Automatic Topic Labeling in Asynchronous Conversations

Shafiq Joty and Giuseppe Carenini and Raymond Ng*

1 Introduction and Related Work

Asynchronous conversations are conversations where participants collaborate with each other at different times (e.g., email, blog, forum). The huge amount of textual data generated everyday in these conversations calls for automated methods of conversational text analysis. *Topic segmentation and labeling* is often considered a prerequisite for higher-level conversation analysis and has been shown to be useful in many NLP applications including summarization, information extraction and conversation visualization. *Topic segmentation* in asynchronous conversation refers to the task of clustering the sentences into a set of coherent topical clusters. In [6, 7], we presented unsupervised and supervised topic segmentation models for asynchronous conversations, which to our knowledge are the state-of-the-art. This study concerns *topic labeling*, that is, given a set of topical clusters in a conversation, the task is to assign appropriate topic labels to the clusters. For example, five topic labels in a Slashdot [2] blog conversation about releasing a new game called Daggerfall are *game contents and size*, *game design*, *bugs/faults*, *other gaming options* and *performance/speed issues*. Such topic labels can serve as a concise summary of the conversation and can also be used for indexing.

Ideally, topic labels should be meaningful, semantically similar to the underlying topic, general (i.e., broad coverage of the topic) and discriminative (or exclusive) when there are multiple topics [10]. Traditionally, the top K terms in a multinomial topic model (e.g., LDA [3]) are used to represent a topic. However, as pointed out by [10], at the *word-level*, topic labels may become too general that it can impose cognitive difficulties on a user to interpret the meaning of the topic by associating the words together. On the other hand, if the labels are expressed at the *sentence-level*, they may become too specific to cover the whole theme of the topic. Based on these observations, [10] and other recent studies (e.g., [9]) advocate for *phrase-level* topic labels. This is also consistent with the monologue corpora built as part of the Topic Detection and Tracking (TDT) project [1] as well as with our own email and blog conversational corpora in which human annotators without specific instructions spontaneously generated labels at the phrase-level. Considering all these factors, we also treat phrase-level as the right level of granularity for a label in this work.

Few prior studies have addressed the topic labeling problem in different settings [10, 9]. Common to their approaches is that they first mine topics in the form of topic-word distributions from the *whole corpus* using topic models like LDA. Then, they try to label the topics (i.e., topic-word distributions) with an appropriate label using the statistical association metrics (e.g., point-wise mutual information, t-test) computed from either the source corpus or an external knowledge base (e.g., Wikipedia). In contrast, our task is to label the topical clusters in a given *conversation*, where topics are closely related and distributional variations are subtle (e.g., *game contents and size, game design*). Therefore, corpus-based statistical association metrics are not reliable in our case. Also at the conversation-level, the topics are too specific to find their labels in an external source. To our knowledge, none has studied this problem before. Therefore, there is no standard corpus and no agreed-upon evaluation metrics available.

Our contributions aim to remedy these problems. First, we present a blog and an email corpora annotated with topics. Second, we propose to generate topic labels using an *extractive* approach, that finds the most representative phrases from the text without relying on an external source. Since, graph-based key phrase ranking has proved to be the state-of-the-art (unsupervised) method [11], we adopt the same framework. We propose a novel biased random walk model that exploits the fact that the leading sentences in a topic often carry the most informative clues for its label. However, the phrases extracted only by considering the sentences in a topic may ignore the global aspect of the conversation. As another contribution, we propose to re-rank the phrases extracted from the whole conversation with respect to the individual topics and include the relevant ones. Experimental results show that our approach outperforms other ranking models including a general random walk model (i.e., TextRank) proposed by [11], a lead-oriented model and a frequency-based model, and including the relevant conversation-level phrases improves the performance.

2 Corpora and Evaluation Metric

In [6], we presented our email corpus annotated with topics by three annotators. This contains 40 email threads (conversations) from the W3C corpus at an average of 5 emails per thread. Recently, we have built a new blog corpus containing 20 conversations from Slashdot, each annotated with topics by three annotators. The number of comments per blog conversation varies from 30 to 101 with an average of 60.3. The annotators first read the conversation and listed the topics discussed in a conversation by a short description which provides a high-level overview of the topic. Then, they assigned the most appropriate topic to each sentence in the conversation. The short high-level descriptions of the topics serve as reference (or gold) topic labels in our experiments. Note that, the annotators can disagree on the number of topics (specific vs. generic), on the topic assignment of the sentences and on the topic labels. In total, the annotators found 269 and 562 topics (or clusters) in email and blog corpora, respectively.

The traditional approach to evaluate key phrase extraction methods is to compute the precision, recall and F-measure for *each cluster* based on exact matches. However, as claimed by [8], this approach is not flexible enough, as it ignores the near-misses. Besides, in our case, there is only one reference (or gold) topic label per cluster, therefore, the per cluster precision, recall and F-measure metrics are not applicable. Rather, the n-gram-based evaluation metrics used in summarization and machine translation (e.g., ROUGE, BLUE) that account for partial matches, are more appropriate in our scenario. Recently, [8] evaluated the utility of different n-gram-based metrics for key phrase extraction and showed that the metric *R-precision* correlates most with the human judgments. Formally, *R-precision*=$\frac{n_o}{max(n_r,n_c)}$, where n_r and n_c are the number of words in the reference and candidate phrases,

*{rjoty,carenini,rng}@cs.ubc.ca, Department of Computer Science, University of British Columbia, Vancouver.

respectively, and n_o is the number of overlapping (stemmed) words between the candidate and the reference phrases. Note that, this metric evaluates a single candidate phrase against a reference phrase. But, we may choose to produce the top K phrases as output of our system. In that case, we want to evaluate K candidates against one reference phrase. Therefore, we have adapted the metric as: *weighted R-precision*=$\sum_{k=1}^{K} \frac{n_o}{max(n_r,n_c^k)} \times P(c_k)$, where, $P(c_k)$ is the normalized score of the k-th candidate phrase c_k.

3 Our Approach

In the *preprocessing* step, we tokenize the text and apply a *syntactic filter* to select the terms (e.g., nouns, verbs) we want to rank. One simple approach to term ranking would be to use the frequency, but as shown later (see Freq-BL in Section 4), this approach leads to poor results. Our approach is based on the intuition that the leading sentences in a cluster carry the most informative clues for the topic labels, since this is where the speaker most likely will try to signal a topic shift and describe the new topic. Therefore, we may wish to rank the terms based on their relevance to the leading sentences. Let L_t denotes the set of leading sentences in topic t. We consider the first and second sentences as the leading sentences in our experiments. We can measure the relevance of a term w to L_t by: $\rho(w|L_t) = log(tf_{w,L_t} + 1)log(tf_{w,t} + 1)$, where tf_{w,L_t} and $tf_{w,t}$ are the number of times word w appears in L_t and t, respectively. But, this model (see Lead-BL in Section 4) does not consider the inter-word relation. Our intuition is that a term that is related to other high scoring terms in the cluster should also have a high score. Since, the labels for different topics should be *discriminative*, a high scoring term in one cluster should not have high scores in other clusters. This idea is captured by the following combined model, where $P(w|L_t)$, the score of a word w given a set of leading sentences L_t in topic t, is expressed as the convex combination of its relatedness with other words in the topic and its relevance to the leading sentences L_t, i.e., $\rho(w|L_t)$.

$P(w|L_t) = \lambda \sum_{y \in C_t} \frac{e(y,w)}{\sum_{z \in C_t} e(y,z)} P(y|L_t) + (1-\lambda) \frac{\rho(w|L_t)}{\sum_{z \in C_t} \rho(z|L_t)}$, where the value of λ ($0 \leq \lambda \leq 1$), which we call "bias", is a trade-off between the two components and should be set empirically. C_t is the set of words in topic t. We define the edge weights as: $e(y,w) = tf^t_{w,y} \times log(T/(0.5 + t^{'}_{w,y}))$, where T denotes the total number of topics, and $tf^t_{w,y}$ and $t^{'}_{w,y}$ are the number of times terms w and y co-occur in a window of size s in topic t and in topics except t, respectively. This metric for computing edge weights is intended to give discriminative labels. The equation for ranking words above can be written in matrix notation as: $\pi = [\lambda Q + (1-\lambda)R]^T \pi = A^T \pi$, where Q and R are square matrices such that $Q_{i,j} = e(i,j)/\sum_{j \in C_t} e(i,j)$ and $R_{i,j} = \rho(j|L_t)/\sum_{z \in C_t} \rho(z|L_t)$ for all i, respectively. Here A is a stochastic matrix, and therefore, it can be treated as the transition matrix of a Markov chain. Imagine performing a random walk on the graph, where at every time step, with probability λ a transition is made to the related words in the topic and with probability $1-\lambda$ a transition is made to the words that are relevant to the leading sentences. The vector π we are looking for is the stationary distribution of this Markov chain and is the (normalized) eigenvector for the eigenvalue 1. For larger matrices π can be efficiently computed by an iterative method called *power method*.

Once we have the ranked list of words in a topic, we select the top M key-words for *post-processing*. In the post-processing, we mark the M key-words in the text, and collapse the sequences of adjacent key-words into a *key-phrase*. The score of a key phrase is determined by taking the maximum score of its constituents (i.e., words). Notice that, the above method extracts phrases only from the cluster and does not consider the full conversation. It fails to find a label if it appears outside the cluster. Therefore, we propose to include the relevant conversation-level phrases with the topic-level ones. To do that, we apply a general random walk model [11] to the whole conversation and extract conversation-level phrases using the same way as described above. Then, we re-rank the conversation-level phrases (i.e., taking maximum score of its constituents) with respect to a topic by using the score of the words (given by biased random walk model) in that cluster. Finally, we use Maximum Marginal Relevance (MMR) [4] to produce the final list of K key phrases as topic labels that are relevant but not redundant to achieve the broad coverage of the topic.

4 Preliminary Results

Table (a) shows the *weighted R-precision* results (in percentage) of different models for different values of K on the two corpora. All the models follow the same preprocessing and post-processing steps. We experimented with five different *syntactic filters* in the preprocessing, and the best performance we achieved was with the filter that selects nouns and adjectives. The results presented here are based on this filter. The bias λ and the window size s in our proposed model and the value of M in the post-processing were empirically set to 0.15, 2 and 25% of the total number of words, respectively, based on a development set. On Blog, the frequency-based baseline *Freq-BL* performs better than the lead-oriented baseline *Lead-BL*, but on email, *Lead-BL* beats *Freq-BL*. The General TextRank *Gen-TR* fails to beat either one or both of the baselines. Our biased TextRank model *Bias-TR* appears to improve the performance consistently. We get further improvement in this model (*Bias-TR+*), when we include the relevant conversation-level phrases. In general, given that the results are expressed in percentage, all these methods perform rather unsatisfactory. This is due to the fact that most of the human-authored labels in our corpora are abstractive in nature. Only 9.81% of the labels in blog and 12.74% of the labels in email appear verbabtim in the conversation. Also recall that, annotators can disagree on number of topics, topic assignments and topic labels. Table (b) shows the performance of our proposed models and annotator agreement with respect to different annotators. To compute agreement on topic labels between two annotations, we first map the clusters of one annotation to the clusters of another in a way that maximizes the total overlap (see [5]), then we measure the R-precision between the labels of the paired clusters. The goal of this preliminary work was to verify to what extent an extractive approach could contribute to the labeling task. We are currently working on exploiting the output of the extractive method within more abstractive approaches.

	K=1		K=2		K=3		K=4		K=5	
Models	Blog	Email	Blog	Email	Blog	Email	Blog	Email	Blog	Email
Freq-BL	13.87	12.97	11.81	10.96	10.63	9.73	9.96	8.96	9.63	8.28
Lead-BL	12.79	15.59	10.84	12.40	9.69	10.96	8.98	10.14	8.56	9.76
Gen-TR	12.83	9.95	10.88	8.43	9.44	8.15	8.55	7.22	8.02	6.63
Bias-TR	14.15	17.72	12.67	14.95	11.67	13.15	10.99	12.31	10.72	11.94
Bias-TR+	14.38	17.37	13.61	15.22	12.89	13.62	12.10	12.39	11.62	11.90

(a) Performance of different models for different values of K on the two corpora.

	Annotator 1		Annotator 2		Annotator 3	
Models	Blog	Email	Blog	Email	Blog	Email
Bias-TR	16.29	16.31	12.40	17.62	13.35	19.34
Bias-TR+	16.48	16.67	11.80	17.52	13.99	18.00
Human	20.93	30.98	19.25	34.30	20.38	37.32

(b) Performance of proposed models (K=1) with respect to different annotators. Human agreement with annotator i means how much other two annotators on average agree with i.

References

[1] http://projects.ldc.upenn.edu/TDT/.

[2] http://slashdot.org.

[3] D. Blei, A. Ng, and M. Jordan. Latent Dirichlet Allocation. *JMLR*, 3:993–1022, 2003.

[4] J. Carbonell and J. Goldstein. The Use of MMR, Diversity-based Reranking for Reordering Documents and Producing Summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336, Melbourne, Australia, 1998. ACM.

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

[6] S. Joty, G. Carenini, G. Murray, and R. Ng. Exploiting Conversation Structure in Unsupervised Topic Segmentation for Emails. In *Proceedings of the conference on Empirical Methods in Natural Language Processing*, pages 388–398, Massachusetts, 2010. ACL.

[7] S. Joty, G. Carenini, G. Murray, and R. Ng. Supervised Topic Segmentation of Email Conversations. In *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*, pages 530–533, Barcelona, 2011. AAAI.

[8] S. Kim, T. Baldwin, and M. Kan. Evaluating N-gram based evaluation metrics for automatic keyphrase extraction. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 572–580, Beijing, China, 2010. ACL.

[9] J. Lau, K. Grieser, D. Newman, and T. Baldwin. Automatic Labelling of Topic Models. In *Proceedings of the 49th annual meeting on Association for Computational Linguistics*, pages 1536–1545, Portland, USA, 2011. ACL.

[10] Q. Mei, X. Shen, and C. Zhai. Automatic Labeling of Multinomial Topic Models. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 490–499, California, USA, 2007. ACM.

[11] R. Mihalcea and D. Radev. *Graph-based Natural Language Processing and Information Retrieval*. Cambridge University Press, 2011.