Email mining: Tasks, common techniques, and tools

Article *in* Knowledge and Information Systems · October 2013

DOI: 10.1007/s10115-013-0658-2

Guanting Tang
Simon Fraser University

Knowl Inf Syst
DOI 10.1007/s10115-013-0658-2

SURVEY PAPER

Email mining: tasks, common techniques, and tools

Guanting Tang · Jian Pei · Wo-Shun Luk

Received: 17 January 2012 / Revised: 4 March 2013 / Accepted: 6 April 2013

Abstract Email is one of the most popular forms of communication nowadays, mainly due to its efficiency, low cost, and compatibility of diversified types of information. In order to facilitate better usage of emails and explore business potentials in emailing, various data mining techniques have been applied on email data. In this paper, we present a brief survey of the major research efforts on email mining. To emphasize the differences between email mining and general text mining, we organize our survey on five major email mining tasks, namely spam detection, email categorization, contact analysis, email network property analysis and email visualization. Those tasks are inherently incorporated into various usages of emails. We systematically review the commonly used techniques and also discuss the related software tools available.

Keywords Email · Data mining · Tools · Classification · Clustering · Social network analysis

1 Introduction

Emails exist for only about 50 years since MIT's "Compatible Time-Sharing System"(or "CTSS MAIL" [84]) in early 1960s, which was designed for multiple users logging into a central system to share and store documents from remote terminals. The popularity of emails grows at a terrific speed due to its high efficiency, extremely low cost and compatibility with many different types of information. As one of the most widespread communication approaches nowadays, emails are broadly used in our daily lives. For example, co-workers discuss work through emails; friends share social activities and experiences via emails; business companies distribute advertisements by emails. Radicati and Hoang [60] reported, "The number of worldwide email accounts is expected to increase from an installed base of 3.1 billion in 2011 to nearly 4.1 billion by year-end 2015." Yarow [91] reported that 1.9 billion emailers sent 107 trillion emails in the first quarter of 2010, on average 294 billion emailsper day.

G. Tang · J. Pei (✉) · W.-S. Luk
School of Computing Science, Simon Fraser University, Burnaby, BC, Canada
e-mail: jpei@cs.sfu.ca

Published online: 04 June 2013

In order to facilitate better usages of emails and explore business potential in emailing, email mining, which applies data mining techniques on emails, has been conducted extensively and achieved remarkable progress in both research and practice. Particularly, emails can be regarded as a mixed information cabinet containing both textual data and human social, organizational relations.

Email content as textual and non-textual data Emails distinguish themselves from general text data in many documents in two aspects.

- Emails are often much shorter and more briefly written than many other documents, such as stories and user manuals. Emails often contain some faddish words or abbreviations that may not appear in traditional dictionaries. Standard text mining techniques may not be effective when they are applied to email mining tasks.
- In addition to textual data, emails may contain richer types of data, such as URL links, HTML markups and pictures. Although some studies, such as [1,2,30,52,56], simply delete those non-textual data entities in the data preprocessing stage, those richer types of data may be useful in certain tasks, such as email spam detection. To fully take advantage of those non-textual data in emails is an interesting and challenging problem.

Emails representing human social, organizational relations The emailing activities themselves represent rich human social and organization relations, which connect people into communities and complex systems. Understanding the organizational structures or relationships among people within a big organization can be very useful in real life. One example is the investigation of the bankruptcy cases of large companies, such as Enron in 2001 and WorldCom in 2006. Investigating these scandals increases the demand for studying human relations from email corpus, which can help to understand the relationships among people, identify the key roles and support forensic analysis in law enforcement. Another example is how to find resources of interest like people with special expertise in a big organization. Campbell et al. [15] mentioned, “email is a valuable source of expertise.” Automatic expertise discovery using emails can improve organizational efficiency.

Five major tasks have been well investigated in email mining, namely spam detection, email categorization, contact analysis, email network property analysis and email visualization.

Spam detection is to detect unsolicited bulk emails. It is the most important task in email mining. According to [58], “the US workforce loses more than $71 billion a year in lost productivity to managing spam.” Effectively detecting spam emails not only can reduce financial losses, but also can improve email users’ satisfactions. Spam detection methods can be divided into two categories: detecting from email contents and detecting from email senders.

Email categorization (or email filing) is to organize emails into different categories. Categorizing emails manually becomes a heavy burden for email users when the amount of emails grows fast. Dabbish and Kraut [25] showed that the lack of good email categorization leads to massive negative effects on both personal and organizational performance. A good automatic email categorization tool can save people’s effort in organizing and finding emails.

Contact analysis is to identify special email contacts or groups of email contacts by analyzing the contacts’ characteristics from the email contents or email social relations. It can be further divided into two subtasks: contact identification and contact categorization.

- *Contact identification* is to find email contacts with special characteristics. It can serve various purposes. For example, it can help to find people with special expertise in a big

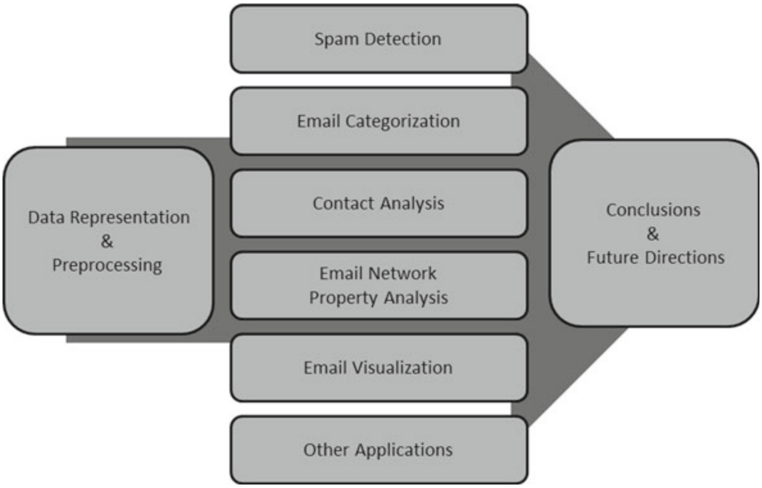

Spam Detection
Email Categorization
Contact Analysis
Email Network Property Analysis
Email Visualization
Other Applications
Data Representation & Preprocessing
Conclusions & Future Directions

Fig. 1 Framework

organization or distinguish whether the sender of a suspicious email is the email account owner himself or somebody else with malicious intention.

- *Contact categorization* is to assign email contacts into groups so that the contacts within each group have certain common characteristics. It is useful in practice. For instance, the groups of users found by contact categorization can be used to suggest email receivers when composing emails. Categorizing email contact by email exchange frequencies and email contents are the most popular ways.

Email network property analysis is to analyze the critical properties of an email network, such as general network structures, relation strength and organizational structures. Due to the lack of large public email corpus, this is a direction not well explored. Since the US Federal Energy Commission made the Enron email corpus public during the company crisis investigation in 2000, more and more studies have been conducted. Understanding email network properties can provide us information about how people communicate with each other in large communities.

Email visualization is to use visualization techniques to help users identify, retrieve and summarize useful information hidden in numerous emails. Email visualization can help to tackle the challenges like how to optimize email interface to provide users more convenience and how to display useful hidden information in one's email account. A good user interface design can provide users understandable access to their email accounts and benefit users in synthesizing information and deriving insights from emails.

This survey is organized mainly according to email mining tasks. The framework is shown in Fig. 1. For each major mining task, we review the commonly used techniques and software tools when available. Before pursuing any tasks, we describe the methods used in data representation and preprocessing. We wrap up the survey by discussing some possible future research directions.

We are aware of several existing surveys or reviews on the topic. There are three critical differences between our work and the existing surveys.

- We review *all major email mining tasks* and the corresponding techniques and software tools. Some existing surveys only focus on one specific task. For example, Boykin

Table 1 Tasks discussed in the existing survey papers and ours

Paper ID	Spam detection	Email categorization	Contact analysis	Email network property analysis	Email visualization	Other tasks
[44]	X	X	X			X
[31]	X	X	X			X
[10]	X					
[8]	X					
[48]		X				
[86]		X				
Ours	X	X	X	X	X	X

and Roychowdhury [10] and Blanzieri and Bryl [8] compared different techniques and their evaluation methods used in spam detection. Koprinska et al. [48] and Wang and Cloete [86] reviewed methods applied in email categorization.

- As shown in Fig. 1, we organize our survey in a *task oriented way*, which is different from the previous surveys. Katakis et al. [44] organized their review based on mining techniques. They briefly introduced email mining techniques and applications separately. Ducheneaut and Watts [31] organized their paper according to email usages. They identified three major email usages as metaphors and reviewed the practical requirements, research problems and existing solutions for each metaphor.
- We include *email network property analysis* and *email visualization* in this survey, which are more and more interesting and important aspects. To the best of our knowledge, none of the existing reviews discuss these issues.

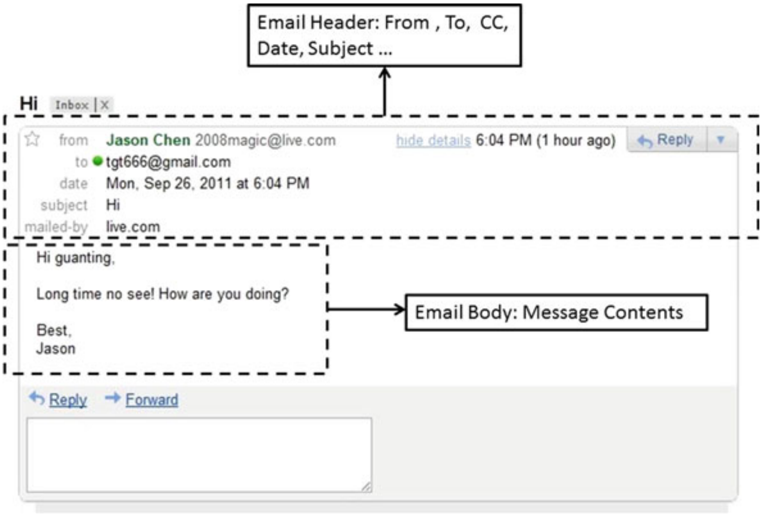
Table 1 shows the tasks discussed in the existing survey papers as well as ours. As can be seen, our paper has a larger coverage than the others and covers more recent studies.

The rest of the paper is organized as follows. In Sect. 2, we discuss email data representation and preprocessing. The five major email mining tasks are discussed in Sects. 3, 4, 5, 6 and 7, respectively. Section 8 reviews some other email mining tasks. Section 9 concludes the paper and discusses some future directions.

2 Data representation and preprocessing

An email message contains two parts: the header and the body, as exemplified in Fig. 2. The header part consists of a set of fields, such as "From," "To," "CC," "Subject" and "Date." Different email service providers have different sets of fields for displaying the header part. The "From," "To" and "Subject" fields are common. Usually, the body part is made of unstructured text, sometimes together with graphic elements, URL links, HTML markups and attachments. Data representation here refers to the appropriate methods used to record the information in emails, such as sender/receiver information, subjects and contents. A suitable representation serves better in the data analysis stage and also helps to improve the results of the mining task. Data representation is often a part of the data preprocessing step, which processes"raw" emails, removes noise, such as stop words, from original emails to facilitate the mining tasks.

In this section, we discuss two common approaches for email representation, namely the feature-based approach and the social structure-based approach. We will also briefly review the major data preprocessing methods for emails.


Email Header: From , To, CC, Date, Subject ...
Hi
Inbox | X
hide details
Reply
Email Body: Message Contents
Reply
Forward

from Jason Chen 2008magic@live.com 6:04 PM (1 hour ago)
to tgt666@gmail.com
date Mon, Sep 26, 2011 at 6:04 PM
subject Hi
mailed-by live.com

Hi guanting,

Long time no see! How are you doing?

Best,
Jason

Fig. 2 Email message structure

2.1 The feature-based approach

The feature-based approach represents an email using some features. The most prevalent method is the vector space model [68]. An email is presented as a vector. The dimensions are a set of features extracted from the email. Formally, an email e_i with n features can be represented as a vector $e_i = [w_{i,1}, w_{i,2}, \ldots, w_{i,n}]$, where $w_{i,j}$ $(1 \leq j \leq n)$ is the value on a particular feature.

The most commonly used features include the following types.

- Terms are usually extracted from the “Subject” field and the email body. The terms can be single words, phrases or n-grams. Standard document preprocessing steps, such as removing stop words and stemming, are applied on emails, too. Some previous studies [2,14,30,52,56,92] treat words extracted from email bodies as the features, after removing stop words and stemming by standard methods, such as the Porter stemmer algorithm [81]. In order to present email contents in a more concise and effective way, there are some works that focus on feature reduction. For example, Gomez et al. [37] proposed a framework to extract highly discriminative features. Methods based on principal component analysis (“PCA”) [42] and latent dirichlet allocation model (“LDA”) [9] are designed to select the discriminative features.
- Structure information statistics refer to the statistics that can reflect the email senders’ writing styles. For example, structural statistics, such as total number of unique words, special tokens and average sentence lengths, are used [22,27,48,50].
- Values of email header fields can be used as features. The values of fields like “From,” “To” and “CC” are used to represent email users most of the time. Some studies directly use the original email addresses. For example, Campbell et al. [15], Rowe et al. [65] and Roth et al. [64] treated different email addresses as unique users. Some studies tokenize and analyze the email addresses before using them to represent users. For instance, Schwartz and Wood [71] extracted usernames from email addresses. Users with the same username are represented as one user. Sometimes, the values of the “IP Addresses” field are used

to represent users, too. For example, Taylor [78] took unique IP addresses as different senders. The values of "Date" field can be used as a criteria for spam detection. It includes the date and local time when an email is sent. For instance, Stolfo et al. [76] treated the average number of email messages each user sends in different periods (day, evening, and nights) of a day as features to track the user's abnormal behaviors.

The features can be of binary values or weights.

- Binary values ("$0/1$") are used to indicate the presence of a particular feature. For example, Sahami et al. [66] and Androutsopoulos et al. [2] used a binary variable to indicate whether a word appears in an email or not.
- Weights are used to emphasize the importance of the features. The term frequency-inverse document frequency is a popular weighting scheme. Term frequency ("TF") [74] captures the importance of a term to a document. In email mining, it is the total number of times a term appears in an email. Term frequency-inverse document frequency ("TF·IDF") [67] reflects the importance of a term to a document in a document corpus. Given a term t_j in an email e_i from an email corpus E, let $TF(t_j, e_i)$ be the term frequency of t_j in e_i, $|E|$ be the total number of emails in E, and EF_{t_j} be the total number of emails t_j appears. The TF·IDF function of t_j in e_j can be written as

$$\text{TF} \cdot \text{IDF}(t_j, e_i) = \text{TF}(t_j, e_i) \log \frac{|E|}{EF_{t_j}}.$$

  For example, Cohen [18] and Segal and Kephart [72] used the TF weighting scheme. Sasaki and Shinnou [69] used the TF·IDF weighting scheme. Drucker et al. [30] used both of them to present the importance of the term features.

2.2 The social structure-based approach

As we said in Sect. 1, emailing activities represent human social, organizational relations. The social structure-based approaches extract a social network formed by emailing activities.

Graphs are widely used in modeling social network structures. They are also used in modeling email social networks. An email corpus can be modeled as a social network graph $G = (V, E)$, where V is the set of email addresses as nodes and E is the email interactions as edges. For example, Tyler et al. [80] built an undirected, unweighted graph from the headers of email logs to model the email network structure. The nodes are the email addresses. An edge is created if the number of emails between two nodes is greater than a pre-defined threshold.

We can add attributes to the graph when we want to include more information, such as the importance of a person, the strength of the relationship between two people and the most common topics between two people.

Particularly, the importance of a person can be presented as the score of a node. The scoring function should be designed by considering the related factors, such as the total number of emails the node receives, the average response time of an email sent by the node and the total number of cliques the node involved. For instance, Rowe et al. [65] measured the social importance of a person by a social score, which is a weighted combination of some factors, such as the number of emails and the average response time.

The strength of the relationship between two people in emails is often presented as the weight of an edge. The weighting function should be designed by considering the related factors, such as the frequency of two people exchanging emails and the time when the email interaction happens. For example, Roth et al. [64] proposed an "interaction rank" to assign an edge weight, which indicates how well two people are connected in a certain community.

In interaction rank, each email is weighted by a function about the time when the email is sent. The interaction rank is calculated by summing up all the weights of emails.

Common topics between two people can be learned from the content of emails between them. The common topics can be used to support email mining tasks, such as expert finding. The topics are usually generated by clustering methods or by topic models. For example, Mccallum et al. [53] built the author-recipient-topic model on top of the LDA model to generate topics for different email sender–receiver pairs.

We can add edge directions to a graph when we want to consider the emails sent and received by a user, respectively. Usually, a directed graph is constructed, the edges are pointed from the email senders to the receivers. The in-/out-degrees of the nodes can be used in many different ways. For example, Lam and Yeung [49] used the in-/out-degrees as a factor in classifying the spam emails. Yoo et al. [92] used the in-/out-degrees as a factor to measure the social importance of a node.

3 Spam detection

Spamming emails are the unwanted emails, which are sent to a large number of email accounts [89]. Spamming emails are also known as "unsolicited bulk emails" [1] or junk emails. Cormack and Lynam [21] defined email spam as "unsolicited, unwanted emails that were sent indiscriminately, directly or indirectly, by a sender having no current relationship with the recipient." Commercial advertisement email is one of the most common spams. More and more business companies choose email as a way to distribute advertisements, since sending emails costs very little comparing to other channels.

There are numerous spamming emails being sent every day. *Internet Threats Trend Report Q1 2010* [39] reported, "spam levels averaged 83 % of all email traffic" in the first quarter of 2010. Claburn [17] mentioned, "the cost of spam in terms of lost productivity has reached 21.58 billion dollars annually" in 2004. This cost should be much higher today considering the rapidly growing speed of spamming. Spam detection becomes a must have task for email service providers.

In the context of email mining, spam detection is to identify unsolicited bulk emails using data mining techniques. In general, based on the information mainly used, spam detection methods can be divided into two categories, namely content-based detection and sender-based detection. Content-based detection is to identify spamming emails according to the email content. Sender-based detection is to identify spamming emails using the email sender information. We will review the techniques used in the two categorizes, respectively, in the following two subsections.

3.1 Content-based detection

Classification and semi-supervised clustering are the most often used techniques in content-based spam detection.

3.1.1 Classification methods

The classification problem in email content detection can be defined as follows. Given a training email corpus $E = \{(e_1, l_1), (e_2, l_2), \dots, (e_n, l_n)\}$, where e_i $(1 \leq i \leq n)$ is an email, l_i $(l_i \in \{\text{spam, non-spam}\})$ is the label for e_i, we want to build a classifier that can predict whether an unlabeled email is spam or not. An email e_i here is often represented as a vector.

The dimensions of the vector are the term features extracted from the email. Other features may be used in addition to the term features sometimes.

Naïve Bayes classifiers The naïve Bayes classifiers [41] have been widely used in email spam detection. The first attempt was made by [66]. The naïve Bayes methods assume that the values of the features are statistically independent from each other.

Since the cost of misclassifying a non-spam email as a spam one is much higher than the other way, people often label an email as spam using a substantially high confidence threshold. For example, Sahami et al. [66] put an email in the spam class if the probability of being spam is $>99.9\,\%$. Androutsopoulos et al. [1,2] set up a threshold λ. An email is spam if $\frac{P(\text{spam})}{P(\text{non-spam})} > \lambda$, where $P(\text{spam})$ and $P(\text{non-spam})$ are the probabilities that the email is spam and non-spam, respectively. They set λ to different values (1, 9, and 999) in their experiments. The results show that the larger λ, the higher spam detection precision. The spam detection precision is about 89–91 % when λ is set to be 1; it can be about 99 % when λ is set to be 999.

In addition to using the terms extracted from the emails as the features, the precision of the naïve Bayes classifiers in spam prediction can be further improved by adding some other features, such as whether an email has attachments or not, and the domain of an email address. Sahami et al. [66] compared three different feature settings, namely “words only,” “words + phrase” and “words + phrases + other features.” The experimental results show that the “words only” feature setting can predict more than 97 % spam emails correctly. The precision can be further improved to 100 % by adding phrases as features and some other features.

Support vector machines classifiers The support vector machines (SVM) classifiers [23] are binary. In email spam detection, emails are separated into two classes (“spam” and “non-spam”) by a hyperplane. The goal is to find a hyperplane, which can maximize the margin between the spam and non-spam classes.

Rios and Zha [63] compared the SVM, random forest and naïve Bayes classifiers. The SVM and random forest classifiers show comparable performance in most cases and are better than the naïve Bayes classifiers. Experiments were conducted to test the effectiveness of the three techniques, at relatively low false positive rates. Emails used in the experiments were extracted from a range of public sources and private corpora. Their results show that, when setting the false positive rate threshold to 1 %, the true positive rates of both SVM and random forest classifiers are consistently $>70\,\%$, and the true positive rate of naïve Bayes classifiers varies from <50–85 %. Drucker et al. [30] discovered from their experiments that the SVM classifiers using binary feature values have better performance than Ripper, Rocchio and boosting decision tree classifiers. The purpose of their experiments is to find a classifier with the lowest error rate. Messages used in the experiments were collected from AT&T staffs’ emails. Experimental results show that the SVM classifier using binary feature values has an error rate of about 2 %, while the error rates of the other classifiers are more than 3 %.

Rule-based classifiers The major idea of the rule-based classifiers is to classify the emails by a set of “IF-THEN” rules. Different from other classifiers, feature vectors of emails are not required here. Drucker et al. [30] used Ripper [19] to induce the classification rules. An example rule used is

IF “word FREE appears in subject” OR “word !!!! appears in subject”
THEN “the email is spam.”

Ripper has false alarm rates about 8–17 % while setting miss rate to 5 %. False alarm rate is defined as the number misclassified spam samples divided by the total number of spam samples. Miss rate is defined as the number misclassified non-spam samples divided by the total number of non-spam samples.

Androutsopoulos et al. [1] built a "keyword-based" classifier, which searches for the special patterns in the subject filed or the body field of the email. They identified 58 patterns for the spam emails, such as

body contains ',000' AND body contains '!!' AND body contains '$'.

Emails containing those patterns are considered spamming. This classifier can achieve a spam precision of 95 %.

Other classifiers Stolfo et al. [75] proposed the content-based classifier, which computes the distances between an unlabeled email and the spam/non-spam classes. The email is assigned the same label as the label of its closer class. An email is represented as a vector. The distance between an email and a class is defined as the cosine distance between the email vector and the centroid vector of the training emails in that class. The accuracy of the method varies from 70 to 94 % when different parts of emails are used as training/testing sets.

Rios and Zha [63] used a *random forest classifier* [12] to classify the spamming emails. A random forest is made up of many classification trees [13]. The k-th classification tree is a classifier denoted by $h(e_u, \Theta_k)$, where e_u is an unlabeled input email vector, Θ_k is a randomly generated vector. Θ_k is generated by selecting random features of the training emails for each node. The Θs of different classification trees in the forest are independent of each other, but are generated by the same distribution. For an unlabeled email, each classification tree provides a prediction, either "spam" or "non-spam." It is called a vote. The label receiving more votes is assigned to the unlabeled email.

3.1.2 Semi-supervised clustering methods

The semi-supervised clustering problem in content-based spam detection can be defined as follows. Given an email corpus $E = \{(e_1, l_1), (e_2, l_2), \ldots, (e_n, l_n)\}$, where e_i $(1 \le i \le \text{n})$ is an email, l_i ($l_i \in$ {spam, non-spam}) is the optional label for e_i, we want to partition the emails into clusters, which are labeled by "spam" or "non-spam." For an unlabeled email, we want to predict its label according to the label of the cluster it belongs to. The email e_i $(1 \le i \le \text{n})$ is represented as a vector. The dimensions of the vector are the term features extracted from the email. Other features may be used in addition sometimes.

The K-means algorithm [51] is one of the widely used clustering algorithms. The idea of the k-means algorithm can be described as follows. Initially, the algorithm randomly picks k instances as the k cluster centers. Then, it assigns the remaining instances to the nearest cluster centers and calculates the new center candidate for each cluster after that. The algorithm repeats the instance assignment and center reevaluation step until the cluster centers stabilize. The cosine distance is used here.

Sasaki and Shinnou [69] proposed a spam detection system. The spherical k-means algorithm [29] is used to cluster all the existing emails. For a new incoming email, the cosine distances between the email and the cluster centers are calculated. The label of the closest cluster will be assigned to the email. The label of a cluster is decided according to the ratio of the spamming emails in it. A cluster is considered as a spamming one if the ratio of spamming emails is over 70 %.

Notes The classification methods applied in content-based detection use a training email corpus to build classifiers. The semi-supervised clustering methods use a limited number of labeled emails to assign labels to the unlabeled ones by leveraging email clusters. Comparing to the classification methods, the semi-supervised clustering methods are more adaptive to new data, since the clusters are updated each time after a label is assigned to a new email.

3.2 Sender-based detection

The classification, semi-supervised clustering and email sender reputation analysis are the most often used techniques in sender-based spam detection.

3.2.1 Classification methods

The classification problem in sender-based spam detection is similar to the content-based detection problem. The difference between them is the features used for classification. In sender-based detection, the email sender's information, such as the writing style and the user name of the email sender, is used as the major features. In content-based detection, terms extracted from the emails are the major features.

***K*-nearest neighbor (K-NN) classifiers** The idea of the *k*-nearest neighbor (K-NN) classifiers [73] used here is as follows. For an unlabeled email, the classifier searches for the *k* nearest training emails according to a certain distance function. Then, the unlabeled email is given the same label of the class, to which most of the *k* nearest training emails belong.

Lam and Yeung [49] proposed a K-NN classification-based approach for spam detection. The sender features, such as the numbers of emails received and sent by the email user, respectively, and the number of interactive neighbors a user has, are extracted from a social network built from the email logs. A Gaussian function-based similarity function is used to calculate the similarity score between two senders. The mean K-NN similarity score used to label an unlabeled email is the mean of the distances between the email sender and her k-nearest neighbors. The positive/negative sign of the score can be used to classify whether the email is spam or not. The magnitude can be used to show the confidence of the classification. Emails with high scores are more likely to be non-spam. Experiment results show that with 3 % of the senders being labeled, the detection rate can be 99 and only 0.5 % of false positives.

Naïve Bayes classifiers Stolfo et al. [75] used the naïve Bayes classifier [41] as a component of the malicious email tracking (EMT) system, which classifies an unlabeled email to be malicious or not. Features about users, such as the domain name from the email address and the size of the email body, are used.

3.2.2 Semi-supervised clustering methods

The semi-supervised clustering problem in sender-based spam detection is similar to the content-based detection problem. The difference between them is the features used for clustering. In content-based detection, terms extracted from emails are the major features. In sender-based detection, email senders' information, such as user names from email addresses, is used as the major features. Sometimes, people use a distance threshold to ensure sufficient similarities among the emails within one cluster.

Based on the assumption that both the spamming and the non-spamming email senders have a list of people whom they often contact with, Gomes et al. [36] proposed a novel distance-based clustering algorithm to cluster the spamming email senders/receivers. Each email sender is represented as a binary vector, where the dimensions are all the contacts. The distance between two senders is calculated by the cosine similarity. The distance between a sender and a cluster is then defined as the distance between the sender and the centroid of the cluster. The email sender will be assigned to the closest cluster. The email receivers will be clustered in the same way. For a cluster, the probability of sending/receiving spamming emails is calculated by the average probability of sending/receiving spamming emails of all the nodes in that cluster. It is used as the spam probability of the emails sent/received by the people in the cluster. Experiment results show this method has a precision above 60 % with a small portion (0.27 %) of emails being analyzed.

3.2.3 Email sender reputation analysis methods

The email sender reputation analysis here refers to detecting spamming emails using the email senders' reputation scores, which can be inferred from a reputation network or calculated by some other factors, such as the IP address the email is sent from.

Golbek and Hendler [34] propose an email scoring mechanism based on a reputation social network to show the spam probability of an email. People in the reputation network can rate others' reputation. Each person is connected to the ones she/he rated in the past. The reputation score of an individual is calculated by a weighted average of the neighbors' reputation ratings. Experiment results show the precision of this method is above 82 %. Taylor [78] describes a simplified version of the spam detection method used by Gmail. An event log is used to record all the existing emails' labels ("spam" or "non-spam"). An email sender's reputation score is calculated by the percentage of non-spamming emails sent by this sender. The emails sent by senders with good reputation scores are more likely to be non-spam.

Notes Comparing to the classification and semi-supervised clustering methods, the sender reputation analysis methods can be subjective in some cases. For example, in the reputation network used by Golbek and Hendler [34], people rate others' reputation. Since the rating process is subjective, the reputation score used to determine whether an email is spam or not is subjective, too.

3.3 Software tools

Golbek and Hendler [34] developed TrustMail, an email client prototype. It shows an email's reputation score in a folder view, in addition to other standard fields, such as subject and date. A folder view is the user interface when an email user opens the email client. The email's reputation score can help email users to know whether an email is spam or not even before opening the message. The system can also help email users to judge the importance of an email from an unfamiliar sender by the email's reputation score.

3.4 Challenges

In terms of email contents, spamming emails disguise themselves by various forms, such as using lots of spamming URL links, pictures and attaching large size attachments. These forms may also change over time for better cloaking. It is hard to find a universal set of features to describe spamming (or non-spamming) emails. It is difficult to achieve a classifier good for all emails, either. Some studies compare selections of features and classifiers. For example, Androutsopoulos et al. [2] compared different feature settings between two classifiers on a personal email data set. Cormack and Lynam [20] performed a study of 11 spam detection classifiers on a personal email corpus. The results show classifiers with different feature settings have quite different performance.

In terms of email senders, although a black-white list is an efficient way with high accuracy in identifying spammers, it only works for the contacts appeared in the list. Some studies try to extend the black-white list concept by using sender reputations. For example, Taylor [78] proposed to calculate a sender's reputation score. Golbek and Hendler [34] built a reputation network to help users to determine if a sender is a spammer or not. Those methods, however, may have difficulty in determining a brand new sender or a disguised spammer.

4 Email categorization

Email categorization is to assign emails into different categories according to some conditions. Neustaedter et al. [57] defined it as "the process of going through unhandled emails and deciding what to do with them."

Bälter [3] pointed out that email users are daunted by categorizing emails, as more and more unorganized emails piled up in their mailboxes. Bellotti et al. [4] showed that about 10 % of the time people spending on emails is used to categorize email messages. Large amounts of uncategorized emails affect both the personal and organizational performance. It is in great need to develop some automatic methods that can help people to categorize their emails properly.

We will discuss the major techniques used in email categorization in this section.

4.1 Classification methods

The classification problem in email categorization is similar to the one in spam detection. The difference between them is the number of classes that a classifier has to predict. In spam detection, emails only have two classes, "spam" and "non-spam." In email categorization, however, emails usually have more classes, such as "work" and "entertainment."

TF·IDF classifiers The TF·IDF classifiers [67] are one of the most popular classifiers used in email categorization. The major idea is to calculate the distances between an uncategorized email and all the existing categories, and then assign the email to the category with the shortest distance. TF·IDF is used as the similarity measure.

Segal and Kephart [72] proposed a TF·IDF classifier. For an uncategorized email, the classifier suggests the top *n* categories that it most likely belongs to. An email is represented as a word-frequency vector; and a category is represented by a centroid vector, which is a weighted word-frequency vector calculated using the TF·IDF principle. The similarity score between an uncategorized email and a category is calculated by a variation of cosine distance, named $SIM4$ [67]. Experiments were conducted to evaluate the performance of the classifier. Email accounts from 6 researchers at IBM Thomas J. Watson Research Center were used. The results show that the accuracies of TF·IDF classifier are in the range of 60–90 %.

Cohen [18] proposed a different TF·IDF classifier. An email is represented as a weighted vector. The value of each dimension is calculated based on the TF·IDF weight of a term. A category is represented as a vector by summing up the emails in that category and subtracting the emails in the "mis-classified" folder of the user's existing email account. The similarity score between an uncategorized email and a category is calculated by the inner product of the email vector and the category vector. A threshold t_c is used to control the error on the existing emails of the email account. A newly incoming email is classified into a category if the similarity score between the email and the category is $< t_c$. Three different email corpora, which were obtained from the author's own email account, were used in experiments. The results show that the classifier can obtain an error rate lower than 7 % with more than 200 training examples.

Naïve Bayes classifiers McCallum and Nigam [54] used the generative models and the naïve Bayes classifiers. There are two kinds of generative models: the multi-variate Bernoulli model and the multinomial model. Both of them assume that emails are generated by a parametric model, in which each word is generated by a certain probability. The parameter of the model is calculated from the training data. An unlabeled email is assigned to a category that has the highest probability to generate it. The difference between the two models is the approaches of presenting the values of the email vector's dimensions: the multi-variate Bernoulli model uses the binary values and the multinomial model uses the weights, such TF·IDF weighting.

Rennie [61] used a multinomial model and a naïve Bayes classifier to tackle email categorization. All emails are assumed to be generated by a multinomial model parameterized by θ. An uncategorized email will be assigned to a category that has the highest probability to generate it. A Naïve Bayes classifier is used. Experiments were carried out to test the accuracy of the method. The data set used was collected from 4 volunteers. The results show that the accuracies of the proposed method are in the range of 79–91 %.

Support vector machine classifiers The "one-vs-rest" methodology is applied to use the SVM classifiers to classify emails to more than two categories. The major idea is, for n ($n > 2$) categories, the SVM classifiers are applied n times. Each time, a SVM classifier decides whether the unlabeled email belongs to a certain category c_j or not, and the probability of the email to belong to the category is also calculated. For each email, the categories are ranked according to the probabilities that the email is assigned to. The email may be assigned to the top k categories.

Klimt and Yang [47] studied the email categorization problem on the Enron email corpus using SVM classifiers. They proposed multiple categories for each email using a threshold for each category. The categories passing the thresholds are presented to the user. The threshold is a local optimal score, which is calculated by SCut [90] and evaluated by the $F1$ scores [62]. Several experiments are conducted to learn the features. For example, one experiment is to compare the usefulness of different parts of the email, when they are used as classification features. The results show that the email body is the most useful feature, with micro-average $F1$ score of 0.69 (read from the bar chart in the paper) and macro-average score of 0.54 (read from the bar chart in the paper). Another experiment is to learn the correlation between the number of emails a user has and the classifier's performance. The results show that the number of emails does not effect the classifier's performance much, but the number of categories has clear effect on the classifier's performance. Users with fewer categories (<5 categories) tend to have higher $F1$ scores (micro-average $F1$ scores >0.65 and macro-average scores >0.45).

Notes The naïve Bayes and SVM classifiers have similar performance according to the experiment results by Koprinska et al. [48]. Brutlag and Meek [14] compared the performance of the TF·IDF and SVM classifiers. Their results show that the TF·IDF classifiers offer better performance for the emails from the light folders, while the SVM classifiers have better performance for the emails from the heavy folders. Here, a light folder refers to a folder contains a small number of emails.

4.2 Survey and quantitative analysis methods

To evaluate the need and the effectiveness of email categorization, user surveys and quantitative analysis are used.

Whittaker and Sidner [87] presented a quantitative analysis of 20 users' email accounts and 34 h of interviews. They explored three main functions of emails and addressed the problem of how to manage large amounts of emails. The three main email functions explored are task management, personal archiving and asynchronous communication. Users' opinions on each task are gathered. For the problem of handling large amounts of emails, three different user strategies are identified: users not using folders, users using folders and cleaning up the inbox on a daily base, and users using folders and cleaning up the inbox periodically. Redesigning the email interface to support the main functions is suggested. For the asynchronous communication, the ability to track the conversation history is important. For the personal archiving, an automatic and dynamic tool is needed. For the task management, the important things have to be easily seen by users, and a reminder is needed.

Neustaedter et al. [57] investigated the email categorization problem with a focus on the interface design by conducting contextual interviews and distributing surveys. They tried to answer questions about email categorization in various aspects, such as what kinds of email people often categorize, how emails are dealt with during the categorizing process and when users categorize their emails. Their results confirm the fact that people do need a more efficient means to handle the email categorization task, and the importance of an email depends on the social context, such as the sender and time. They suggested that the email interface should provide more socially salient information about senders, receivers and time for sorting and searching emails, which can help the email categorization.

4.3 Software tools

Segal and Kephart [72] developed MailCat, an intelligent assistant tool to help users with the email categorizing task. For an unlabeled email, it recommends the top 3 categories that the email most likely belongs to. A shortcut button is also provided to save users' effort of moving the email. MailCat keeps updating its classifier according to whether a recommendation is accepted by the user or not. The accuracy of its prediction is about 60–90 %.

Rennie [61] developed Ifile, an effective and efficient email categorizing system, which has been adopted by the EXMH email client already. It assigns an unlabeled email to a category that the email is most likely to be generated from. The prediction accuracy of Ifile is about 79–91 %.

4.4 Challenges

The challenges in email categorization come from the "personal characteristics" of emails. First, users' email categorization preferences change over time. The proposed techniques must be suitable to work in dynamic circumstances. Second, although most existing email categorization methods can deal with the condition that different people have different categorization habits, they tend to be conservative. For example, Segal and Kephart [72], Rennie [61] and Klimt and Yang [47] categorized emails based on the user's existing folders. In other words, they can only find the most relevant folders for emails even when creating a new folder is necessary.

5 Contact analysis

In the context of email mining, contacts are email senders and receivers in an email corpus. Contact analysis is to identify special contacts or contact groups by analyzing the contacts' characteristics from email contents or email network structures. It has two subtasks, namely contact identification and contact categorization.

Contact identification is to discover an email contact's attributes from her email contents or email social network properties. It is widely used in practice. For example, if one wants to buy a laptop and needs advice on laptop selection, with the help of contact identification, she can easily find the right person to consult, which saves her efforts of asking people around.

Contact categorization is to assign an email contact to one or more categories according to the contact's characteristics. It can be used for various purposes, such as an email receiver suggestion service provided by email service providers like Gmail. Using contact categorization, people can get suggestions on appropriate email receivers while they are composing emails. This service helps to lower the probability of forgetting certain email receivers when one writes an email to several persons.

5.1 Contact identification

Classification and social network analysis are the most often used methods in contact identification.

5.1.1 Classification methods

The classification problem in contact identification can be defined as follows. We extract contacts $C = \{c_1, c_2, \ldots, c_m\}$ from a given training email corpus $E = \{e_1, e_2, \ldots, e_n\}$, where e_i $(1 \leq i \leq n)$ is an email, c_j $(1 \leq j \leq m)$ is an email contact. For an email e_u $(e_u \notin E)$, we want to predict its contacts C_u $(C_u \subseteq C)$.

Support vector machines classifiers Corney et al. [22] used an SVM classifier to identify possible disguised senders of a suspect email. The "one-vs-rest" methodology mentioned in Sect. 4.1 is used to suggest more than one possible disguised sender. Features of writing style, such as the average word length, the total number of tab characters used and the average sentence length, are used as stylistic features for senders. Different combinations of the stylistic features are tested in the experiments. The accuracy varies from 60 to 80 %. The results show that it is possible to identify disguised email senders effectively.

Other classifiers Carvalho and Cohen [16] explored the problem of suggesting potential receivers of an email, given its current content and specified receivers so far. Both TF·IDF and K-NN classifiers are evaluated for this task.

In the TF·IDF classifier, the potential receivers are suggested based on a final ranking score, which is produced by computing the cosine distance between the centroid vector of a candidate and the current email content.

In the K-NN classifier, the k emails that are most similar to the current email are found. The dimensions of an email are the words extracted form the email corpus, weighted by the TF·IDF scheme. The similarity between two emails is calculated by the cosine similarity. Then, the potential receivers are suggested based on the final ranking score.

Experiments were designed to test the mean average precisions of K-NN and TF·IDF classifiers under different combinations of email parts. The Enron email data set was used. The results show that the K-NN classifier has a slightly better accuracy than the TF·IDF classifier. The mean average precisions of K-NN classifier in different settings are in the range of 33–46 %, while the mean average precisions of the TF·IDF classifier are in the range of 30–44 %.

5.1.2 Social network analysis (SNA) methods

Social network analysis (SNA) is "a process of quantitative and qualitative analysis of a social network" [79]. SNA analyzes the human relationships from the mathematical aspect. It maps and measures the relationships or flows between people, organizations and other connected entities.

Campbell et al. [15] recommended experts on a given topic based on the analysis of an expert graph built from some selected emails in three steps.

1. Collect all emails related to a topic. A topic contains more than one word. An email is considered to be related to a topic if it contains at least one word under that topic.
2. Build an expert graph based on the emails found from the first step. In the graph, the nodes are the contacts and the directed edges are created by the “From” and “To” fields of the emails.
3. Get expert ratings for all candidates in the expert graph. Candidates with top k expert ratings are recommended. A modified version of the HITS [46] algorithm is used. The expert ratings of contacts on the topic are the “authority” scores of nodes in HITS, which are initially set to 1s, and then updated by the total number of edges pointed to the nodes.

Experiments were conducted to examine the effectiveness of the proposed approach using two data sets containing both emails and expertise ratings from two organizations. The precisions on these two data sets are 52 and 67 %, respectively, and the recalls are 38 and 33 %, respectively.

Hőlzer et al. [38] used a SNA method to identify possible email aliases for a given email address in an email social network.

An email social network is represented by an undirected graph $G = (V, E)$, which is built from an email corpus. A source s_i ($s_i \in S$, where S is a set of sources) is a Web page, which contains a list of email addresses. For example, the conference organizers Web page of KDD 2012 (http://www.kdd.org/kdd2012/organizers.shtml) can be considered as a source, which contains a list of email addresses of all the conference organizers. V is a set of unique email addresses, V_{s_i} is the subset of email addresses in s_i, v_m is an email address. An edge e_{nm} is created if v_n and v_m appear in s_i. The total number of sources containing both v_n and v_m (t_{nm}) is recorded as well.

Three ranking scores, namely geodesic, multiple collocation and combined, are used to find the top k most possible aliases for a given node v_o. The geodesic score between two nodes v_a and v_o is defined as the length of the shortest path between them.

Based on the assumption that two aliases appear together on more web pages indicates a stronger relationship, the multiple collocation score of two nodes v_a and v_o is defined as

$$mulcol_{ao} = \frac{1}{2} + \frac{1}{2^{t_{ao}}}$$

Based on the assumption that the relationship strength between two aliases is inversely correlated with the number of email addresses in one source, the source size score of two node v_a and v_o is defined as $sousiz_{ao} = 1 - \frac{1}{|s_{ao}|}$, where $|s_{ao}|$ is the total number of email addresses on the Web page that contains both v_a and v_o. The combined score refers to an integration of the multiple collocation score and the source size score, defined as

$$comb_{ao} = \text{Max}\left(1 - \sum_{a=1}^{t_{ao}} \frac{1}{\alpha * |s_{ao}|}, \frac{1}{2}\right),$$

where α is the maximum number of sources collocating two aliases. Different data sets should have different values for α.

An email address data set derived from CMU Web pages was used to evaluate the three ranking scores. The experiment results on a small subset of aliases (6 email addresses) show that the combined score has better precisions (about 30–40 %) than the other two scores (about 10–25 %), while the recall varies from 20 to 100 %.

Rowe et al. [65] designed an SNA algorithm to extract a social hierarchy from an email corpus. The social hierarchy can be used to better understand the organizational structure and relationships between people within an organization. The higher level a contact is in the hierarchy, the more important role that contact plays in the organization.

The general idea of this algorithm is to assign the contacts to different levels of a social hierarchy according to their social importance scores. A social importance score here is a combination of two types of element, namely information flow and communication network. The information flow elements include the number of emails a contact has and the average response time of emails of the contact. The communication network elements include the social network structure factors, such as the number of cliques a contact involved. A communication network here is an undirected graph, where the nodes are contacts, and an edge is created if two contacts exchange more than a certain number of emails.

The experiment conducted on the Enron North American West Power Traders Division email data set reproduces the top part of the hierarchy in that division with high accuracy.

Notes The classification methods pay more attention to email contents using terms and writing stylistics from email bodies as features. The social network analysis methods focus on the structure of the network built from the email corpus.

5.2 Contact categorization

Clustering and social network analysis are the most often used methods in contact categorization.

5.2.1 Clustering methods

The clustering problem in contact categorization can be defined as follows. Given an email corpus $E = \{e_1, e_2, \ldots, e_n\}$, where e_i $(1 \leq i \leq n)$ is an email, We extract contacts $C = \{c_1, c_2, \ldots, c_m\}$ from E and cluster the contacts C into groups.

Girvan–Newman algorithm The Girvan–Newman algorithm [33] is a popular clustering method used in community detection of a network. The general idea of the algorithm is to repeatedly calculate the betweenness [32] of all edges in a network, and then remove the edges with the highest betweeness. The process terminates if the left network structure satisfies some pre-defined conditions.

Tyler et al. [80] identified the communities within an organization using the email logs. An undirected network graph is built according to the “From” and “To” fields of emails. The nodes are the contacts. The edges are the directed email connections between two nodes. The Girvan–Newman algorithm is applied in the clustering process. They defined a community as a component of size at least 6, the edge removal process of a component in the graph will stop if the size of a component is less than 6. The quality of communities is verified by interviewing 16 people from 7 different communities; 62.5 % of them are satisfied with the results.

Other algorithms Roth et al. [64] suggested email receivers, such as whether a contact is missing or a contact is wrongly added, when a user is composing an email. Contacts with similar email interacting behaviors are clustered into one group. Contacts within one group are suggested if one or more contacts in that group appear.

An email network is built as a directed weighted hyper-graph. The nodes are the contacts. A hyper-edge is created if an email is sent from one contact to a group of several contacts. The weight of an edge is determined by the recency and frequency of email interactions between the contact and the group. A contact's interaction score is calculated based on the weights of the edges linked to the contact. Contacts with similar interaction scores are clustered into one group. The purpose of the experiment was to test the capability of the proposed method in predicting a user's future email interactions by using the existing email network. Real user data from Gmail was used. The results show that this method has precisions in the range of 40–95 % and recalls in the range of 20–95 %.

5.2.2 Social network analysis (SNA) methods

Johansen et al. [40] found groups of people having common interests for a specified email contact a, based on the email volumes and frequencies between a and other contacts. They used a connection value $C_{(a,b)}$ to measure the relationship strength between the specified contact a and another contact b. Three algorithms are proposed based on three different approaches of calculating the connection values. Contacts are assigned to one group if their connection values with a pass a pre-defined threshold τ.

The first two algorithms, namely the basic algorithm and the frequency-based algorithm, share a connection value definition, but have different parameter settings. A connection value $C_{(a,b)}$ between the specified contact a and another contact b is defined as

$$C_{(a,b)} = \begin{cases} C_{(a,b)} + \mu & \text{if } a \text{ receives an email from } b, \\ C_{(a,b)} + \lambda & \text{if } a \text{sends an email to } b. \end{cases}$$

where μ and λ are parameters. λ is used to weight the number of incoming and outgoing emails, which is defined as the ratio of the number of outgoing emails against the number of incoming emails. The basic algorithm and the frequency-based algorithm have different choices of μ. In the basic algorithm, μ is set to 1. In the frequency-based algorithm, μ varies as the frequency of connections between two contacts changes, which is based on the assumption that the relationship strength between two contacts is stronger if the email connections between them happen more frequently.

There is one common problem of the previous two algorithms. Once a contact is in one group, she would never be removed from that group. The third algorithm, namely the decaying frequency-based algorithm, is proposed to overcome this drawback. It is an extension of the frequency-based algorithm and has two steps to calculate the connection values and updates the connection values on a daily base. In the first step, $C_{(a,b)}$ between the specified contact a and another contact b is calculated in the same way that the frequency-based algorithm does. In the second step, the $C_{(a,b)}$ from the previous step is modified as

$$C_{(a,b)} = \begin{cases} C_{(a,b)} & \text{if } 0 \leq C_{(a,b)} \leq \tau - 1, \\ C_{(a,b)} - \frac{C_{(a,b)} - (\tau - 1)}{\delta} & \text{otherwise} \end{cases}$$

where δ is a decay coefficient, which varies based on the purpose of the algorithm. For example, one contact is allowed to stay in one group longer without any email connection if the value of δ is chosen to be larger.

Experiments were carried out to test the usefulness of the algorithms. In particular, the accuracies of identifying the senders of high priority emails were examined. The results show that these algorithms can effectively identify the senders of high priority emails with accuracy higher than 90 %.

Keila and Skillicorn [45] used a matrix decomposition method, namely Singular Value Decomposition (SVD) [35], to discover communities from the email corpus of an organization. Contacts are considered to be similar if they have similar word usage patterns.

A matrix is built to record the word usage profiles for all contacts. The rows of the matrix correspond to the contacts. The columns correspond to the word usage profile of a certain contact. A word index is built based on each word's global frequency rank in the whole email corpus. Entries of the matrix are the words' indexes. All emails sent by the same contact are aggregated. For each row, words are recorded in descending order of their local frequency ranks, which is the number of times a word used by a specific contact.

For example, for a contact c_i, if the word "job" is the most frequent word used by her, and "job" is the $5th$ most frequently appeared word in the whole email corpus. The entry of the first column of row i is 5.

The SVD method decomposes a matrix A, which contains n rows and m columns, as follows:

$$A = CWF$$

where matrix C is of size $n \times k$, matrix F is of size $k \times m$, and a diagonal matrix W is of size $k \times k$. C and F' are orthonormal, the diagonal of W is non-increasing and $k \leq m$. Points having similar distances from the origin are considered to be from the same community.

Notes The clustering methods used here focus on the email network structures. Email interaction-related factors, such as the number of exchanged emails and the frequency of email exchanges, are the major considered factors. The email contents and the email network structures are used in the social network analysis methods according to different criterion of creating communities.

We notice that graphs are used a lot in the contact analysis task. Table 2 shows the different types of graphs used in different subtasks. Both the directed and undirected graphs are used.

5.3 Software tools

Roth et al. [64] developed two Gmail Labs features, namely "Don't forget Bob!" and "Got the wrong Bob?" The "Don't forget Bob!" feature suggests contacts for the "To:" field to a user after she inputs more than two receivers in that field. The "Got the wrong Bob?" feature detects whether there is any current recipients entered by the user that can be replaced by a more related contact.

Stuit and Wortmann [77] developed a tool, namely Email Interaction Miner (EIM), to investigate and visualize the relationships among people from a user specified email folder. A real-life case study of EIM in a Dutch gas transport company (GTS) shows the business value of EIM. For example, the results of EIM provide GTS with organizational insights, which can help to improve the interactions between people at different levels and human resource allocated for future projects.

5.4 Challenges

Challenges in contact analysis vary in different applications. For example, in expert finding applications, challenges lie in the human judgements of "experts." Most existing methods find their perceived experts instead of real experts. In contact distinguishing applications, the brevity of emails is a problem. Sometimes, an email is too short for the algorithms to learn its

Table 2 Graphs used in contact analysis

Types	Contact identification	Contact categorization
Directed graphs	Campbell et al. [15]	Roth et al. [64]
Undirected graphs	Hőlzer et al. [38], Rowe et al. [65]	Tyler et al. [80]

contact's writing style. In community detection applications, how to choose the right criteria for creating communities is the major challenge.

6 Email network property analysis

An email network is a social network made up of email contacts as nodes and email interactions as edges. One node can be an email address or several email addresses belonging to the same person. Edges are created according to some criteria. For example, one edge is created if two contacts exchange emails more than a certain number of times.

There are two types of email network, namely personal email networks and complete email networks. A personal (or egocentric) email network refers to a network built from one user's email account. It is shaped like a star network, centered by the node of the email account owner. Edges may exist between the other nodes in the network. A complete (or whole) email network is a network built from an email data corpus of an organization. It can be viewed as a combination of many local email networks.

Similar to other social networks, an email network records rich information about communication among people. Understanding the properties of email networks not only provides us clues to develop new features for email services, but also helps us to know how people communicate with each other in virtual social communities.

In this section, we will discuss two types of existing methods studying email network properties, namely social network analysis and classification methods.

6.1 Social network analysis (SNA) methods

Bird et al. [6] studied the email network properties of the email archive of an OSS system (the Apache system), with focus on how developers participate in email activities in the OSS project development.

They modeled a complete email network. Different email addresses owned by the same person are combined into one node. Each email message has a unique message id. A response message can be easily distinguished by its header, which contains the phrase "in-response-to" and the message id of the previous message. A directed edge from node i to node j is created when i posts a message and j responds to it. For example, if Alice posts a message and Jim responds, a directed edge is created from Alice to Jim.

The out-degree of a node i is the number of persons who reply to i's messages. The in-degree of i is the number of persons whose messages i responds. The number of participants with respect to the out-/in-degree of the person follows the power-law distribution.

In order to understand the relationship between the email activities and the OSS software development, they explored three aspects.

The first aspect is the activity correlation between the efforts spent on the software development and the email activity. The efforts spent on the software development refer to the changes on either source code or documents of the project. The Spearman's rank correlation [55] is used. It is defined as follows.

$$\rho = \frac{\sum_i (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_i (x_i - \overline{x})^2 \sum_i (y_i - \overline{y})^2}},$$

where $X = \{x_1, x_2, \ldots, x_n\}$ and $Y = \{y_1, y_2, \ldots, y_n\}$ are two ranking lists, x_n and y_n represent the elements' corresponding ranks in the lists, respectively. The results show that the more efforts one spends on software development, the more email activities he undertakes.

Second, to measure a person's social importance, the out-/in-degree and the betweenness measurements are used. A node v's betweenness in graph G is

$$BW(v) = \sum \frac{G_{ivj}}{G_{ij}},$$

where i, j are the nodes in $G, i \neq j, i \neq v, j \neq v$; G_{ivj} is the number of shortest paths from i to j via v; and G_{ij} is the total number of shortest pathes from i to j. The results show that the more changes one contributes to the software, the more significant role in the network the person plays.

Last, the statistics of the changes, including the source code changes and the document changes, the in-/out-degree and the betweenness, emphasize the strong correlation between social network importance and the source code/document changing contributions.

Bird et al. [7] applied the similar analysis approach on the Postgres system email archive and reported some similar observations.

Karagiannis and Vojnovic [43] explored the user behavioral patterns of email usage in an enterprise email data corpus over a period of 3 months. They conducted statistical analysis to answer two questions.

1. What are the factors that cause people to reply to an email? Both the reply time and the reply probability are analyzed. The reply time is the time difference between receiving an email and the corresponding reply. For the reply time, the time-related factors, such as the recency of an email, the processing time and the actual time when the email is received, are examined. For reply probability, the size of an email and the sender-related factors, such as the organizational level of the sender and the number of emails from the sender to the receiver, are considered. The results show that most of the factors mentioned above are significant.
2. What do we get if we increase the number of contacts in the email network by adding the friends of the existing friends? They discovered that, for about 80 % of the users in the data set, their friends offer <100 new contacts. This interesting discovery indicates that there may be only a small number of overlaps between the newly added contacts and the existing friends, and going through the duplicate contacts detection process may not be needed.

De Choudhury et al. [26] studied the problem of email network inference and relevance on two email corpora, one is the Enron email corpus, and the other one is an email data set from a large university in the United States.

An email network is defined as $G(V, E_s; \tau)$, where V represents the contacts, E_s represents the edges with weights $> \tau$ within a certain time period s. By adjusting the threshold τ, we can generate different email networks for the same email corpus. The weight of an edge $e_{i,j}$ within s is calculated by $w^s_{i,j} = \sqrt{w_{i,j} \cdot w_{j,i}}$, where $w_{i,j}$ is the total number of emails from nodes i to j in period s.

For each data set, different networks are generated by varying threshold τ. The changes of properties in different networks are analyzed at both the network-level and node-level.

At the network-level, the variation of the network size for different τ is explored. The results show that as τ increases, the numbers of edges in both data sets have similar decreasing trends; but the numbers of nodes have very different decreasing rates, the number of nodes on Enron corpus decreases much faster.

At the node-level, some properties, such as the node's degree, the neighbor degree and the size of two-hop neighborhood, are explored for networks with different τ. The results suggest that there is no clearly preferred value for τ for generating the email networks.

6.2 Classification methods

Lockerd and Selker [50] used the SVM classifier to classify the social resources of unlabeled emails. The social resources here refer to the senders' activity motivations of sending emails, such as informing, inquiring, and planning. The features used include terminating punctuation, URLs, the frequency of emoticons used and some others. The model has an accuracy of 50–70 %.

Wang et al. [85] classified the outgoing emails of an enterprise email system into "official" and "private" categories based on social network analysis. A private social network and an official social network are constructed using the "To," "Cc" and "Bcc" fields of some labeled emails ("private" and "official"). Vertices of the networks are contacts, edges are created if there are any email exchanges between the contacts. Several social feature-based measurements, such as the domain divergence of an email's recipients and the percentage of employee recipients, are developed to enforce the classification results. An SVM classifier with cross-validation is used to train and test the data sets. Experiment results show that this method has an accuracy $>$90 %.

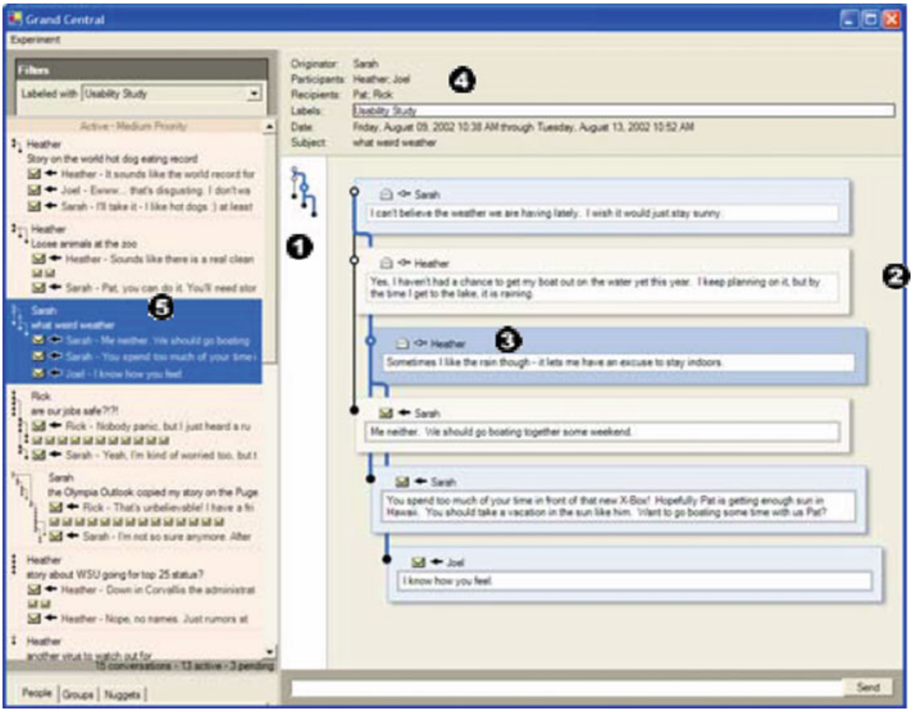
Notes The social network analysis is the most commonly used approach. It analyzes the email network properties from the statistical point of view and provides us with information from the network-level. The classification technique here can be viewed as a tool to provide or predict social contextual information at the individual (email/person) level.

6.3 Software tools

Lockerd and Selker [50] developed DriftCatcher, an email client integrated with social information. The users can easily catch the drift of their personal email networks. It displays the user's social context in various ways. For instance, the font size of an email sender's name is based on the relationship strength between the user and the sender. The background color suggests the social resource type, such as blue means inform/share. A user study conducted on 30 users shows the DriftCatcher email client helps people in processing emails. The most useful social features are response time and relationship strength.

7 Email visualization

In general, email visualization is to use visualization techniques to assist users to identify, retrieve and summarize useful information hidden in large amounts of email messages. The existing studies on email visualization mainly focus on identifying user-email system interaction issues in the existing products and proposing innovative improvements and novel functions. The improvements on user experience are often measured by survey and quantitative analysis.



Labeled with: Usability Study

Active - Medium Priority

Heather
Story on the world hot dog eating record
Heather - It sounds like the world record for
Joel - Ewww... that's disgusting. I don't wa
Sarah - I'll take it - I like hot dogs :) at least

Heather
Loose animals at the zoo
Heather - Sounds like there is a real clean
Sarah - Pat, you can do it. You'll need stor

Sarah
what weird weather
Sarah - Me neither. We should go boating
Sarah - You spend too much of your time i
Joel - I know how you feel

Rick
are our jobs safe?!?!
Rick - Nobody panic, but I just heard a ru
Sarah - Yeah, I'm kind of worried too, but I

Sarah
the Olympia Outlook copied my story on the Puge
Rick - That's unbelievable! I have a fri
Sarah - I'm not so sure anymore. After

Heather
story about WSU going for top 25 status?
Heather - Down in Corvallis the administrat
Heather - Nope, no names. Just rumors at

Heather
another virus to watch out for

15 conversations - 13 active - 3 pending

Originator: Sarah
Participants: Heather; Joel
Recipients: Pat; Rick
Labels: Usability Study
Date: Friday, August 09, 2002 10:38 AM through Tuesday, August 13, 2002 10:52 AM
Subject: what weird weather

Sarah
I can't believe the weather we are having lately. I wish it would just stay sunny.

Heather
Yes, I haven't had a chance to get my boat out on the water yet this year. I keep planning on it, but by the time I get to the lake, it is raining.

Heather
Sometimes I like the rain though - it lets me have an excuse to stay indoors.

Sarah
Me neither. We should go boating together some weekend.

Sarah
You spend too much of your time in front of that new X-Box! Hopefully Pat is getting enough sun in Hawaii. You should take a vacation in the sun like him. Want to go boating some time with us Pat?

Joel
I know how you feel.

Fig. 3 Thread-based user interface design. ① shows the overall tree structure of a conversation; ② displays the details of a conversation; ③ is an example of a message header; ④ contains a summary of a conversation; and ⑤ highlights the selected conversation. (extracted from [82])

7.1 Survey and quantitative analysis methods

Venolia and Neustaedter [82] proposed a thread-based design. The sequence and replying relationships among email conversations are clearly displayed to users in this design. An email conversation (also known as an email "thread") is a group of email messages, which includes an original email message and all its replies. Their design is shown in Fig. 3 [82]. Messages are put in a chronological ordered tree structure. The user experience results show that all participants understand that emails are sorted chronologically; and most users feel that displaying emails in tree structures is easy to read.

Whittaker et al. [88] carried out a study from the email re-finding point of view. Email re-finding is the process of finding out previous emails again. They tried to obtain some implications to optimize the existing user interface design. User access behaviors, such as sorting, folder-access, scrolling, tag-access, searching, opening message and operation duration, were studied. Particularly, they noticed that a user often has to scroll up and down and view multiple pages in order to find the target email. Their results indicate that scrolling can be enhanced in the current search-based interface by displaying all emails in mailbox on one page, so that the user does not need to go to different pages to access and view the messages; and the current threading approach can be extended to "super-threading," which includes multiple threads having similar topics.

Perer and Smith [59] designed three visualizations, shown in Fig. 4 [59], namely correspondent treemaps, correspondent crowds and author lines, to help email users better navigate

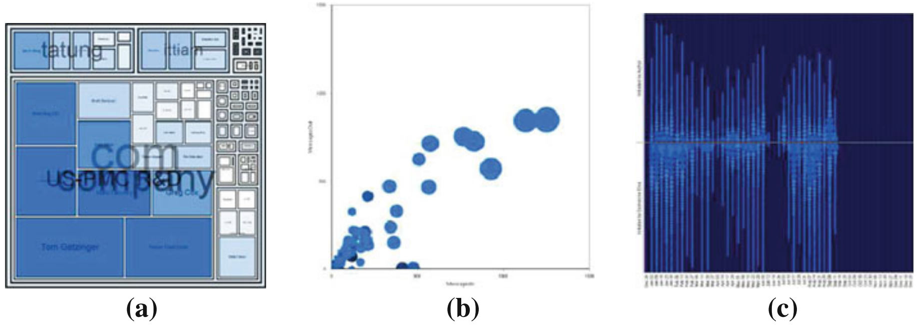


Fig. 4 Three visualizations (extracted from Perer and Smith [59]). **a** Correspondent treemaps. **b** Correspondent crowds. **c** Author lines

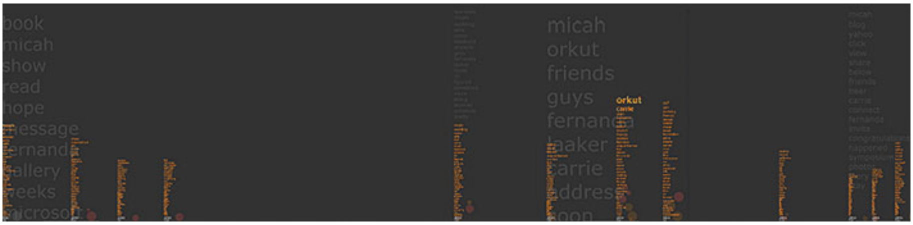


Fig. 5 Themail user interface design (extracted from Viégas et al. [83])

and capture their email archives. Correspondent treemaps, as shown in Fig. 4a, organize contacts from an email account into hierarchies according to the domain hierarchy implication of email addresses. For example, all the ".org" contacts would be put into one outer box, which contains smaller boxes representing contacts from each organization. Correspondent crowds, as shown in Fig. 4b, are generated based on the number of email exchanges between a contact and the email account owner. Each circle represents a contact, the larger the size of a circle, the more email exchanges between the contact and the email account owner. Author lines, as shown in Fig. 4c, reflect the weekly email activities of the email account owner in terms of sending and replying messages. User feedback shows that all the visualizations are easy to understand, and the visualizations are regarded valuable for both analysts and end users.

7.2 Software tools

Viégas et al. [83] proposed a novel email client called Themail, which provides visualization of topic changes between an email account owner and her contacts over a period of time. As shown in Fig. 5 [83], topics are placed in columns and chronologically ordered. Topics are displayed in different sizes and colors according to their frequencies and typicalities. With the help of Themail, one user can easily know the topics and topic changes that she is discussing with her contacts. The user study results show that the participants are quite excited to use Themail, and 87 % participants are happy to use Themail as their email reader.

8 Other applications

In addition to the major tasks discussed so far, there are some other applications worth mentioning.

8.1 Automatic email answering

More and more companies use email as one of their customer service channels, which answer customer inquiries about products or services. Those companies usually hire employees to answer questions from customers manually. However, same or similar questions are often asked again and again by different customers. An automatic email answering system can help those companies to save some labor force in email customer service.

The automatic email answering problem can be described as finding approaches to analyze the incoming emails and then reply them with appropriate answers automatically.

Bickel and Scheffer [5] developed an automatic email answering system. The automatic email answering problem here is considered as a problem of learning mappings from new inquiries (emails of questions) to existing replies (emails of answers). Existing "inquiry-reply" pairs with similar replies are clustered into one group. The reply in the pair, which is in the center of the cluster, is called the "template" of the reply for that cluster. An SVM classifier is used to find the right cluster for a newly incoming inquiry. An email customer service data set provided by a large online store was used to evaluate the performance of the classifier. The experiment results show that the classifier has precisions about 40–90 % and recalls about 5–40 %.

Scheffer [70] developed an email answering assistance system. Standard answer sets for different inquiries are manually defined. Classifiers, such as SVM classifiers and naïve Bayes classifiers, are used to classify the newly incoming inquiries based on the standard answer sets. Area under curve ("AUC") is used to measure the performance. The curve here refers to the receiver operating characteristic ("ROC") [11] curve. The data set used in the experiments was provided by the TELES European Internet Academy. The experiment results show that the SVM classifier has a better performance. It can identify specialized questions with a few positive examples. For instance, given 7 positive examples, it has an AUC between 80 and 95 %. However, the Naïve Bayes classifier performs poorly when documents have very different lengths.

8.2 Adding email social features

Email social features the features that provide users interfaces to perform social network Web site activities, such as finding email contacts' corresponding social network accounts and tracking their friends' updates in social network Web sites.

Adding social features to email services is an interesting, yet non-trivial task. It has already attracted lots of attention from large email service providers. According to [28], "the biggest Web email services, including Yahoo Inc., Microsoft Corp. and Time Warner Inc.'s AOL unit, are adding features that allow users to perform such sociable functions as tracking friends."

Cui et al. [24] formulate an interesting account mapping problem, which can help email users to find their email contact correspondents accurately in social network Web sites. A hybrid account mapping approach, which combines both profile (account name) matching and social structure (email social network structure and social network Web site structure) matching is proposed to solve this problem. Experiments were conducted to evaluate the effectiveness of the approach. Real data sets collected from two volunteers were used. The matching accuracy of this approach is around 50 %.

Table 3 Techniques and tasks used in research papers

Tasks	Classification	Clustering	Social network analysis	Others
Spam detection	[66]	[69]		[34]
	[2]	[36]		[78]
	[1]			
	[69]			
	[63]			
	[30]			
	[75]			
Email categorization	[72]			[57]
	[18]			[87]
	[61]			
	[47]			
Contact analysis	[22]	[80]	[15]	[77]
	[16]	[64]	[38]	
			[40]	
			[65]	
			[45]	
Email network property analysis	[50]		[6]	
			[7]	
			[43]	
			[26]	
Email visualization				[82]
				[88]
				[59]
				[83]
Others	[5]			[24]
	[70]			

9 Conclusions and future directions

In this paper, we present a brief but still comprehensive survey on email mining. We introduce two email data representation approaches that are often conducted in the preprocessing phase. Then, we identify five email mining tasks, namely spam detection, email categorization, contact analysis, email network property analysis and email visualization. For each task, we discuss the commonly used techniques. Table 3 summarizes the techniques and tasks in different studies. We also discuss some research challenges for the individual tasks. We briefly mention the corresponding software tools for each task.

As email is one of the most popular forms of communication nowadays, email mining is invaluable in practice. Although good progress has been achieved, there are still dramatic potentials for future work. We discuss two important directions as examples here.

9.1 Novel egocentric networks

More and more people communicate with each other through social network Web sites, such as Facebook and Twitter. Integrating social network features of existing social network Web sites to the email networks will lead to novel egocentric networks. These novel egocentric networks can help people to manage their personal email networks and social network accounts more efficiently.

For example, how to choose the most attractive social network posts for an email user? Let us assume an email user has on average 100 email contacts, and each contact has on average 3 social network accounts. If each contact on average has 1 post of each social network account every day, then there would be a total of 300 daily posts on average. Going through those posts one by one could cost the email user a lot of time. By email mining on the novel egocentric network, we can recommend the most important social network posts to the email user based on the topics of the posts and the relationship strength between the email user and the contacts who are involved in the posts.

9.2 Email monetization

Following the success of search engine monetization, email monetization is an interesting direction of great business protential.

For example, what advertisements displayed alongside an email are more attractive to the email user? We notice that different email service providers have different ad displaying strategies. A systematic evaluation and comparison of different ad displaying strategies and their effectiveness would be useful for the advertisers to choose the right email service providers to distribute their ads.

Acknowledgments We are grateful to the anonymous reviewers for their very useful comments and suggestions. This research is supported in part by an NSERC Discovery Grant, a BCFRST NRAS Endowment Research Team Program project and a GRAND NCE project. All opinions, findings, conclusions and recommendations in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

References

1. Androutsopoulos I, Koutsias J, Chandrinos K, Spyropoulos C, (2000)An experimental comparison of naive bayesian and keyword-based anti-spam filtering with personal e-mail messages, In: Proceedings of the 23rd annual international special interest group on information retrieval (SIGIR) conference on research and development in information retrieval, SIGIR'00, ACM, New York, NY, USA, pp 160–167
2. Androutsopoulos I, Paliouras G, Karkaletsis V, Sakkis G, Spyropoulos, C, Stamatopoulos P (2000) Learning to filter spam e-mail: a comparison of a naive bayesian and a memory-based approach. Computing Research Repository (CoRR) cs.CL/0009009
3. Bälter O (2000) Keystroke level analysis of email message organization. In: Proceedings of the SIGCHI conference on Human factors in computing systems, CHI'00, ACM, New York, NY, USA, pp 105–112
4. Bellotti V, Ducheneaut N, Howard M, Smith I, Grinter RE (2005) Quality versus quantity: e-mail-centric task management and its relation with overload. Hum Comput Interact 20:89–138
5. Bickel S, Scheffer T (2004) Learning from message pairs for automatic email answering. In: Proceedings of the European conference on machine learning (ECML), pp 87–98
6. Bird C, Gourley A, Devanbu P, Gertz M, Swaminathan A (2006a), Mining email social networks. In: Proceedings of the 2006 international workshop on mining software repositories, MSR'06, ACM, New York, NY, USA, pp 137–143
7. Bird C, Gourley A, Devanbu P, Gertz M , Swaminathan A (2006b) Mining email social networks in postgres. In: Proceedings of the 2006 international workshop on mining software repositories, MSR'06, ACM, New York, NY, USA, pp 185–186

8. Blanzieri E, Bryl A (2008) A survey of learning-based techniques of email spam filtering. Artif Intell Rev 29:63–92
9. Blei DM, Ng AY, Jordan MI (2003) Latent dirichlet allocation. J Mach Learn Res 3:993–1022
10. Boykin PO, Roychowdhury VP (2004) Personal email networks: an effective anti-spam tool. Computing Research Repository (CoRR) cond-mat/0402143
11. Bradley A (1997) The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recogn 30:1145–1159
12. Breiman L (2001) Random forests. Mach Learn 45:5–32
13. Breiman L, Friedman J, Stone CJ, Olshen RA (1984) Classification regression trees, 1st edn. Wadsworth and Brooks, Monterey, CA
14. Brutlag JD, Meek C (2000) Challenges of the email domain for text classification. In: Proceedings of the seventeenth international conference on machine learning, ICML'00, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp 103–110
15. Campbell CS, Maglio PP, Cozzi A, Dom B (2003) Expertise identification using email communications. In: Proceedings of the twelfth international conference on Information and knowledge management, CIKM'03, ACM, New York, NY, USA, pp 528–531
16. Carvalho VR, Cohen WW (2008) Ranking users for intelligent message addressing. In: Proceedings of the IR research, 30th European conference on Advances in information retrieval, ECIR'08, Springer, Berlin, Heidelberg, pp 321–333
17. Claburn T (2005) Spam costs billions. Website http://www.informationweek.com/news/59300834
18. Cohen W (1996) Learning rules that classify e-mail. In: Papers from the association for the advancement of artificial intelligence (AAAI) spring symposium on machine learning in information access, AAAI Press, pp 18–25
19. Cohen WW (1995) Fast effective rule induction. In: Proceedings of the twelfth international Conference on machine learning, Morgan Kaufmann, pp 115–123
20. Cormack G, Lynam T (2004) A study of supervised spam detection applied to eight months of personal e-mail
21. Cormack G, Lynam T (2005) Spam corpus creation for trec. In: Proceedings of the second conference on email and anti-spam (CEAS), Mountain View, CA
22. Corney MW, Anderson AM, Mohay GM, de Vel O (2001) Identifying the authors of suspect email
23. Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20:273–297
24. Cui Y, Pei J, Tang G, Jiang D, Luk W-S, Hua M (2011) Finding email correspondents in online social networks. World Wide Web J, 2012, Springer, Netherlands, pp 1–24
25. Dabbish LA, Kraut RE (2006) Email overload at work: an analysis of factors associated with email strain. In: Proceedings of the (2006) 20th anniversary conference on computer supported cooperative work, CSCW'06. ACM, New York, NY, USA, pp 431–440
26. De Choudhury M, Mason WA, Hofman JM, Watts DJ (2010) Inferring relevant social networks from interpersonal communication. In: Proceedings of the 19th international conference on World wide web, WWW'10, ACM, New York, NY, USA, pp 301–310
27. de Vel O, Anderson A, Corney M, Mohay G (2001) Multi-topic e-mail authorship attribution forensics. In: Proceedings of the workshop on data mining for security applications, 8th ACM conference on computer security (CCS)
28. Delaney KJ, Vara V (2007) Will social features make email sexy again? Wall Str J, (18 Oct)
29. Dhillon IS, Modha DS (2001) Concept decompositions for large sparse text data using clustering. Mach Learn 42:143–175
30. Drucker H, Wu D, Vapnik V (1999) Support vector machines for spam categorization. IEEE Trans Neural Netw 10(5):1048–1054
31. Ducheneaut N, Watts LA (2005) In search of coherence: a review of e-mail research. Hum Comput Interact 20:11–48
32. Freeman LC (1977) A set of measures of centrality based on betweenness. Sociometry 40(1):35–41
33. Girvan M, Newman MEJ (2002) Community structure in social and biological networks. Proc Natl Acad Sci 99(12):7821–7826
34. Golbeck J, Hendler JA (2004) Reputation network analysis for email filtering. In: Proceedings of the first conference on email and anti-spam (CEAS), Mountain View, CA
35. Golub GH, van Van Loan CF (1996) Matrix computations, 3rd edn. The Johns Hopkins University Press, Baltimore, MD
36. Gomes LH, Castro FDO, Almeida RB, Bettencourt LMA, Almeida VAF, Almeida JM (2005) Improving spam detection based on structural similarity. Computing Research Repository (CoRR) abs/cs/0504012
37. Gomez JC, Boiy E, Moens M-F (2012) Highly discriminative statistical features for email classification. Knowl Inf Syst 31(3):23–57

38. Hőlzer R, Malin B, Sweeney L (2005) Email alias detection using social network analysis. In: Proceedings of the international conference on knowledge discovery and data mining (KDD) workshop on link discovery: issues, approaches, and applications, ACM Press
39. Internet Threats Trend Report Q1 2010 (2010), Company Press
40. Johansen L, Rowell M, Butler K, Mcdaniel P (2007) Email communities of interest. In: Proceedings of the fourth conference on email and anti-spam (CEAS), Mountain View, CA
41. John GH, Langley P (1995) Estimating continuous distributions in bayesian classifiers. In: Proceedings of the eleventh conference on uncertainty in artificial intelligence, UAI'95, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp 338–345
42. Jolliffe IT (1986) Principal component analysis. Springer, New York
43. Karagiannis T, Vojnovic M (2009) Behavioral profiles for advanced email features. In: Proceedings of the 18th international conference on world wide web', WWW'09, ACM, New York, NY, USA, pp 711–720
44. Katakis I, Tsoumakas G, Vlahavas I (2007) Web data management practices: emerging techniques and technologies. IGI Publishing, Hershey, PA
45. Keila PS, Skillicorn DB (2005) Structure in the enron email data set. Comput Math Organ Theory 11:183–199
46. Kleinberg JM (1999) Authoritative sources in a hyperlinked environment. J ACM 46:604–632
47. Klimt B, Yang Y (2004) The enron corpus: A new data set for email classification research. In: The European conference on machine learning (ECML), pp 217–226
48. Koprinska I, Poon J, Clark J, Chan J (2007) Learning to classify e-mail. Inf Sci 177:2167–2187
49. Lam H-Y, Yeung D-Y (2007) A learning approach to spam detection based on social networks. In: Proceedings of the fourth conference on email and anti-spam (CEAS), Mountain View, CA
50. Lockerd A, Selker T (2003) DriftCatcher: The implicit social context of email. In: Proceedings of the ninth IFIP TC13 international conference on human–computer interaction (INTERACT) 2003, pp 1–5
51. MacQueen JB (1967) Some methods for classification and analysis of multivariate observations. In: Cam LML, Neyman J (eds) Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, vol 1, University of California Press, pp 281–297
52. McArthur R, Bruza P (2003) Discovery of implicit and explicit connections between people using email utterance. In: Proceedings of the eighth conference on European conference on computer supported cooperative work (ECSCW) 2003, Kluwer Academic Publishers, Norwell, MA, USA, pp 21–40
53. Mccallum A, Corrada-emmanuel A, Wang X (2004) The author-recipient-topic model for topic and role discovery in social networks: experiments with enron and academic email, Technical report, University of Massachusetts Amherst
54. McCallum A, Nigam K (1998) A comparison of event models for naive bayes text classification. In: Proceedings of the association for the advancement of artificial intelligence (AAAI) workshop on learning for text categorization, AAAI Press, pp 41–48
55. Myers JL, Well AD (2003) Research design and statistical analysis, 2nd edn. Lawrence Erlbaum, Hillsdale, NJ
56. Nagwani NK, Bhansali A (2010) An object oriented email clustering model using weighted similarities between emails attributes. Int J Res Rev Comput Sci 1(2):1–6
57. Neustaedter C, Brush AJB, Smith MA (2005) Beyond "from" and "received": exploring the dynamics of email triage. In: ACM CHI'05 extended abstracts on human factors in computing systems, CHI EA'05, ACM, New York, NY, USA, pp 1977–1980
58. Nucleus Research Inc. (2007) Spam, the repeat offender. Notes and reports
59. Perer A, Smith MA (2006) Contrasting portraits of email practices: visual approaches to reflection and analysis. In: Proceedings of the working conference on advanced visual interfaces, AVI'06, ACM, New York, NY, USA, pp 389–395
60. Radicati S, Hoang Q (2010) Email statistics report, 2011–2015. Company Press
61. Rennie JDM (2000) Ifile: An application of machine learning to e-mail filtering. In: Proceedings of the international conference on knowledge discovery and data mining (KDD) workshop on text mining
62. Rijsbergen CJV (1979) Information retrieval, 2nd edn. Butterworth-Heinemann, Newton, MA
63. Rios G, Zha H (2004) Exploring support vector machines and random forests for spam detection. In: Proceedings of the first conference on email and anti-spam (CEAS), Mountain View, CA
64. Roth M, Ben-David A, Deutscher D, Flysher G, Horn I, Leichtberg A, Leiser N, Matias Y, Merom R (2010) Suggesting friends using the implicit social graph. In: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD'10, ACM, New York, NY, USA, pp 233–242
65. Rowe R, Creamer G, Hershkop S, Stolfo SJ (2007) Automated social hierarchy detection through email network analysis. In: Proceedings of the 9th WebKDD and 1st SNA-KDD (2007) workshop on Web mining and social network analysis, WebKDD/SNA-KDD'07. ACM, New York, NY, USA, pp 109–117

66. Sahami M, Dumais S, Heckerman D, Horvitz E (1998) A bayesian approach to filtering junk e-mail
67. Salton G, McGill MJ (1986) Introduction to modern information retrieval. McGraw-Hill, Inc., New York, NY
68. Salton G, Wong A, Yang CS (1997) A vector space model for automatic indexing. In: Sparck Jones K, Willett P (eds) Readings in information retrieval. Morgan Kaufmann Publishers Inc., San Francisco, CA, pp 273–280
69. Sasaki M, Shinnou H (2005) Spam detection using text clustering. In: Proceedings of the 2005 international conference on cyberworlds (CW), IEEE Computer Society, Washington, DC, USA, pp 316–319
70. Scheffer T (2004) Email answering assistance by semi-supervised text classification. Intell Data Anal 8:481–493
71. Schwartz MF, Wood DCM (1993) Discovering shared interests using graph analysis. Commun ACM 36:78–89
72. Segal RB, Kephart JO (1999) Mailcat: an intelligent assistant for organizing e-mail. In: Proceedings of the sixteenth national conference on artificial intelligence and the eleventh innovative applications of artificial intelligence conference innovative applications of artificial intelligence', AAAI'99/IAAI'99, American Association for Artificial Intelligence, Menlo Park, CA, USA, pp 925–926
73. Silverman BW, Jones MC (1951) E. fix and j.l. hodges (1951): an important contribution to nonparametric discriminant analysis and density estimation: commentary on fix and hodges (1951). Int Stat Rev/Revue Internationale de Statistique 57(3):233–238
74. Sparck Jones K (1988) A statistical interpretation of term specificity and its application in retrieval. Taylor Graham Publishing, London
75. Stolfo SJ, Hershkop S, Wang K, Nimeskern O, Hu C-W (2003a), A behavior-based approach to securing email systems. In: Proceedings of the Computer network security, second international workshop on mathematical methods, models, and architectures for computer network security, MMM-ACNS 2003, St. Petersburg, Russia, September 21–23, 2003 (Lecture Notes in Computer Science) vol 2776. Springer
76. Stolfo SJ, Hershkop S, Wang K, Nimeskern O, Hu C-W (2003b), Behavior profiling of email. In: Proceedings of the 1st NSF/NIJ conference on intelligence and security informatics, ISI'03, Springer, Berlin, Heidelberg, pp 74–90
77. Stuit M, Wortmann H (2012) Discovery and analysis of email-driven business processes. Inf Syst 37(2):142–168
78. Taylor B (2006) Sender reputation in a large webmail service. In: Proceedings of the third conference on email and anti-spam (CEAS), Mountain View, CA
79. Techopedia.com (n.d.) Social network analysis (SNA). Website http://www.techopedia.com/definition/3205/social-network-analysis-sna
80. Tyler JR, Wilkinson DM, Huberman BA (2003) Email as spectroscopy: automated discovery of community structure within organizations. In: Communities and technologies, Kluwer, B.V., Deventer, The Netherlands, pp 81–96
81. van Rijsbergen C, Robertson S, Porter M (1980) New models in probabilistic information retrieval
82. Venolia GD, Neustaedter C (2003) Understanding sequence and reply relationships within email conversations: a mixed-model visualization. In: Proceedings of the SIGCHI conference on human factors in computing systems (CHI'03), ACM, New York, NY, USA, pp 361–368
83. Viégas FB, Golder S, Donath J (2006) Visualizing email content: portraying relationships from conversational histories. In: Grinter R, Rodden T, Aoki P, Cutrell E, Jeffries R, Olson G (eds) Proceedings of the SIGCHI conference on human factors in computing systems, CHI'06, ACM, New York, NY, USA, pp 979–988
84. Vleck TV (2001) The history of electronic mail. Website http://www.multicians.org/thvv/mail-history.html
85. Wang M-F, Jheng S-L, Tsai M-F, Tang C-H (2011) Enterprise email classification based on social network features. In: Proceedings of the international conference on advances in social networks analysis and mining, 2011, IEEE Computer Society, Washington, DC, USA, pp 532–536
86. Wang X-L, Cloete I (2005) Learning to classify email: a survey. In: Proceedings of the international conference on machine learning and, cybernetics, 2005, vol 9, pp 5716–5719
87. Whittaker S, Sidner C (1996) Email overload: exploring personal information management of email. In: Proceedings of the special interest group on computer human interaction (SIGCHI) conference on Human factors in computing systems: common ground, CHI'96, ACM, New York, NY, USA, pp 276–283
88. Whittaker S, Matthews T, Cerruti J, Badenes H, Tang J (2011) Am I wasting my time organizing email? A study of email refinding. In: Proceedings of the 2011 annual conference on human factors in computing systems, CHI'11. ACM, New York, NY, USA, pp 3449–3458
89. Wikipedia (2012) E-mail spam. Website http://en.wikipedia.org/wiki/E-mail_spam

90. Yang Y (2001) A study on thresholding strategies for text categorization. In: Proceedings of the 24th ACM international conference on research and development in information retrieval. ACM Press, pp 137–145
91. Yarow J (2011) 107,000,000,000,000. Website http://www.businessinsider.com/internet-statistics-2011-1-2011-1
92. Yoo S, Yang Y, Lin F, Moon I-C (2009) Mining social networks for personalized email prioritization. In: Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining, KDD'09, ACM, New York, NY, USA, pp 967–976

Author Biographies

Guanting Tang received her B.Sc. degree in computer science from Shanghai Ocean University, China, in 2007 and her M.Sc. degree in computer science from Georgia Southwestern State University, USA, in 2008. She is currently a Ph.D. candidate at the School of Computing Science at Simon Fraser University, Canada. Her research interests include data mining, information retrieval, natural language processing and applications in business intelligence.

Jian Pei is a Professor at Simon Fraser University, Canada. His expertise is on developing effective and efficient data analysis techniques for novel data intensive applications. Particularly, he is currently interested in and actively working on developing various techniques of data mining, Web search, information retrieval, data warehousing, online analytical processing and database systems, as well as their applications in social networks, health-informatics and business intelligence. Since 2000, he has published one textbook, two monographs and over 180 research papers in refereed journals and conferences and has served in the organization committees and the program committees of over 170 international conferences and workshops. He is the editor-in-chief of the IEEE Transactions on Knowledge and Data Engineering and an associate editor or editorial board member of several major academic journals in his fields. He is a senior member of ACM and IEEE, and an ACM Distinguished Speaker.

Wo-Shun Luk is a Professor at the School of Computing Science at Simon Fraser University, Canada. His research current interests include data warehouse, OLAP/BI, Mobile Data Analytics and Information Extraction.