

A Generative Model for Statistical Determination of Information Content from Conversation Threads

Yingjie Zhou
Rensselaer Polytechnic
Institute
110 8th Street
Troy, New York 12180
zhouy5@rpi.edu

Malik Magdon-Ismail
Rensselaer Polytechnic
Institute
110 8th Street
Troy, New York 12180
magdon@cs.rpi.edu

Mark Goldberg
Rensselaer Polytechnic
Institute
110 8th Street
Troy, New York 12180
goldberg@cs.rpi.edu

William A. Wallace
Rensselaer Polytechnic
Institute
110 8th Street
Troy, New York 12180
wallaw@rpi.edu

ABSTRACT

We present a generative model for determining the information content of a message without analyzing the message content. Such a tool is useful for automated analysis of the vast contents of online communication which are extensively contaminated by uninformative, spam, and broadcast. Content analysis is not feasible in such a setting. We propose a purely statistical methodology to determine the information value of a message, which we denote the Information Content Factor (ICF). Underlying our methodology is the definition of information in a message as the message's ability to generate conversation. The generative nature of our model allows us to estimate the ICF of a message without prior information on the participants. We test our approach by applying it to separating spam/broadcast messages from non-spam/non-broadcast. Our algorithms achieve 94% accuracy when tested against a human classifier which analyzed content.

Categories and Subject Descriptors

H.4.3 [Information Systems Applications]: Communications Applications—*Electronic mail, Bulletin boards*; H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval—*Information filtering*; H.1.1 [Models and Principles]: Systems and Information Theory—*Value of information*

General Terms

Algorithms, Experimentation

Keywords

Information Content Factor, broadcast, reply process, conversation thread, content analysis

1. INTRODUCTION

Copyright is held by the author/owner(s).
WWW2008, April 21–25, 2008, Beijing, China.

With ever-increasing Internet accessibility, various electronic media, such as online forums, message boards, blogs, and emails, are available for people to exchange ideas and opinions worldwide. People utilize these tools to communicate with strangers, friends, or experts, to just socialize or to seek help. In general, a person initiates a topic, which could be an activity, a question, a rumor, an advertisement, or a proposal; if this topic is “interesting”, it may trigger replies from people who have access to the message, otherwise, it may be completely ignored by the recipients. In fact a good working measure of a message's interest value could be the number of replies generated. Here, we explore this idea further by giving a quantitative implementation of this intuition.

The volume of such electronic data has increased tremendously. The enormous data can easily overwhelm people interested in analyzing the data for social science purposes [2, 8]. Needless to say, the data contains valuable information. For example, the sentiments from a stock message board have been analyzed to show that they could influence the stock market [12, 1, 6]. On the other hand, huge amounts of spam and noise also exist in the data. Consider a stock analyst observing a stock message board to extract useful tips. It is not feasible to analyze every post given that there is much spam. How should the analyst determine which posts stand a good chance of being “interesting”? Removing the spam from the data set is as important as identifying the important messages. When studying interactions between people (eg. social group dynamics) by looking at senders and recipients of messages, the spam should be removed since it does not represent interactions. In the case of email, for example, the spam and broadcasts connect the sender to one or more recipients but no interaction occurs; therefore, when a social network is constructed from an email data set, the existence of a significant number of spam and broadcasts will distort the communication pattern that forms the basis for Social Network Analysis. The task of distinguishing useful information from spam among millions of messages is difficult [6]. A straightforward method to separate the in-

formative messages from uninformative ones is to examine the content of the messages; however, for large data sets this method is not practical.

We propose a generative model to determine the information value of a message, which we call the Information Content Factor (ICF). Our approach does not examine message content. We take as input, a set of conversation threads which have been preprocessed from the raw digital data. A conversation thread is defined as a collection of messages in response to a message. The message, which initiates the conversation, is called the root message. The parent-child relationship between messages is determined by the reply function. All replies to a message are children of that message, and a message is the parent of its replies. Thus, a root message generates a tree of replies (the thread). The depth of the thread is the depth of the tree. The total number of replies to the root message is the summation of messages at each generation summed over all generations. The general intuition behind our generative model is that the more replies, the larger the ICF of the root message is.

We recognize the need to differentiate between the transfer of information and communication. As noted in the Shannon-Weaver Model [10], information transfer is a technical process while communication involves the transfer of meaning from source to destination and is a human cognitive process. The “importance” of a message lies in its transfer of meaning. We postulate that important topics generate more information transfer – participants in the information transfer process are motivated to add value to important messages¹. Therefore, we argue that the more informative or important a message is, the more replies it induces. We propose the ICF, which ranges from 0 to 1, to measure how informative a message is based upon the reply structure to that message. The ICF is computed from the thread structure. The ICF can be used to separate the informative messages from uninformative messages without examining the content. We apply this methodology to identify broadcasts in the Enron email data set, and we test against a human who has access to the content. Our approach gives a 94% success rate, treating the human as ground truth.

The outline of the paper is as follows. Section 2 describes two general reply processes and their expected number of replies. We then present the email reply process, which is a mixture of these two processes in Section 2.3. Finally we apply the methodology to Enron emails to identify broadcasts. We conclude with suggestions for future research.

2. GENERATIVE STATISTICAL MODEL

We assume a message with ICF 1 will be replied with probability 1 by each of the recipients of that message, and a message with ICF 0 has no replies. More generally, the ICF is related to the probability of obtaining a reply. There are two parts to our generative model. The first determines the probability that a message is replied given its ICF b . The second determines how the ICF of a reply is related to the ICF of the parent message. Intuitively the higher b , the more likely a reply, and the ICF of a reply should be smaller than the ICF of the parent. We define the ICF-propagation function $f(b)$ to capture the decay in ICF from

parent message to child message, $0 < f(b) < 1$. Thus, for a message with ICF b ,

$$P[\text{reply}] = b,$$

$$ICF[\text{reply}] = bf(b)$$

For example, if the ICF of the root message is b , the ICF of each message at depth 1 will be $bf(b)$, and the ICF of each message at depth 2 will be $bf(b)f(bf(b))$, and so on. One interesting special case is $f(b) = f$, where f is a constant decay factor. The ICF of a message at depth i will be bf^i . Assume a sender S initiates a message M . We consider two cases: the message M has one recipient; and, the message M has multiple recipients. Let R denote the recipient set. In the case of multiple recipients, $R = \{R_1, R_2, \dots, R_i, \dots, R_n\}$, where $n \geq 2$ is the number of recipients.

We give the generative model for three reply processes below. The first is for the single recipient reply process. The second is for the multiple recipients in the “Reply All” framework in which the recipients can either choose to reply to all the recipients (including the sender) or not to reply at all. In forums, message boards, or blogs, it is appropriate to assume that the root message is for all the active members who are willing to reply, and each replied message is again open to all the active members. In emails, it is slightly different. The third reply process is a mixture of these two in which the recipient may choose one of 3 actions: not to reply, reply only to sender (initiating a single recipient sequence); or, reply to all (re-initiating the same process). This third process is more applicable to email data which will be our application. The general idea behind the generative model will become clear from these three examples. The specific details are however application dependent and it should be possible to extend our framework to accommodate different domains.

2.1 Single Recipient Reply Process

In this reply process, the sender S initiates the root message to the recipient R , and R may take 2 actions: reply to S or don’t reply. If R chooses to reply to S , and S again has two options, reply to R or don’t reply, and so on. The conversation between S and R continues until one of them fails to reply.

2.1.1 Single Recipient Reply Process Example

Figure 1 illustrates an example of such a conversation between S and R . We assume the ICF of the root message is b . The ICF, depth, probability of each message are indicated in the figure. For example, when S initiates a message to R at *depth* = 0, the probability p that R replies is b . If R replies, the ICF of this replied message is $bf(b)$, and its depth is 1 in the reply process, and so on.

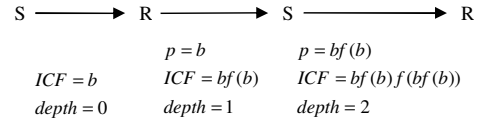


Figure 1: Reply process with one recipient

2.1.2 Expected Number of Replies

¹This material is based upon communications with our colleague, James H. Watt, Professor, Department of Language, Literature, and Communication, RPI.

This reply process is recursive with decreasing ICF. The recursion shows that the expected number of replies of the parent message is a function of the expected number of replies of the child messages. Let X denote the total number of replied messages to the root message with one recipient. Let $\mathcal{E}(b) = E[X|b]$ be the expected number of replies to the root message with ICF b .

We derive $\mathcal{E}(b)$ recursively as

$$\mathcal{E}(b) = b(1 + \mathcal{E}(bf(b))) \quad (1)$$

The first term is the expected number of replies to the root message, and the second recursive term captures the expected number of messages for the single recipient reply process initiated by the first reply. It turns out that it is hard to solve (1) analytically. We give an approximate recursion to calculate $\mathcal{E}(b)$. First we approximate $\mathcal{E}(b)$ when b is small using a Taylor series expansion to second order, and then use (1) to calculate $\mathcal{E}(b)$ recursively.

The Taylor series expansion of $\mathcal{E}(b)$ at $b = 0$ is given by

$$\mathcal{E}(b) = \mathcal{E}(0) + \mathcal{E}'(0)b + \frac{\mathcal{E}''(0)b^2}{2} + \dots \quad (2)$$

Since b is small, we ignore the orders higher than 2. When $b = 0$, the probability of reply for each recipient is 0, therefore by definition, $\mathcal{E}(0) = 0$. To find $\mathcal{E}'(0)$ and $\mathcal{E}''(0)$, the first and second derivative of $\mathcal{E}(b)$, $\mathcal{E}'(b)$ and $\mathcal{E}''(b)$, should be obtained first.

$$\mathcal{E}'(b) = 1 + \mathcal{E}(bf(b)) + b(f(b) + bf'(b))\mathcal{E}'(bf(b)) \quad (3)$$

$$\begin{aligned} \mathcal{E}''(b) = & 2f(b)\mathcal{E}'(bf(b)) + b(4f'(b) + bf''(b))\mathcal{E}'(bf(b)) \\ & + b(f(b) + bf'(b))^2\mathcal{E}''(bf(b)) \end{aligned} \quad (4)$$

Therefore, we have $\mathcal{E}'(0) = 1$ and $\mathcal{E}''(0) = 2f(0)$ from (3) and (4). Thus, $\mathcal{E}(b)$ can be calculated numerically as in Algorithm 1. The expected number of replies to any of the messages in the stream can be obtained by replacing b with corresponding ICF for that message.

Algorithm 1 Numerical analysis of $\mathcal{E}(b)$

```

if  $b \leq 10^{-5}$  then
   $\mathcal{E}(b) \leftarrow b + b^2 f(0)$ 
else
   $\mathcal{E}(b) \leftarrow b(1 + \mathcal{E}(bf(b)))$ 
end if

```

2.2 Multiple Recipient Reply Process

In this reply process, the sender S initiates the root message to the recipient set $R = \{R_1, R_2, \dots, R_i, \dots, R_n\}$, where $n \geq 2$. R_i may take 2 actions: reply to the sender and the other recipients or not to reply. We assume that the recipients choose to reply or not independent on the recipients. The conversation among $\{S\} \cup R$ along a particular message path dies when a recipient fails to reply. The conversation ends when every message path dies.

2.2.1 Multiple Recipient Reply Process Example

Figure 2 illustrates an example of such a conversation between S and $R = \{R_1, R_2, R_3\}$. We assume the ICF of the root message is b . The ICF, depth, probability of each message are indicated in the figure. For example, when S

initiates a message to R at $depth = 0$, the probability p that each of R_1 , R_2 , and R_3 replies is b . If R_i replies, the ICF of this replied message is $bf(b)$, and its depth is 1 in the reply process, and so on.

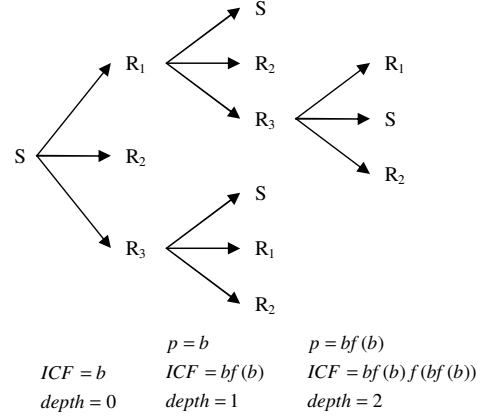


Figure 2: Reply process with multiple recipients

2.2.2 Expected Number of Replies

This reply process is also recursive with decreasing ICF. Let Y denote the total number of reply messages to the root message. Let $\mathcal{F}(n, b) = E[Y|n, b]$ denote the expected number of replies to the root message with ICF b .

We derive $\mathcal{F}(n, b)$ recursively as

$$\mathcal{F}(n, b) = nb(1 + \mathcal{F}(n, bf(b))) \quad (5)$$

The logic behind this expression is that S starts n independent threads of the same form (note the factor n). For each thread, the expected number of messages is $1 + \mathcal{F}(n, bf(b))$ with probability b because R_i starts exactly the same process with lower ICF $bf(b)$. Follow the same procedures shown in Section 2.1.2, we can obtain $\mathcal{F}(n, 0) = 0$, $\mathcal{F}'(n, 0) = n$ and $\mathcal{F}''(n, 0) = 2n^2 f(0)$. When b is small, $\mathcal{F}(n, b)$ can be approximated by Taylor series to second order. $\mathcal{F}(n, b)$ can be calculated numerically as in Algorithm 2.

Algorithm 2 Numerical analysis of $\mathcal{F}(n, b)$

```

if  $b \leq 10^{-5}$  then
   $\mathcal{F}(n, b) \leftarrow nb + n^2 b^2 f(0)$ 
else
   $\mathcal{F}(n, b) \leftarrow nb(1 + \mathcal{F}(n, bf(b)))$ 
end if

```

2.3 Mixed Reply Process

A mixed reply process is a mixture of the single recipient and multiple recipient reply processes. In the mixed reply process, sender S initiates the root message to the recipient set $R = \{R_1, R_2, \dots, R_i, \dots, R_n\}$, where $n \geq 2$. R_i may take 3 actions: reply to the sender only ("Reply"), reply to the sender and all other recipients ("Reply All"), or not to reply. We assume that each recipient acts independently. We denote the probability of reply to the sender only as $p_1 = bp_1^*$, and the probability of reply to the sender and the other recipients as $p_2 = bp_2^*$, where p_1^* and p_2^* denote the

probability of reply using “Reply” and “Reply All” option respectively when the ICF is 1, and $p_1^* + p_2^* = 1$. The conversation among $\{S\} \cup R$ along a particular message path dies when a recipient fails to reply. The conversation ends when every message path dies.

2.3.1 Mixed Reply Process Example

Figure 3 illustrates an example of such a conversation between S and $R = \{R_1, R_2, R_3\}$. In this particular reply process, at depth 1 R_1 chooses “Reply All”, R_2 chooses “Not Reply”, and R_3 chooses “Reply”. We assume the ICF of the root message is b . The ICF, depth, probability of each message are indicated in the figure. For example, when S initiates a message to R_1, R_2 , and R_3 at *depth* = 0, the probability of replying to the sender only, $p_1 = bp_1^*$, and the probability of replying to the sender and the recipients, $p_2 = bp_2^*$, and the probability of no action is $1 - p_1 - p_2$, which is $1 - b$. The ICF of this replied message is $bf(b)$, and its depth is 1 in the reply process, and so on. What we should notice is that once a “Reply” option is chosen, the reply process followed will be the single recipient reply process.

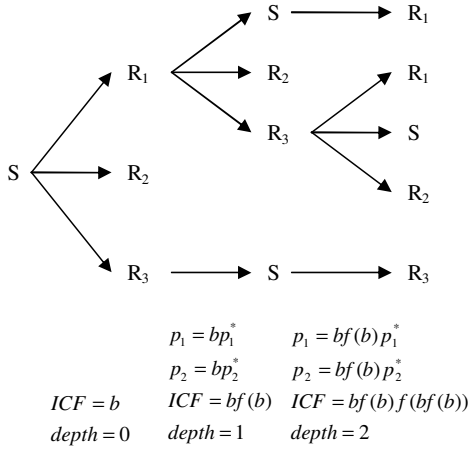


Figure 3: Mixed reply process

2.3.2 Expected Number of Replies

Assume sender S initiates a message M to a list of recipients R . Let $R = \{R_1, R_2, \dots, R_i, \dots, R_n\}$ denote the recipient set with n recipients. Let $R'_i = \{R_1, R_2, \dots, R_{i-1}, R_{i+1}, \dots, R_n\}$ denote $R \setminus \{R_i\}$. We assume that when a message is replied, two options, “Reply” and “Reply All”, are to be used, which correspond to “S” and $\{S\} \cup R'_i$ respectively as the recipient(s) in the reply message of R_i . Let p denote the probability of reply, $p = p_1 + p_2$, in which $p_1 = bp_1^*$ is the probability of reply using “Reply” option, and $p_2 = bp_2^*$ is the probability of reply using “Reply All” option.

Of the three actions, do nothing, Reply, Reply All, the last two actions may result in more replied messages. Figure 3 shows an example of the reply process. Take recipient R_3 as an example, at depth 1 R_3 chooses “Reply”, the reply process followed is the single recipient reply process; the probability of reply to a parent message is the summation of probability of “Reply” and “Reply All”, i.e., $p = bp_1^* + bp_2^* = b(p_1^* + p_2^*) = b$. On the other hand, R_1 chooses “Reply All”, the reply process followed is the recursive process with a

lower ICF.

Let Z denote number of replies of this process, and X denote number of replies when a “Reply” option is selected when multiple recipients exist. Let $\mathcal{G}(n, b) = E[Z|n, b]$ be the expected number of replies to the root message. Since the “Reply All” option leads to a recursive reply process with a lower ICF, $\mathcal{G}(n, b)$ is given recursively as

$$\mathcal{G}(n, b) = n\{bp_1^*(1 + \mathcal{E}(bf(b))) + bp_2^*(1 + \mathcal{G}(n, bf(b)))\} \quad (6)$$

The logic behind this expression is that S starts n independent threads. The term $bp_1^*(1 + \mathcal{E}(bf(b)))$ captures the expected number of messages for the single recipient reply process initiated by the “Reply” option at depth 1; the term $bp_2^*(1 + \mathcal{G}(n, bf(b)))$ captures the expected number of messages for the mixed reply process initiated by the “Reply All” option at depth 1 because an exactly same process with lower ICF $bf(b)$ is started.

Since $\mathcal{E}(b) = b(1 + \mathcal{E}(bf(b)))$ from (1), $\mathcal{G}(n, b)$ can be written as:

$$\mathcal{G}(n, b) = np_1^*\mathcal{E}(b) + nbp_2^* + nbp_2^*\mathcal{G}(n, bf(b)) \quad (7)$$

Again, we get an approximate recursion for $\mathcal{G}(n, b)$ using a second order Taylor expansion. We can obtain $\mathcal{G}(n, 0) = 0$, $\mathcal{G}'(n, 0) = n$ and $\mathcal{G}''(n, 0) = 2np_1^*f(0) + 2n^2p_2^*f(0)$. The calculation of $\mathcal{G}(n, b)$ is shown in Algorithm 3. The expected number of replies for any message in the thread can be obtained by replacing b with the corresponding ICF of the message.

Algorithm 3 Numerical analysis of $\mathcal{G}(n, b)$

```

if  $b \leq 10^{-5}$  then
     $\mathcal{G}(n, b) \leftarrow nb + nb^2p_1^*f(0) + n^2b^2p_2^*f(0)$ 
else
     $\mathcal{G}(n, b) \leftarrow np_1^*\mathcal{E}(b) + nbp_2^* + nbp_2^*\mathcal{G}(n, bf(b))$ 
end if

```

2.4 Statistical Determination of ICF

Given a thread for root message M , we would like to determine the ICF b of message M . We select b to match the observed tree. The approach we propose here is to select b to match the expected number of messages and the observed number of messages. This can be done at every level of the tree, treating each node as the root of the subtree-thread. The ICF of this subtree-root is determined from b and the ICF propagation function.

For a given root message with ICF b , assume the depth of the reply process is m , and there are n_i messages at each depth i . Let x_{ij} denote the total number of observed replies to the j^{th} message at depth i , b_i denote the ICF of messages at depth i , and $E[X_{ij}|n_{ij}, b_i]$ denote the expected number of replies to this message. We select b to minimize the summation of the squared difference between expected and real number of replies for every message in the reply process. Thus, we define the error function

$$\Sigma(b) = \sum_{i=0}^m \sum_{j=1}^{n_i} (x_{ij} - E[X_{ij}|n_{ij}, b_i])^2, \quad (8)$$

where b_i is a function of b as follows:

$$b_0 = b$$

$$b_i = b_{i-1}f(b_{i-1}) \quad i > 0$$

For each root message in an email data set, its conversation threads could be obtained by examining the parent-children relationship between emails. The observed number of replies to each message in the conversation thread is thus easy to obtain. Meanwhile, the expected number of replies for every message has been derived numerically above. The best fit b of the root message can thus be obtained from (8) using nonlinear optimization.

The ICF b is then selected as $\text{argmin } \Sigma(b)$. A second approach, which we postpone to a full version of this paper is to maximize the likelihood of the observed thread.

3. DETECTING BROADCASTS IN ENRON EMAILS

The methodology is applied to a subset of Enron emails to detect broadcast messages. A broadcast is defined as an email which is sent to multiple recipients, but the conversation triggered by this email dies down quickly. The purpose of detecting broadcasts is to eliminate the emails that inspire little or no interaction between sender and recipients and hence are misleading for Social Network Analysis. In this paper, only those root messages with 5 or more recipients are tested.

3.1 A Brief Description of Enron Email Data

Enron Corporation was founded in 1985. It became the seventh largest business organization in the USA in fifteen years [9, 11]. Enron's stock price was as high as \$90 in August of 2000, however, Enron declared bankruptcy in December 2001 without any warning [9, 11]. It was the biggest bankruptcy in American history at that time; thousands of people lost their jobs; and hundreds of thousands of people lost their retirement funds. After Enron's bankruptcy numerous investigations were conducted by authorities. Many employees' emails were also collected and released by Federal Energy Regulatory Commission (FERC) to the public for investigation [5]. Among the many interesting messages, it is expected to have many spam/broadcasts.

The data set we are testing on is extracted from the March 2, 2004 Version of Enron emails posted by Cohen [3]. The March 2, 2004 Version contains 517,431 messages dated from November 1998 to June 2002 organized into 150 employee folders. We identified 156 employees from this data set, and most of them were senior managers of Enron [14]. Because the communication among these 156 employees was our interest, 21,693 emails among these 156 employees were extracted from the March 2, 2004 Version. The conversation threads are constructed and tested by our methodology.

3.2 Constructing Email Threads

Methods for threading emails into conversations have been discussed in previous research [4, 7, 13]. Although it is argued that language processing should be applied to threading electronic messages [4, 7], we adopt a simpler but efficient method.

In an email system, usually two options, "Reply" and "Reply All", are available for replying a message. We ignore the slight possibility that neither of them is used in replying. We assume that when one of these two options is used to reply a message, the subject will not be changed except a "Re:" may be added. We examine the "Subject", "From", "To",

"Cc", and "Date" headers to construct the parent-children relationship. If the "Subject" header of a message contains "Re:", we consider it as a child message. To find its parent message, we compare header fields of two messages. If the "Reply" option is used, the recipient of the replied message will be the sender of the parent message; and if the "Reply All" option is used, the recipient of the replied message will be the sender and the other recipients of the parent message. For both options, the sender of the child message should be one of the recipients of the parent message. The "Date" field will be used as a time constraint to determine the parent-children relationship for any two messages, since the response time should not be long. We use 96 hours as the response time window. From the data set, 1116 threads in which the root message has 5 or more recipients are constructed.

3.3 Experiments

In this experiment, we assume the ICF propagation function $f(b) = f$ is a constant ranging from 0 to 1. Nine settings, $f \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$ are tested. The probabilities of "Reply" and "Reply All" when ICF is 1 (p_1^* and p_2^*) are approximated by their relative frequencies of having been used. It turns out p_1^* and p_2^* don't differ significantly, therefore, $p_1^* = p_2^* = 0.5$ is used in this experiment. We randomly select 50 threads as a training set, and another 50 as a testing set. For each thread in the test and training sets, we read the content of the email to determine if it is a broadcast message; if a message is to inform of a decision, a result, news, a meeting time, or anything that doesn't require a reply, we categorize it as a broadcast, otherwise it is considered as a normal message. The ICF of each thread at each f setting is calculated by minimizing (8). A threshold is then chosen to determine if a thread is a broadcast, i.e., a thread is a broadcast if the ICF of the thread is not larger than the threshold, and it is a normal message otherwise.

Let C and H be the variables indicating if a message is a broadcast from the statistical method and the content of the message respectively. They can be either 0 or 1, in which 0 represents normal message, and 1 represents broadcast. Let T denote the threshold. For any message i , C_i is defined as:

$$C_i(T) = \begin{cases} 0 & \text{if } ICF_i > T \\ 1 & \text{otherwise} \end{cases} \quad (9)$$

The error Err is defined as the cumulative absolute difference between C_i and H_i in (10). The threshold T is determined by minimizing the error function as shown in (11).

$$Err = \sum_{i=0}^{50} (C_i(T) - H_i)^2 \quad (10)$$

$$T^* = \text{argmin} \sum_{i=0}^{50} (C_i(T) - H_i)^2 \quad (11)$$

3.4 Results

The result shows that the magnitude of f doesn't effect the error Err greatly for a given threshold. Table 1 shows the nine f values with their corresponding optimal threshold T^* and error Err . When $f \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8\}$, the error Err is 2; however, when $f = 0.9$, Err is 3.

A linear relationship between f and T is clear from Figure 4. The regression analysis shows the adjusted $R^2 = 99.4\%$

Table 1: f and its corresponding threshold T

f	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
T^*	0.2813	0.2580	0.2395	0.2237	0.2092	0.1955	0.1809	0.1651	0.1480
Err	2	2	2	2	2	2	2	2	3

with slope -0.1596 and intercept 0.2910. Therefore, $T = 0.2910 - 0.1596 * f$ can be used to estimate the threshold for a given f value. However, we notice that the error when $f = 0.9$ is larger than the error when f takes the other eight values. We further investigated the effect of f on the ICF with an example.

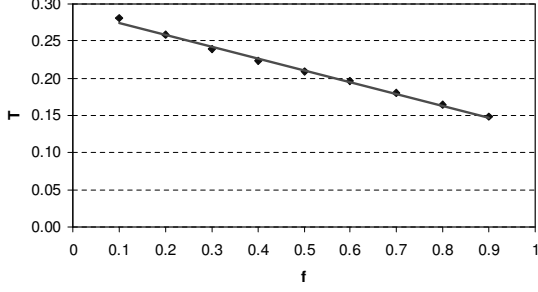


Figure 4: f and its associated threshold T

In this example four threads are illustrated in Figure 5. All of their root messages have 5 recipients, but the realized replies are different. Thread (a) shows one of the five recipients replies to the sender of the root message; thread (b) has two of the five recipients reply, one chooses “Reply” and the other chooses “Reply All” option; thread (c) shows one of the recipients replies to the sender, and the sender follows up with another message; thread (d) shows one of the recipients replies to the sender and the other recipients, and one of them follows up.

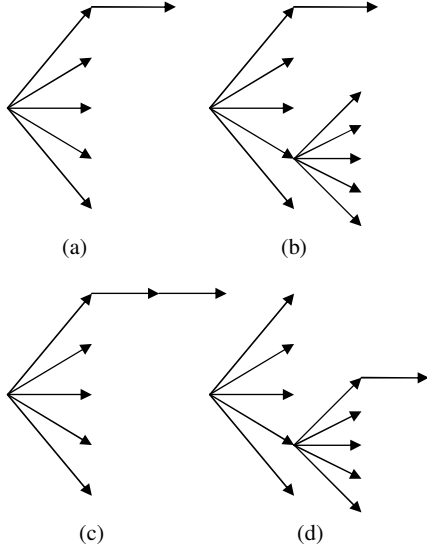


Figure 5: Four threads as an example

The ICFs for thread (a), (b), (c), and (d) are shown in Figure 6 for $f \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$. As

expected, the ICF decreases when f increases, and the rate of decrease is almost a constant for each thread. However, the decreasing rate varies from thread to thread. For instance, thread (a) has a flatter slope compared with (b), (c), and (d) as shown in Figure 6. As a result, the ICF of (a) is very close to that of (b) at $f = 0.9$ although they are quite apart from each other at $f = 0.1$ because thread (b) expects more replies when f is large. Thread (b), (c), and (d) cluster when f is small but separate when f is large. Threads (c) and (d) intersect when f is around 0.55 because the ICF of (d) is more sensitive with high f values. The difference in slopes is identified as the reason that error Err changes with f . Compared with two replies in (b), (c), and (d), thread (a) has only one reply; therefore, (a) should be distant from the others in terms of ICF. From Figure 6, we notice that when f is small, (a) is indeed separated from the other three threads, therefore, small f values are recommended.

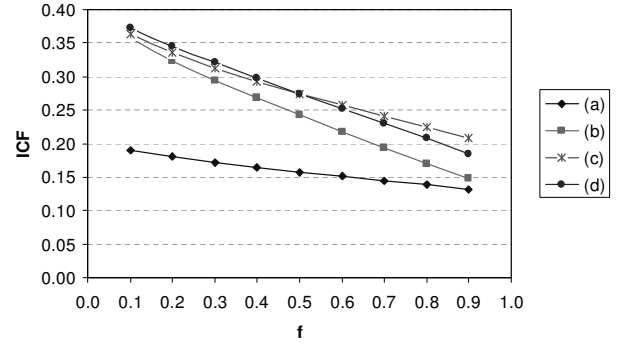


Figure 6: ICF vs. f for four threads

Another reason that we recommend small f values (≤ 0.3) is justified as follows. Consider the thread “ $A \rightarrow B \rightarrow A \rightarrow B$ ”, three emails between A and B. Communication patterns like this are very common in real life. Intuition suggests that the ICF of the root message should be high (≥ 0.9). Table 2 lists the f values and their corresponding ICFs for this thread. The table shows that when ICF is at least 0.9, $f \leq 0.3$. On the other hand, f cannot be too small since we also need to differentiate the interesting root messages which have triggered heated discussions. Therefore, $f \in [0.1, 0.3]$ is our recommendation.

The Relative Operating Characteristic, or ROC curve, is plotted in Figure 7, for $f = 0.1$, with the training set data. The X axis is the false positive rate (FPR), which means it is categorized as a normal message from the content but it is classified as a broadcast using our methodology; and the Y axis is true positive rate (TPR), which means the message is categorized as a broadcast from both the content and our methodology. The area under the ROC curve is larger than 90%, which proves that our methodology is very effective (on the training set). We applied the combination of $f = \{0.1, 0.2, 0.3\}$ and its associated

Table 2: f and its corresponding ICF for thread $A \rightarrow B \rightarrow A \rightarrow B$

f	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
ICF	0.9796	0.9396	0.8920	0.8395	0.7824	0.7200	0.6513	0.5754	0.4914

threshold $T^* = \{0.2813, 0.2580, 0.2395\}$ to the test data set, which produced 3 disagreement out of 50 threads with accuracy 94%. The confusion matrix is shown in Table 3, in which “B” represents Broadcast and “NB” represents non-broadcast. Since the error doesn’t deviate much from the error of the training data, our method is believed to be robust.

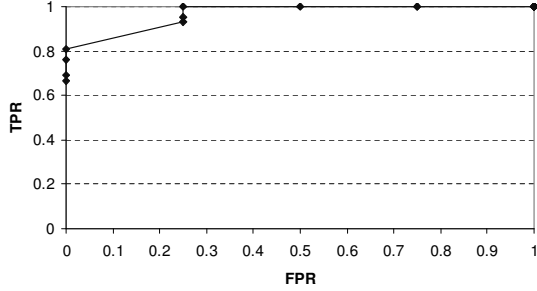


Figure 7: ROC curve of the training data when $f = 0.1$

Table 3: Confusion matrix for the test data

		Human	
		B	NB
our algorithm	B	41	2
	NB	1	6

4. CONCLUSIONS

We have developed a statistical method to evaluate how informative a message is by the conversation it triggered. This method is then applied to a subset of Enron email data to detect the broadcast messages. We conclude that the threshold to differentiate the broadcast from the normal message is a linear function of information decay factor f , and $f \in [0.1, 0.3]$ is recommended. The method is proved to be effective and robust in detecting broadcast messages by applying it on both the training and testing data. The proposed method, in general, helps to process the data for various analyses and achieve a better understanding of the data. Our future research includes applying the methodology to detecting interesting topics from conversation threads.

5. ACKNOWLEDGEMENTS

This material is based upon work partially supported by the National Science Foundation under Grants No. 0324947, No. 0323324, No. 0634875, No. 0522672, and by the ONR Grant No. N00014-06-1-0466. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or U.S. Government.

6. REFERENCES

- [1] W. Antweiler and M. Z. Frank. Is all that talk just noise? the information content of internet stock message boards. *Journal of Finance*, 59(3):1259–1294, 2004.
- [2] H. Berghel. Cyberspace 2000: dealing with information overload. *Communications of the ACM*, 40(2):19–24, 1997.
- [3] W. W. Cohen. Enron email dataset. <http://www.cs.cmu.edu/~enron/>.
- [4] D. E. Comer and L. L. Peterson. Conversation-based mail. *ACM Transactions on Computer Systems*, 4(4):299–319, 1986.
- [5] Federal Energy Regulatory Commission. Addressing the 2000-2001 western energy crisis. <http://www.ferc.gov/industries/electric/indus-act/wec/enron/info-release.asp>.
- [6] B. Gu, P. Konana, A. Liu, B. Rajagopalan, and J. Ghosh. Predictive value of stock message board sentiments. In *the Social Science Research Network Electronic Paper Collection*. Social Science Electronic Publishing, Inc., 2007.
- [7] D. D. Lewis and K. A. Knowles. Threading electronic mail: A preliminary study. *Information Processing and Management*, 33(2):209–217, 1997.
- [8] R. M. Losee, Jr. Minimizing information overload: the ranking of electronic messages. *Journal of Information Science*, 15(3):179–189, 1989.
- [9] B. McLean and P. Elkind. *Smartest Guys in the Room: The Amazing Rise and Scandalous Fall of Enron*. Portfolio, 1st edition, 2003.
- [10] C. E. Shannon and W. Weaver. *The Mathematical Theory of Communication*. University of Illinois Press, Urbana, IL, 1st edition, 1949.
- [11] M. Swartz and S. Watkins. *Power Failure: The Inside Story of the Collapse of Enron*. Doubleday, 1st edition, 2003.
- [12] R. Tumarkin and R. Whitelaw. News or noise? internet message board activity and stock prices. *Financial Analysts Journal*, 57:41–51, 2001.
- [13] G. D. Venolia and C. Neustaedter. Understanding sequence and reply relationships within email conversations: a mixed-model visualization. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 361–368, New York, NY, 2003. ACM Press.
- [14] Y. Zhou, M. Goldberg, M. Magdon-Ismael, and W. A. Wallace. Strategies for cleaning organizational emails with an application to enron email dataset. In *5th Conference of North American Association for Computational Social and Organizational Science*, Emory, Atlanta, GA, June 7–9 2007.