Exploiting Conversation Features for Finding Topics in Emails

Shafiq Joty, Giuseppe Carenini, Gabriel Murray, and Raymond Ng
Department of Computer Science
University of British Columbia
Vancouver, B.C. Canada V6T1Z4
{rjoty,carenini,gabrielm,rng}@cs.ubc.ca

Abstract

Our ongoing research addresses the task of finding topics at the sentence level in email conversations. We first describe how the existing topic models can be applied to this problem. Then we demonstrate why the existing methods are inadequate for this task and what more we need to consider. With an experiment we further show that conversation structure in the form of fragment quotation graph can be helpful for finding topics. To this end, we propose a novel graph-theoretic framework to solve the problem. Crucial to our proposed approach is that it captures the rich conversation features and integrates the strengths of the supervised approach with the unsupervised technique.

1 Conversations and Topic Modeling

With the increasing growth of web technologies, in our daily lives, we experience with conversations in many different modalities (e.g., emails, blogs, forums). Effective processing of these can be of great strategic value. For example, *text summarization* can provide direct and quick access to the information content and can help corporate managers to make decisions.

Our definition of 'topic' is something about which the participants of a conversation discuss or argue. For example, an email thread about arranging a conference can have topics such as 'location and time', 'registration', 'food menu', 'workshops', etc. Multiple topics seem to occur naturally in social interactions, whether synchronous (e.g., chats, meetings) or asynchronous (e.g., emails, blogs) conversations. In multi-party chat [2] report an average of 2.75 discussions. In our current, still limited development set containing 5 email threads from the BC3 corpus (www.cs.ubc.ca/nest/lci/bc3.html), we found an average of 3.5 topics per thread.

In this research we deal with the *topic assignment* problem. This involves clustering the sentences of an email thread into a set of coherent topical clusters. It is often considered as a prerequisite for other higher-level conversation analysis (i.e., identifying dialog acts, adjacency pairs, and rhetorical relations) and the applications of the derived structure are broad, encompassing text summarization, information ordering, automatic question answering, information retrieval.

Topic modeling for emails is challenging due to the characteristics of emails, especially its asynchronous nature and the way people use it. We noticed the following facts about email conversations: topics in emails do not change in a sequential way (unlike newspaper articles or meetings), people tend to use informal and short sentences, different people have different styles of writing, and processing only headers does not help much as they are often misleading. Therefore, the existing methods which are successful in written monologue and dialog may not succeed when they are applied to find topics in email conversations. In the next section we describe how the existing methods can be applied to this problem and pinpoint their problems.

2 Existing Methods

To our knowledge there is no previous research that tries to find topics at the sentence level in emails, though the closely related task of topic segmentation (aka finding topic boundaries) in monologue and dialog has received extensive attention. Broadly, there are two different research avenues that have been tried to tackle this problem namely 1) Lexical chain based topic segmentation (LCSeg) [3] and 2) Probabilistic topic models (e.g., Latent Dirichlet Allocation (LDAs)) [4].

LDA specifies the following distribution over words within a document:

$P(w_i) = \sum_{j=1}^{T} P(w_i|z_i = j)P(z_i = j)$

where T is the number of topics. $P(w_i|z_i = j)$ is the probability of word w_i under topic j and $P(z_i = j)$ is the probability that j^{th} topic was sampled for the i^{th} word token. We refer the multinomial distributions $\phi^{(j)} = P(w|z_i = j)$ and $\theta^{(d)} = P(z)$ as topic-word distribution and document-topic distribution respectively. There is a Dirichlet (α) prior on θ and a Dirichlet (β) prior on ϕ. This framework can be directly applied to an email corpus by considering each email as a document. Using LDA we get $z = P(z_i = j|w_i)$ (i.e., topic assignments for words). By assuming the words in a sentence occur independently we can estimate the the topic assignments for sentences as follows:

$P(z_i = j|s_k) = \prod_{w_i \in s_k} P(z_i = j|w_i)$ where, s_k is the k^{th} sentence for which we can assign the topic by: $j^* = argmax_j P(z_i = j|s_k)$.

In order to apply **LCSeg** we arrange the emails in an email thread based on their temporal relation (i.e., arrival time) and apply the off-the-shelf tool LCSeg [3] to get the topic boundaries. Here note that this method considers the minimal conversation structure (i.e., temporal relation).

Both LDA and LCSeg rely on the Bag of Word assumption. We argue that these are inadequate for finding topics in emails especially when topics are closely related and distributional variations are subtle. To better identify the topics we need to consider the email specific features. We observe in our corpus that people often use quotations to answer to the same topic. Another important finding is that participants often use each other's name to make disentanglement easier. When people reply to multiple recipients they usually mention the name of the person being referred to. Another key feature in any discourse is 'topic shift cue words' like "now", "however", etc. that people often use to shift from one topic to another. We [1] previously showed how the conversation structure can be captured efficiently at the finer granularity level (i.e., fragment level) using the 'fragment quotation graph (FQG)' (see figure 1). Based on an analysis of the quotation embedded in emails, the graph provides a fine representation of the referential structure of a conversation. To capture the feature that people often quote to talk to the same topic, we are using the FQG. Next section describes how LCSeg can be applied to this structure.

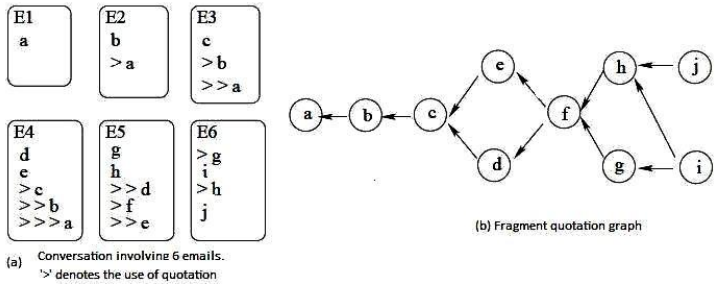

Figure 1: Fragment Quotation Graph for emails

3 LCSeg with Conversation Structure

To examine whether conversation structure does help or not we apply LCSeg on FQG. We extract different paths of the FQG, then we apply LCSeg on these paths separately. A fragment can be in multiple paths and this will cause a sentence to be in multiple segments found by LCSeg. To consolidate these segments we form a graph where the nodes V represent the sentences and the edge weights $w(u, v)$ represent the number of times sentence u and sentence v fall in the same segment. Then the consolidation problem becomes a graph partitioning problem which we solve using the *Normalized Cut* criterion.

4 Evaluation

In table 1 we show the preliminary results of different systems applied to our development set containing five email threads. The widely used metric of agreement in NLP κ-statistics is not applicable as two topic assignments may have different number of topics. We adopt the more appropriate metrics introduced in [2] namely 1-to-1 and loc_3. We can notice that LCSeg performs better than LDA, and LCseg applied to FQG performs better than LCSeg applied to the temporal ordering of emails. This clearly shows that conversation structure in the form of FQG helps finding topics in emails.

	LDA			LCSeg			LCSeg+FQG		
Scores	Max	**Mean**	Min	Max	**Mean**	Min	Max	**Mean**	Min
1-to-1	0.67	**0.49**	0.30	0.8	**0.54**	0.35	0.87	**0.60**	0.37
loc_3	0.74	**0.52**	0.33	0.83	**0.62**	0.42	0.87	**0.655**	0.36

Table 1: Performance of different systems on our development set

5 Proposed Solution (Current work)

Our proposed solution considers a rich feature set. We characterize each pair of sentences with: 1) Topic features (LSA, LDA), 2) Conversation features (*distance between the two sentences in the FQG*, speaker, *mention of names*, time, subject of the email, "reply to" relation in email), and 3) Lexical features (tf*idf, *Cue words*). Our approach is "Classify (supervised)" then "cut (unsupervised)". We use a binary classifier learned from a training set, to decide, given any two sentences, whether they should be in the same topic or not.

In the next step we form an undirected graph $G = (V, E)$, where the nodes V represent the sentences of an email thread and the edge weights $w(u, v)$ denote the class (i.e., same topic) membership probability for the two sentences u and v. Once we have the graph, the problem of finding topics can be formulated as a graph partitioning problem which we can solve using the normalized cut criterion.

References

[1] G. Carenini, R. T. Ng, and X. Zhou. Summarizing email conversations with clue words. In *Proceedings of the 16th international conference on World Wide Web*, pages 91–100. ACM New York, NY, USA, 2007.

[2] M. Elsner and E. Charniak. You talking to me? a corpus and algorithm for conversation disentanglement. In *Proceedings of ACL-08: HLT*, pages 834–842, Ohio, June 2008. ACL.

[3] M. Galley, K. Mckeown, E. F. Lussier, and H. Jing. Discourse segmentation of multi-party conversation. In *in 41st Annual Meeting of ACL*, pages 562–569, 2003.

[4] M. Steyvers and T. Griffiths. *Latent Semantic Analysis: A Road to Meaning*, chapter Probabilistic topic models. Laurence Erlbaum,