Exploiting Conversation Structure in Unsupervised Topic Segmentation for Emails

Shafiq Joty and Giuseppe Carenini and Gabriel Murray and Raymond T. Ng
{rjoty, carenini, gabrielm, rng}@cs.ubc.ca
Department of Computer Science
University of British Columbia
Vancouver, BC, V6T 1Z4, Canada

Abstract

This work concerns automatic topic segmentation of email conversations. We present a corpus of email threads manually annotated with topics, and evaluate annotator reliability. To our knowledge, this is the first such email corpus. We show how the existing topic segmentation models (i.e., Lexical Chain Segmenter (LCSeg) and Latent Dirichlet Allocation (LDA)) which are solely based on lexical information, can be applied to emails. By pointing out where these methods fail and what any desired model should consider, we propose two novel extensions of the models that not only use lexical information but also exploit finer level conversation structure in a principled way. Empirical evaluation shows that LCSeg is a better model than LDA for segmenting an email thread into topical clusters and incorporating conversation structure into these models improves the performance significantly.

1 Introduction

With the ever increasing popularity of emails and web technologies, it is very common for people to discuss issues, events, agendas or tasks by email. Effective processing of the email contents can be of great strategic value. In this paper, we study the problem of *topic segmentation for emails*, i.e., grouping the sentences of an email thread into a set of coherent topical clusters. Adapting the standard definition of topic (Galley et al., 2003) to conversations/emails, we consider a topic is something about which the participant(s) discuss or argue or express their opinions. For example, in the email thread shown in Figure 1, according to the majority of our annotators, participants discuss three topics (e.g., 'telecon cancellation', 'TAG document', and 'responding to I18N'). 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 (Elsner and Charniak, 2008) report an average of 2.75 discussions active at a time. In our email corpus, we found an average of 2.5 topics per thread.

Topic segmentation is often considered a prerequisite for other higher-level conversation analysis and applications of the extracted structure are broad, encompassing: summarization (Harabagiu and Lacatusu, 2005), information extraction and ordering (Allan, 2002), information retrieval (Dias et al., 2007), and intelligent user interfaces (Dredze et al., 2008). While extensive research has been conducted in topic segmentation for monologues (e.g., (Malioutov and Barzilay, 2006), (Choi et al., 2001)) and synchronous dialogs (e.g., (Galley et al., 2003), (Hsueh et al., 2006)), none has studied the problem of segmenting asynchronous multi-party conversations (e.g., email). Therefore, there is no reliable annotation scheme, no standard corpus, and no agreed-upon metrics available. Also, it is our key hypothesis that, because of its asynchronous nature, and the use of quotation (Crystal, 2001), topics in an email thread often do not change in a sequential way. As a result, we do not expect models which have proved successful in monologue or dialog to be as effective when they are applied to email conversations.

Our contributions in this paper aim to remedy

Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pages 388–398,
MIT, Massachusetts, USA, 9-11 October 2010. ©2010 Association for Computational Linguistics

these problems. First, we present an email corpus annotated with topics and evaluate annotator agreement. Second, we adopt a set of metrics to measure the local and global structural similarity between two annotations from the work on multi-party chat disentanglement (Elsner and Charniak, 2008). Third, we show how the two state-of-the-art topic segmentation methods (i.e., LCSeg and LDA) which are solely based on lexical information and make strong assumptions on the resulting topic models, can be effectively applied to emails, by having them to consider, in a principled way, a finer level structure of the underlying conversations. Experimental results show that both LCSeg and LDA benefit when they are extended to consider the conversational structure. When comparing the two methods, we found that LCSeg is better than LDA and this advantage is preserved when they are extended to incorporate conversational structure.

2 Related Work

Three research areas are directly related to our study: a) text segmentation models, b) probabilistic topic models, and c) extracting and representing the conversation structure of emails.

Topic segmentation has been extensively studied both for monologues and dialogs. (Malioutov and Barzilay, 2006) uses the minimum cut model to segment spoken lectures (i.e., monologue). They form a weighted undirected graph where the vertices represent sentences and the weighted links represent the similarity between sentences. Then the segmentation problem can be solved as a graph partitioning problem, where the assumption is that the sentences in a segment should be similar, while sentences in different segments should be dissimilar. They optimize the 'normalized cut' criterion to extract the segments. In general, the minimization of the normalized cut criterion is NP-complete. However, the linearity constraint on text segmentation for monologue allows them to find an exact solution in polynomial time. In our extension of LCSeg, we use a similar method to consolidate different segments; however, in our case the linearity constraint is absent. Therefore, we approximate the optimal solution by spectral clustering (Shi and Malik, 2000). Moving to the task of segmenting dialogs, (Galley et al., 2003) first proposed the lexical chain based unsupervised segmenter (LCSeg) and a supervised segmenter for segmenting meeting transcripts. Their supervised approach uses C4.5 and C4.5 rules binary classifiers with lexical and conversational features (e.g., cue phrase, overlap, speaker, silence, and lexical cohesion function). Their supervised approach performs significantly better than LCSeg. (Hsueh et al., 2006) follow the same approaches as (Galley et al., 2003) on both manual transcripts and ASR output of meetings. They perform segmentation at both coarse (topic) and fine (subtopic) levels. For the topic level, they achieve similar results as (Galley et al., 2003), with the supervised approach outperforming LCSeg. However, for the subtopic level, LCSeg performs significantly better than the supervised one. In our work, we show how LCSeg performs when applied to the temporal ordering of the emails in a thread. We also propose its extension to leverage the finer conversation structure of emails.

The probabilistic generative topic models, such as LDA and its variants (e.g., (Blei et al., 2003), (Steyvers and Griffiths, 2007)), have proven to be successful for topic segmentation in both monologue (e.g., (Chen et al., 2009)) and dialog (e.g., (Georgescul et al., 2008)). (Purver et al., 2006) uses a variant of LDA for the tasks of segmenting meeting transcripts and extracting the associated topic labels. However, their approach for segmentation does not perform better than LCSeg. In our work, we show how the general LDA performs when applied to email conversations and describe how it can be extended to exploit the conversation structure of emails.

Several approaches have been proposed to capture an email conversation . Email programs (e.g., Gmail, Yahoomail) group emails into threads using headers. However, our annotations show that topics change at a finer level of granularity than emails. (Carenini et al., 2007) present a method to capture an email conversation at the finer level by analyzing the embedded quotations in emails. A fragment quotation graph (FQG) is generated, which is shown to be beneficial for email summarization. In this paper, we show that topic segmentation models can also benefit significantly from this fine conversation structure of email threads.

3 Corpus and Evaluation Metrics

There are no publicly available email corpora annotated with topics. Therefore, the first step was to develop our own corpus. We have annotated the BC3 email corpus (Ulrich et al., 2008) with topics[1]. The BC3 corpus, previously annotated with sentence level speech acts, meta sentence, subjectivity, extractive and abstractive summaries, is one of a growing number of corpora being used for email research. The corpus contains 40 email threads from the W3C corpus[2]. It has 3222 sentences and an average of 5 emails per thread.

3.1 Topic Annotation

Topic segmentation in general is a nontrivial and subjective task (Hsueh et al., 2006). The conversation phenomenon called 'Schism' makes it even more challenging for conversations. In schism a new conversation takes birth from an existing one, not necessarily because of a topic shift but because some participants refocus their attention onto each other, and away from whoever held the floor in the parent conversation and the annotators can disagree on the birth of a new topic (Aoki et al., 2006). In the example email thread shown in Figure 1, a schism takes place when people discuss about 'responding to I18N'. All the annotators do not agree on the fact that the topic about 'responding to I18N' swerves from the one about 'TAG document'. The annotators can disagree on the number of topics (i.e., some are specific and some are general), and on the topic assignment of the sentences[3]. To properly design an effective annotation manual and procedure we performed a two-phase pilot study before carrying out the actual annotation. For the pilot study we picked five email threads randomly from the corpus. In the first phase of the pilot study we selected five university graduate students to do the annotation. We then revised our instruction manual based on their feedback and the source of disagreement found. In the second phase we tested with a university postdoc doing the annotation.

For the actual annotation we selected three computer science graduates who are also native speakers of English. They annotated 39 threads of the BC3 corpus[4]. On an average they took seven hours to annotate the whole dataset.

BC3 contains three human written abstract summaries for each email thread. With each email thread the annotators were also given an associated human written summary to give a brief overview of the corresponding conversation. The task of finding topics was carried out in two phases. In the first phase, the annotators read the conversation and the associated summary and list the topics discussed. They specify the topics by a short description (e.g., "meeting agenda", "location and schedule") which provides a high-level overview of the topic. The target number of topics and the topic labels were not given in advance and they were instructed to find as many topics as needed to convey the overall content structure of the conversation.

In the second phase the annotators identify the most appropriate topic for each sentence. However, if a sentence covers more than one topic, they were asked to label it with all the relevant topics according to their order of relevance. If they find any sentence that does not fit into any topic, they are told to label those as the predefined topic 'OFF-TOPIC'. Wherever appropriate they were also asked to make use of two other predefined topics: 'INTRO' and 'END'. INTRO (e.g., 'hi', 'hello') signifies the section (usually at the beginning) of an email that people use to begin their email. Likewise, END (e.g., 'Cheers', 'Best') signifies the section (usually at the end) that people use to end their email. The annotators carried out the task on paper. We created the hierarchical thread view ('reply to' relation) using 'TAB's (indentation) and each participant's name is printed in a different color as in Gmail.

Table 1 shows some basic statistics computed on the three annotations of the 39 email threads[5]. On

[1] The BC3 corpus had already been annotated for email summarization, speech act recognition and subjectivity detection. This new annotation with topics will be also made publicly available at http://www.cs.ubc.ca/labs/lci/bc3.html

[2] http://research.microsoft.com/en-us/um/people/nickcr/w3c-summary.html

[3] The annotators also disagree on the topic labels, however in this work we are not interested in finding the topic labels.

[4] The annotators in the pilot and in the actual study were different so we could reuse the threads used in pilot study. However, one thread on which the pilot annotators agree fully, was used as an example in the instruction manual. This gives 39 threads left for the actual study.

[5] We got 100% agreement on the two predefined topics 'IN-

average we have 26.3 sentences and 2.5 topics per thread. A topic contains an average of 12.6 sentences. The average number of topics active at a time is 1.4. The average entropy is 0.94 and corresponds (as described in detail in the next section) to the granularity of the annotation. These statistics (number of topics and topic density) indicate that the dataset is suitable for topic segmentation.

	Mean	Max	Min
Number of sentences	26.3	55	13
Number of topics	2.5	7	1
Avg. topic length	12.6	35	3
Avg. topic density	1.4	3.1	1
Entropy	0.94	2.7	0

Table 1: Corpus statistics of human annotations

Metrics	Mean	Max	Min
1-to-1	0.804	1	0.31
loc_k	0.831	1	0.43
m-to-1	0.949	1	0.61

Table 2: Annotator agreement in the scale of 0 to 1

3.2 Evaluation Metrics

In this section we describe the metrics used to compare different human annotations and system's output. As different annotations (or system's output) can group sentences in different number of clusters, metrics widely used in classification, such as the κ statistic, are not applicable. Again, our problem of topic segmentation for emails is not sequential in nature. Therefore, the standard metrics widely used in sequential topic segmentation for monologues and dialogs, such as P_k and $WindowDiff(WD)$, are also not applicable. We adopt the more appropriate metrics 1-to-1, loc_k and m-to-1, introduced recently by (Elsner and Charniak, 2008). The 1-to-1 metric measures the global similarity between two annotations. It pairs up the clusters from the two annotations in a way that maximizes (globally) the total overlap and then reports the percentage of overlap. loc_k measures the local agreement within a context of k sentences. To compute the loc_3 metric for the m-th sentence in the two annotations, we consider the previous 3 sentences: m-1, m-2 and m-3, and mark them as either 'same' or 'different' depending on their topic assignment. The loc_3 score between two annotations is the mean agreement on these 'same' or 'different' judgments, averaged over all sentences. We report the agreement found in 1-to-1 and loc_k in Table 2. In both of the metrics we get high agreement, though the local agreement (average of 83%) is little higher than the global agreement (average of 80%).

If we consider the topic of a randomly picked sentence as a random variable then its entropy measures the level of detail in an annotation. If the topics are evenly distributed then the uncertainty (i.e., entropy) is higher. It also increases with the increase of the number of topics. Therefore, it is a measure of how specific an annotator is and in our dataset it varies from 0 [6] to 2.7. To measure how much the annotators agree on the general structure we use the m-to-1 metric. It maps each of the source clusters to the single target cluster with which it gets the highest overlap, then computes the total percentage of overlap. This metric is asymmetrical and not a measure to be optimized[7], but it gives us some intuition about specificity (Elsner and Charniak, 2008). If one annotator divides a cluster into two clusters then, the m-to-1 metric from fine to coarse is 1. In our corpus by mapping from fine to coarse we get an m-to-1 average of 0.949.

4 Topic Segmentation Models

Developing automatic tools for segmenting an email thread is challenging. The example email thread in Figure 1 demonstrates why. We use different colors and fonts to represent sentences of different topics[8]. One can notice that email conversations are different from written monologues (e.g., newspaper) and dialogs (e.g., meeting, chat) in various ways. As a communication media Email is distributed (unlike face to face meeting) and asynchronous (unlike

TRO' and 'END'. In all our computation (i.e., statistics, agreement, system's input) we excluded the sentences marked as either 'INTRO' or 'END'

[6] 0 uncertainty happens when there is only one topic found

[7] hence we do not use it to compare our models.

[8] 2 of the 3 annotators agree on this segmentation. Green represents topic 1 ('telecon cancellation'), orange indicates topic 2 ('TAG document') and magenta represents topic 3 ('responding to I18N')

chat), meaning that different people from different locations can collaborate at different times. Therefore, topics in an email thread may not change in sequential way. In the example, we see that topic 1 (i.e., 'telecon cancellation') is revisited after some gaps.

The headers (i.e., subjects) do not convey much information and are often misleading. In the example thread, participants use the same subject (i.e., 20030220 telecon) but they talk about 'responding to I18N' and 'TAG document' instead of 'telecon cancellation'. Writing style varies among participants, and many people tend to use informal, short and ungrammatical sentences. These properties of email limit the application of techniques that have been successful in monologues and dialogues.

LDA and *LCSeg* are the two state-of-the-art models for topic segmentation of multi-party conversation (e.g., (Galley et al., 2003), (Hsueh et al., 2006), (Georgescul et al., 2008)). In this section, at first we describe how the existing models of topic segmentation can be applied to emails. We then point out where these methods fail and propose extensions of these basic models for email conversations.

4.1 Latent Dirichlet Allocation (LDA)

Our first model is the probabilistic LDA model (Steyvers and Griffiths, 2007). This model relies on the fundamental idea that documents are mixtures of topics, and a topic is a multinomial distribution over words. The generative topic model 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. (Blei et al., 2003) refined this basic model by placing a Dirichlet (α) prior on θ. (Griffiths and Steyvers, 2003) further refined it by placing a Dirichlet (β) prior on ϕ. The inference problem is to find ϕ and θ given a document set. Variational EM has been applied to estimate these two parameters directly. Instead of estimating ϕ and θ, one can also directly estimate the posterior distribution over $z = P(z_i = j|w_i)$ (topic assignments for words). One efficient estimation technique uses Gibbs sampling to estimate this distribution.

This framework can be directly applied to an email thread 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 topic assignments for sentences as follows:

$$P(z_i = j|s_k) = \prod_{w_i \in s_k} P(z_i = j|w_i)$$

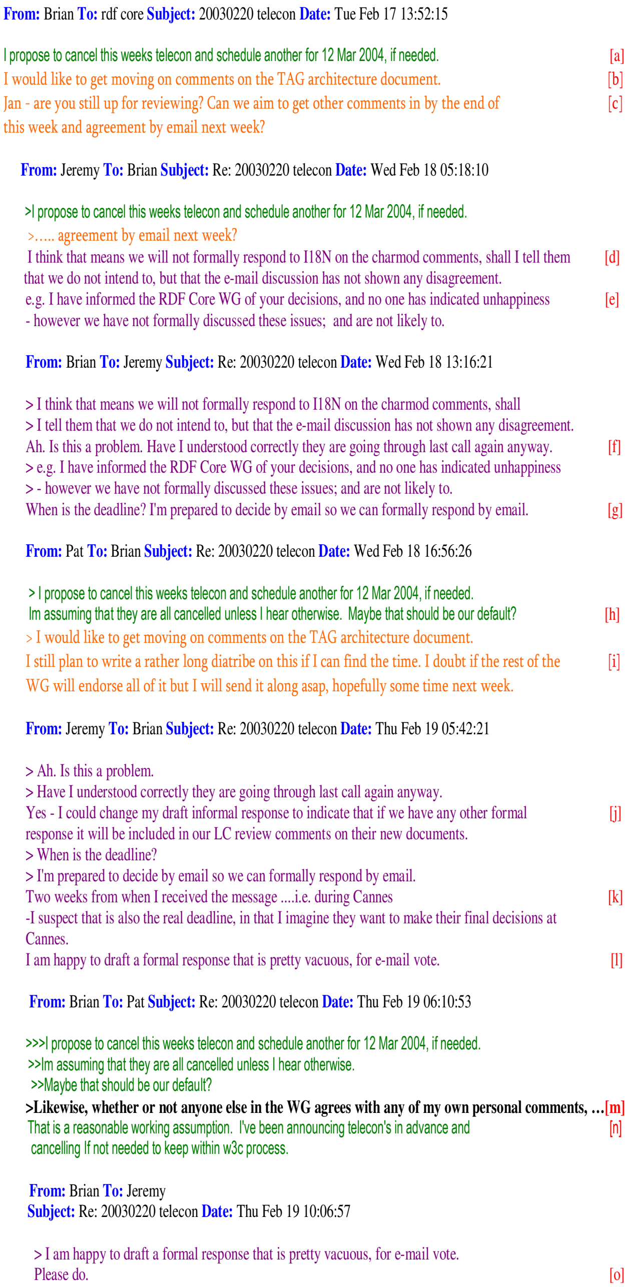
where, s_k is the k^{th} sentence for which we can assign the topic by: $j^* = argmax_j P(z_i = j|s_k)$.

4.2 Lexical Chain Segmenter (LCSeg)

Our second model is the lexical chain based segmenter LCSeg, (Galley et al., 2003). LCSeg assumes that topic shifts are likely to occur where strong term repetitions start and end[9]. LCSeg at first computes 'lexical chains' for each non-stop word based on word repetitions. It then ranks the chains according to two measures: 'number of words in the chain' and 'compactness of the chain'. The more compact (in terms of number of sentences) and the more populated chains get higher scores.

The algorithm then works with two adjacent analysis windows, each of a fixed size k which is empirically determined. For each sentence boundary, LCSeg computes the cosine similarity (or lexical cohesion function) at the transition between the two windows. Low similarity indicates low lexical cohesion, and a sharp change signals a high probability of an actual topic boundary. This method is similar to TextTiling (Hearst, 1997) except that the similarity is computed based on the scores of the 'lexical chains' instead of 'term counts'. In order to apply LCSeg on email threads we arrange the emails based on their temporal relation (i.e., arrival time) and apply the LCSeg algorithm to get the topic boundaries.

[9] One can also consider other lexical semantic relations (e.g., synonym, hypernym, hyponym) in lexical chaining. However, Galley et al., (Galley et al., 2003) uses only repetition relation as previous research results (e.g., (Choi, 2000)) account only for repetition.



From: Brian **To:** rdf core **Subject:** 20030220 telecon **Date:** Tue Feb 17 13:52:15

I propose to cancel this weeks telecon and schedule another for 12 Mar 2004, if needed. [a]
I would like to get moving on comments on the TAG architecture document. [b]
Jan - are you still up for reviewing? Can we aim to get other comments in by the end of [c]
this week and agreement by email next week?

From: Jeremy **To:** Brian **Subject:** Re: 20030220 telecon **Date:** Wed Feb 18 05:18:10

>I propose to cancel this weeks telecon and schedule another for 12 Mar 2004, if needed.
>..... agreement by email next week?
I think that means we will not formally respond to I18N on the charmod comments, shall I tell them [d]
that we do not intend to, but that the e-mail discussion has not shown any disagreement.
e.g. I have informed the RDF Core WG of your decisions, and no one has indicated unhappiness [e]
- however we have not formally discussed these issues; and are not likely to.

From: Brian **To:** Jeremy **Subject:** Re: 20030220 telecon **Date:** Wed Feb 18 13:16:21

> I think that means we will not formally respond to I18N on the charmod comments, shall
> I tell them that we do not intend to, but that the e-mail discussion has not shown any disagreement.
Ah. Is this a problem. Have I understood correctly they are going through last call again anyway. [f]
> e.g. I have informed the RDF Core WG of your decisions, and no one has indicated unhappiness
> - however we have not formally discussed these issues; and are not likely to.
When is the deadline? I'm prepared to decide by email so we can formally respond by email. [g]

From: Pat **To:** Brian **Subject:** Re: 20030220 telecon **Date:** Wed Feb 18 16:56:26

> I propose to cancel this weeks telecon and schedule another for 12 Mar 2004, if needed.
Im assuming that they are all cancelled unless I hear otherwise. Maybe that should be our default? [h]
> I would like to get moving on comments on the TAG architecture document.
I still plan to write a rather long diatribe on this if I can find the time. I doubt if the rest of the [i]
WG will endorse all of it but I will send it along asap, hopefully some time next week.

From: Jeremy **To:** Brian **Subject:** Re: 20030220 telecon **Date:** Thu Feb 19 05:42:21

> Ah. Is this a problem.
> Have I understood correctly they are going through last call again anyway.
Yes - I could change my draft informal response to indicate that if we have any other formal [j]
response it will be included in our LC review comments on their new documents.
> When is the deadline?
> I'm prepared to decide by email so we can formally respond by email.
Two weeks from when I received the message ....i.e. during Cannes [k]
-I suspect that is also the real deadline, in that I imagine they want to make their final decisions at
Cannes.
I am happy to draft a formal response that is pretty vacuous, for e-mail vote. [l]

From: Brian **To:** Pat **Subject:** Re: 20030220 telecon **Date:** Thu Feb 19 06:10:53

>>>I propose to cancel this weeks telecon and schedule another for 12 Mar 2004, if needed.
>>Im assuming that they are all cancelled unless I hear otherwise.
>>Maybe that should be our default?
>Likewise, whether or not anyone else in the WG agrees with any of my own personal comments, ...[m]
That is a reasonable working assumption. I've been announcing telecon's in advance and [n]
cancelling If not needed to keep within w3c process.

From: Brian **To:** Jeremy
Subject: Re: 20030220 telecon **Date:** Thu Feb 19 10:06:57

> I am happy to draft a formal response that is pretty vacuous, for e-mail vote.
Please do. [o]

Figure 1: Sample thread from the BC3 corpus. Each different color/font indicates a different topic. Right most column specifies the fragments (sec 4.4).

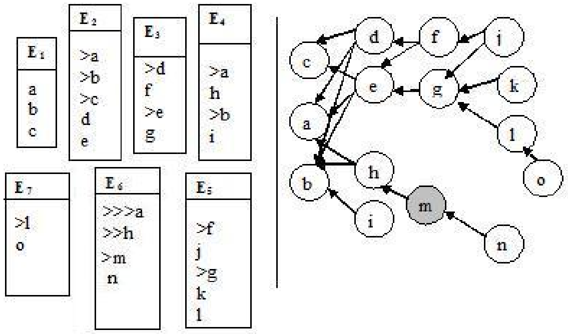


Figure 2: Fragment Quotation Graph for emails

4.3 Limitation of Existing Approaches

The main limitation of the two models discussed above is that they take the bag-of-words (BOW) assumption without considering the fact that an email thread is a multi-party, asynchronous conversation[10]. The only information relevant to LDA is term frequency. LCSeg considers both term frequency and how closely the terms occur in a document. These models do not consider the word order, syntax and semantics. However, several improvements of LDA over the BOW approach have been proposed. (Wallach, 2006) extends the model beyond BOW by considering n-gram sequences. (Griffiths et al., 2005) presents an extension of the topic model that is sensitive to word-order and automatically learns the syntactic as well as semantic factors that guide word choice. (Boyd-Graber and Blei, 2010) describes another extension to consider syntax of the text. As described earlier, one can also incorporate lexical semantics (i.e., synonym, hypernym, hyponym) into the LCSeg model. However, we argue that these models are still inadequate for finding topics in emails especially when topics are closely related (e.g., ‘extending the meeting’ and ‘scheduling the meeting’) and distributional variations are subtle. To better identify the topics in an email thread we need to consider the email specific conversation features (e.g., reply-to relation, usage of quotations). As can be seen in the example (Figure 1), people often use quotations to talk about the same topic. In fact in our corpus we found an average quotation usage of 6.44 per thread. Therefore,

[10] though in LCSeg we provide minimal conversation structure in the form of temporal relation between emails.

we need to leverage this useful information in a principled way to get the best out of our models. Specifically, we need to capture the conversation structure at the fragment (quotation) level and to incorporate this structure into our models.

In the next section, we describe how one can capture the conversation structure at the fragment level in the form of Fragment Quotation Graph (henceforth, FQG). In Section 4.5 and 4.6 respectively, we show how the LDA and LCSeg models can be extended so that they take this conversation structure into account for topic segmentation.

4.4 Extracting Conversation Structure

We demonstrate how to build a FQG through the example email thread involving 7 emails shown in Figure 1. For convenience we do not show the real content but abbreviate them as a sequence of fragments.

In the first pass by processing the whole thread we identify the new (i.e., quotation depth 0) and quoted (i.e., quotation depth > 0) fragments based on the usage of quotation ($>$) marks. For instance, email E_3 contains two new fragments (f, g), and two quoted fragments (d, e) of depth 1. E_2 contains abc and de. Then in the second step, we compare the fragments with each other and based on the overlap we find the distinct fragments. If necessary we split the fragments in this step. For example, de in E_2 is divided into d and e distinct fragments when compared with the fragments of E_3. This process gives 15 distinct fragments which constitute the vertices of the FQG. In the third step, we compute the edges, which represent referential relations between fragments. For simplicity we assume that any new fragment is a potential reply to its neighboring quoted fragments. For example, for the fragments of E_4 we create two edges from h ((h,a),(h,b)) and one edge from i ((i,b)). We then remove the redundant edges. In E_6 we found the edges (n,h), (n,a) and (n,m). As (h,a) is already there we exclude (n,a). The FQG with all the redundant edges removed is shown at the right in Figure 2. If an email does not contain quotes then the fragments of that email are connected to the fragments of the source email to which it replies.

The advantage of the FQG is that it captures the conversation at finer granularity level in contrast to the structure found by the 'reply-to' relation at the email level, which would be merely a sequence from E_1 to E_7 in this example. Another advantage of this structure is that it allows us to find the 'hidden fragments'. Hidden fragments are quoted fragments (shaded fragment m in fig 2 which corresponds to the fragment made bold in fig 1), whose original email is missing in the user's inbox. (Carenini et al., 2007) study this phenomenon and its impact on email summarization in detail.

4.5 Regularizing LDA with FQG

The main advantage of the probabilistic (Bayesian) models is that they allow us to incorporate multiple knowledge sources in a coherent way in the form of priors (or regularizer). We want to regularize LDA in a way that will force two sentences in the same or adjacent fragments to fall in the same topical cluster. The first step forwards this aim is to regularize the topic-word distribution with a word network such that two connected words get similar topic distributions. Then we can easily extend it to fragments. In this section, at first we describe how one can regularize the LDA model with a word network, then we extend this by regularizing LDA with FQG.

Assume we are given a word network as an undirected graph with nodes (V) representing the words and the edges (E) representing the links between words. We want to regularize the LDA model such that two connected words u, v have similar topic-word distributions (i.e., $\phi_j^{(u)} \approx \phi_j^{(v)}$ for $j = 1 \ldots T$). Note that the standard conjugate Dirichlet prior on ϕ is limited in that all words share a common variance parameter, and are mutually independent except normalization constraint (Minka, 1999). Therefore it does not allow us to encode this knowledge. Very recently, (Andrzejewski et al., 2009) shows how to encode 'must-link' and 'cannot-link' (between words) into the LDA model by using a Dirichlet Forest prior. We reimplemented this model; however, we only use its capability of encoding 'must-links'. Therefore, we just illustrate how to encode 'must-links' here. Interested readers can see (Andrzejewski et al., 2009) for the method of encoding 'cannot-links'.

Must links such as $(a, b), (b, c)$, or (x, y) in Figure 3(A) can be encoded into the LDA model by using a Dirichlet Tree (henceforth, DT) prior. Like the traditional Dirichlet, DT is also a conjugate to the multinomial but under a different parameterization.

Instead of representing a multinomial sample as the outcome of a K-sided die, in this representation we represent a sample as the outcome of a finite stochastic process. The probability of a leaf is the product of branch probabilities leading to that leaf. The words constitute the leaves of the tree.

DT distribution is the distribution over leaf probabilities. Let ω^n be the DT edge weight leading into node n, $C(n)$ be the children of node n, L be the leaves of the tree, I the internal nodes, and $L(n)$ be the leaves in the subtree under n. We generate a sample ϕ^k from Dirichlet_Tree(Ω) by drawing a multinomial at each internal node $i \in I$ from Dirichlet($\omega^{C(i)}$) (i.e., the edge weights from i to its children). The probability density function of DT($\phi^k|\Omega$) is given by:

$$DT(\phi^k|\Omega) \approx \left(\prod_{l \in L} \phi_l^{k^{\omega^l - 1}}\right) \left(\prod_{i \in I} \left(\sum_{j \in L(i)} \phi_j^k\right)^{\Delta(i)}\right)$$

Here $\Delta(i) = \omega^i - \sum_{j \in C(i)} \omega^j$ (i.e., the difference between the in-degree and out-degree of internal node i. Note that if $\Delta(i) = 0$ for all $i \in I$, then the DT reduces to the typical Dirichlet distribution.

Suppose we have the following (Figure 3(A)) word network. The network can be decomposed into a collection of chains (e.g., (a,b,c), (p), and (x,y)). For each chain having number of elements more than one (e.g., (a,b,c), (x,y)), we have a subtree (see Figure 3(B)) in the DT with one internal node (blank in figure) and the words as leaves. We assign $\lambda\beta$ as the weights of these edges where λ is the regularization strength and β is the hyperparameter of the symmetric Dirichlet prior on ϕ. The root node of the Dirichlet tree then connects to the internal node i with weight $|L(i)|\beta$. The other nodes (words) which form single element chains (e.g, (p)) are connected to the root directly with weight β. Notice that when $\lambda = 1$ (i.e., no regularization), $\Delta(i) = 0$ and our model reduces to the original LDA. By tuning λ we control the strength of regularization.

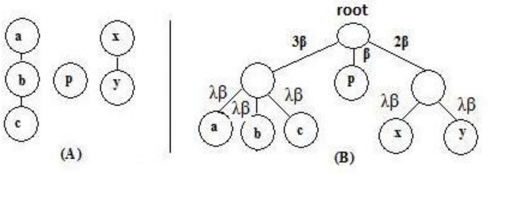


Figure 3: Incorporating word network into DT

To regularize LDA with FQG, we form the word network where a word is connected to the words in the same or adjacent fragments. Specifically, if word $w_i \in frag_x$ and word $w_j \in frag_y$ ($w_i \neq w_j$), we create a link (w_i, w_j) if $x = y$ or $(x, y) \in E$, where E is the set of edges of the FQG. Implicitly by doing this we want two sentences in the same or adjacent fragments to have similar topic distributions, and fall in the same topical cluster.

4.6 LCSeg with FQG

If we examine the FQG carefully, different paths (considering the fragments of the first email as root nodes) can be interpreted as subconversations. As we walk down a path topic shifts may occur along the pathway. We incorporate FQG into the LCSeg model in three steps. First, we extract the paths of a FQG. We then apply LCSeg algorithm on each of the extracted paths separately. This process gives the segmentation decisions along the paths of the FQG. Note that a fragment can be in multiple paths (e.g., f, g, in Figure 2) which will cause its sentences to be in multiple segments found by LCSeg. Therefore, as a final step we need a consolidation method. Our intuition is that sentences in a consolidated segment should fall in same segments more often when we apply LCSeg in step 2. To consolidate the segments found, we form a weighted undirected graph where the vertices V represent the sentences and the edge weights $w(u, v)$ represent the number of times sentence u and v fall in the same segment. The consolidation problem can be formulated as a *N-mincut* graph partitioning problem where we try to optimize the *Normalized Cut* criterion:

$$Ncut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(B, A)}{assoc(B, V)}$$

where $cut(A, B) = \Sigma_{u \in A, v \in B} w(u, v)$ and $assoc(A, V) = \Sigma_{u \in A, t \in V} w(u, t)$ is the total connection from nodes in partition A to all nodes in the graph and $assoc(B, V)$ is similarly defined. However, solving this problem turns out to be NP-hard. Hence, we approximate the solution following (Shi and Malik, 2000) which has been successfully applied to image segmentation in computer vision.

This approach makes a difference only if FGQ contains more than one path. In fact in our corpus we found an average paths of 7.12 per thread.

Avg. Topic	LDA	LDA +FQG	LCSeg	LCSeg +FQG	Speaker	Block 5
Number	2.10	1.90	2.2	2.41	4.87	5.69
Length	13.3	15.50	13.12	12.41	5.79	4.60
Density	1.83	1.60	1.01	1.39	1.37	1.00
Entropy	0.98	0.75	0.81	0.93	1.88	2.39

Table 3: Corpus statistics of different system's annotation

5 Experiments

We ran our four systems *LDA, LDA+FQG, LCSeg, and LCSeg+FQG* on the dataset[11]. The statistics of these four annotations and two best performing baselines (i.e., 'Speaker' and 'Block 5' as described below) are shown in Table 3. For brevity we just mention the average measures. Comparing with Table 1, we see that these fall within the bounds of the human annotations.

We compare our results in Table 4, where we also provide the results of some simple baseline systems. We evaluated the following baselines and report the best two in Table 4.

All different: Each sentence is a separate topic.

All same: The whole thread is a single topic.

Speaker: The sentences from each participant constitute a separate topic.

Blocks of $k (= 5, 10, 15)$**:** Each consecutive group of k sentences is a topic.

Most of these baselines perform rather poorly. **All different** is the worst baseline with mean 1-to-1 score of 0.10 (max: 0.33, min: 0.03) and mean loc_3 score of 0.245 (max: 0.67, min: 0). **Block 10** has mean 1-to-1 score of 0.35 (max: 0.71, min: 0.13) and mean loc_3 score of 0.584 (max: 0.76, min: 0.31). **Block 15** has mean 1-to-1 score of 0.32 (max: 0.77, min: 0.16) and mean loc_3 score of 0.56 (max: 0.82, min: 0.38). **All same** is optimal for threads containing only one topic, but its performance rapidly degrades as the number of topics in a thread increases. It has mean 1-to-1 score of 0.28 (max: 1[12], min: 0.11) and mean loc_3 score of 0.54 (max: 1, min: 0.34).

As shown in Table 4, **Speaker** and **Blocks of** 5 are two strong baselines especially for the loc_3. In general, our systems perform better than the baselines, but worse than the gold standard. Of all the systems, the basic LDA model performs very disappointingly. In the local agreement it even fails to beat the baselines. A likely explanation is that the independence assumption made by LDA when computing the distribution over topics for a sentence from the distribution over topics for the words causes sentences in a local context to be excessively distributed over topics. Another possible explanation for LDA's disappointing performance is the limited amount of data available for training. In our corpus, the average number of sentences per thread is 26.3 (see table 1) which might not be sufficient for the LDA models.

If we compare the performance of the regularized LDA (in the table LDA+FQG) with the basic LDA we get a significant (p=0.0002 (1-to-1), p=9.8e-07 (loc_3)) improvement in both of the measures [13]. This supports our claim that sentences connected by referential relations in the FQG usually refer to the same topic. The regularization also prevents the local context from being overly distributed over topics.

A comparison of the basic LCSeg with the basic LDA reveals that LCSeg is a better model for email topic segmentation (p=0.00017 (1-to-1), p<2.2e-16 (loc_3)). One possible reason is that LCSeg extracts the topics keeping the local context intact. Another reason could be the term weighting scheme employed by LCSeg. Unlike LDA, which considers only 'repetition', LCSeg also considers how tightly the 'repetition' happens. When we incorporate the conversation structure (i.e., FQG) into LCSeg (in the table LCSeg+FQG), we get a significant improvement in the 1-to-1 measure over the basic LCSeg (p=0.0014). Though the local context (i.e., loc_3) suf-

[11] For a fair comparison of the systems we set the same topic number per thread for all of them. If at least two of the annotators agree on the topic number we set that number, otherwise we set the floor value of the average topic number. λ is set to 20 in LDA+FQG.

[12] The maximum value of 1 is due to the fact that for some threads some annotators found only one topic

[13] Tests of significance were done by paired t-test with df=116

	Baselines		Systems				Human
Scores	Speaker	Block 5	LDA	**LDA+FQG**	LCSeg	**LCSeg+FQG**	
Mean 1-to-1	0.52	0.38	0.57	**0.62**	0.62	**0.68**	0.80
Max 1-to-1	0.94	0.77	1.00	1.00	1.00	1.00	1.00
Min 1-to-1	0.23	0.14	0.24	0.24	0.33	0.33	0.31
Mean loc_3	0.64	0.57	0.54	**0.61**	**0.72**	**0.71**	0.83
Max loc_3	0.97	0.73	1.00	1.00	1.00	1.00	1.00
Min loc_3	0.27	0.42	0.38	0.38	0.40	0.40	0.43

Table 4: Comparison of Human, System and best Baseline annotations

fers a bit, the decrease in performance is minimal and it is not significant. The fact that LCSeg is a better model than LDA is also preserved when we incorporate FQG into them (p=2.140e-05 (1-to-1), p=1.3e-09 (loc_3)). Overall, LCSeg+FQG is the best model for this data.

6 Future Work

There are some other important features that our models do not consider. The 'Speaker' feature is a key source of information. A participant usually contributes to the same topic. The best baseline 'Speaker' in Table 4 also favours this claim. Another possibly critical feature is the 'mention of names'. In multi-party discussion people usually mention each other's name for the purpose of disentanglement (Elsner and Charniak, 2008). In our corpus we found 175 instances where a participant mentions other participant's name. In addition to these, 'Subject of the email', 'topic-shift cue words' can also be beneficial for a model. As a next step for this research, we will investigate how to exploit these features in our methods.

We are also interested in the near future to transfer our approach to other similar domains by hierarchical Bayesian multi-task learning and other domain adaptation methods. We plan to work on both synchronous (e.g., chats, meetings) and asynchronous (e.g., blogs) domains.

7 Conclusion

In this paper we presented an email corpus annotated for topic segmentation. We extended LDA and LCSeg models by incorporating the fragment quotation graph, a fine-grain model of the conversation, which is based on the analysis of quotations. Empirical evaluation shows that the fragment quotation graph helps both these models to perform significantly better than their basic versions, with LCSeg+FQG being the best performer.

Acknowledgments

We are grateful to the 6 pilot annotators, 3 test annotators and to the 3 anonymous reviewers for their helpful comments. This work was supported in part by NSERC PGS award, NSERC BIN project, NSERC discovery grant and Institute for Computing, Information and Cognitive Systems (ICICS) at UBC.

References

James Allan, 2002. *Topic detection and tracking: event-based information organization*, pages 1–16. Kluwer Academic Publishers, Norwell, MA, USA.

David Andrzejewski, Xiaojin Zhu, and Mark Craven. 2009. Incorporating domain knowledge into topic modeling via dirichlet forest priors. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML'09)*, pages 25–32, New York, NY, USA. ACM.

Paul M. Aoki, Margaret H. Szymanski, Luke D. Plurkowski, James D. Thornton, Allison Woodruff, and Weilie Yi. 2006. Where's the "party" in "multi-party"?: analyzing the structure of small-group sociable talk. In *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work (CSCW '06)*, pages 393–402, New York, NY, USA. ACM.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *JMLR*, 3:993–1022.

Jordan L. Boyd-Graber and David M. Blei. 2010. Syntactic topic models. *CoRR*, abs/1002.4665.

G. Carenini, R. T. Ng, and X. Zhou. 2007. 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.

Harr Chen, S. R. K. Branavan, Regina Barzilay, and David R. Karger. 2009. Global models of document structure using latent permutations. In *NAACL'09*, pages 371–379, Morristown, NJ, USA. ACL.

Freddy Y. Y. Choi, Peter Wiemer-Hastings, and Johanna Moore. 2001. Latent semantic analysis for text segmentation. In *In Proceedings of EMNLP*, pages 109–117, Pittsburgh, PA USA.

Freddy Y. Y. Choi. 2000. Advances in domain independent linear text segmentation. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, pages 26–33, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

David Crystal, 2001. *Language and the Internet*. Cambridge University Press.

Gaël Dias, Elsa Alves, and José Gabriel Pereira Lopes. 2007. Topic segmentation algorithms for text summarization and passage retrieval: an exhaustive evaluation. In *AAAI'07: Proceedings of the 22nd national conference on Artificial intelligence*, pages 1334–1339. AAAI Press.

Mark Dredze, Hanna M. Wallach, Danny Puller, and Fernando Pereira. 2008. Generating summary keywords for emails using topics. In *IUI '08*, pages 199–206, New York, NY, USA. ACM.

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

Michel Galley, Kathleen McKeown, Eric Fosler-Lussier, and Hongyan Jing. 2003. Discourse segmentation of multi-party conversation. In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 562–569, Morristown, NJ, USA. Association for Computational Linguistics.

M. Georgescul, A. Clark, and S. Armstrong. 2008. A comparative study of mixture models for automatic topic segmentation of multiparty dialogues. In *ACL-08:HLT*, pages 925–930, Ohio, June. ACL.

Thomas L. Griffiths and Mark Steyvers. 2003. Prediction and semantic association. In *Advances in Neural Information Processing Systems*. MIT Press.

Thomas L. Griffiths, Mark Steyvers, David M. Blei, and Joshua B. Tenenbaum. 2005. Integrating topics and syntax. In *In Advances in Neural Information Processing Systems*, pages 537–544. MIT Press.

Sanda Harabagiu and Finley Lacatusu. 2005. Topic themes for multi-document summarization. In *SIGIR '05:*, pages 202–209, New York, NY, USA. ACM.

Marti A. Hearst. 1997. Texttiling: segmenting text into multi-paragraph subtopic passages. *Comput. Linguist.*, 23(1):33–64, March.

Pei Hsueh, Johanna D. Moore, and Steve Renals. 2006. Automatic segmentation of multiparty dialogue. In *Proceedings of the European Chapter of the Association for Computational Linguistics (EACL)*, Trento, Italy. ACL.

Igor Malioutov and Regina Barzilay. 2006. Minimum cut model for spoken lecture segmentation. In *Proceedings of the ACL'06*, pages 25–32, Sydney, Australia, July. ACL.

T. Minka. 1999. The dirichlet-tree distribution. Technical report, Justsystem Pittsburgh Research Center.

Matthew Purver, Konrad P. Körding, Thomas L. Griffiths, and Joshua B. Tenenbaum. 2006. Unsupervised topic modelling for multi-party spoken discourse. In *Proceedings of the ACL'06*, pages 17–24, Sydney, Australia. ACL.

Jianbo Shi and Jitendra Malik. 2000. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905.

M. Steyvers and T. Griffiths, 2007. *Latent Semantic Analysis: A Road to Meaning*, chapter Probabilistic topic models. Laurence Erlbaum.

J. Ulrich, G. Murray, and G. Carenini. 2008. A publicly available annotated corpus for supervised email summarization. In *EMAIL-2008 Workshop*, pages 428–435. AAAI.

Hanna M. Wallach. 2006. Topic modeling: beyond bag-of-words. In *ICML '06*, pages 977–984, NY, USA.