Journal of King Saud University – Computer and Information Sciences (2015) 27, 46–57

King Saud University

Journal of King Saud University – Computer and Information Sciences

www.ksu.edu.sa
www.sciencedirect.com

Clustering and classification of email contents

Izzat Alsmadi [a,*], Ikdam Alhami [b]

[a] Department of Computer Science, Boise State University, USA
[b] Yarmouk University, Jordan

Received 3 September 2013; revised 12 February 2014; accepted 13 March 2014
Available online 7 January 2015

KEYWORDS

Emails classification;
Document similarity;
Document classification;
Feature extraction;
Subject classification;
Content classification

Abstract Information users depend heavily on emails' system as one of the major sources of communication. Its importance and usage are continuously growing despite the evolution of mobile applications, social networks, etc. Emails are used on both the personal and professional levels. They can be considered as official documents in communication among users. Emails' data mining and analysis can be conducted for several purposes such as: Spam detection and classification, subject classification, etc. In this paper, a large set of personal emails is used for the purpose of folder and subject classifications. Algorithms are developed to perform clustering and classification for this large text collection. Classification based on NGram is shown to be the best for such large text collection especially as text is Bi-language (i.e. with English and Arabic content).

1. Introduction

Emails are used by most humans on earth. It is estimated that there are more than 3 billion email accounts of almost half of the world population. They are expected to reach 4 billion by the year 2015 (Email Statistics Report, 2011). Even kids are allowed under certain conditions to have email accounts supervised by parents.

Spam in emails is one of the most complex problems in email services. Spam emails are those unwanted, unsolicited emails that are not intended for specific receiver and that are sent for either marketing purposes, or for scam, hoaxes, etc. It is estimated that in 2009 more than 97% of emails were classified as spam (Elements of Computer Security, 2010). This is why many research papers which studied or analyzed emails focused on this aspect (i.e. the classification of emails into spam or not). However, the struggle between spammers and spam detection tools is continuous where each side is trying to create new ways to overcome the techniques developed by the other.

Some local papers that conducted spam assessment (e.g. Abdullah Al-Kadhi, 2011 paper) showed that the problem is serious. Authors conducted surveys to assess the current status of Spam distribution in KSA. Authors tried also to summarize major reasons of spreading of spam messages and emails including: Sexual contents, commercials, phishing, religious reasons, etc. Of course major disadvantage of spam spread is the overconsumption and bandwidth and resources for no good purposes.

* Corresponding author.
E-mail address: izzatalsmadi@boisestate.edu (I. Alsmadi).
Peer review under responsibility of King Saud University.

ELSEVIER Production and hosting by Elsevier

http://dx.doi.org/10.1016/j.jksuci.2014.03.014

In this aspect, an email spam-based classifier is not only expected to accurately classify spam emails as spams, but also expected to classify non-spam emails as non-spam or normal. This is since both are considered conditions for evaluating the quality of its classification or prediction. Four prediction metrics are used then to evaluate the quality of email prediction. True Positive (TP) indicates that the spam detection tool predicts that the email is spam and truly it was a spam. True Negative (TN) indicates that the tool or the email system predicts that the email is normal and not spam and correctly it was so. False Positive (FP) indicates that by mistake the tool predicts that a good email is spam (aka false alarms). Last, False Negative (FN) indicates also another mistake where it is predicted that a spam email is normal. As such, a perfect detection system should have the values: TP 100%, TN 100%, FP 0%, and FN 0%. In reality such perfect situation is impossible and impractical. TP and FP complement each other for 100% (i.e. their total should be 100%). Same thing is applied for TN and FN.

The challenge of some email detection systems is that if it is restricted through many spam-detection roles, TP may go high, but at the account of getting many false alarms. On the opposite very lean rules may get very high TN but at the account of FN.

Another challenge in emails' spam detection is speed. In security, speed or performance is always in a trade off with security where too many roles may slow down the system.

In addition to spam based classification, papers that conducted research in emails discussed other aspects such as: Automatic subject or folder classification, priority based filtering of email messages, emails and contacts clustering, etc. Some papers evaluated replies in emails to classify emails on different threads. Currently some email servers such as Gmail combine email together if they came as a reply.

Following are some of the focuses in the research of email analysis (Based on our review of papers related to research papers in data mining in emails' datasets):

1. Generally, email analysis can be classified under text categorization in its most activities. Algorithms such as: VSM, KNN, Ripper, Maximum Entropy (MaxEnt), Winnow, ANN are examples of algorithms used in email analysis.
2. A major research subject in email classification is to classify emails into spam or no spam emails. This can be further used for the real time prediction of spam emails.
3. Some email classification research papers tried to classify emails based on the gender of the sender given some of the common aspects that may distinguish emails from females or males.
4. Email classification can be also used to automatically assign emails to predefined folders.
5. Rather than spam and non spam emails, emails can be also classified into: Interesting and uninteresting emails.
6. Features are extracted from the email content or body, title or subject or some of the other Meta data that can be extracted from the emails such as: sender, receiver, BCC, date of sending, receiving, number of receivers, etc. The method to extract feature can be based on words, bags of words, etc.
7. Email clustering is also considered to cluster emails into different subjects or folders.
8. The time information in emails (e.g. when: sent, received, etc.) is used also in some research papers to classify emails.
9. Some research papers tried to classify emails based on similar threads or subjects. Some email systems such as Gmail connect emails related to each other (e.g. by reply or forward events) together.

In this paper, a personal email archive of more than 19,000 normal messages is used for analysis and evaluation. The focus is to study the email content and address and classify each email into one of three: Personal, professional and other based on sender, content and header.

The rest of the paper is organized as the following: Section two presents several research papers in email analysis. Section three presents goals and approaches. Section four presents experiment and analysis and paper is concluded with conclusion section.

2. Related work

As mentioned earlier, collecting an archive of emails for analysis can be done for several purposes. One of the major goals is spam detection. This sub section describes some research papers related to spam email classification.

2.1. Spam–non-spam email classification

We selected some papers, based on citation, related to spam detection or filtering. Those papers are: Zhuang et al., 2008; Blanzieri and Bryl, 2008; Webb et al., 2006; Mishne et al., 2005; Sculley and Wachman, 2007; Zhou et al., 2010; Pérez-Díaz et al., 2012; Xie et al., 2006; Katakis et al. 2007; Bogawar et al. 2012; Ozcaglar 2008. Different papers discussed the using of different algorithms and also applying the algorithms in different places between email senders and receivers.

Zhuang et al.'s (2008) paper focused on trying to find Botnets. Botnets are groups responsible for spreading spam emails. Methods are evaluated to detect such sources of spam campaigns that share some common features. Spammers however try to change spam emails through some intended mistakes or obfuscations especially in popular filtered keywords. Certain finger prints are defined where all emails that have those finger prints are then clustered together.

Blanzieri and Bryl (2008) presented a technical report in 2008 to survey learning algorithms for spam filtering. The paper discussed several aspects related to spam filtering such as the proposals to change or modify email transmission protocols to include techniques to eliminate or reduce spams. Some methods focused only on content while others combined header or subject with content. Some other email characters such as size, attachments, to, from, etc. were also considered in some cases. Feature extraction methods were also used for both email content, attached and embedded images.

Webb et al.'s (2006) paper talked about web spam and how to use email spam detection techniques to detect spam web pages. Similar to the approaches to detect spam in emails, web pages are scanned for specific features that may classify them as spam pages such as using irrelevant popular words, keywords stuffing, etc. Mishne et al.'s (2005) paper represents another example of web or link spam research paper. Blogs, social networks, news or even e-commerce websites now allow

users to publish their comments or feedback. Spammers use such ability to post spam messages through those posts. Hence spam detection techniques should be also used to allow automatic detection of such posts.

Sculley and Wachman (2007) discussed also algorithms such as VSM for email, blogs, and web and link spam detection. The content of the email or the web page is analyzed using different natural language processing approaches such as: Bags of words, NGram, etc. The impact of a tradeoff parameter in VSM is evaluated using different setting values for such parameter. Results showed that VSM performance and prediction accuracy is high when the value of this parameter is high.

Zhou et al. (2010) proposed a spam-based classification scheme of three categories. In addition to typical spam and not spam categories, a third undetermined category is provided to give more flexibility to the prediction algorithm. Undecided emails must be re-examined and collect further information to be able then to judge whether they are spam or not. Authors used Sculley and Cormack (2008); UCI Machine Learning Repository as their experimental email dataset (machine learning repository).

Pérez-Díaz et al.'s (2012) paper 2012 evaluates applying rough set on spam detection with different rule execution schemes to find the best matching one. UCI Spam base is used in the experimental study (machine learning repository).

Xie et al.'s (2006) paper 2006 tried to summarize features that can identify Botnets or spam proxies that are used to send a large number of spam emails. Authors looked at network related behaviors that can possibly identify such spam proxies.

2.2. Other email data analysis research goals

In this section, we will describe some papers related to the analysis of email messages for purposes other than spam detection.

Kiritchenko and Matwin (2001) presented a paper on email classification through combining labeled and unlabeled data. Similar to many other papers, VSM is showed to be the best classifier in terms of prediction or classification performance. Text classification is used to classify emails into different folders based on predefined categories. Authors tried to define classes as interesting and uninteresting categories. An initial list of manually labeled emails can be used for the future automatic training and classification. VSM is showed to benefit from the co-training process proposed in the paper.

Enron email database is used in several research papers in email classification (http://www.cs.cmu.edu/~enron) (Klimt and Yang, 2004). Shetty and Adibi's (2005) paper used Enron email database in email classification based on graph entropy modeling. The entropy tried to select the most interesting nodes (that represent emails) in the graph. Edges represent messages between different email users.

Yoo et al. (2009) discussed personalized email prioritization in email messages and social networks or groups. Goals such as clustering contacts and classification (Using Newman clustering method) were evaluated in relation with email messages and social networks.

Klimt and Yang (2004) studied relationships in email messages such as the relations between contacts and messages or threads of messages. A thread of messages includes several emails exchanged between two or more persons through several email messages. Enron dataset is used in this study similar to many other relevant research papers in this area where it is considered as the largest publically available email messages dataset. For this specific paper, another small email dataset (CMU) is used.

McCallum and Wang's (2007) paper is also in the area of social networks and email analysis with the goal of topical analysis and classification based on relations between people. The Author-Recipient-Topic Model tries to build relation in emails and social networks between these three concepts, entities or elements. Papers tried also to study trends in email and social network relations such as spouses, team members in classes, companies, etc.

Carmona-Cejudo et al.'s (2011) paper is related to real time email classification and introduced GNUsmail open source for email folder classification. The application is developed to parse emails from different email clients and perform some data mining analysis using WIKA data mining tool. In email folder classification is also based on the time of email messages (Bekkerman et al., 2004). The paper used Enron and SRI email datasets for the case study. Some new classification methods such as: MaxEnt were evaluated in the paper. The major decision to make in all email classification papers is what features to select. Features can be related to email title, from or to addresses or can be related to the content; words, sequence of words, etc. Natural language processing activities such as parsing and stemming are then involved to parse email contents and eliminate any words that may not be relevant for the classification process.

Bird's (2004) paper discussed an approach to predict response on emails based on mined data. Example of response prediction can be related to for example the most appropriate person to respond to an email. Latent Semantic Indexing (LSI) Information Retrieval (IR) methods can be used to parse and extract features from emails. Artificial Neural Networks (ANNs) are used and showed to have very good results in terms of prediction accuracy.

2.3. Ontology classification of email contents

Ontologies are proposed for several purposes related to the reusability of knowledge, knowledge sharing and analysis and also to separate commonalities from differences in the different knowledge areas.

In the specific research subject of ontology classification or knowledge extraction of Email contents, there have been some research papers that tried to propose and introduce concepts usually found in Email contents. Such ontology can be also used for email validation or spam detection.

For example, Taghva et al.'s (2003) paper proposed email concepts' extraction using Ecdysis Bayesian email classifier. Authors extracted email contents based on features collected from the extracted or trained data and also from DOE inclusionary or exclusionary records (Office of Civilian Radioactive Waste Management, 1992). Inclusionary concepts include: Organization, Department, Email Agent, and Message Topics. Exclusionary concepts include: Email Characteristics, Count Characteristics, and Attachment Type Characteristics. Each one of those entities includes several related attributes. Protégé ontological tool (http://protege.stanford.edu/) was used to build and show the ontology. In

our case, MIME parser is used to parse from emails many attributes of those described in Taghva et al. ontology.

Yang and Callan (2008) in 2008 presented also ontology to extract concepts from a corpus of public comments (Mercury and Polar Bear datasets). NGram mining is used to identify candidate concepts. Wordnet and surface text pattern matching are used to identify relationships among the concepts. Wordnet keywords are used to guide organization of concepts into intended hierarchal relationships. Part of Speech (POS) tagger from Stanford University is used as a text parser. Authors then used NGram based on words.

Beseiso et al.'s (2012) paper proposed a method for concepts' extraction from email systems. Authors discussed one of the challenges of emails concepts' extraction as in most cases; users' emails are domains specific and highly dependent on the person, their profession, interests, etc. Authors extended NEPOMUK Message Ontology and defined email general concepts and domain specific concepts. Authors used Enron and custom email datasets for evaluation.

Aloui and Neji's (2010) paper proposed a system for automatic email classification and question answering. The approach proposed three clusters of emails based on their general subjects: Procedural, social and cognitive functions. The paper extended an approach in the paper of Lê and Lê (2002). The 10 categories include: Requesting, Thinking, Discussing, Confirming, Referring, Clarifying, Complimenting, Complaining, Greeting and Sharing.

Text clustering and classification can be used for a wide spectrum of applications. For example, Altwaijry and Algarny's (2012) paper used text classification methods to classify network income data and traffic and classify such data into threat (harmful) or non-threat data. A Naive Bayesian (NB) classifier is used. Such classifier is proved to be effective for classification in several different areas. Authors used public KDD IDS dataset for testing and training.

Another major application area for classification especially in information retrieval systems includes image classification (De and Sil, 2012). In this specific paper, authors used fuzzy logic to assign soft class labels to the different images in the collected dataset. Such image classification can be used for search engines query and in most cases images are associated with embedded text or text located around those images.

3. Goals and approaches

In this section, a summary of tasks followed in this paper to utilize a personal large content of emails for emails' data mining is described.

1. Data collection stage:

In this paper, a Gmail personal email of 19,620 emails (excluding spam or junk emails) is collected. General statistics about the emails' dataset is collected from Google report provided for Gmail accounts' users. Total number of contacts is 2400 (Based on Google activity report). Total number of distinct terms in the emails' dataset is 303,381. Google includes also other information in activity report related to conversations. 12,711 is the activity conversations value for the author personal email. This indicates the emails that include more than one single flow from the email (sending or receiving).

An open source software tool is used to parse those emails into .EML extension text files (https://code.google.com/p/got-your-back/).

2. Emails parsing and pre-processing: A MIME parser is then used to parse information from those emails to generate a dataset that include one record for each email with the following information parsed: Email file name, email body, from, subject, and sending date.
3. Emails' dataset data mining.

A tool is self developed to further parse all text from all emails and calculate frequency of words. More than 420,000 words are collected. Frequency of words varies from 1 to about 100,000 times.

We selected words of frequency of above 100 in the whole email set. Stemming is also applied in the term frequency table to stem out some of the generic terms that are usually excluded in most natural language processing activities such as: You, is, a, your, I, at, be, will, on, PM, AM, are, that, this, with, have, for, from, etc. Although those terms have a high frequency in the frequency table, hence they are excluded as those may not be relevant to the further process such as: Feature extraction, text categorization, etc.

Five classes are proposed to label the nature of emails users may have: Personal, Job, Profession, Friendship, and Others.

We tried also to use clustering to assist in classification. Rather than labeling emails manually by users, we can cluster sets of emails based on some aspects through algorithms and then we need only to pick a name for developed clusters to come up with an email classification scheme. There are several approaches that can be used for clustering unstructured data to create vector space or bag of words model (Salton and McGill, 1983). Most repeated words or top frequency words are used to represent document features. From the complete email dataset words and their frequency will be collected. Stemming is then applied to remove irrelevant words or words, pronouns, verbs, adjectives that are used to connect and complete statements and hence cannot uniquely categorize a statement or a document. Using the vector space model various classical clustering algorithms such as the k-means algorithm and its variants, spherical k-means, hierarchical agglomerative can be then used. In generating the VSM, we adopted the steps described in Dhillon et al.'s (2001) paper to generate VSM to represent words by emails matrix where rows represent top frequency words and columns represent different emails. If the popular word exists in the subject email, value is one else value is zero. The model can be reversed where top frequency words can be in columns and rows can represent different emails.

Due to the large number of documents, a complete clustering process can be time consuming.

The following algorithm is developed first to perform elementary clustering to save time in initial clustering evaluation:

- Pick a random document from the emails' collection (call it seed1).
- Evaluate the similarity of seed1 to every other email in the collection via cosine similarity.
- Save the 100 most similar emails as the seed1 cluster for cosine similarity.

- Repeat for multiple seed emails.

4. Experiments and analysis

Typically, most frequent terms are collected in most natural language processing techniques for several goals such as: Clustering, classification, concept extraction, text summarization, etc. Table 1 shows the most frequent terms in the email collection after stemming or eliminating irrelevant terms or, part of speech terms that cannot be useful to distinguish emails from each other based on any classification scheme.

Most frequent terms can be also used for features or concepts' extractions. Most frequent terms can be largely divided into two categories: Generic that can be found in all emails, and bespoke which are tailored to the email owner profession, personnel, etc. Fig. 1 shows the most frequent words and their frequency in log based format. Some words are not correctly displayed in the chart as they are originally written in Arabic language.

Two methods will be evaluated and compared: Term frequency and WordNet (wordnet.princeton.edu) are used to process emails' clustering and classification. Term frequency document clustering and classification are widely used in natural language frequency and information retrieval. The Vector Space Model (VSM) approach models a two dimensional array between documents and terms. Further, there are two approaches for such model representation. In the binary case, zero or one are the only data elements in the array to indicate whether the term exists in the specific document or not (Binary weighting). In the second model representation, the number of occurrences of the word or the term is included to give further information on the number of times such term occurs in each document (Raw term weighting). In popular terms approach, document similarity between two documents is calculated based on the cumulative distance between terms in the two documents. Table 2 shows a sample of applying simple K-means clustering on the terms-emails VSM to cluster emails' popular terms in three possible clusters based on the distance of each term from the centroids of the three clusters. In this specific approach most terms are shown to be in cluster0.

Inverse Document Frequency (IDF) is used to evaluate the importance of words in their documents or the documents that they appear in. IDF is calculated for popular words based on the formula IDF = $\log(N/n)$ where N is the total number of documents (in our case emails) and n is the number of document or emails that include the subject popular term.

Table 1 Top most common words in the data set.

Word	Frequency	Word	Frequency
Izzat	25,936	Email	6364
Paper	12,199	Papers	5589
Information	9815	Dear	5310
Please	14,017	Research	5278
Software	13,511	Mohammed	5181
Computer	8101	Alsmadi	5129
University	7703	Technology	4975
2013	7246	Journal	4868
About	6959	Engineering	4777
Science	4352	Send	4554
International	4142	Message	4473

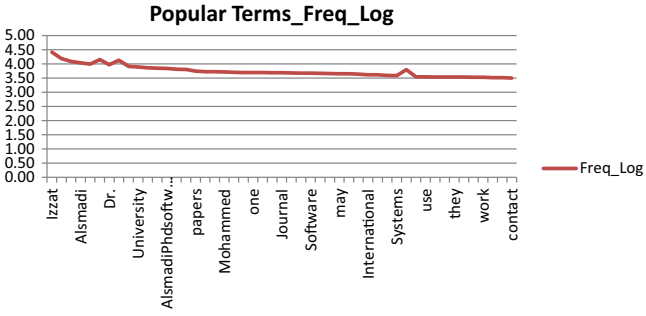

Fig. 1 Top words' frequency in the email collection (log based).

Table 2 Clustering based on term frequency: excerpt.

Attribute	Cluster0	Cluster1	Final cluster
Izzat	0.700	0.134	Cluster0
PM	0.350	0.031	Cluster0
2012	0.000	0.000	Cluster3
Please	0.500	0.258	Cluster0
From	0.350	0.155	Cluster0
Software	0.200	0.052	Cluster0
Paper	0.400	0.031	Cluster0
Alsmadi	0.800	0.134	Cluster0
Information	0.100	0.186	Cluster1
Dr.	0.500	0.165	Cluster0
AM	0.400	0.083	Cluster0
Computer	0.400	0.144	Cluster0
University	0.400	0.165	Cluster0
2013	0.000	0.000	Cluster3

$$\log \frac{Number\ of\ Document}{Document\ Frequency}$$

Fig. 2 Method to calculate IDF.

The equation below (Fig. 2) shows the method used to calculate IDF. Document or email frequency is calculated in a separate method (Fig. 3).

Further, documents' frequency uses another method (GetWordFrequency) that calculates frequency of words in the documents.

Table 3 shows popular terms in the email dataset and their IDF. Most popular terms are usually given zero value.

Fig. 4 shows the 8 selected clusters with similarity of each email with the centroid email. Horizontal line represents email number in the cluster (just random number) and vertical axis represents similarity value between that email and its cluster centroid. Similarity value is then calculated between 0 and 1. A cluster with higher similarity values (e.g. cluster 7) is then

1. For each document in the collection, Do:
2. For each word in the current document, calculate words frequency
3. Normalize word or term frequency based on total terms in document
4. Output is terms in documents with their normalized frequency

Fig. 3 Method to calculate documents' frequency.

Table 3 Popular terms and their IDF values.

Term	IDF	Term	IDF
Izzat	0.00	About	1.39
2012	0.69	Alsmadi Phd software	1.10
Please	1.10	الله	2.08
From	1.10	Email	1.39
Software	0.69	Conference	2.30
Paper	1.10	Papers	2.08
Alsmadi	0.69	Dear	0.69
Information	1.39	Research	1.79
Dr.	1.10	Mohammed	2.20
AM	1.39	Need	1.61
Computer	1.39	Technology	1.61
University	1.10	Information	1.61
2013	1.79	Journal	2.08
على	2.08	Please	1.39

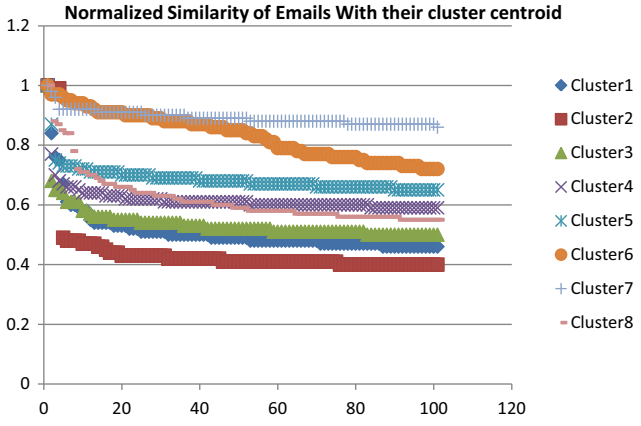

Fig. 4 Normalized similarity of emails with their cluster centroid.

a homogenous cluster. On the contract, cluster 2 has the lowest similarity values with the least good cluster in terms of similarity between emails in the cluster.

K-means clustering is used to cluster data points into different clusters where distance between elements of the cluster and its centroid is minimized. We assumed that distance is the complement of similarity that is calculated between emails. Random centroids are selected and the algorithm selects those that have a small overall distance between all data elements (one at a time) and the centroid. Fig. 3 shows different overall distances for clusters of a fixed number of data points in each cluster which is 150. The algorithm is similar to that explained earlier where closest 150 data points are selected to each randomly selected centroid. Highest and lowest similarities represent those of the 150 emails or data points that are closest to the selected email.

The table (see Fig. 5 and Table 4) below shows results from one round of running the experiment with 5 randomly selected emails from the pool as centroids. According to the overall distance cluster 2 is the best and cluster 4 is the worst. The process can be repeated for all emails as centroids and then select the best 4, or any number, of clusters with their best team selection according to elements' distance from the center.

4.1. Classification based on WordNet class

WordNet is a popular lexical database for English language. In this lexicon, language constructs: Nouns, verbs, adjectives, adverbs are grouped into synonyms. Clustering is then applied based on term frequency and WordNet.

Based on WordNet English language and lexicon, we developed a method to measure similarity between emails' content and body. A matrix of all emails is built to calculate similarity as a percentage between all emails one to one. Table 5 shows a sample of data in the very large table that is constructed from the 19,620 emails in one to one cases. Similarity value between each two-emails-couple varies between zero and one.

Emails are then clustered based on similarity values into different scales: 90–100% – A, 80–90% B, etc. In most clustering algorithms, distance is used as the major factor to cluster different elements. Distance is seen as the opposite or the complement of similarity where the distance between the element and itself is zero and maximum distance between two elements is normalized to 1.

Standard K-means clustering is based on considering clusters' centroids (e.g. CC0, CC1, CC2, etc). For each element, it is assigned to a CC based on its closest distance. There are many similarity distance measures to include such as: Euclidian or Cosine, Manhattan, etc. Elements or data points are typically expected to be vectors.

On the other hand, relational K-means is proposed to deal with non-vector data. Based on relational K-means, we developed an algorithm to take the similarity, or the complement of similarity, matrix between emails in the dataset as input and generate clusters as output. User can specify initially the number of clusters to generate. The relational K-means clustering in this case is similar to that described in Szalkai (2013).

Based on relational K-means clustering and the distance matrix, the clustering algorithm produced results shown in Figs. 6–8 for the number of clusters: 5, 6 and 7, respectively.

Each cluster is named with a unique number. K-means is used to evaluate which selection (i.e. 5, 6 or 7 clusters) can show better results in terms of emails-clusters' distribution.

In Fig. 9, mean and standard deviation are shown below for the 5, 6, and 7-clusters' selections respectively. The figure shows that 5 clusters selection is better than the others where both mean and standard deviation indicate homogenous distribution of emails in the different clusters.

Clustering the whole set of more than 19,000 emails using the approaches described earlier can be time consuming. For example in the similarity TF/IDF approach, more than 30,000 unique terms should be evaluated in more than 19,000 cycles for the documents to best estimate the proper cluster. The Word-net approach can be also time consuming given the large number of unique terms and also the large number of emails.

4.2. Cluster and classification evaluation

Classification based on the clustering process can be used to indirectly evaluate the quality of the clustering process.

In this section, three classification methods will be used to classify emails from the dataset into one of different classes. A Support Vector Machine (SVM) classification model is

0	0.25	0.28	0.3	0.27	0.27	0.29	0.29	0.3	0.26	0.27	0.31	0.29	0.31	0.38	0.38	0.28	0.48
0.25	0	0.17	0.16	0.18	0.18	0.2	0.19	0.21	0.15	0.16	0.15	0.17	0.17	0.37	0.38	0.19	0.55
0.28	0.17	0	0.14	0.15	0.14	0.13	0.13	0.12	0.13	0.12	0.12	0.11	0.1	0.38	0.39	0.13	0.52
0.3	0.16	0.14	0	0.17	0.16	0.17	0.17	0.16	0.15	0.15	0.15	0.15	0.15	0.37	0.38	0.17	0.55
0.27	0.18	0.15	0.17	0	0.01	0.02	0.03	0.04	0	0.01	0.02	0.03	0.04	0.2	0.21	0.03	0.39
0.27	0.18	0.14	0.16	0.01	0	0	0.01	0.02	0.01	0	0.01	0.01	0.02	0.19	0.2	0.01	0.37
0.29	0.2	0.13	0.17	0.02	0	0	0	0.02	0.01	0.01	0.01	0.01	0.02	0.2	0.2	0	0.37
0.29	0.19	0.13	0.17	0.03	0.01	0	0	0.02	0.01	0.01	0.01	0	0.02	0.2	0.2	0	0.38
0.3	0.21	0.12	0.16	0.04	0.02	0.02	0.02	0	0.02	0.02	0.02	0.02	0.02	0.19	0.19	0.01	0.38
0.26	0.15	0.13	0.15	0	0.01	0.01	0.01	0.02	0	0.01	0.02	0.01	0.03	0.2	0.2	0.01	0.49
0.27	0.16	0.12	0.15	0.01	0	0.01	0.01	0.02	0.01	0	0.01	0	0.02	0.2	0.21	0.01	0.48
0.31	0.15	0.12	0.15	0.02	0.01	0.01	0.01	0.02	0.02	0.01	0	0.01	0.03	0.23	0.24	0.01	0.49
0.29	0.17	0.11	0.15	0.03	0.01	0.01	0	0.02	0.01	0	0.01	0	0.02	0.21	0.21	0.01	0.47
0.31	0.17	0.1	0.15	0.04	0.02	0.02	0.02	0.02	0.03	0.02	0.03	0.02	0	0.24	0.24	0.02	0.5
0.38	0.37	0.38	0.37	0.2	0.19	0.2	0.2	0.19	0.2	0.2	0.23	0.21	0.24	0	0.01	0.2	0.17
0.38	0.38	0.39	0.38	0.21	0.2	0.2	0.2	0.19	0.2	0.21	0.24	0.21	0.24	0.01	0	0.2	0.18
0.28	0.19	0.13	0.17	0.03	0.01	0	0	0.01	0.01	0.01	0.01	0.01	0.02	0.2	0.2	0	0.37
0.48	0.55	0.52	0.55	0.39	0.37	0.37	0.38	0.38	0.49	0.48	0.49	0.47	0.5	0.16	0.18	0.37	0
0.29	0.19	0.13	0.17	0.03	0.01	0	0	0.01	0.01	0.01	0.01	0.01	0.02	0.2	0.2	0	0.38
0.29	0.2	0.12	0.17	0.03	0.01	0	0	0.01	0.02	0.01	0.01	0.01	0.02	0.19	0.2	0	0.37
0.22	0.14	0.1	0.13	0.15	0.14	0.14	0.14	0.12	0.13	0.13	0.13	0.12	0.11	0.34	0.35	0.14	0.55
0.28	0.15	0.1	0.13	0.16	0.14	0.14	0.14	0.12	0.14	0.13	0.13	0.13	0.11	0.41	0.42	0.14	0.56
0.16	0.23	0.21	0.23	0.27	0.27	0.29	0.28	0.27	0.35	0.35	0.35	0.36	0.37	0.36	0.36	0.28	0.45
0.19	0.21	0.17	0.22	0.24	0.22	0.22	0.21	0.21	0.3	0.29	0.29	0.28	0.25	0.35	0.37	0.21	0.46
0.29	0.33	0.27	0.36	0.35	0.34	0.35	0.35	0.35	0.42	0.42	0.42	0.42	0.35	0.44	0.44	0.35	0.52
0.23	0.27	0.25	0.32	0.31	0.3	0.32	0.31	0.31	0.38	0.38	0.39	0.39	0.35	0.37	0.38	0.31	0.46
0.45	0.42	0.4	0.45	0.39	0.38	0.38	0.37	0.36	0.35	0.35	0.38	0.35	0.32	0.43	0.43	0.37	0.59
0.87	0.88	0.85	0.85	0.87	0.87	0.86	0.85	0.87	0.89	0.89	0.88	0.87	0.81	0.81	0.83	0.86	0.76

Fig. 5 An excerpt of emails' distance matrix (0 for the email with itself).

Table 4 Randomly selected clusters with their overall distances.

Cluster	Email (selected as centroid)	Highest similarity	Lowest similarity	Overall distance
1	26170.eml	47675.eml,0.58	102812.eml,0.47	76.32
2	111736.eml	111728.eml,0.92	82361.eml,0.73	37.78
3	114692.eml	114662.eml,0.63	100277.eml,0.45	77.32
4	1–77648.eml	20026.eml,0.67	95197.eml,0.23	81.11

Table 5 Sorting emails based on 1-1 most similar relations.

EmailA	EmailB	Similarity
email6	email7	0.996917
email6	email11	0.996263
email5	email10	0.995885
email6	email17	0.994474
email6	email19	0.994216
email6	email8	0.993634
email6	email10	0.993382
email6	email20	0.992123
email6	email13	0.990391
email6	email12	0.989116
email5	email11	0.98774
email5	email6	0.98773
email6	email5	0.98773
email6	email9	0.987135
email5	email7	0.97857
email5	email12	0.976986

formed. The three SVM models that are evaluated can be summarized into:

1. Top 100 words-VS-emails before removing stop words

In this scenario, the top 100 most frequent words in the whole emails' collection are collected. An SVM matrix is then formulated programmatically between those words and emails. Columns represent top frequent words while rows represent emails. Values represent the number of occurrences of the word in the email. In this part any word with letters three and above will be considered. This means that stop words are not removed.

If we want to construct a matrix for testing with the complete set of emails of more than 19,000, the matrix can be very large for data computation. As such, 100 emails are selected

```
0;->;email1;email2;email3;email4;email5;email6;email7;email8;email9;email10;email11;email12
;email13;email14;email17;email19;email20;email21;email22;email23;email24;email25;email26;
email36;email52;email55;email60;email68;email71;email76;email82;email97;email99
1;->;email27;email31;email37;email40;email41;email42;email45;email51;email53;email54;email58;
email61;email62;email66;email67;email72;email73;email74;email75;email77;email78;email79;email85;email87
2;->;email59;email69;email70;email81;email83;email88;email89;email92;email98;email100
3;->;email28;email50;email63;email93
4;->;email15;email16;email18;email29;email30;email32;email33;email34;email35;email38;email39;email43;
email44;email46;email47;email48;email49;email56;email57;email64;email65;email80;email84;email86;
email90;email91;email94;email95;email96
```

Fig. 6 Relational K-means clustering: 5 clusters.

```
0;->;email59;email69;email70;email81;email83;email88;email89;email92;email98;email100
1;->;email15;email16;email18;email33;email35;email43;email44;email47;email48;email56;
email57;email65;email77;email79;email80;email90;email94;email96
2;->;email31;email37;email40;email41;email42;email45;email51;email53;email54;email58;
email61;email62;email66;email67;email68;email72;email73;email74;email75;email78;email85;email87
3;->;email1;email2;email3;email4;email5;email6;email7;email8;email9;email10;email11;email12;
email13;email14;email17;email19;email20;email21;email22;email23;email24;email25;email26;email36;
email52;email55;email60;email71;email76;email82;email97;email99
4;->;email27;email29;email30;email32;email34;email38;email39;email46;
email49;email64;email84;email86;email91;email95
5;->;email28;email50;email63;email93
```

Fig. 7 Relational K-means clustering: 6 clusters.

```
0;->;email28;email50;email63;email93
1;->;email29;email30;email32;email46;email55;email71;email86;email91;email95;email99
2;->;email27;email31;email37;email40;email41;email42;email45;email51;email53;email54;email58;
email61;email62;email66;email67;email72;email73;email74;email75;email77;email78;email85;email87
3;->;email33;email34;email35;email38;email47;email48;email64;email65;email80;email84;email90;email94
4;->;email59;email69;email70;email81;email83;email88;email89;email92;email98;email100
5;->;email1;email2;email3;email4;email5;email6;email7;email8;email9;email10;email11;email12;email13;
email14;email17;email19;email20;email21;email22;email23;email24;email25;email26;email36;email52;
email60;email68;email76;email82;email97
6;->;email15;email16;email18;email39;email43;email44;email49;email56;email57;email79;email96
```

Fig. 8 Relational K-means clustering: 7 clusters.

```
     mean              0.5879     0.5637     0.5767       0.58     0.5517
     std. dev.         0.2305     0.2213     0.2242     0.2376     0.2343

mean              0.6595     0.6295     0.6249     0.6492     0.6555     0.0931
std. dev.         0.1441     0.1454     0.1525     0.1357     0.1469     0.0238

mean              0.7707     0.7505     0.7549     0.5266     0.7678     0.0924
std. dev.         0.0688     0.0622     0.0602     0.1153     0.0594     0.0228
```

Fig. 9 Mean and standard deviation for the different clusters.

randomly from the emails' dataset. The 100 top words Vs 100 emails matrix is then constructed programmatically. As mentioned earlier, values in the matrix represent the number of time the word is repeated in the particular email (Fig. 10).

Figs. 11 and 12 show two rounds of testing emails' classification based on a new classifier called LibSVM (available in WEKA 3.7). In order to specify class labels required by the classification algorithm, KNN clustering algorithm is used for three clusters with randomly selected centroids. This is why prediction accuracy may vary from one cycle to another based on the quality of the randomly selected centroids.

While evaluated methods showed always 100% TP rate, its FP rate was always high. This is because Arabic terms that were included in the SVM matrix were not correctly recognized – due to encoding problem with WEKA data mining tool and its inability to recognize Arabic terms.

Process can be repeated many times and based on prediction performance we can set the best clustering scheme.

2. Top 100 words-VS-emails after removing stop words

In this section, same previous process is repeated. The only difference is that the selection of the top 100 frequent words is considered after removing stop words. Since the majority of the text in the emails is in English, English stop words are used. Arabic stop words are not included as most frequent words appear to be in English.

email	and	the	for	that	you	your	are	with	2010	from	will
1-27328.e	0	0	0	0	0	0	0	0	0	0	0
1-27330.e	0	0	0	0	0	0	0	0	0	0	0
1-27331.e	0	0	0	0	0	0	0	0	0	0	0
1-27959.e	7	0	5	2	5	4	0	6	6	5	0
1-27990.e	16	16	2	13	2	3	4	0	3	2	2
1-27997.e	0	0	1	0	0	2	0	0	0	0	0
1-28046.e	0	2	0	1	1	2	0	0	0	2	1
1-28047.e	0	0	0	0	0	0	0	0	0	0	0
1-28074.e	1	0	1	1	1	2	0	0	0	1	0
1-28075.e	22	24	7	19	6	6	7	2	3	2	3
1-28118.e	1	1	0	0	0	3	0	0	0	1	0
1-28131.e	0	0	0	0	0	0	0	0	0	0	0
1-28135.e	1	0	1	0	0	0	0	0	0	0	0
1-28175.e	1	1	0	0	1	3	0	1	1	1	0
1-28184.e	0	0	1	1	0	0	0	0	0	0	0
1-28190.e	0	0	0	0	0	0	0	0	0	0	0
1-28193.e	0	0	0	0	1	2	0	1	0	0	0
1-28231.e	1	1	3	1	2	3	0	0	0	0	1
1-28234.e	0	0	1	1	1	0	1	0	0	0	0
1-28266.e	1	1	1	0	4	8	0	5	1	2	0

Fig. 10 A screen shot of the SVM output.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         319               80.1508 %
Incorrectly Classified Instances        79               19.8492 %
Kappa statistic                          0.1536
Mean absolute error                      0.1323
Root mean squared error                  0.3638
Relative absolute error                 55.1191 %
Root relative squared error            105.3604 %
Coverage of cases (0.95 level)          80.1508 %
Mean rel. region size (0.95 level)      33.3333 %
Total Number of Instances              398

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.898    0.797      1.000    0.887      0.285    0.551     0.797     cluster2
                 0.122    0.000    1.000      0.122    0.217      0.318    0.561     0.285     cluster1
                 0.000    0.000    0.000      0.000    0.000      0.000    0.500     0.035     cluster3
Weighted Avg.    0.802    0.699    0.807      0.802    0.731      0.282    0.551     0.675

=== Confusion Matrix ===

   a   b   c   <-- classified as
 310   0   0 |   a = cluster2
  65   9   0 |   b = cluster1
  14   0   0 |   c = cluster3
```

Fig. 11 Emails' classification prediction characteristics: LibSVM classifier (Trial 1).

Initial notice from this experiment is that removing stop words cause removing most of the top 100 words or terms appeared in the earlier experiments. In other words, most of the terms in this set are shown to be different from those of the earlier experiments that did not include the exclusion of stop words.

Same process is repeated in this experiment as that of the earlier one. Fig. 13 shows a run of classification experiment after removing stop words. TP is still at 100% in the largest cluster. FP is reduced in comparison with earlier experiments.

3. NGram terms-VS-emails

As an alternative to using top frequent words in the SVM columns, NGram can be used. NGram (e.g. 3 g, 4 g, 5 g, etc.) is a process that includes dividing the whole text content into sub-terms based on the gram size. For example, in the line Hello dear, the 3 g output will be: Hel, ell, llo, lo, o d, de, dea, and ear. Spaces can be considered or ignored. In addition, usually stop words are included in the NGram process. The whole number of NGram in the complete set will be huge. As such, to

```
Correctly Classified Instances         368               92.4623 %
Incorrectly Classified Instances        30                7.5377 %
Kappa statistic                          0
Mean absolute error                      0.0503
Root mean squared error                  0.2242
Relative absolute error                 51.9459 %
Root relative squared error            103.47   %
Coverage of cases (0.95 level)          92.4623 %
Mean rel. region size (0.95 level)      33.3333 %
Total Number of Instances              398

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               1.000    1.000    0.925      1.000    0.961      0.000    0.500     0.925     cluster3
               0.000    0.000    0.000      0.000    0.000      0.000    0.500     0.065     cluster1
               0.000    0.000    0.000      0.000    0.000      0.000    0.500     0.010     cluster2
Weighted Avg.  0.925    0.925    0.855      0.925    0.888      0.000    0.500     0.859

=== Confusion Matrix ===

   a   b   c   <-- classified as
 368   0   0 |   a = cluster3
  26   0   0 |   b = cluster1
   4   0   0 |   c = cluster2
```

Fig. 12 Emails' classification prediction characteristics: LibSVM classifier (Trial 2).

```
Correctly Classified Instances         388               97.4874 %
Incorrectly Classified Instances        10                2.5126 %
Kappa statistic                          0.6551
Mean absolute error                      0.0251
Root mean squared error                  0.1585
Relative absolute error                 25.7174 %
Root relative squared error             72.5524 %
Coverage of cases (0.95 level)          97.4874 %
Mean rel. region size (0.95 level)      50      %
Total Number of Instances              398

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               1.000    0.500    0.974      1.000    0.987      0.698    0.750     0.974     cluster3
               0.500    0.000    1.000      0.500    0.667      0.698    0.750     0.525     cluster1
Weighted Avg.  0.975    0.475    0.976      0.975    0.971      0.698    0.750     0.952

=== Confusion Matrix ===

   a   b   <-- classified as
 378   0 |   a = cluster3
  10  10 |   b = cluster1
```

Fig. 13 Emails' classification experiment after removing stop words.

produce a smaller size SVM with possible significant content, top 1000 g are selected for SVM columns' matrix.

One experiment is selected on 4 g size. In this experiment, all 4 g are parsed. Top 1000 4-grams based on their occurrences in the emails are selected. Then same steps of previous experiments are repeated.

Table 6 below shows top 20 4 g of the dataset (shown as a sample).

Fig. 14 below shows prediction performance from using 4-gram classification.

It can be seen that while TP is a little less than 100%, FP rate is significantly improved in all clusters in 4 g experiment in comparison with previous experiments. Dealing with stop words' issues and bi-language text, NGram approaches can

Table 6 Top 20: 4 g.

NGram	Freq.	NGram	Freq.
tion	2864	form	610
atio	2172	cati	585
mail	1115	your	556
ment	870	2010	552
ence	795	nter	551
ahoo	759	sion	546
yaho	742	icat	543
that	736	with	537
ions	654	arch	534

```
Correctly Classified Instances         352               88.4422 %
Incorrectly Classified Instances        46               11.5578 %
Kappa statistic                          0.7354
Mean absolute error                      0.0771
Root mean squared error                  0.2776
Relative absolute error                 25.3109 %
Root relative squared error             71.2825 %
Coverage of cases (0.95 level)          88.4422 %
Mean rel. region size (0.95 level)      33.3333 %
Total Number of Instances              398

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0.957    0.148    0.936      0.957    0.946      0.821    0.904     0.926     cluster3
               0.898    0.093    0.759      0.898    0.822      0.763    0.902     0.706     cluster1
               0.000    0.000    0.000      0.000    0.000      0.000    0.500     0.060     cluster2
Weighted Avg.  0.884    0.125    0.836      0.884    0.859      0.757    0.880     0.819

=== Confusion Matrix ===

   a   b   c   <-- classified as
 264  12   0 |   a = cluster3
  10  88   0 |   b = cluster1
   8  16   0 |   c = cluster2
```

Fig. 14 4-Gram emails' classification results.

be in many cases better than typical words-frequency stemming and classification.

5. Conclusion

Documents' classification in general and emails' classification in particular utilize several natural language processing and data mining activities such as: Text parsing, stemming, classification, clustering, etc. There are many goals or reasons why to cluster or classify emails whether in real time or historical. This may include reasons such as: Spam detection, subject or folder classification, etc.

In this paper a personal large dataset of emails is collected and assembled. Several approaches are evaluated to cluster the emails based on their contents. Manual or supervised classification can be much more reliable and effective. However, in many cases, there is a need to perform such process automatically. Several clustering methods were evaluated based on samples' selection or based on utilizing the complete emails' dataset.

Classification algorithms are conducted to evaluate the performance of experimented cluster algorithms. True Positive TP rate is shown to be very high in all cases. However, FP rate was shown to be the best in case of NGram based clustering and classification. Such accuracy can also depend on the number of folders in the classification scheme.

The major challenge we noticed in the analysis process is the large number of emails and the large number of unique terms that are used as inputs to the clustering and classification processes. It is desirable in future that email servers or applications should include different types of pre-defined folders. The first category includes the general traditional folders: Mailbox, sent, trash, etc. It should also allow users to add new folders that can be user defined as well as intelligent or context aware. In convergence with social networks, users should be able to classify emails based on senders or content into different groups.

References

Abdullah Al-Kadhi, Mishaal, 2011. Assessment of the status of spam in the Kingdom of Saudi Arabia. J. King Saud Univ. Comput. Inf. Sci. 23, 45–58.

Aloui, Awatef, Neji, Mahmoud, 2010. Automatic classification and response of E-mails. Int. J. Digital Soc. (IJDS) 1 (1).

Altwaijry, Hesham, Algarny, Saeed, 2012. Bayesian based intrusion detection system. J. King Saud Univ. Comput. Inf. Sci. 24, 1–6.

Ron Bekkerman, Andrew McCallum, Gary Huang, 2004. Automatic Categorization of Email into Folders: Benchmark Experiments on Enron and SRI Corpora.

Beseiso, Majdi, Rahim Ahmad, Abdul, Ismail, Roslan, 2012. A new architecture for email knowledge extraction. Int. J. Web Semantic Technol. (IJWesT) 3 (3).

Christian Bird, 2004. Predicting Email Response using Mined Data, <http://www.cabird.com/papers/mlpaper.pdf> (last accessed 2014).

Enrico Blanzieri, Anton Bryl, 2008. A survey of learning-based techniques of email spam filtering, Technical Report # DIT-06-056.

Pranjal S. Bogawar, Kishor K. Bhoyar, 2012. Email mining: a review, IJCSI Int. J. Comput. Sci. Issues 9(1), No 1, January 2012.

Carmona-Cejudo, José M., Baena-García, Manuel, Morales Bueno, Rafael, Gama, João, Bifet, Albert, 2011. Using GNUsmail to compare data stream mining methods for on-line email classification. J. Mach. Learn. Res. Proc. Track 17, 12–18.

De, Indrajit, Sil, Jaya, 2012. Entropy based fuzzy classification of images on quality assessment. J. King Saud Univ. Comput. Inf. Sci. 24, 165–173.

Dhillon, I.S., Guan, Y., Fan, J., 2011. Efficient clustering of very large document collections. In: Data Mining for Scientific and Engineering Applications, Kluwer Academic Publishers, Dordrecht, pp. 357–381.

Elements of Computer Security, 2010. David Salomon. Springer-Verlag, London Limited.

Email Statistics Report, 2011–2015 – Executive Summary, Radicati Group, 2011. <http://www.radicati.com/wp/wp-content/uploads/2011/05/Email-Statistics-Report-2011-2015-Executive-Summary.pdf>.

Enron email dataset, http://people.cs.umass.edu/≃ronb/enron_dataset.html.

Ioannis Katakis, Grigorios Tsoumakas, Ioannis Vlahavas, 2007. Email Mining: Emerging Techniques for Email Management, Web Data Management Practices: Emerging Techniques and Technologies, IGI.

Svetlana Kiritchenko, Stan Matwin, 2001. Email classification with co-training. In: CASCON '01: Proceedings of the 2001 conference of the Centre for Advanced Studies on Collaborative research.

Klimt, Bryan, Yang, Yiming, 2004. The Enron corpus: a new dataset for email classification research. ECML, 217–226.

Thao Lê, Quynh Lê, 2002. 'The Nature of Learners' email communication. In: Proceedings of the International Conference on Computers in Education.

McCallum, Andrew, Wang, Xuerui, 2007. Andrés Corrada-Emmanuel, Enron and academic email. J. Artif. Intell. Res. 30, 249–272.

Mishne, G., Carmel, D., Lempel, R., 2005. Blocking blog spam with language model disagreement. In Proc. 1st AIRWeb, Chiba, Japan.

Office of Civilian Radioactive Waste Management, 1992. Programs records management: Receipt and handling of program records and records packages, September 1992. Accession Number: HQF.930510.0048.

Cagri Ozcaglar, 2008. Classification of email messages into topics using latent dirichlet allocation, Master thesis, Rensselaer Polytechnic Institute Troy, New York.

Pérez-Díaz, Noemí, Ruano-Ordás, David, Méndez, José R., Gálvez, Juan F., Fdez-Riverola, Florentino, 2012. Rough sets for spam filtering: selecting appropriate decision rules for boundary e-mail classification. Appl. Soft Comput. 12, 3671–3682.

Salton, G., McGill, M.J., 1983. Introduction to Modern Retrieval. McGraw-Hill Book Company.

Sculley, D., Gordon V. Cormack, 2008. Filtering Email Spam in the Presence of Noisy User Feedback, CEAS.

Sculley, D., Gabriel M. Wachman, 2007. Relaxed online VSMs for spam filtering, SIGIR 2007 Proceedings.

Jitesh Shetty, Jafar Adibi, 2005. Discovering Important Nodes through Graph Entropy the Case of Enron Email Database, KDD'2005, Chicago, Illinois.

Szalkai, B., 2013. An implementation of the relational k-means algorithm. ArXiv e-prints.

Kazem Taghva, Julie Borsack, Jeffrey S. Coombs, Allen Condit, Steven Lumos, Thomas A. Nartker, 2003. Ontology-based Classification of Email, ITCC, IEEE Computer Society, pp. 194–198.

UCI Machine Learning Repository, Spambase, <http://archive.ics.uci.edu/ml/datasets/Spambase>.

Steve Webb, James Caverlee, Calton Pu, 2006. Introducing the Webb Spam Corpus: using Email spam to identify web spam automatically, CEAS.

Mengjun Xie, Heng Yin, Haining Wang, 2006. An effective defense against email spam laundering, CCS'06, October 30–November 3, Alexandria, Virginia, USA.

Yang, H., Callan, J., 2008. Ontology generation for large email collections, Proceedings of the 2008 international conference on digital government research, pp. 254–261.

Shinjae Yoo, Yiming Yang, Frank Lin, Il-Chul Moon, 2009. Mining Social Networks for Personalized Email Prioritization, KDD'09, June 28–July 1, Paris, France.

Bing Zhou, Yiyu Yao, Jigang Luo, 2010. A three-way decision approach to email spam filtering. Canadian Conference on AI, pp. 28–39.

Zhuang, L., Dunagan, J., Simon, D.R., Wang, H.J., Tygar, J.D., 2008. Characterizing Botnets from Email Spam Records, LEET'08 Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats Article No. 2.