length = 2* computation expands the scope of influence of each node over other nodes in the graph.

Table 4 compares graph entropy model result with some other conventional techniques such as betweenness centrality. Clearly betweenness centrality capture those nodes that are in the center of the graph but not necessarily those with higher authorities. Though since Louis Kitchen had a crucial role in Enron and several VPs and

Table 2: Most Important Nodes $length = 2$

Rank	Name	Designation at Enron
1	Greg Whalley	President
2	Kenneth Lay	CEO
3	Louise Kitchen	President
4	Mike Grigsby	Manager
5	Harry Arora	VP

Table 3: List of people with high number of sent emails

Rank	Name	Designation at Enron
1	Jeff Dasovich	N/A
2	Kay Mann	Employee
3	Sara Shackleton	N/A
4	Tana Jones	N/A
5	Chris Germany	N/A

Managers used to report to her she is in the betweenness centrality list as well.

In addition, we compare the result of our model with a simple frequency counting of those individuals who have sent most emails comparing to the rest of the Enron employees. Table 3 illustreates these people.

5. RELATED WORK

As illustrated by our experiments the main focus of our work is to find important nodes in a graph. We further use this to find relations and connections among entities and individuals. Our approach does not look for similarities among individuals as a classification task such as the work by Getoor et. al [8].

Graph entropy has different definition in various literature depending on the nature of the data [13] [11] [14]. We use a different notion of graph entropy and consider dependencies among links as well. Similar notion of such definition is introduced in [16].

In his recent work [3] Borgatti address the problem of discovering *key players* in a network. His approach is based on measuring explicitly the contribution of a set of actors to the cohesion of a network. In addition, he identifies two separate conceptions or functions of key players which reflect different analytical goals, and develop separate measures of suitability for each type of goal. In addition our approach has a fundamental difference with [3]. While Borgatti finds key players in a network, we try to find leaders. Our example at the beginning of this paper illustrates that key players are different with influential nodes. Freeman [7] in his work address centrality issues in social networks. As we discussed earlier the concept of centrality is close to key players and it is different with our view and definition of important nodes.

In [20] they address the problem of most important nodes in the network. One major difference of this work with our work is that we do not consider the Google referral type of links. Their example of bibliographical is based on reference which make the problem somewhat different. Famous works such as Google Page Rank [17] and HITS [12] are also in this category.

In [5] a linear model is used based on well-known electrical circuits formulas to represent a graph. They produce approximate, but high-quality connection subgraphs in real time on very large graphs. [9] also uses the same approach and exploits Kirchhoff laws to model the social network graph. Other approaches such as [15] which use betweenness and centrality to find crucial central nodes.

Table 4: Most Important Nodes Betweenness Centrality

Rank	Name	Designation at Enron
1	Bill Williams	Broker
2	Steven Merris	N/A
3	Eric Linder	Employee
4	Kay Mann	Employee
5	Louise Kitchen	President

Another issue is that event based graph entropy is scalable. We do not need to explore more than 3 or 4 levels to find important nodes. We can explore the graph around those nodes and run the engine recursively to find more important nodes. Approaches similar to betweenness centrality are also effective but may fail when applied to large networks, since the order of the algorithm is at least N^2 where N is the number of nodes.

Scale-free network also has been discussed extensively recently in literatures. One of the major line of work in scale free networks is gossip modelling and finding the *most influential nodes* to either broadcast a gossip or to prevent a virus distribution over a given scale free network. Kempe et al in their work [10] they consider the problem of selecting influential nodes. Using an analysis framework based on submodular functions, they show that a natural greedy strategy obtains a solution that is provably within 63% of optimal for several classes of models. Our work differentiates from this work since they do not have the notion of order in their model and their definition of *most influential nodes* is somewhat different with our *definition of important nodes*.

Our work is inspired by [6] in which he introduces the notion of order in networks and graphs. However we used a different approach comparing to [6].

6. CONCLUSION

The Enron email dataset is the largest real email dataset present in the public domain; other datasets haven't been public because of privacy concerns. This dataset contains all kind of emails personal and official. This is a valuable resource for many diverse fields. Social network analysis, link discovery are the most relevant fields where this can be used.

In this work, we defined and addressed the problem of important nodes and finding closed group around them. Additional contributions are the following:

- We proposed a novel yet simple intuitive way to measure the graph entropy as event based graph entropy. We showed that approaches like betweenness centrality lead to poor answers when our network consist leaders and followers.
- We provided a systematic way to find important nodes in a graph based on their effect on graph entropy.
- Moreover, we implemented our algorithms in a working prototype, complete with an interactive Java-based interface, on a real graph that we derived from the Enron Dataset. The graph has about 150 nodes and more then quarter million links.

In this paper we focused on using event based entropy to find influential nodes in a graph. We tested an evaluated the results on the Enron graph. Enron graph being a representation of a real life ex organization there was some evidence available to validate some facts revealed from our experiments. There are certain basic assumptions on which the entropy model claims its results, like the

evidence data is complete and there is no noise in the data. The result gets deteriorated when used on noisy data. Our main focus was to exhibit how an entropy model can act as a good means for detecting influential nodes in a graph.

There are several lines of ongoing and future work, such as, determining group leaders by measuring their entropy over time to capture the change in such entropy in a given period. In addition we would like to exploit sampling, randomization and data streams techniques to deal with very large datasets.

7. ACKNOWLEDGMENTS

This work was supported in part by the Department of the Navy, Office of Naval Research under contract N00173-05-1-G006. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the Office of Naval Research. The authors would like to thank Hans Chalupsky and Eduard Hovy for their invaluable comments.

8. REFERENCES

[1] L. Adamic and B. Huberman. *Information dynamics in a networked world*. Lecture Notes in Physics. Springer, 2003.

[2] S. Borgatti and R. Chase. *Net Draw*. http://www.analytictech.com/, 2004.

[3] S. Borgatti. Identifying sets of key players in a network. In *Computational, Mathematical and Organizational Theory.*, 2005.

[4] H. Buhrman, M. Li, J. Tromp, and P. Vitnyi. Kolmogorov random graphs and the incompressibility method. *SIAM Journal on Computing*, 29(2):590–599, 2000.

[5] C. Faloutsos, K. McCurley, and A. Tomkins. Fast discovery of connection subgraphs. In *Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2004.

[6] J. D. Farley. Breaking al qaeda cells: A mathematical analysis of counterterrorism operations (a guide for risk assessment and decision making). *Studies in Conflict & Terrorism*, 26:399411, 2003.

[7] L. Freeman. Centrality in social networks: Conceptual clarification. *Social Networks*, 1:215–239, 1979.

[8] L. Getoor, E. Segal, B. Taskar, and D. Koller. Probabilistic models of text and link structure for hypertext classification. In *IJCAI01 Workshop on Text Learning: Beyond Supervision*, Seattle, Washington, 2001.

[9] B. Huberman and W. Fang. Discovering communities in linear time: a physics approach. In *KDD*. ACM, 2004.

[10] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence in a social network. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2003.

[11] J. Kieffer and E. Yang. Ergodic behavior of graph entropy. *ERA American Mathematical Society*, 3(1):11–16, 1997.

[12] J. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of ACM*, 46, number =, 1999.

[13] J. Korner. Bounds and information theory. *SIAM Journal on Algorithms and Discrete Mathematics*, (7):560–570, 1986.

[14] M. Li and P. Vitanyi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer Verlag, 1997.

[15] M. Newman. Who is the best connected scientist? a study of scientific coauthorship networks. In E. Ben-Naim, H. Frauenfelder, and Z. Toroczkai, editors, *Complex Networks*, pages 337–370. Springer, 2004.

[16] C. Nobel and D. J. Cook. Graph-based anomaly detection. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2003.

[17] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: bringing order to the web. Technical report, Stanford, 1998.

[18] T. Senator. Evidence extraction and link discovery. 2002.

[19] J. Shetty and J. Adibi. Enron email dataset. Technical report, USC Information Sciences Institute, http://www.isi.edu/ adibi/Enron/Enron.htm, 2004.

[20] S. White and P. Smyth. Algorithms for estimating relative importance in networks. In *ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 266–275, 2003.