Towards Topic Labeling with Phrase Entailment and Aggregation

Yashar Mehdad **Giuseppe Carenini** **Raymond T. Ng** **Shafiq Joty**
Department of Computer Science, University of British Columbia
Vancouver, BC, V6T 1Z4, Canada
{mehdad, carenini, rng, rjoty}@cs.ubc.ca

Abstract

We propose a novel framework for topic labeling that assigns the most representative phrases for a given set of sentences covering the same topic. We build an entailment graph over phrases that are extracted from the sentences, and use the entailment relations to identify and select the most relevant phrases. We then aggregate those selected phrases by means of phrase generalization and merging. We motivate our approach by applying over conversational data, and show that our framework improves performance significantly over baseline algorithms.

1 Introduction

Given text segments about the same topic written in different ways (*i.e.*, language variability), topic labeling deals with the problem of automatically generating semantically meaningful labels for those text segments. The potential of integrating topic labeling as a prerequisite for higher-level analysis has been reported in several areas, such as summarization (Harabagiu and Lacatusu, 2010; Kleinbauer et al., 2007; Dias et al., 2007), information extraction (Allan, 2002) and conversation visualization (Liu et al., 2012). Moreover, the huge amount of textual data generated everyday specifically in conversations (*e.g.*, emails and blogs) calls for automated methods to analyze and re-organize them into meaningful coherent clusters.

Table 1 shows an example of two human written topic labels for a topic cluster collected from a blog[1], and possible phrases that can be extracted from the topic cluster using different approaches. This example demonstrates that although most approaches (Mei et al., 2007; Lau et al., 2011; Branavan et al., 2007) advocate extracting phrase-level topic labels from the text segments, topically related text segments do not always contain one keyword or key phrase that can capture the meaning of the topic. As shown in this example, such labels do not exist in the original text and cannot be extracted using the existing probabilistic models (*e.g.*, (Mei et al., 2007)). The same problem can be observed with many other examples. This suggests the idea of aggregating and generating topic labels, instead of simply extracting them, as a challenging scenario for this field of research.

Text: a: Where do you think the term "Horse laugh" comes from? b: And that rats also giggled when tickled. c: My hypothesis- if an animal can play, it can "laugh" or at least it is familiar with the concept of "laughing". Many animals play. There are various sorts of humour though. Some involve you laughing because your brain suddenly made a lots of unexpected connections.
Possible extracted phrases: animals play, rats have, laugh, Horse laugh, rats also giggle, rats
Human-authored topic labels: animals which laugh, animal laughter

Table 1: Topic labeling example.

Moreover, to generate a label for a topic we have to be able to capture the overall meaning of a topic. However, most current methods disregard semantic relations, in favor of statistical models of word distributions and frequencies. This calls for the integra-

[1] http://slashdot.org

Proceedings of NAACL-HLT 2013, pages 179–189,
Atlanta, Georgia, 9–14 June 2013.

tion of semantic models for topic labeling.

Towards the solution of the mentioned problems, in this paper we focus on two novel contributions:

1. Phrase aggregation. We propose to generate topic labels using the extracted information by producing the most representative phrases for each text segment. We perform this task in two steps. First, we generalize some lexically diverse concepts in the extracted phrases. Second, we aggregate and generate new phrases that can semantically imply more than one original extracted phrase. For example, the phrase "*rats also giggle*" and "*horse laugh*" should be merged into a new phrase "*animals laugh*". Although our method is still relying on extracting phrases, we move beyond current extractive approaches, by generating new phrases through generalization and aggregation of the extracted ones.

2. Building a multidirectional entailment graph over the extracted phrases to identify and select the relevant information. We set such problem as an application-oriented variant of the Textual Entailment (TE) recognition task (Dagan and Glickman, 2004), to identify the information that are semantically equivalent, novel, or more informative with respect to the content of the others. In this way, we prune the redundant and less informative text portions (*e.g.*, phrases), and produce semantically informed phrases for the generation phase. In the case of the example in Table 1, we eliminate phrases such as "*rats have*", "*rats*" and "*laugh*" while keeping "*animal play*", "*Horse laugh*" and "*rats also giggle*".

The experimental results over conversational data sets show that, in all cases, our approach outperforms other models significantly. Although conversational data are known to be challenging (Carenini et al., 2011), we choose to test our method on conversations because this is a genre in which topic modeling is critically needed, as conversations lack the structure and organization of, for instance, edited monologues. The results indicate that our framework is sufficiently robust to deal with topic labeling in less structured, informal genres (when compared with edited monologues). As an additional result of our experiments, we show that the identification and selection phase using semantic relations (entailment graph) is a necessary step to perform the final step (*i.e.,* the phrase aggregation).

2 Topic Labeling Framework

Each topic cluster contains the sentences that can semantically represent a topic. The task of clustering the sentences into a set of coherent topic clusters is called topic segmentation (Joty et al., 2011), which is out of the scope of this paper. Our goal is to generate an understandable label (*i.e.*, a sequence of words) that could capture the semantic of the topic, and distinguish a topic from other topics (based on definition of a good topic label by (Mei et al., 2007)), given a set of topic clusters. Among possible choices of word sequences as topic labels, in order to balance the granularity, we set phrases as valid topic labels.

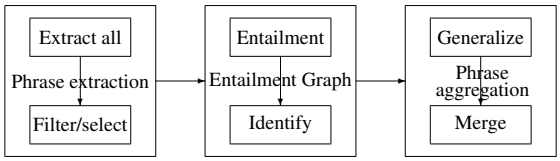

Figure 1: Topic labeling framework.

As shown in Figure 1, our framework consists of three main components that we describe in more details in the following sections.

2.1 Phrase extraction

We tokenize and preprocess each cluster in the collection of topic clusters with lemmas, stems, part-of-speech tags, sense tags and chunks. We also extract n-grams up to length 5 which do not start or end with a stop word. In this phase, we do not include any frequency count feature in our candidate extraction pipeline. Once we have built the candidates pool, the next step is to identify a subset containing the most significant of those candidates. Since most top systems in key phrase extraction use supervised approaches, we follow the same method (Kim et al., 2010b; Medelyan et al., 2008; Frank et al., 1999).

Initially, we consider a set of features used in the other systems to determine whether a phrase is likely to be a key phrase. However, since our dataset is conversational (more details in Section 3), and the text segments are not long, we aim for a classifier with high recall. Thus, we only use TFxIDF (Salton and McGill, 1986), position of the first occurrence (Frank et al., 1999) and phrase length as our features. We merge the training and test data released

for SemEval-2010 Task #5 (Kim et al., 2010b), which consists of 244 scientific articles and 3705 key phrases, to train a Naive Bayes classifier in order to learn a supervised model. We then apply our model to extract the candidate phrases from the collected candidates pool.

As a further step, to increase the coverage (recall) of our extracted phrases and to reduce the number of very short phrases (frequent keywords), we choose the chunks containing any of the extracted keywords. We add those chunks to our extracted phrases and eliminate the associated keywords.

2.2 Entailment graph

So far, we have extracted a pool of key phrases from each topic cluster. Many such phrases include redundant information which are semantically equivalent but vary in lexical choices. By identifying the semantic relations between the phrases we can discover the information in one phrase that is semantically equivalent, novel, or more/less informative with respect to the content of the other phrase.

We set this problem as a variant of the Textual Entailment (TE) recognition task (Mehdad et al., 2010b; Adler et al., 2012; Berant et al., 2011). We build an entailment graph for each topic cluster, where nodes are the extracted phrases and edges are the entailment relations between nodes. Given two phrases (ph_1 and ph_2), we aim at identifying and handling the following cases:
i) ph_1 and ph_2 express the same meaning (*bidirectional* entailment). In such cases one of the phrases should be eliminated;
ii) ph_1 is more informative than ph_2 (*unidirectional* entailment). In such cases, the entailing phrase should replace or complement the entailed one;
iii) ph_1 contains facts that are not present in ph_2, and vice-versa (the "*unknown*" cases in TE parlance). In such cases, both phrases should remain.

Figure 2 shows how entailment relations can help in selecting the phrases by removing the redundant and less informative ones. For example, the phrase "*animals laugh*" entails "*rats giggle*", "*Horse laugh*" and "*Mice chuckle*",[2] but not "*Animals play*".

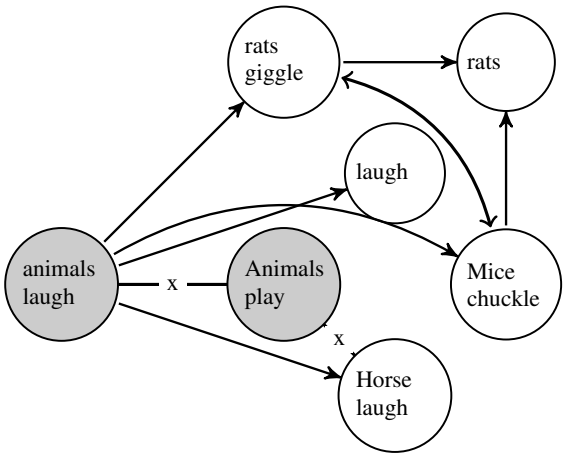

Figure 2: Building an entailment graph over phrases. Arrows and "x" represent the entailment direction and unknown cases respectively.

So we can keep "*animals laugh*" and "*Animals play*" and eliminate others. In this way, TE-based phrase identification method can be designed to distinguish meaning-preserving variations from true divergence, regardless of lexical choices and structures.

Similar to previous approaches in TE (*e.g.*, (Berant et al., 2011; Mehdad et al., 2010b; Mehdad et al., 2010a)), we use supervised method. To train and build the entailment graph, we perform the following three steps.

2.2.1 Training set collection

In the last few years, TE corpora have been created and distributed in the framework of several evaluation campaigns, including the Recognizing Textual Entailment (RTE) Challenge[3] and Cross-lingual textual entailment for content synchronization[4] (Negri et al., 2012). However, such datasets cannot directly support our application. Specifically, our entailment graph is built over the extracted phrases (with max. length of 5 tokens per phrase), while the RTE datasets are composed of longer sentences and paragraphs (Bentivogli et al., 2009; Negri et al., 2011).

In order to collect a dataset which is more similar to the goal of our entailment framework, we decide to select a subset of the sixth and seventh RTE challenge main task (*i.e.*, RTE within a Corpus). Our

[2]Assuming that "animals laugh" is interpreted as "all animals laugh".

[3]http://pascallin.ecs.soton.ac.uk/Challenges/RTE/

[4]http://www.cs.york.ac.uk/semeval-2013/task8/

dataset choice is based on the following reasons: *i*) the length of sentence pairs in RTE6 and RTE7 is shorter than the others, and *ii*) RTE6 and RTE7 main task datasets are originally created for summarization purpose which is closer to our work. We sort the RTE6 and RTE7 dataset pairs based on the sentence length and choose the first 2000 samples with a equal number of positive and negative examples. The average length of words in our training data is 6.7 words. There are certainly some differences between our training set and our phrases. However, the collected training samples was the closest available dataset to our purpose.

2.2.2 Feature representation and training

Working at the phrase level imposes another constraint. Phrases are short and in terms of syntactic structure, they are not as rich as sentences. This limits our features to the lexical level. Lexical models, on the other hand, are less computationally expensive and easier to implement and often deliver a strong performance for RTE (Sammons et al., 2011).

Our entailment decision criterion is based on similarity scores calculated with a phrase-to-phrase matching process. Each example pair of phrases (ph_1 and ph_2) is represented by a feature vector, where each feature is a specific similarity score estimating whether ph_1 entails ph_2.

We compute 18 similarity scores for each pair of phrases. In order to adapt the similarity scores to the entailment score, we normalize the similarity scores by the length of ph_2 (in terms of lexical items), when checking the entailment direction from ph_1 to ph_2. In this way, we can check the portion of information/facts in ph_2 which is covered by ph_1.

The first 5 scores are computed based on the exact lexical overlap between the phrases: word overlap, edit distance, ngram-overlap, longest common subsequence and Lesk (Lesk, 1986). The other scores were computed using lexical resources: WordNet (Fellbaum, 1998), VerbOcean (Chklovski and Pantel, 2004), paraphrases (Denkowski and Lavie, 2010) and phrase matching (Mehdad et al., 2011). We used WordNet to compute the word similarity as the least common subsumer between two words considering the synonymy-antonymy, hypernymy-hyponymy, and meronymy relations. Then, we calculated the sentence similarity as the sum of the similarity scores of the word pairs in Text and Hypothesis, normalized by the number of words in Hypothesis. We also use phrase matching features described in (Mehdad et al., 2011) which consists of phrasal matching at the level on ngrams (1 to 5 grams). The rationale behind using different entailment features is that combining various scores will yield a better model (Berant et al., 2011).

To combine the entailment scores and optimize their relative weights, we train a Support Vector Machine binary classifier, SVMlight (Joachims, 1999), over an equal number of positive and negative examples. This results in an entailment model with 95% accuracy over 2-fold and 5-fold cross-validation, which further proves the effectiveness of our feature set for this lexical entailment model. The reason that we gained a very high accuracy is because our selected sentences are a subset of RTE6 and RTE7 with a shorter length (less number of words) which makes the entailment recognition task much easier than recognizing entailment between paragraphs or complex long sentences.

2.2.3 Graph edge labeling

We set the edge labeling problem as a two-way classification task. Two-way classification casts multidirectional entailment as a unidirectional problem, where each pair is analyzed checking for entailment in both directions (Mehdad et al., 2012). In this condition, each original test example is correctly classified if both pairs originated from it are correctly judged ("*YES-YES*" for bidirectional,"*YES-NO*" and "*NO-YES*" for unidirectional entailment and "*NO-NO*" for unknown cases). Two-way classification represents an intuitive solution to capture multidimensional entailment relations. Moreover, since our training examples are labeled with binary judgments, we are not able to train a three-way classifier.

2.2.4 Identification and selection

Assigning all entailment relations between the extracted phrase pairs, we are aiming at identifying relevant phrases and eliminating the redundant (in terms of meaning) and less informative ones. In order to perform this task we follow a set of rules based on the graph edge labels. Note that since entailment

#	Merging patterns
1	merge ($cw1_{1_{(C_{POS=[N\|V\|J])}}}..w1_n$, $cw2_{1_{(C_{POS=[N\|V\|J])}}}..w2_n$) = $w1_1..w1_n$ and $w2_2..w2_n$
E.g.	merge (*challenging situation* , *challenging problem*) = *challenging situation and problem*
2	merge ($w1_1..cw1_{n_{(C_{POS=[N\|V\|J])}}}$, $w2_1..cw2_{n_{(C_{POS=[N\|V\|J])}}}$) = $w1_1..w1_{n-1}$ and $w2_1..w2_n$
E.g.	merge (*wet Mars* , *warm Mars*) = *wet and warm Mars*
3	merge ($w1_1..cw1_{n_{(C_{POS=[N\|V\|J])}}}$, $cw2_{1_{(C_{POS=[N\|V\|J])}}}..w2_n$) = $w1_1..w1_n\ w2_2..w2_n$
E.g.	merge (*interesting story* , *story continues*) = *interesting story continues*
4	merge ($cw1_{1_{(C_{POS=[N\|V\|J])}}}..w1_n$, $w2_1..cw2_{n_{(C_{POS=[N\|V\|J])}}}$) = $w2_1..w2_n\ w1_2..w1_n$
E.g.	merge (*LHC shutting down* , *details about LHC*) = *details about LHC shutting down*
5	merge ($w1_{1_{C_{pos}}}, cw1_{2_{(C_{POS=[N\|V\|J])}}}, w1_{3_{C_{pos}}}$, $w2_{1_{C_{pos}}}, cw2_{2_{(C_{POS=[N\|V\|J])}}}, w2_{3_{C_{pos}}}$) = $w1_1$ and $w2_1\ w2_2\ w2_3$ and $w1_3$
E.g.	merge (*technology grow fast* , *media grow exponentially*) = *technology and media grow exponentially and fast*

Table 2: Phrase merging patterns.

is a transitive relation, our entailment graph is transitive *i.e.*, if entail(ph_1,ph_2) and entail(ph_2,ph_3) then entail(ph_1,ph_3) (Berant et al., 2011).

Rule 1) If there is a chain of entailing nodes, we keep the one which is in the root of the chain and eliminate others (e.g. "*animals laugh*" in Figure 2);

Rule 2) Among the nodes that are connected with bidirectional entailment (semantically equivalent nodes) we keep only the one with more outgoing bidirectional and unidirectional entailment relations, respectively;

Rule 3) Among the nodes that are connected with unknown entailment (novel information with respect to others) we keep the ones with no incoming entailment relation (*e.g.*, "*Animals play*" in Figure 2).

Although deleting might be harsh, in our current framework, we only rely on the performance of an entailment model which gives us a yes/no entailment decision. In future, we are planning to improve our entailment graph by weighting the edges. In this way, we can take advantage of the weights to make a more conservative decision in pruning the entailment chains.

2.3 Phrase aggregation

Once we have identified and selected the informative phrases, the generation of topic labels can be done in two steps. First, we generalize the phrases containing the concepts that are lexically connected. Second, we merge the phrases with a set of hand written linguistically motivated patterns.

2.3.1 Phrase generalization

In this step, we generalize phrases that contain concepts which are lexically connected. For this purpose, we search in phrases for different words with the same part-of-speech and sense tag. Then, we find the link between those words in WordNet. If they are connected and the shortest path connecting them is less than 3 (estimated over the development set), we replace both by their common parent in the WordNet. In the case that they belong to the same synset, we can replace one by another. Note that we limit our search to nouns and verbs. For example, "*rat*" and "*horse*" can be replaced by "*animal*", or "*giggle*" and "*chuckle*" can be replaced by "*laugh*". The motivation behind the generalization step is to enrich the common terms between the phrases in favor of increasing the chance that they could merge to a single phrase. This also helps to move beyond the limitation of original lexical choices.

2.3.2 Phrase merging

The goal is to merge the phrases that are connected, and to generate a human readable phrase that contains more information than a single extracted phrase. Several approaches have been proposed to aggregate and merge sentences in Natural Language Generation (NLG) (*e.g.* (Barzilay and Lapata, 2006; Cheng and Mellish, 2000)), however most of them use syntactic structure of the sentences. To merge phrases at the lexical level, we set few common linguistically motivated aggregation patterns such as: simple conjunction, and conjunction via shared participants (Reiter and Dale, 2000).

Table 2 demonstrates the merging patterns, where wi_j is the jth word (or segment) in phrase i, cw is the common word (or segment) in both phrases and C_{POS} is the common part-of-speech tag of the corresponding word. To illustrate, pattern 1

looks for the first segment of each phrase (wi_1). If they are same (cwi_1) and share the same POS tag (C_{POS}), then we aggregate the first phrase ($w1_1..w1_n$) and the second phrase removing the first element ($w2_{\mathbf{2}}..w2_n$) by using the connective "*and*". For instance, the aggregation of "*animals laugh*" and "*animals play*" results in "*animals laugh and play*". The rest of the patterns follow the same logic and for the sake of brevity we avoid illustrating each pattern. These patterns are among the most common domain and application independent methods by which two phrases/sentences can be aggregated, as described in the NLG literature (Reiter and Dale, 2000).

In our aggregation pipeline, we group the phrases based on their lexical overlap (number of common words). The merging process is conducted over each group in descending order (larger number of words in common), in order to increase the chance of merging rules application. Then, we perform the merging over the resulting generated phrases from each group. If our phrases cannot be merged (*i.e.*, do not match merging patterns), we select them as labels for the topic cluster.

3 Datasets and Evaluation Metrics

3.1 Datasets

To verify the effectiveness of our approach, we experiment with two different conversational datasets. Our interest in dealing with conversational texts derives from two reasons. First, the huge amount of textual data generated everyday in these conversations validates the need of text analysis frameworks to process such conversational texts effectively. Second, conversational texts pose challenges to the traditional techniques, including redundancies, disfluencies, higher language variabilities and ill-formed sentence structure (Liu et al., 2011).

Our conversational datasets are from two different asynchronous media: email and blog. For email, we use the dataset presented in (Joty et al., 2010), where three individuals annotated the publicly available BC3 email corpus (Ulrich et al., 2008) with topics. The corpus contains 40 email threads (or conversations) at an average of 5 emails per thread. On average it has 26.3 sentences and 2.5 topics per thread. A topic has an average length of 12.6 sentences. In total, the three annotators found 269 topics in a corpus of 1,024 sentences.

There are no publicly available blog corpora annotated with topics. For this study, we build our own blog corpus containing 20 blog conversations of various lengths from Slashdot, each annotated with topics by three human annotators.[5] The number of comments per conversation varies from 30 to 101 with an average of 60.3 and the number of sentences per conversation varies from 105 to 430 with an average of 220.6. The annotators first read a conversation and list the topics discussed in the conversation by a short description (e.g., Game contents or size, Bugs or faults) which provides a high-level overview of the topic. Then, they assign 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. The target number of topics was not given in advance and the annotators were instructed to find as many topics as needed to convey the overall content structure of the conversation. The annotators found 5 to 23 topics per conversation with an average of 10.77. The number of sentences per topic varies from 11.7 to 61.2 with an average of 27.16. In total, the three annotators found 512 topics in our blog corpus containing 4,411 sentences overall.

Note that our annotators performed topic segmentation and labeling independently. In the email corpus, the three annotators found 100, 77 and 92 topics respectively (269 in total), and in the blog corpus, they found 251, 119 and 192 topics respectively (562 in total). For the evaluation, there is a single gold standard per topic written by each annotator. Table 1 shows a case in which two annotators selected the same topical cluster and so we have two labels for the same cluster.

3.2 Evaluation metrics

Traditionally, key phrase extraction is evaluated using precision, recall and f-measure based on exact matches on all the extracted key phrases with gold standards for a given text. However, as claimed by (Kim et al., 2010a), this approach is not flexible enough as it ignores the near-misses. Moreover, in the case of topic labeling, most of the human written

[5]The new blog corpus annotated with topics will be made publicly available for research purposes.

topic labels cannot be found in the text. Recently, (Kim et al., 2010a) evaluated the utility of different n-gram-based metrics for key phrase extraction and showed that the metric *R-precision* correlates most with human judgments. *R-precision* normalizes the approximate matching score by the maximum number of words in the reference and candidate phrases. Since this penalize our aggregation phase, where the phrases tend to be longer than original extracted phrase, we decide to use *R-f1* as our evaluation metric which considers length of both reference and candidate phrases.

$$R{-}precision = \frac{1}{k}\sum_{i=1}^{k}\frac{overlap(cand_i, ref)}{\#words(cand_i)}$$

$$R{-}recall = \frac{1}{k}\sum_{i=1}^{k}\frac{overlap(cand_i, ref)}{\#words(ref)}$$

$$R{-}f1 = \frac{2 * R{-}precision * R{-}recall}{(R{-}precision + R{-}recall)}$$

The metric described above only considers word overlap and ignores other semantic relations (e.g., synonymy, hypernymy) between words. However, annotators write labels of their own and may use words that are not directly from the conversation but are semantically related. Therefore, we propose to also use another variant of *R-f1* that incorporates semantic relation between words. To calculate the *Semantic R-f1*, we count the number of overlaps not only when they have the same form, but also when they are connected in WordNet with a synonymy, hypernymy, hyponymy and entailment relation.

Its worth noting that the generalizations phase and the evaluation method are completely independent. In the generalization step, we try to generalize the phrases which are automatically extracted from the text segments. While, in the evaluation, we compare the human written gold standards with the system output. Therefore, using WordNet in the generalization step does not bias the results in the evaluation.

4 Experiments and Results

4.1 Experimental settings

We conduct our experiments over the blog and email datasets described in Section 3.1, after eliminating the development set from the test datasets. In our experiments, the development set was used for the pattern extraction and the shortest path threshold connecting the words in Wordnet in the generalization phase. Our test dataset consists of 461 topics (*i.e.*, clusters and their associated topic labels) from 20 blog conversations and 242 topics from 40 email conversations.

For preprocessing our dataset we use OpenNLP[6] for tokenization, part-of-speech tagging and chuncking. For sense disambiguation, we use the extended gloss overlap measure with the window size of 5, developed by (Pedersen et al., 2005). We also apply Snowball algorithm (Porter, 2001) for stemming.

We compare our approach with two strong baselines. The first baseline Freq-BL ranks the words according to their frequencies and select the top 5 candidates applying Maximum Marginal Relevance algorithm (Carbonell and Goldstein, 1998) using the same pre- and post-processing as the work by (Mihalcea and Tarau, 2004). The second baseline Lead-BL, ranks the words based on their relevance to the leading sentences.[7] The ranking criteria is $log(tf_{w,L_t} + 1) \times log(tf_{w,t} + 1)$, where tf_{w,L_t} and $tf_{w,t}$ are the number of times word w appears in a set of leading sentences L_t and topic cluster t, respectively (Allan, 2002). The log expressions, as the ranking criterion, assign more weights to the words in the topic segment, that also appear in the leading sentences. This is because topics tend to be introduced in the first few sentences of a topical cluster. We also measure the performance of our framework at each step in order to compare the effectiveness of each phase independently or in combination.

4.2 Results

We evaluate the performance of different models using the metrics *R-f1* and *Semantic R-f1* (*Sem-R-f1*), described in Section 3.2. Table 4 shows the results in percentage for different models. The results show that our framework outperforms the baselines signif-

[6] http://opennlp.sourceforge.net/

[7] The key intuitions for this baseline is the leading sentences of a topic cluster carry the most informative clues for the topic labels. Based on our development set, when we consider the first three sentences, the coverage of content words that appear in human labeled topics are 39% and 49% for blog and email, respectively.

Blog		Email	
Human-authored	**system generated**	**Human-authored**	**system generated**
Shutting down the LHC	*story about the LHC shutting down* (#3)	*How it affects coding*	*it screws my coding*
typical shutdown and upgrade times	*typical and scheduled shutdown* (#2)	*Opinions and preferences of tools*	*opinion about what tools*
MARS was warm and wet 3B years ago	*Mars was warm and wet early history* (#3)	*white on black for disabled users*	*white text on black background* (#3)
Moon Treaty and outer space treaty	*Moon and Outer Space Treaty* (#2)	*Contact with Steven*	*email to Steven Pemberton* (#3)

Table 3: Successful examples of human-authored and system generated labels for blog and email datasets. The number near some examples refers to the aggregation patterns in Table 2.

	R-f1		Sem-R-f1	
Models	blog	email	blog	email
Lead-BL	13.5	14.0	34.5	30.1
Freq-BL	15.3	13.1	34.7	29.1
Extraction-BL	13.9	16.0	31.6	33.2
Entailment	12.2	15.6	30.8	33.3
Extraction+Aggregation	15.1	**18.5**	**35.5**	**37.6**
Extraction+Entailment+ Aggregation	**17.9**	**20.4**	**38.7**	**41.6**

Table 4: Results for candidate topic labels on blog and email corpora.

icantly[8] in both datasets.

On the blog corpus, our key phrase extraction method (*Extraction-BL*) fails to beat the other baselines (Lead-BL and Freq-BL) in majority of cases (except R-f1 for Lead-BL). However, in the email dataset, it improves the performance over both baselines in both evaluation metrics. This might be due to the shorter topic clusters (in terms of number of sentences) in email corpus which causes a smaller number of phrases to be extracted.

We also observe the effectiveness of the aggregation phase. In all cases, there is a significant improvement ($p < 0.05$) after applying the aggregation phase over the extracted phrases (*Extraction+Aggregation*).

Note that there is no improvement over the extraction phase after the entailment (*Entailment* row). This is mainly due to the fact that the entailment phase filters the equivalent phrases. This affects the results negatively when such filtered phrases share many common words with our human-authored phrases. However, the results improve more significantly ($p < 0.01$) when the aggregation is conducted after the entailment. This demonstrates that, the combination of these two steps are beneficial for topic labeling over conversational datasets.

In addition, the differences between the results using *R-f1* and *Sem-R-f1* metrics suggests the need for more flexible automatic evaluation methods for this task. Moreover, although the same trend of improvement is observed in blog and email corpora, the differences between their performance suggest the investigation of specialized methods for various conversational modalities.

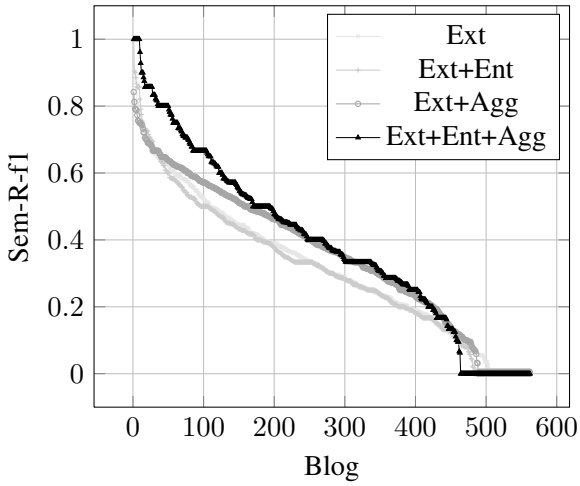

Figure 3: *Sem-R-f1* results distribution after each phase of our pipeline for blog corpus. The x-axis represents the examples sorted based on their *Sem-R-f1* score.

To further analyze the performance, in Figure 3, we show the *Sem-R-f1* results distribution for our blog dataset.[9] We can observe that the aggregation after the entailment phase (bold curve) clearly increase the number of correct labels, while such improvement can be only achieved when the entailment relations is used to identify the relevant phrases. This further highlights the need of semantics in this task. Comparing both datasets, this effect is more dominant in blogs. We believe that this is due to the length of topic clusters. Presumably, building an entailment graph over a greater pool of

[8]The statistical significance tests was calculated by approximate randomization described in (Yeh, 2000).

[9]For brevity's sake we do not show the email dataset graph.

original phrases is more effective to filter the redundant information and identify the relevant phrases.

5 Discussion

After analyzing the results and through manual verification of some cases, we observe that our approach led to some interestingly successful examples. Table 3 shows few generated labels and the human written topics for such cases.

In general, given that the results are expressed in percentage, it appears that the performance is still far from satisfactory level. This leaves an interesting challenge for the research community to tackle. However, this is not always due to the weakness of our proposed model. We have identified three different system independent sources of error:[10]

Type 1: Abstractive human-authored labels: the nature of our method is based on extraction (with the exception of our simple generalization phase) and in many cases the human-written labels cannot be extracted from the text and require more complex generalizations. In fact, only 9.81% of the labels in blog and 12.74% of the labels in email appear verbatim in their respective conversations. For example:

Human-authored label: *meeting schedule and location*

Generated phrases: *meeting, Boston area, mid October*

Type 2: Evaluation methods: in this work, we proposed a semantic method to evaluate our system. However, the current evaluation methods fail to capture the meaning. For example:

Human-authored label: *Food choices*

Generated phrase: *I would ask what people want to eat*

Type 3: Subjective topic labels: often is not easy for human to agree on one label for a topic cluster.[11] For example:

Human-authored label 1: *Member introduction*

Human-authored label 2: *Bio of Len*

Generated phrases: *own intro, Len Kasday, chair*

In light of this analysis, we conclude that a more comprehensive evaluation method (e.g., human evaluation) could better reveal the potential of our system in dealing with topic labeling, specially on conversational data.

[10]There are many examples of such cases, however for brevity we just mention one example for each type.

[11]The mean R-precision agreements computed based on one-to-one mappings of the topic clusters are 20.22 and 36.84 on blog and email data sets, respectively.

6 Conclusion

In this paper, we study the problem of automatic topic labeling, and propose a novel framework to label topic clusters with meaningful readable phrases. Within such framework, this paper makes two main contributions. First, in contrast with most current methods based on fully extractive models, we propose to aggregate topic labels by means of generalizing and merging techniques. Second, beyond current approaches which disregard semantic information, we integrate semantics by means of building textual entailment graphs over the topic clusters. To achieve our objectives, we successfully applied our framework over two challenging conversational datasets. Coherent results on both datasets demonstrate the potential of our approach in dealing with topic labeling task.

Future work will address both the improvement of our aggregation phase and ranking the output candidate phrases for each topic cluster. On one hand, we plan to accommodate more sophisticated NLG techniques for the aggregation and generation phase. Incorporating a better source of prior knowledge in the generalization phase (e.g., YAGO or DBpedia) is also an interesting research direction towards a better phrase aggregation step. On the other hand, we plan to apply a ranking strategy to select the top candidate phrases generated by our framework.

Acknowledgments

We would like to thank the anonymous reviewers and Frank Tompa for their valuable comments and suggestions to improve the paper, and the NSERC Business Intelligence Network for financial support. Yashar Mehdad also would like to acknowledge the early discussions on the related topics with Matteo Negri.

References

Meni Adler, Jonathan Berant, and Ido Dagan. 2012. Entailment-based text exploration with application to the health-care domain. In *Proceedings of the ACL 2012 System Demonstrations*, ACL '12, pages 79–84,

Stroudsburg, PA, USA. Association for Computational Linguistics.

James Allan. 2002. Topic detection and tracking: event-based information organization.

Regina Barzilay and Mirella Lapata. 2006. Aggregation via set partitioning for natural language generation. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, HLT-NAACL '06, pages 359–366, Stroudsburg, PA, USA. Association for Computational Linguistics.

Luisa Bentivogli, Ido Dagan, Hoa Trang Dang, Danilo Giampiccolo, and Bernardo Magnini. 2009. The fifth pascal recognizing textual entailment challenge. In *In Proc Text Analysis Conference (TAC09*.

Jonathan Berant, Ido Dagan, and Jacob Goldberger. 2011. Global learning of typed entailment rules. In *Proceedings of ACL*, Portland, OR.

SRK Branavan, Pawan Deshpande, and Regina Barzilay. 2007. Generating a table-of-contents. In *ACL*, volume 45, page 544.

Jaime Carbonell and Jade Goldstein. 1998. 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*, SIGIR '98, pages 335–336, New York, NY, USA. ACM.

Giuseppe Carenini, Gabriel Murray, and Raymond Ng. 2011. Methods for mining and summarizing text conversations.

Hua Cheng and Chris Mellish. 2000. Capturing the interaction between aggregation and text planning in two generation systems. In *In Proceedings of the 1st International Natural Language Generation Conference, 186193, Mitzpe*.

Timothy Chklovski and Patrick Pantel. 2004. Verbocean: Mining the web for fine-grained semantic verb relations. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 33–40, Barcelona, Spain, July. Association for Computational Linguistics.

I. Dagan and O. Glickman. 2004. Probabilistic textual entailment: Generic applied modeling of language variability.

Michael Denkowski and Alon Lavie. 2010. Meteor-next and the meteor paraphrase tables: improved evaluation support for five target languages. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, WMT '10, pages 339–342, Stroudsburg, PA, USA. Association for Computational Linguistics.

Gaël Dias, Elsa Alves, and José Gabriel Pereira Lopes. 2007. Topic segmentation algorithms for text summarization and passage retrieval: an exhaustive evaluation. In *Proceedings of the 22nd national conference on Artificial intelligence - Volume 2*, AAAI'07, pages 1334–1339. AAAI Press.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.

Eibe Frank, Gordon W. Paynter, Ian H. Witten, Carl Gutwin, and Craig G. Nevill-Manning. 1999. Domain-specific keyphrase extraction. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, IJCAI '99, pages 668–673, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Sanda Harabagiu and Finley Lacatusu. 2010. Using topic themes for multi-document summarization. *ACM Trans. Inf. Syst.*, 28(3):13:1–13:47, July.

T. Joachims. 1999. Making large-scale svm learning practical. LS8-Report 24, Universität Dortmund, LS VIII-Report.

Shafiq Joty, Giuseppe Carenini, Gabriel Murray, and Raymond T. Ng. 2010. Exploiting conversation structure in unsupervised topic segmentation for emails. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, Stroudsburg, PA, USA. Association for Computational Linguistics.

Shafiq Joty, Gabriel Murray, and Raymond T. Ng. 2011. Supervised topic segmentation of email conversations. In *In ICWSM11*. AAAI.

Su Nam Kim, Timothy Baldwin, and Min-Yen Kan. 2010a. 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, Stroudsburg, PA, USA. Association for Computational Linguistics.

Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin. 2010b. Semeval-2010 task 5: Automatic keyphrase extraction from scientific articles. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, SemEval '10, pages 21–26, Stroudsburg, PA, USA. Association for Computational Linguistics.

Thomas Kleinbauer, Stephanie Becker, and Tilman Becker. 2007. Combining multiple information layers for the automatic generation of indicative meeting abstracts. In *Proceedings of the Eleventh European Workshop on Natural Language Generation*, ENLG '07, pages 151–154, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jey Han Lau, Karl Grieser, David Newman, and Timothy Baldwin. 2011. Automatic labelling of topic models. In *ACL*, pages 1536–1545.

Michael Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of the 5th annual international conference on Systems documentation*, SIGDOC '86, pages 24–26, New York, NY, USA. ACM.

F. Liu, Y. Liu, C. Busso, S. Harabagiu, and V. Ng. 2011. *Identifying the Gist of Conversational Text: Automatic Keyword Extraction and Summarization*. Ph.D. thesis, THE UNIVERSITY OF TEXAS AT DALLAS.

Shixia Liu, Michelle X. Zhou, Shimei Pan, Yangqiu Song, Weihong Qian, Weijia Cai, and Xiaoxiao Lian. 2012. Tiara: Interactive, topic-based visual text summarization and analysis. *ACM Trans. Intell. Syst. Technol.*, 3(2):25:1–25:28, February.

O. Medelyan, I.H. Witten, and D. Milne. 2008. Topic indexing with wikipedia. In *Proceedings of the AAAI WikiAI workshop*.

Y. Mehdad, A. Moschitti, and F.M. Zanzotto. 2010a. Syntactic semantic structures for textual entailment recognition. Association for Computational Linguistics.

Yashar Mehdad, Matteo Negri, and Marcello Federico. 2010b. Towards cross-lingual textual entailment. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 321–324, Stroudsburg, PA, USA. Association for Computational Linguistics.

Yashar Mehdad, Matteo Negri, and Marcello Federico. 2011. Using bilingual parallel corpora for cross-lingual textual entailment. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 1336–1345, Stroudsburg, PA, USA. Association for Computational Linguistics.

Yashar Mehdad, Matteo Negri, and Marcello Federico. 2012. Detecting semantic equivalence and information disparity in cross-lingual documents. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2*, ACL '12, pages 120–124, Stroudsburg, PA, USA. Association for Computational Linguistics.

Qiaozhu Mei, Xuehua Shen, and ChengXiang Zhai. 2007. Automatic labeling of multinomial topic models. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '07, pages 490–499, New York, NY, USA. ACM.

R. Mihalcea and P. Tarau. 2004. TextRank: Bringing order into texts. In *Proceedings of EMNLP-04and the 2004 Conference on Empirical Methods in Natural Language Processing*, July.

Matteo Negri, Luisa Bentivogli, Yashar Mehdad, Danilo Giampiccolo, and Alessandro Marchetti. 2011. Divide and conquer: crowdsourcing the creation of cross-lingual textual entailment corpora. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 670–679, Stroudsburg, PA, USA. Association for Computational Linguistics.

Matteo Negri, Alessandro Marchetti, Yashar Mehdad, Luisa Bentivogli, and Danilo Giampiccolo. 2012. Semeval-2012 task 8: cross-lingual textual entailment for content synchronization. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics - Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, SemEval '12, pages 399–407, Stroudsburg, PA, USA. Association for Computational Linguistics.

T. Pedersen, S. Banerjee, and S. Patwardhan. 2005. Maximizing Semantic Relatedness to Perform Word Sense Disambiguation. Research Report UMSI 2005/25, University of Minnesota Supercomputing Institute, March.

Martin F. Porter. 2001. Snowball: A language for stemming algorithms.

Ehud Reiter and Robert Dale. 2000. Building natural language generation systems.

Gerard Salton and Michael J. McGill. 1986. Introduction to modern information retrieval.

Mark Sammons, V.G.Vinod Vydiswaran, and Dan Roth. 2011. Recognizing Textual Entailment. In Daniel M. Bikel and Imed Zitouni, editors, *Multilingual Natural Language Applications: From Theory to Practice*. Prentice Hall, Jun.

Jan Ulrich, Gabriel Murray, and Giuseppe Carenini. 2008. A publicly available annotated corpus for supervised email summarization.

Alexander Yeh. 2000. More accurate tests for the statistical significance of result differences. In *Proceedings of the 18th conference on Computational linguistics - Volume 2*, COLING '00, pages 947–953, Stroudsburg, PA, USA. Association for Computational Linguistics.