Reminiscing Through Personal Email

Mikhail Sushkov
Stanford University
msushkov@stanford.edu

John Doherty
Stanford University
doherty1@stanford.edu

ABSTRACT

Personal email stores a wealth of information about our actions and thoughts through the years, providing rich material for reminiscence. However, browsing through thousands of messages is tedious and impractical without a proper summary of the data. We present a system that gives a high-level overview of an email archive by extracting the underlying topics of the emails and presenting the user with a series of messages that best represent these topics. Topics are found using Latent Dirichlet Allocation (LDA), and the representative set of emails is extracted by modeling the set of email messages as a factor graph and solving a Constraint Satisfaction Problem (CSP) over the set.

1. INTRODUCTION

The Stanford MobiSocial Group has developed a program called Muse [1], which analyzes personal email archives to label outgoing messages according to common emotions and sentiments [2]. Users are presented with a timeline over their years of email usage illustrating the trends in emotions and life events, providing the opportunity for reminiscence and recollection of past occurrences. While classifying email messages according to a set of emotions provides interesting insights for self-reflection, it is beneficial to also explore the underlying semantic structure of a personal email archive to extract information about the evolution of topics through time. This would give users an alternative view of their emails, and would provide information about their communication patterns and trends. This project aims to extend the current system to perform summarization of a user’s email archive, and then to use the learned topics, along with email metadata, to find a set of important events in the user’s life. The final output will be a set of emails determined to be most representative of the messages within a particular topic.

2. DATA

The training data used is the corpus of Sarah Palin’s emails released in 2011. The data set contains over 15,000 email messages from Sarah Palin’s time as Alaska governor. The emails range in dates from December 2006 to September 2008.

3. PREPROCESSING

The data is cleaned through the following preprocessing steps: conversion to lowercase; removal of terms with under 3 characters; removal of rarest 5 terms and most frequent 30 terms; removal of messages with less than 7 words; stop-word removal; tokenization. Apache Lucene is used to perform these steps [3]. After preprocessing, the data set contains 11,605 messages.

4. SUMMARY EXTRACTION

4.1 Initial Approach: Clustering

K-means clustering is applied as a first attempt at extracting summary information from the messages. Emails are converted into vector space by computing tf-idf scores for each word and storing them as sparse vectors [4]. Then, messages are clustered using k-means++, with cosine similarity as the measure of distance between emails [5, 6]. The k-means++ algorithm differs from traditional k-means only at the initialization step, and has been shown to find a more optimal solution [5]. The Apache Commons Java implementation of k-means++ is used for this step [7]. For computational efficiency, each email is represented as a vector of terms with the top 200 tf-idf scores.

4.1.1. Clustering Results

To evaluate clustering performance, a random sample of 50 email pairs was chosen, and each pair was manually labeled as 0 or 1 (0 if the messages were judged to belong to different clusters, and 1 if the messages were judged to belong to the same cluster). The judging criterion was the perceived overlap between terms. Clustering performance was evaluated by checking whether each of the 50 pairs of emails was placed into the same or different clusters. Fig. 1 summarizes the results for different values of k, the number of clusters.

k	Error Rate	Runtime (s)
5	0.36	261
15	0.19	754
25	0.15	1177

Figure 1. Clustering performance on labeled test set.

Given the user-facing nature of this system, the above metric is insufficient by itself to judge clustering performance. A better evaluation metric is to examine the set of terms that each cluster represents, and to evaluate whether the given clustering of terms makes sense. Fig. 2 presents the terms with the highest tf-idf scores for each cluster center, for several clusters after running k-means++ with k = 25.

[alerts, business, center, education, fish, gov.state.ak.us, http, important, law, letter, mail, national, policy, research, spam, states, web, webmail, wiche, writing]

[465, agia, bill, budget, comment, dnr, energy, gas, gov.state.ak.us, juneau, personal, phone, pipeline, press, privileged, project, public, sharon, staff, work]

[comment, concerns, dear, gov.state.ak.us, govweb.alaska.gov, important, mail, mailto:webmail, opinions, person, received, respond, reviewed, staff, suggestion, unable, valuable, web, webmail, writing]

[aces, business, cavuto, conf, economists, featured, gas, goofy, ill, natural, oil, pipeline, pis, project, show, speakers, sponsor, support, understanding, women]

Figure 2. Terms describing several representative cluster centers after running k-means++ with k = 25.

It is clear that the terms given by the cluster centroids are noisy and do not do a great job of summarizing the data set, giving somewhat ambiguous results. Moreover, the runtime of nearly 20 minutes for k = 25 makes this approach prohibitively slow. Thus, it is necessary to explore more sophisticated techniques to summarize email archives.

4.2 Topic Modeling

Much recent work has been aimed at extracting information from text corpora using topic modeling [8]. This approach assumes that a collection of documents can be represented by a few underlying themes, or collections of terms. One of the most well-known topic-modeling techniques is LDA [9]. LDA is a generative model that assumes that for a collection of documents, a mixture of topics produces the words in the documents with some probabilities (fig. 3). The goal is to find the set of topics most likely to have generated the given set of documents [10]. Thus, running LDA on the training set will generate a set of topics that best summarize the email corpus.

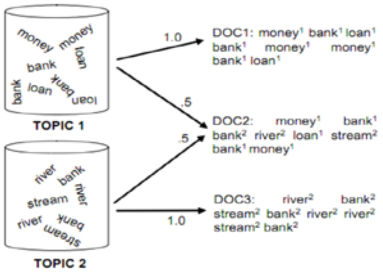
money
money
loan
bank
bank
loan
bank
loan
TOPIC 1
river
bank
river
stream
river
bank
stream
TOPIC 2
1.0
.5
.5
1.0
DOC1: money^1 bank^1 loan^1 bank^1 money^1 money^1 bank^1 loan^1
DOC2: money^1 bank^1 bank^2 river^2 loan^1 stream^2 bank^1 money^1
DOC3: river^2 bank^2 stream^2 bank^2 river^2 river^2 stream^2 bank^2

Figure 3. Generative process visualization, with probabilities of each document being drawn from each topic [11].

4.2.2 LDA Results

LDA outputs a set of topics, represented by terms in the corpus, and each message has a probability distribution over all topics. For example, a particular message could come from topic 0 with probability 0.4, topic 1 with probability 0.1, and so forth. Most email messages in the data set come from a single topic with a high probability (greater than 0.5), and from other topics with much lower probabilities. Thus it is reasonable to think of each message as originating from a single topic. This allows us to compare the performance of clustering and LDA by evaluating LDA using the same test set of 50 emails used to evaluate k-means++. A pair of emails labeled 0 would be correctly separated with LDA if the highest-probability topics for the messages are different. Similarly, a pair of emails labeled 1 would be grouped correctly by LDA if the highest-probability topics are the same. Fig. 4 summarizes these results when running the MALLET implementation of LDA on the Sarah Palin data set with 25 topics [12].

Number of Topics	Error Rate	Runtime (s)
5	0.20	80
15	0.10	90
25	0.06	115

Figure 4. LDA performance on labeled test set.

[military, national, service, page, security, army, day, support, soldiers, august, guard, veterans, home, department, air, family, september, honor, allen, policy]

[court, law, states, people, federal, u.s, united, case, government, rights, president, republican, party, supreme, attorney, campaign, general, national, amendment, mccain]

[gas, oil, energy, pipeline, tax, project, natural, north, companies, million, resources, development, costs, plan, cost, year, production, alaskans, line, years]

[hunting, game, moose, fish, wildlife, property, area, year, hunt, wolves, control, people, years, subsistence, quot, page, bears, hunters, board, wolf]

[health, care, public, services, department, benefits, safety, insurance, federal, program, section, medical, change, climate, act, provide, system, hss, mental, programs]

Figure 5. Terms describing several representative topics after running LDA on the Sarah Palin data set with 25 topics.

Similarly to the case of clustering, it is difficult to judge the quality of the results without examining the terms that represent each topic. Fig. 5 lists the top 20 representative terms for topics learned from running LDA on the data set, with 25 topics, and fig. 6 shows the evolution of the topics *military* and *court* through time.

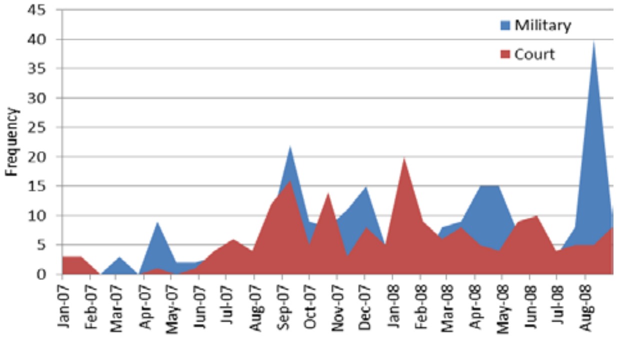

Figure 6. Trends in selected topics from Figure 6.

The results are less noisy and more coherent than those obtained through clustering – the keywords that represent each topic are clearly related, and the topics are distinct from each other in terms of their keywords.

Moreover, the error rate of 0.06 and runtime of under 2 minutes for the best-performing run of LDA (with 25 topics) significantly outperforms the error rate of 0.15 and runtime of nearly 20 minutes for the best-performing run of clustering with k = 25.

5. REPRESENTATIVE TOPIC EMAILS

While it is informative to view the summary topics of an email archive, users may want to drill down into the specific emails that produced a particular topic. Thus, it is beneficial to extract several messages that best represent each topic. The goal is to take a set of emails from a topic and find the messages that are both strongly related to the content of that topic and offer a snapshot of the topic over time.

To find an optimal subset of representative emails, several data points are available: topic probabilities (from LDA), and email timestamp (from email metadata). The topic probability is the probability of the email being associated with a given topic, as determined by LDA. The optimal subset is the group of emails that maximize both the average topic probability and time spread. Time spread is a measure of how well the time distribution of a subset of representative emails matches the time distribution of the whole set of emails in that topic, and will be explained in more detail below.

A variable-based model can be used to solve this problem. In this case, the problem can be modeled using a factor graph and solved using constraint satisfaction problem (CSP) algorithms.

To model this subset selection as a factor graph, each variable in the graph is associated with an email and has a binary domain representing whether or not it is included in the solution. Additionally, there several high-arity potentials connected to every email in the graph. There are three potentials: the constraint restricting the size of the subset, the potential for the average topic probability, and the potential for time spread (fig. 7). These factors are presented for conceptual purposes because in reality, scores are just computed over emails in the subset and never over all the emails in the factor graph.

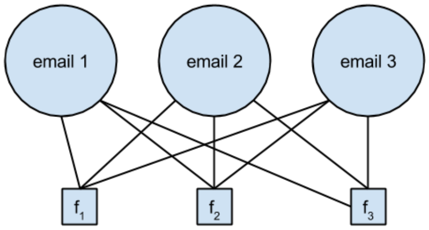

Figure 7. Factor graph for subset extraction.

Given this model, the next step is to choose an algorithm to find an optimal solution. While it would be possible to use a library or generic algorithm for this purpose, it is easier and more efficient to modify a traditional factor graph algorithm for the purposes of this problem.

The specific algorithm used to extract the email subset is based on local search. Since the size of the representative subset remains constant, it is reasonable to start with a completed assignment and modify it to find a maximum-weight assignment. The algorithm is as follows:

```
FindSubset(Emails, SubsetSize, Iterations):
  Sort Emails by topic probability
  Subset = Emails[0 ≤i ≤SubsetSize]
  Loop for Iterations:
    Loop over 0 ≤i ≤SubsetSize:
      Subset[i] = ArgMax e ∈Emails Score(Subset ∪ e)
```

Score(subset) is a scoring function to evaluate the current partial solution; it is based on average topic probability and time spread. Average topic probability is a straightforward evaluation metric, but time spread is more ambiguous. Time spread is evaluated by splitting the time range of the email set into *SubsetSize* number of buckets. The time range spanned by each bucket is based on the density of emails in that time range. For example, in fig. 10, the algorithm would attempt to place more buckets in regions of frequency spikes. Additionally, buckets placed within one of these spikes would cover a smaller time range than those placed outside of the spikes. With the time range divided into these intervals, time spread can be evaluated by computing the number of these buckets that would be filled by emails in *Subset*.

5.1 Email Extraction Results

The algorithm is effective at choosing emails that span the time range while also maintaining a high average topic probability. To check the accuracy we compare the results to other possible approaches. For example, a naive approach to would be to form a subset out of the emails with the highest probabilities for a topic. While this would give the result with the best average topic score, it says nothing about the distribution of emails in time. This can be seen in fig. 8. Here the blue line represents the frequency of emails over time in a single topic. The bottom light blue dots show points in time where the naive approach extracts emails.

On the other hand, the factor graph approach finds the emails represented by the dark blue dots. Clearly these emails are better spaced in time and the density of the emails tends to line up with the spikes of the frequency plot. In addition to producing a superior time spread, this approach also maintains a high average topic probability. While the naive approach will always produce the maximum average topic probability, the factor graph approach is only 11% worse on average over 25 topics.

Fig. 9 illustrates the representative set extraction for an example topic in the training set. It is clear that the representative emails extracted for the given topic are very relevant to the topic itself, and represent the progression of the topic in time.

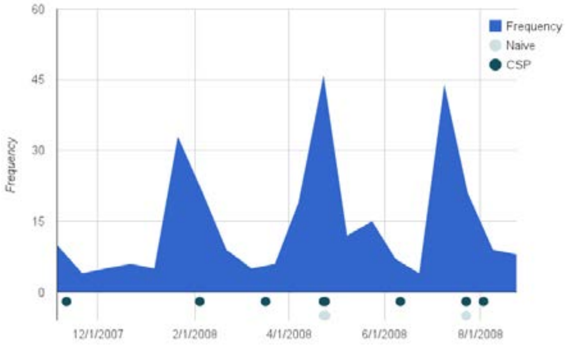

Figure 8. Example of time spreading using CSP approach. Line shows frequency of emails in example topic and dots show emails extracted using naive and CSP approaches.

Topic: *[gas oil energy project agia pipeline laa senator bill rep natural]*

Subject: Halcro took a shot at me on his blog today **Date:** 2/4/2008
…Conoco blows hot air about gas production By Ivy Frye The Alaska Gasline Inducement Act (AGIA) remains the subject debate…
…Alaska 's first … process for building a natural gas pipeline…

Subject: Kensington Gold Project **Date:** 4/23/2008
…also use the existing water treatment facility which has been shown to be effective in meeting state and federal water quality standards…

Subject: Fw: Legislative Action Alert! **Date:** 7/23/2008
…fewer permitting obstacles in Canada , and, as an independent pipeline company…
…Award of an AGIA license to TC Alaska allows the state to assist smaller pipeline projects that can provide gas…

Figure 9. Example emails extracted from the topic *gas* using the CSP approach.

6. EVALUATION: ENRON EMAILS

To further evaluate the performance of the system, the pipeline was run on a subset of the Enron email corpus. After the preprocessing steps described in section 3, the data set contained 39,741 messages.

Fig. 10 illustrates the topics extracted from the Enron data set using LDA, and fig. 11 illustrates the time spread for representative set extraction and average topic probability. Fig. 12 illustrates the representative set extraction for an example topic in the Enron email set.

[market, year, companies, million, stock, trading, investment, buy, financial, price, high, earnings, billion, capital, report, long, investors, money, term, week]

[travel, net, fares, hotel, rates, miles, travelocity, specials, city, hotels, sheraton, san, book, fare, deals, hilton, car, visit, airport, offers]

[intended, recipient, database, data, corp, error, confidential, dbcaps, sender, alias, privileged, unknown, prohibited, contract, delete, distribution, strictly, attachments, disclosure, named]

[management, group, services, risk, trading, president, power, market, industry, team, markets, development, program, global, director, technology, work, office, conference, provide]

[image, click, free, price, link, online, cgi, unsubscribe, save, bin, view, list, address, web, order, site, offer, special, service, home]

Figure 10. Terms describing several representative topics after running LDA on the Enron data set with 25 topics.

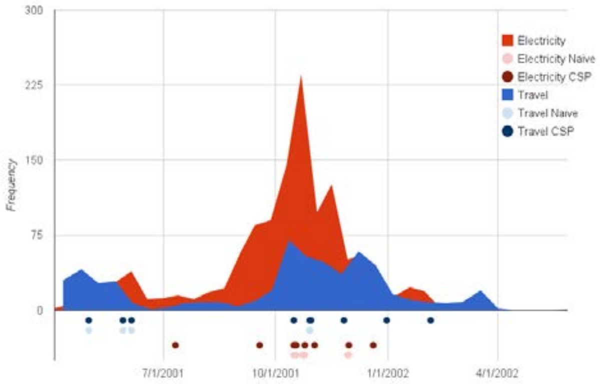

Figure 11. CSP email extraction gives a much better spread over the time range for two topics from the Enron corpus.

Topic: *[enron's stock billion trading financial credit dynegy dow jones shares market million investors year companies debt price news corp percent week copyright earnings investment company's]*

Subject: <blank> **Date**: 23 August 2001
With the U.S. dollar weaking … continue to do so, (which is clearly helpful to **Enron** while we try to sell international assets…)

Subject: Enron **Date**: 24 September 2001
… as an employee and a stock holder of Enron, I'm very concerned... I know you can't tell me where the stock will be in 12 to 18 months…

Subject: Enron Receives Dynegy $1.5B Cash Infusion Tues.
Date: 14 November 2001
...Dynegy Inc. (DYN) provided Enron Corp. (ENE) Tuesday with the $1.5 billion cash infusion envisioned in the companies' merger agreement, a Dynegy spokeswoman said...

Subject: Good News **Date**: 21 November 2001
...Enron Gets Extension on $690 Mln Note Due Next Week (Update1)
...Enron Corp., whose shares had dropped 92 percent …

Figure 12. Representative emails extracted from the topic *enron's stock*.

The keywords that represent each topic for the Enron data set are clearly related, and are significantly different from the keywords of another topic. A clear trend in the topics in fig. 12 can be seen: the first topic is finance-related, the second is travel-related, and so forth.

The representative set of emails for the topic *enron's stock* represent the topic very well, and are far enough apart in time to represent the development of the performance of the company's stock. The representative emails provide the user with a sort of topic storyline, which allows for better understanding of what happened to Enron's stock throughout 2001. Extracting this representative set clearly helps the user to drill down into the topic beyond just the summary keywords.

7. DISCUSSION

In this paper we presented a system for reminiscing through personal email, which builds upon the existing Muse project. The underlying topics of an email archive are extracted using LDA, and then a representative set of emails is found using the factor graph model.

The system was trained on the corpus of Sarah Palin's emails and produced clear and useful results for both the topics learned and the representative email set for each topic. Furthermore, the system was evaluated against the Enron data set, with good results.

8. FUTURE WORK

8.1 More Preprocessing

The following preprocessing improvements will further reduce the noise of the data: spelling correction, named entity recognition, and semantic expansion.

8.2 Extending Email Extraction

The factor graph approach to extracting representative emails is powerful, but only uses a fraction of the available data. The current implementation of this algorithm scores subsets of emails on just their average topic probability and time spread. But the algorithm does not take into account the people tied to the emails or any of the other metadata. Using the communication graph could help find even more relevant emails. Another extension would be to use the topic probability distributions given by LDA. The current algorithm associates an email with its most probable topic, but in LDA emails are assigned a probability distribution over all topics. A similar factor graph algorithm could be used to find trends between topics.

9. REFERENCES

[1] http://mobisocial.stanford.edu/muse

[2] Hangal, Sudheendra, Monica S. Lam, and Jeffrey Heer. "Muse: Reviving memories using email archives." *Proceedings of the 24th annual ACM symposium on User interface software and technology*. ACM, 2011.

[3] http://lucene.apache.org/

[4] Ramos, Juan. "Using tf-idf to determine word relevance in document queries."*Proceedings of the First Instructional Conference on Machine Learning*. 2003.

[5] Arthur, David, and Sergei Vassilvitskii. "k-means++: The advantages of careful seeding." *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 2007.

[6] Dehak, Najim, et al. "Cosine similarity scoring without score normalization techniques." *Proc. Odyssey Speaker and Language Recognition Workshop*. 2010.

[7] http://commons.apache.org/proper/commons-math/

[8] Blei, David M. "Probabilistic topic models." *Communications of the ACM* 55.4 (2012): 77-84.

[9] Blei, David M., Andrew Y. Ng, and Michael I. Jordan. "Latent dirichlet allocation." *the Journal of machine Learning research* 3 (2003): 993-1022.

[10] http://blog.echen.me/2011/06/27/topic-modeling-the-sarah-palin-emails/

[11] Steyvers, Mark, and Tom Griffiths. "Probabilistic topic models." *Handbook of latent semantic analysis* 427.7 (2007): 424-440.

[12] http://mallet.cs.umass.edu/