Supervised Topic Segmentation of Email Conversations

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, Canada

Abstract

We propose a graph-theoretic supervised topic segmentation model for email conversations which combines (i) lexical knowledge, (ii) conversational features, and (iii) topic features. We compare our results with the existing unsupervised models (i.e., LCSeg and LDA), and with their two extensions for email conversations (i.e., LCSeg+FQG and LDA+FQG) that not only use lexical information but also exploit finer conversation structure. Empirical evaluation shows that our supervised model is the best performer and achieves highest accuracy by combining the three different knowledge sources, where knowledge about the conversation has proved to be the most important indicator for segmenting emails.

Introduction and Related Work

We study the problem of *topic segmentation*, i.e., grouping the sentences of an email conversation into a set of coherent topical clusters. Topic segmentation is often considered a prerequisite for other higher-level conversation analysis and applications of the extracted structure are broad, including summarization (Harabagiu and Lacatusu 2005) and information extraction (Allan 2002). While extensive research has been conducted in topic segmentation for monologs (e.g., (Malioutov and Barzilay 2006)) and synchronous dialogs (e.g., (Galley et al. 2003)) segmenting asynchronous conversations (e.g., email) has not received much attention. It is our hypothesis that, because of its asynchronous nature and the use of quotation (Crystal 2001), topics in email conversations often do not change in a sequential way. That is, if you look at the temporal order, the discussion of a topic may appear to intersect with the discussion of other topics. As a result, we do not expect models which have proved successful in monolog or synchronous dialog to be as effective when they are directly applied to emails.

Recently, we (Joty et al. 2010) have shown how existing *unsupervised* topic segmentation models, i.e., LDA (Blei et al. (2003)) and LCSeg (Galley et al. 2003), which are solely based on lexical information, can be effectively applied to email by having them consider a finer level conversation structure (i.e., the Fragment Quotation Graph (FQG) (Carenini, Ng, and Zhou 2007)), generating two novel models **LDA+FQG** and **LCSeg+FQG**, respectively. However, we argue that these models are still limited in terms of accuracy as they ignore other important lexical and conversational features. The contribution of this paper is the development of a novel *supervised* approach that can effectively integrate conversation structure with these additional features to achieve better segmentation accuracy.

Copyright © 2011, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Graph-based Supervised Topic Segmentation

Although the unsupervised methods proposed in our previous work (Joty et al. 2010) have the key advantage of not requiring any labeled data, they can be limited in their ability to learn domain-specific knowledge from a possibly large and diverse set of features. We hypothesize that an even more accurate topic segmenter can be built with a supervised approach that can exploit additional features beyond the conversation structure. Features like 'speaker', 'recipient', 'subject', 'cue words' are arguably useful for segmenting email conversations. We also hypothesize that an effective model needs to combine similarity functions and segmentation decisions of other existing well performing models such as LDA, LCSeg and LSA. Another crucial limitation of the LCSeg+FQG and LDA+FQG models described in (Joty et al. 2010), is that these methods can successfully model the lexical cohesion between sentences in nearby fragments, but tend to fail on distant sentences in the FQG.

On the other hand, the supervised framework serves as a viable option to incorporate a rich feature set, but relies on labeled data. The amount of data required in supervised models to achieve an acceptable performance is always an important factor to consider for choosing supervised vs. unsupervised framework. Our supervised model aims to tackle this issue and remedy the problems of the unsupervised models: firstly, by using a binary classifier to combine all the important features, similarity functions and decisions of other models in a way that require very limited amount of labeled data and, secondly, by forming a complete graph to consider all pair inter-sentence (not fragment) relations.

The supervised model is built on the graph-theoretic framework which has been used in many NLP tasks, including chat disentanglement (Elsner and Charniak 2010). In our case, this method works in three steps. First, a binary classifier which is trained on a training dataset, marks each pair of

sentences of a given (i.e., test) conversation as 'same' or 'different' topics. Second, we form a weighted undirected graph $G = (V, E)$ for the given conversation, where the nodes V represent the sentences and the edge-weights $w(u, v)$ denote the probability (given by the binary classifier) of the two sentences u and v, to be in the 'same' class. Third, the segmentation task is formulated as a graph partitioning problem.

Sentence Pair Classification

In this section, we describe the binary classifier and features used to mark each pair of sentences of an email conversation as 'same' or 'different' topics. Note that as the classifier is defined on the sentence pairs of a conversation, a training conversation containing n sentences gives 1+2+. . .+(n-1)=$\frac{n(n-1)}{2} = O(n^2)$ training examples. Therefore, a training dataset containing m conversations gives $\sum_{i=1}^{m} \frac{n_i(n_i-1)}{2}$ training examples where n_i is the number of sentences in the i^{th} conversation. This quadratic expansion of training examples enables the classifier to achieve its best classification accuracy with very limited amount of labeled data.

Comparison of Classifiers The classifier's accuracy in deciding whether a pair of sentences x and y is in the 'same' or 'different' topics is crucial for the model's performance. By pairing up the sentences of each email conversation in our human annotated corpus (Joty et al. 2010), we got a total of $14,528$ data points of which 58.83% are in the 'same' class (i.e., 'same' is the majority class). Class labels are produced by taking the maximum vote of the three annotators. To select the best classifier, we experimented with a number of classifiers with the full feature set (see table 2). Table 1 shows the performance of the classifiers averaged over *leave one out* procedure (i.e., for a corpus containing m email conversations, train on $m-1$ and test on the rest).

Classifier	Regularizer	Train error	Test error
KNN	-	47.7%	46.7%
SVM (lin)	-	33.2%	32.6%
SVM (rbf)	-	26.4%	34.3%
LR	l_2	**30.6%**	**30.9%**
LR	l_1	32.1%	33.3%
RMLR (rbf)	l_2	10.8%	38.9%

Table 1: Performance of the classifiers using full feature set. Regularizer strength was learned by 10 fold cross validation.

We see that the non-parametric K-Nearest Neighbor (KNN) performs poorly. **Logistic Regression (LR)** with l_2 regularizer performs best in our dataset. Support Vector Machines (SVMs) with 'linear' and 'rbf' kernels perform reasonably well but do not match LR. Ridged Multinomial Logistic Regression (RMLR), kernelized LR with l_2 regularizer, extremely overfits the data. We opted for parametric LR classifier with l_2 regularizer mainly for two reasons: (i) Its accuracy, as shown in Table 1, and (ii) LR with L-BFGS (limited memory BFGS) fitting algorithm (which we use) is both time efficient (quadratic convergence rate; fastest among the listed models) and space efficient ($O(mD)$, where D := number of features).

Features Used Selecting the right set of features is crucial for the classifier's (and the segmenter's) performance. Table 2 summarizes the full feature set and the mean testset accuracy achieved with different subsets of features.

Lexical:	**Acc: 59.6 Pre: 59.7 Rec: 99.8**
$TF.IDF_1$	TF.IDF similarity (k=1).
$TF.IDF_2$	TF.IDF similarity (k=2).
Cue Words	Either one contains a cue word.
QA	x asks a question explicitly using '?' and y contains any of (yes, yeah, okay, ok, no, nope).
Greet	Either one has a greeting word (hi, hello, thank, thanks, thx, tnx.)
Topic:	**Acc: 65.2 Pre: 64.4 Rec: 79.6**
LSA_1	LSA function for x & y (k=1).
LSA_2	LSA function for x & y (k=2).
LDA	LDA decision on x & y.
LCSeg	LCSeg decision on x & y.
LexCoh	Lexical cohesion function of x & y.
Conv.:	**Acc: 65.3 Pre: 66.7 Rec: 85.1**
Gap	The gap between y & x in number of sentence(s).
Speaker	x & y have the same sender.
FQG_1	Distance between x & y in FQG in terms of fragment id (i.e., $\|frag\_id(y) - frag\_id(x)\|$)
FQG_2	Distance between x & y in FQG in terms of number of edges.
FQG_3	Distance between x & y in FQG in number of edges but this time considering it as undirected graph.
Reply-to	x & y are in the same email or one is a reply to the other.
Name	x mentions y or vice versa.
All:	**Acc: 69.1 Pre: 68.4 Rec: 81.5**

Table 2: Features with average performance.

Lexical features encode similarity between two sentences x and y based on their raw contents. Term frequency-based similarity is a widely used feature in previous work (e.g., TextTiling (Hearst 1997)). In order to compute this we consider two analysis windows in a similar fashion to TextTiling. Let X be the window including sentence x and the $k-1$ preceding sentences, and Y be the window including sentence y and the $k-1$ following sentences. We measure the similarity between the two windows by representing them as vectors of **TF.IDF** values of the words and computing the cosine of the angle in between them. Another important task specific feature that proved to be useful in previous research (e.g., (Galley et al. 2003)) is **cue words** that sign the presence of a topical boundary (e.g., 'coming up', 'joining us' in news). As our work concerns conversations (not monolog), we adopt the cue word list derived automatically by (Galley et al. 2003) in the meeting corpus. We use two other lexical features, i.e., Question Answer (**QA**) pairs and **greeting** words with the assumption that if y answers or greets x then it is likely that they are in the same topic.

Topic features are complex and encode information from the existing segmentation models. Choi et al. (2001) used Latent Semantic Analysis (**LSA**) to measure the similarity between two sentences and showed that LSA-based similar-

ity is more accurate than the direct TF.IDF-based similarity since it surmounts the problems of synonymy (e.g., car, auto) and polysemy (e.g., money bank and river bank). To compute LSA, we first form a Word-Document matrix W, where $W_{i,j}$:= *frequency of word i in comment j*×*IDF score of word i*. We perform truncated Singular Value Decomposition (SVD) of W: $W \approx U_k \Sigma_k V_k^T$ and represent each word (i) as k dimensional vector ($\vec{\Lambda}_i^k$). Each sentence s is then represented by the weighted sum of the k-dimensional vectors. Formally, the LSA vector representation for sentence s is $\vec{\Lambda}_s$=$\sum_{i \in s} TF_i \times \vec{\Lambda}_i^k$. In our study, k was set to $\frac{1}{4}\times$number of messages. To compute the LSA-based similarity between sentences x and y, we represent the corresponding windows (i.e., X and Y) as the LSA vectors and compute the cosine similarity as described before. The decisions of **LDA** and **LCSeg** models are also encoded as topic features.

LCSeg computes a lexical cohesion (**LexCoh**) function between two *consecutive* windows based on the scores of the chains that overlap with the windows. (Galley et al. 2003) shows a significant improvement when this function is used as a feature in the supervised sequential segmenter for meetings. However, as our problem is not sequential, we need to compute this function for *any* two given windows X and Y (not necessary consecutive). In order to do that at first we extract all the lexical chains with their scores and spans (i.e., beginning and end sentence numbers) for a conversation. The lexical cohesion function is then computed with:

$$LexCoh(X,Y) = cos(X,Y) = \frac{\sum_{i=1}^{N} w_{i,X} \times w_{i,Y}}{\sqrt{\sum_{i=1}^{N} w_{i,X}^2 \times \sum_{i=1}^{N} w_{i,Y}^2}}$$

where N is the number of chains and

$$w_{i,\Omega} = \begin{cases} rank(C_i) & \text{if chain } C_i \text{ overlaps } \Omega \in \{X,Y\} \\ 0 & \text{otherwise} \end{cases}$$

Conversation features encode conversational properties of an email conversation. 'Time gap' and 'speaker' have been proved to be important for segmenting meetings ((Galley et al. 2003)). We encode similar information in emails by counting the number of sentences between x and y as the **'gap'**, and their senders as the **'speakers'**. The strongest baseline 'Speaker' in the "Experiment" section has proved its effectiveness in emails. Our work (Joty et al. 2010) suggest that fine conversation structure in the form of FQG can be very beneficial when this is incorporated into the existing unsupervised models. We encode this valuable information into our supervised model by computing 3 distance features (FQG_1, FQG_2, FQG_3) on the FQG. If y's message is same as or **reply to** x's message, then it is likely that they discuss the same topic. We use the **'name'** feature since participants often use each other's **name** in multi-party conversations to make disentanglement easier (Elsner and Charniak 2010).

Classification Results Table 2 shows the classifier's performance in terms of overall accuracy, and precision and recall of the 'same' class for different types of feature, averaged over leave one out procedure. Conversation features yield the highest accuracy. Topic features also have proved to be important. Lexical features have poor accuracy (a bit higher than the majority class baseline). However, when we combine all the features we get the best performance.

Figure 1 shows the relative importance of the features based on the absolute values of their coefficients in the LR classifier. Distance in number of edges in the (directed) FQG (FQG_2) is the most effective feature, followed by the decision of the LCSeg segmenter (LCSeg). The other two features on FQG (i.e., FQG_1, FQG_3) are also very relevant.

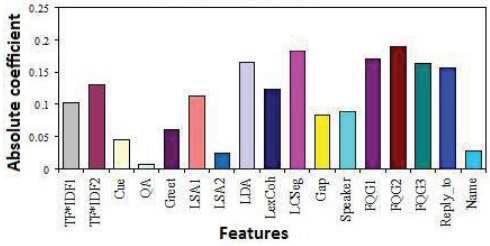

Figure 1: Relative importance of the features

Accuracy vs. amount of labeled data Figure 2 shows the classification error rate, tested on five randomly selected conversations and trained on random sample of different number of conversations. Our approach appears to achieve its best performance with a limited amount of data. The error rate flattens with about 25-30 conversations.

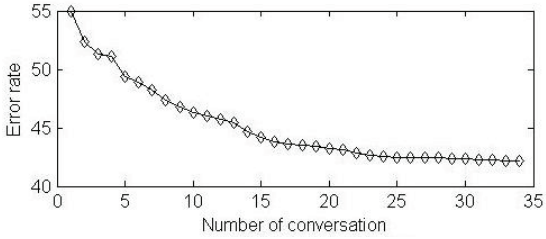

Figure 2: Classifier accuracy vs. amount of data

Graph Partitioning

Given a weighted undirected complete graph $G = (V, E)$, where the nodes V represent the sentences and the edge-weights $w(u, v)$ denote the probability (given by the LR classifier) of the two sentences u and v, to be in the 'same' class, the topic segmentation task can be formulated as a *N-mincut* graph partitioning problem with the intuition that sentences in a segment should be similar to each other, while sentences in different segments should be dissimilar. To do this, 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 $cut(A,B) = \Sigma_{u \in A, v \in B} w(u,v)$ is the total connection from nodes in partition A to nodes in partition B, $assoc(A,V) = \Sigma_{u \in A, t \in V} w(u,t)$ is the total connection from nodes in partition A to all nodes in the graph and

	Baselines		Models					Human
Scores	Speaker	Block 5	LDA	LDA+FQG	LCSeg	LCSeg+FQG	Supervised	
Mean 1-to-1	0.52	0.38	0.57	0.62	0.62	0.68	0.70	0.80
Mean loc_3	0.64	0.57	0.54	0.61	0.72	0.71	0.75	0.83

Table 3: Comparison of Human, System and best Baseline annotations

$assoc(B, V)$ is similarly defined. Previous work on graph-based segmentation (e.g., (Malioutov and Barzilay 2006), (Shi and Malik 2000)) has shown that the 'normalized cut' criterion is more appropriate than just the 'cut' criterion which accounts only for total edge weight connecting A and B and therefore, favors cutting small sets of isolated nodes in the graph. However, solving 'normalized cut' is NP-complete. Hence, we approximate the solution following (Shi and Malik 2000), which is time efficient.

Experiment

Recently, we had humans annotate the BC3 email corpus with topics. The corpus contains 39 email conversations from the W3C corpus. It has 3200 sentences and an average of 5 emails per conversation. The right most column of table 4 shows some basic statistics computed on the three annotations of the corpus. On average, we have 2.5 topics per conversation. A topic contains an average of 12.6 sentences. The average number of topics active at a time is 1.4. We use the 1-to-1 and loc_k agreement metrics from (Elsner and Charniak 2010) to compare different human annotations and model's output. The 1-to-1 metric measures the global similarity between two annotations by pairing up the clusters of the two annotations in a way that maximizes the total overlap and reports the percentage of overlap. loc_k measures the local agreement within a context of k sentences. The annotation procedure, annotation statistics, and the agreements found, are described in detail in (Joty et al. 2010).

We ran our five models (i.e., LDA, LDA+FQG, LCSeg, LCSeg+FQG from our previous work along with the novel supervised method) on our corpus. For a fair comparison we set the same number of topic per conversation in all of them. If at least two of the three annotators agree on the topic number we set that number, otherwise we set the floor value of the average topic number. The average statistics of the five resulting model annotations are shown in Table 4. Comparing with the human annotations, we see that the models' annotations fall within the bounds of the human annotations.

Avg. Topic	LDA	LDA +FQG	LCSeg	LCSeg +FQG	Super vised	Human
Number	2.10	1.90	2.2	2.41	2.41	2.5
Length	13.3	15.50	13.12	12.41	12.41	12.6
Density	1.83	1.60	1.01	1.39	1.12	1.4

Table 4: Corpus statistics of different annotations

We present our results in Table 3. We evaluated the following baselines and report only the best two (i.e., 'Speaker' and 'Block 5') in the table. (i) **All different:** Each sentence is a separate topic. (ii) **All same:** The whole thread is a single topic. (iii) **Speaker:** The sentences from each participant constitute a separate topic. (iv) **Blocks of** $k(= 5, 10, 15, 20)$**:** Each consecutive group of k sentences is a topic.

In general, our models perform better than the baselines, but worse than the gold standard. When we compare our supervised model that combines all the features with the best performing unsupervised model LCSeg+FQG, we get a significant improvement in both metrics ($p<0.01$). This improvement may also be due to the fact that, by constructing a complete graph, this model considers relations between all possible sentence pairs in a thread, which we believe is a key requirement for topic segmentation of email conversations.

Conclusion and Future Work

We presented a novel supervised topic segmentation model for email conversations that incorporates lexical, conversational and topic features. Empirical evaluation shows that our supervised method outperforms all of the unsupervised models even if it is trained on a rather limited amount of data. In future, we will investigate how our models perform in other asynchronous conversations like blogs and fora.

References

Allan, J. 2002. *Topic detection and tracking: event-based information organization*. Kluwer Academic Pub. 1–16.

Blei, D. M.; Ng, A. Y.; and Jordan, M. I. 2003. Latent dirichlet allocation. *JMLR* 3:993–1022.

Carenini, G.; Ng, R. T.; and Zhou, X. 2007. Summarizing email conversations with clue words. In *WWW'07*. ACM.

Choi, F. Y. Y.; Hastings, P. W.; and Moore, J. 2001. Latent semantic analysis for text segmentation. In *EMNLP'01*.

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

Elsner, M., and Charniak, E. 2010. Disentangling chat. *Computational Linguistics* 36:389–409.

Galley, M.; McKeown, K.; Fosler-Lussier, E.; and Jing, H. 2003. Discourse segmentation of multi-party conversation. In *ACL'03*, 562–569. Sapporo, Japan: ACL.

Harabagiu, S., and Lacatusu, F. 2005. Topic themes for multi-document summarization. In *SIGIR '05:*, 202–209.

Hearst, M. A. 1997. TextTiling: segmenting text into multi-paragraph subtopic passages. *Comp. Linguist.* 23(1):33–64.

Joty, S.; Carenini, G.; Murray, G.; and Ng, R. T. 2010. Exploiting conversation structure in unsupervised topic segmentation for emails. In *EMNLP'10*. USA: ACL.

Malioutov, I., and Barzilay, R. 2006. Minimum cut model for spoken lecture segmentation. In *ACL'06*, 25–32.

Shi, J., and Malik, J. 2000. Normalized cuts and image segmentation. *IEEE Tra. Pat. Ana. Mac. Int.* 22(8):888–905.