7-2011

Generating aspect-oriented multi-document summarization with event-aspect model

Peng LI

Yinglin WANG

Wei GAO
Singapore Management University, weigao@smu.edu.sg

Jing JIANG

Generating Aspect-oriented Multi-Document Summarization with Event-aspect model

Peng Li[1] and **Yinglin Wang**[1] and **Wei Gao**[2]and **Jing Jiang**[3]
[1] Department of Computer Science and Engineering, Shanghai Jiao Tong University
[2] Department of Systems Engineering and Engineering Management, Chinese University of Hong Kong
[3] School of Information Systems, Singapore Management University
{lipeng, ylwang@sjtu.edu.cn} {wgao@se.cuhk.edu.hk} {jingjiang@smu.edu.sg}

Abstract

In this paper, we propose a novel approach to automatic generation of aspect-oriented summaries from multiple documents. We first develop an event-aspect LDA model to cluster sentences into aspects. We then use extended LexRank algorithm to rank the sentences in each cluster. We use Integer Linear Programming for sentence selection. Key features of our method include automatic grouping of semantically related sentences and sentence ranking based on extension of random walk model. Also, we implement a new sentence compression algorithm which use dependency tree instead of parser tree. We compare our method with four baseline methods. Quantitative evaluation based on Rouge metric demonstrates the effectiveness and advantages of our method.

1 Introduction

In recent years, there has been much interest in the task of multi-document summarization. In this paper, we study the task of automatically generating aspect-oriented summaries from multiple documents. The goal of aspect-oriented summarization is to present the most important content to the user in a condensed form and a well-organized structure to satisfy the user's needs. A summary should follow a readable structure and cover all the aspects users are interested in. For example, a summary about natural disasters should include aspects about what happened, when/where it happened, reasons, damages, rescue efforts, etc. and these aspects may be scattered in multiple articles written by different news agencies. Our goal is to automatically collect aspects and construct summaries from multiple documents.

Aspect-oriented summarization can be used in many scenarios. First of all, it can be used to generate Wikipedia-like summary articles, especially used to generate introduction sections that summarizes the subject of articles before the table of contents and other elaborate sections. Second, opinionated text often contains multiple viewpoints about an issue generated by different people. Summarizing these multiple opinions can help people easily digest them. Furthermore, combined with search engines and question&answering systems, we can better organize the summary content based on aspects to improve user experience.

Despite its usefulness, the problem of modeling domain specific aspects for multi-document summarization has not been well studied. The most relevant work is by (Haghighi and Vanderwende, 2009) on exploring content models for multi-document summarization. They proposed a HIERSUM model for finding the subtopics or aspects which are combined by using KL-divergence criterion for selecting relevant sentences. They introduced a general content distribution and several specific content distributions to discover the topic and aspects for a single document collection. However, the aspects may be shared not only across documents in a single collection, but also across documents in different topic-related collections. Their model is conceptually inadequate for simultaneously summarizing multiple topic-related document collections. Furthermore, their sentence selection method based on KL-divergence cannot prevent redundancy across different aspects.

In this paper, we study how to overcome these

Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pages 1137–1146,
Edinburgh, Scotland, UK, July 27–31, 2011.

limitations. We hypothesize that comparatively summarizing topics across similar collections can improve the effectiveness of aspect-oriented multi-document summarization. We propose a novel extraction-based approach which consists of four main steps listed below:

Sentence Clustering: Our goal in this step is to automatically identify the different aspects and cluster sentences into aspects (See Section 2). We substantially extend the entity-aspect model in (Li et al., 2010) for generating general sentence clusters.

Sentence Ranking: In this step, we use an extension of LexRank algorithm proposed by (Paul et al., 2010) to score representative sentences in each cluster (See Section 3).

Sentence Compression: In this step, we aim to improve the linguistic quality of the summaries by simplifying the sentence expressions. We prune sentences using grammatical relations defined on dependency trees for recognizing important clauses and removing redundant subtrees (See Section 4).

Sentence Selection: Finally, we select one compressed version of the sentences from each aspect cluster. We use Integer Linear Programming (ILP) algorithm, which optimizes a *global* objective function, for sentence selection (McDonald, 2007; Gillick and Favre, 2009; Sauper and Barzilay, 2009) (See Section 5).

We evaluate our method using TAC2010 Guided Summarization task data sets[1] (Section 6). Our evaluation shows that our method obtains better ROUGE recall score compared with four baseline methods, and it also achieve reasonably high-quality aspect clusters in terms of purity.

2 Sentence Clustering

In this step, our goal is to discover event aspects contained in a document set and cluster sentences into aspects. Here we substantially extend the entity-aspect model in Li et al. (2010) and refer to it as event-aspect model. The main difference between our event-aspect model and entity-aspect model is that we introduce an additional layer of event topics and the separation of general and specific aspects.

Our extension is based upon the following observations. For example, specific events like "Columbine Massacre" and "Malaysia Resort Abduction" can be related to the "Attack" topic. Each event consists of multiple articles written by different news agencies. Interesting aspects may include "what happened, when, where, perpetrators, reasons, who affected, damages and countermeasures," etc[2]. We compared the "Columbine Massacre" and "Malaysia Resort Abduction" data sets and found 5 different kinds of words in the text: (1) stop words that occur frequently in any document collection; (2) general content words describing "damages" or "countermeasures" aspect of attacks; (3) specific content words describing "what happened", "who affected" or "where" aspect of the concrete event; (4) background words describing the general topic of "Attack"; (5) words that are local to a single document and do not appear across different documents. Table 1 shows four sentences related to two major aspects. We found that the entity-aspect model does not have enough capacity to cluster sentences into aspects (See Section 6). So we introduce additional layer to improve the effectiveness of sentence clustering. We also found that their one aspect per sentence assumption is not very strong in this scenario. Although a sentence may belong to a single general aspect, it still contains multiple specific aspect words like second sentence in Table 1. Therefore, We assume that each sentence belongs to both a general aspect and a specific aspect.

2.1 Event-Aspect Model

Stop words can be ignored by LDA model because they can be easily identified using a standard stop word list. Suppose that for a given event topic, there are in total C specific events for which we need to simultaneously generate summaries. We can assume four kinds of unigram language models (i.e. multinomial word distributions). For each event topic, there is a background model $\phi^{\mathcal{B}}$ that generates words commonly used in all documents, and there are A^G general aspect models ϕ^{ga} ($1 \leq ga \leq A^G$), where A^G is the number of general aspects. For each specific event in a topic, there are A^S specific aspect

[1] http://www.nist.gov/tac/2010/Summarization/

[2] http://www.nist.gov/tac/2010/Summarization/Guided-Summ.2010.guidelines.html

countermeasures
Police/GA are/S close/B to/S identifying/GA someone/B responsible/GA for/S the/S attack/B .
Investigators/GA do/S not/S know/B how/S many/S suspects/SA they/S are/S looking/B for/S, but/S reported/B progress/B toward/S identifying/GA one/S of/S the/S bombers/SA .
what happened, when, where
During/S the/S morning/SA rush/D hour/D on/S July/SA 7/SA terrorists/B exploded/SA bombs/SA on/S three/D London/SA subway/D trains/SA and/S a/S double-decker/D bus/SA .
Four/D coordinated/B bombings/SA struck/B central/B London/SA on/SA July/SA 7/SA, three/D in/S subway/D cars/SA and/S one/D on/S a/S bus/SA .

Table 1: Four sentences on "COUNTERMEASURES" and "What, When, Where" aspects from the "Attack" topic. S: stop word. B: background word. GA: general aspect word. SA: specific aspect word. D: document word.

models ϕ^{sa} ($1 \leq sa \leq A^S$), where A^S is the number of specific aspects, and also there are D document models ϕ^d ($1 \leq d \leq D$), where D is the number of documents in this collection. We assume that these word distributions have a uniform Dirichlet prior with parameter β.

We introduce a level distribution σ that controls whether we choose a word from ϕ^{ga} or ϕ^{sa}. σ is sampled from $Beta(\delta_0, \delta_1)$ distribution. We also introduce an aspect distribution θ that controls how often a general or a specific aspect occurs in the collection, where θ is sampled from another Dirichlet prior with parameter α. There is also a multinomial distribution π that controls in each sentence how often we encounter a background word, a document word, or an aspect word. π has a Dirichlet prior with parameter γ.

Let S_d denote the number of sentences in document d, $N_{d,s}$ denote the number of words (after stop word removal) in sentence s of document d, and $w_{d,s,n}$ denote the n'th word in this sentence. We introduce hidden variables $z^{ga}_{d,s}$ and $z^{sa}_{d,s}$ to indicate that a sentence s of document d belongs to which general or specific aspects . We introduce hidden variables $y_{d,s,n}$ for each word to indicate whether a word is generated from the background model, the document model, or the aspect model. We also introduce hidden variables $l_{d,s,n}$ to indicate whether the n'th word in sentence s of document d is generated from the general aspect model. Figure 1 describes the process of generating the whole document collection. The plate notation of the model is shown in Figure 2. Note that the values of δ_0, δ_1, α_1, α_2, β and γ are fixed. The number of general and specific aspects A^G and A^S are also empirically set.

Given a document collection, i.e. the set of all $w_{d,s,n}$, our goal is to find the most likely assignment of $z^{ga}_{d,s}$, $z^{sa}_{d,s}$, $y_{d,s,n}$ and $l_{d,s,n}$ that maximizes distribution $p(\boldsymbol{z}, \boldsymbol{y}, \boldsymbol{l}|\boldsymbol{w}; \alpha, \beta, \gamma, \delta)$, where $\boldsymbol{z}$, $\boldsymbol{y}$, $\boldsymbol{l}$ and $\boldsymbol{w}$ represent the set of all z, y, l and w variables, respectively. With the assignment, sentences are naturally clustered into aspects, and words are labeled as either a background word, a document word, a general aspect word or a specific aspect word.

Inference can be done with Gibbs sampling, which is commonly used in LDA models (Griffiths and Steyvers, 2004).

In our experiments, we set $\alpha_1 = 5$, $\alpha_2 = 3$, $\beta = 0.01$, $\gamma = 20$, $\delta_1 = 10$ and $\delta_2 = 10$. We run 100 burn-in iterations through all documents in a collection to stabilize the distribution of $\boldsymbol{z}$ and $\boldsymbol{y}$ before collecting samples. We take 10 samples with a gap of 10 iterations between two samples, and average over these 10 samples to get the estimation for the parameters.

After estimating all the distributions, we can find the values of each $z^{ga}_{d,s}$ and $z^{sa}_{d,s}$ that gives us sentences clustered into general and specific aspects.

3 Sentence Ranking

In this step, we want to order the clustered sentences so that the representative sentences can be ranked higher in each aspect. Inspired by Paul et al. (2010), we use an extended LexRank algorithm to obtain top ranked sentences. LexRank (Erkan and Radev, 2004) algorithm defines a random walk mod-

1. Draw $\theta_1 \sim \text{Dir}(\alpha_1), \theta_2 \sim \text{Dir}(\alpha_2), \pi \sim \text{Dir}(\gamma)$ Draw $\sigma \sim \text{Beta}(\delta_0, \delta_1)$
2. For each event topic, there is a background model $\phi^{\mathcal{B}}$, and there are general aspect ga, where $1 \leq ga \leq A^G$
   (a) draw $\phi^{\mathcal{B}} \sim \text{Dir}(\beta)$
   (b) draw $\phi^{ga} \sim \text{Dir}(\beta)$
3. For each document collection, there are specific aspect sa, where $1 \leq sa \leq A^S$
   (a) draw $\phi^{sa} \sim \text{Dir}(\beta)$
4. For each document $d = 1, \ldots, D$,
   (a) draw $\phi^d \sim \text{Dir}(\beta)$
   (b) for each sentence $s = 1, \ldots, S_d$
      i. draw $z^{ga} \sim \text{Multi}(\theta_1)$
      ii. draw $z^{sa} \sim \text{Multi}(\theta_2)$
      iii. for each word $n = 1, \ldots, N_{d,s}$
         A. draw $l_{d,s,n} \sim \text{Binomial}(\sigma)$
         B. draw $y_{d,s,n} \sim \text{Multi}(\pi)$
         C. draw $w_{d,s,n} \sim \text{Multi}(\phi^{\mathcal{B}})$ if $y_{d,s,n} = 1$, $w_{d,s,n} \sim \text{Multi}(\phi^d)$ if $y_{d,s,n} = 2$, $w_{d,s,n} \sim \text{Multi}(\phi^{z^{sa}_{d,s}})$ if $y_{d,s,n} = 3$ and $l_{d,s,n} = 1$ or $w_{d,s,n} \sim \text{Multi}(\phi^{z^{ga}_{d,s}})$ if $y_{d,s,n} = 3$ and $l_{d,s,n} = 0$

Figure 1: The document generation process.

el on top of a graph that represents sentences to be summarized as nodes and their similarities as edges. The LexRank score of a sentence gives the expected probability that a random walk will visit that sentence in the long run. A variant is called continuous LexRank improved LexRank by making use of the strength of the similarity links. The continuous LexRank score can be computed using the following formula:

$$L(u) = \frac{d}{N} + (1-d) \sum_{v \in adj[u]} p(u|v) L(v)$$

where $L(u)$ is the LexRank value of sentence u, N is the total number of nodes in the graph, d is a damping factor for the convergence of the method, and $p(u|v)$ is the jumping probability between sentence u and its neighboring sentence v. $p(u|v)$ is defined using content similarity function $sim(u, v)$ between two sentences:

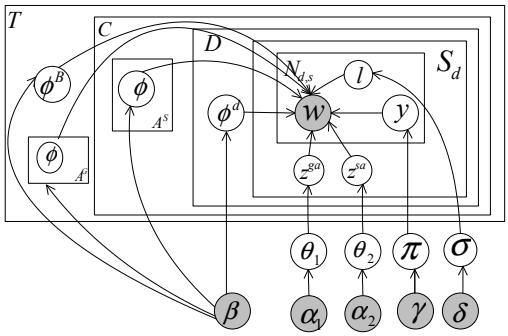

Figure 2: The event-aspect model.

$$p(u|v) = \frac{sim(u, v)}{\sum_{z \in adj[v]} sim(z, v)}$$

The major extension is to modify this jumping probability so as to favor visiting representative sentences. More specifically, we scale $sim(u, v)$ by the likelihood that the two sentences represent the same general aspect ga or specific aspect sa:

$$sim'(u, v) = sim(u, v)[\sum_{ga=1}^{A^G} P(ga|u)P(ga|v) + \sum_{sa=1}^{A^S} P(sa|u)P(sa|v)]$$

where the value $P(ga|u)$ and $P(sa|u)$ can be computed by our event-aspect model. We define $sim(u, v)$ as the $tf * idf$ weighted cosine similarity between two sentences.

We found that sentence ranking is better conducted before the compression because the pre-compressed sentences are more informative and the similarity function in LexRank can be better off with the complete information.

4 Sentence Compression

It has been shown that sentence compression can improve linguistic quality of summaries (Zajic et al., 2007; Gillick et al., 2010). Commonly used "Syntactic parse and trim" approach may produce poor compression results. For example, given the sentence "We have friends whose children go to Columbine, the freshman said", the procedure tries to remove the clause "the freshman said" from the parse tree by using the "SBAR" label to locate the

clause, and will result in "whose children go to Columbine", which is not adequate. Furthermore, some important temporal modifier, numeric modifier and clausal complement need to be retained because they reflect content aspects of the summary. Therefore, we propose the "dependency parse and trim" approach, which prunes sentences based on dependency tree representations, using English grammatical relations to recognize clauses and remove redundant structures. Table 2 shows two examples by removing redundant auxiliary clauses. Below is the sentence compression procedure:

1. Select possible subtree root nodes using grammatical relations, such as clausal complement, complementizer, or parataxis [3].

2. Decide which subtree root node can be the root of clause. If this root contains maximum number of child nodes and the collection of all child edges include object or auxiliary relations, it is selected as the root node.

3. Remove redundant modifiers such as adverbials, relative clause modifiers and abbreviations, participials and infinitive modifiers.

4. Traverse the subtrees and generate all possible compression alternatives using the subtree root node, then keep the top two longest sub sentences.

5. Drop the sub sentences shorter than 5 words.

5 Sentence Selection

After sentence pruning, we prepare for the final event summary generation process. In this step, we select one compressed version of the sentence from each aspect cluster. To avoid redundancy between aspects, we use Integer Linear Programming to optimize a global objective function for sentence selection. Inspired by (Sauper and Barzilay, 2009), we formulate the optimization problem based on sentence ranking information. More specifically, we would like to select exactly one compressed sentence which receives the highest possible ranking score from each aspect cluster subject to a series of constraints, such as redundancy and length. We employed lp_solver [4], an efficient mixed integer programming solver using the Branch-and-Bound algorithm to select sentences.

Original	**Compressed**
When rescue workers arrived, they said, only one of his limbs was visible.	When rescue workers arrived, only one of his limbs was visible.
Two days earlier, a massacre by two students at Columbine High, whose teams are called the Rebels, left 15 people dead and dozens wounded.	Two days earlier, a massacre by two students at Columbine High, left 15 people dead and dozens wounded.

Table 2: Example compressed sentences.

Assume that there are in total K aspects in an event topic. For each aspect j, there are in total R ranked sentences. The variables S_{jl} is a binary indicator of the sentence. That is, S_{jl}= 1 if the sentence is included in the final summary, and S_{jl} = 0 otherwise. l is the ranked position of the sentence in this aspect cluster.

Objective Function

Top ranked sentences are the most relevant corresponding to the related aspects which we want to include in the final summary. Thus we try to minimize the ranks of the sentences to improve the overall responsiveness.

$$\min(\sum_{j=1}^{K}\sum_{l=1}^{R_j} l \cdot S_{jl})$$

Exclusivity Constraints

To prevent redundancy in each aspect, we just choose one sentence from each general or specific aspect cluster. The constraint is formulated as follows:

$$\sum_{l=1}^{R_j} S_{jl} = 1 \qquad \forall j \in \{1 \dots K\}$$

[3] The parataxis relation is a relation between the main verb of a clause and other sentential elements, such as a sentential parenthetical, colon, or semicolon

[4] http://lpsolve.sourceforge.net/5.5/

Redundancy Constraints

We also want to prevent redundancy across different aspects. If sentence-similarity $sim(s_{jl}, s_{j'l'})$ between sentence s_{jl} and $s_{j'l'}$ is above 0.5, then we drop the pair and choose one sentence ranked higher from the pair otherwise. This constraint is formulated as follows:

$$(S_{jl} + S_{j'l'}) \cdot sim(s_{jl}, s_{j'l'}) \leq 0.5$$
$$\forall j, j' \in \{1 \dots K\} \forall l \in \{1 \dots R_j\} \forall l' \in \{1 \dots R_{j'}\}$$

Length Constraints

We add this constraint to ensure that the length of the final summary is limited to L words.

$$\sum_{j=1}^{K} \sum_{l=1}^{R_j} len_{jl} \cdot S_{jl} \leq L$$

where len_{jl} is the length of S_{jl}.

6 Evaluation

In order to systematically evaluate our method, we want to check (1) whether the whole system is effective, which means to quantitatively evaluate summary quality, and (2) whether individual components like clustering and compression algorithms are useful.

6.1 Data

We use TAC2010 Summarization task data set for the summary content evaluation. This data set provides 46 events. Each event falls into a predefined event topic. Each specific event includes an event statement and 20 relevant newswire articles which have been divided into 2 sets: Document Set A and Document Set B. Each document set has 10 documents, and all the documents in Set A chronologically precede the documents in Set B. We just use document Set A for our task. Assessors wrote model summaries for each event, so we can compare our automatic generated summaries with the model summaries. We combine topic related data sets together, then these data sets simultaneously annotated by our Event-aspect model. After labeling process, we run sentence ranking, compression and selection module to get final aspect-oriented summarizations.

6.2 Quality of summary

We use the ROUGE (Lin and Hovy, 2003) metric for measuring the summarization system performance. Ideally, a summarization criterion should be more recall oriented. So the average recall of ROUGE-1, ROUGE-2, ROUGE-SU4, ROUGE-W-1.2 and ROUGE-L were computed by running ROUGE-1.5.5 with stemming but no removal of stop words. We compare our method with the following four baseline methods.

Baseline 1

In this baseline, we try to compare different sentence clustering algorithms in the multi-document summarization scenario. First, we use CLUTO [5] to do K-means clustering. Then we try entity-aspect model proposed by Li et al. (2010) to do sentence clustering. Entity-aspect model is similar with "HIERSUM" content model proposed by Haghighi and Vanderwende (2009). We use the same ranking, compression, and selection components to generate aspect-oriented summaries for comparison.

Baseline 2

In this baseline, we compare our method with traditional ranking and selection summary generation framework (Erkan and Radev, 2004; Nenkova and Vanderwende, 2005) to show that our sentence clustering component is necessary in aspect-oriented summarization system. Also we want check whether sentence ranking combined with greedy based sentence selection can prevent redundancy effectively. We follow LexRank based sentence ranking combined with greedy sentence selection methods. We implement two greedy algorithms (Zhang et al., 2008; Paul et al., 2010). One is to select the top ranked sentence simultaneously by removing 10 redundant neighbor sentences from the sentence similarity graph if the summary length is less then 100 words. This is repeated until the graph cannot be partitioned. The similarity graph building threshold is 0.3, damping factor is 0.2 and error tolerance for Power Method in LexRank is 0.1. The other is to select top ranked sentences as long as the redundancy score (similarity) between a candidate sentence and

[5] http://glaros.dtc.umn.edu/gkhome/cluto/cluto/overview

current summary is under 0.5. This is repeated until the summary reaches a 100 word length limit.

Baseline 3

In this baseline, we compare our ILP based sentence selection with KL-divergence based sentence selection. The KL-divergence formula we use is below,

$$KL(P_S||Q_D) = \sum_w P(w) \log \frac{P(w)}{Q(w)}$$

where $P(S)$ is the empirical unigram distribution of the candidate summary S, and $Q(D)$ is the unigram distribution of document collection D. We only replaced our selection method with the KL-divergence selection method. Other parts are the same. After ranking sentences for each aspect, we add the sentence with the highest ranking score from each aspect sentence cluster as long as the KL-divergence between candidate and current summary does not decrease. This is repeated until the summary reaches a 100 word length limit. To our knowledge, this is the first work to directly compare Integer Linear Programming based sentence selection with KL-divergence based sentence selection in summarization generation framework.

Baseline 4

In this baseline, we directly compare our method with "HIERSUM" proposed by (Haghighi and Vanderwende, 2009). As in Baseline 1, we use entity-aspect model to approximate "HIERSUM" model. We replace unigram distribution of $P(w)$ in KL-divergence with learned distribution estimated by "HIERSUM" model. The KL-divergence based greedy sentence selection algorithm is similar to Baseline 3.

For fair comparison, Baselines 1, 2, 3 and 4 use the same sentence compression algorithm and have the summary length no more then 100 words. In Table 3, we show the average ROUGE recall of 46 summaries generated by our method and four baseline methods. We can see that our method gives better Rouge recall measures then the four baseline methods. For BL-1, we can see that LDA-based sentence clustering is better then k-means. For BL-2, we can see that traditional ranking plus greedy selection summary generation framework is not suitable for the aspect-oriented summarization task. More specifically, greedy-based sentence selection can not prevent redundancy effectively. BL-3 evaluation results showed that ILP-based sentence selection is better then KL-divergence selection in terms of preventing redundancy across different aspects. The measurement performance between BL-3 and BL-4 is close. They use the same KL-divergence based sentence selection, but topic model they use are different, and also BL-3 has a sentence ranking process. The Rouge recall of our method is better than BL-4. It is expected because our event-aspect model can better find the aspects and also prove that our LexRank based sentence ranking combined with ILP-based sentence selection can prevent redundancy.

Due to TAC2010 summarization community just compute ROUGE-2 and ROUGE-SU4 metrics for participants, our ROUGE-2 metric ranked 11 out of 23, ROUGE-SU4 metric ranked 12 out of 23. They use MEAD[6] as their baseline approach. The ROUGE-2 score of our approach achieve 0.06508 higher than MEAD's 0.05929. The ROUGE-SU4 score of our approach achieve 0.10146 higher than MEAD's 0.09112. Many systems that get higher performances leverage domain knowledge bases like Wikipedia or training data, but we didn't. The advantage of our method is that we generate summaries with totally unsupervised framework and this approach is domain adaptive.

6.3 Quality of aspect-oriented sentence clusters

To judge the quality of the aspect-oriented sentence clusters, we ask the human judges to group the ground truth sentences based on the aspect relatedness in each event topic. We then compute the purity of the automatically generated clusters against the human judged clusters. The results are shown in Table 4. In our experiments, we set the number of general aspect clusters A^G is 5 and specific aspect clusters A^S is 3. We can see from Table 4 that our generated aspect clusters can achieve reasonably good performance.

[6] http://www.summarization.com/mead/

Method		Rouge Average Recall				
		ROUGE-1	ROUGE-2	ROUGE-SU4	ROUGE-W-1.2	ROUGE-L
BL-1	k-means	0.21895	0.03689	0.06644	0.06683	0.19208
	entity-aspect	0.26082	0.05082	0.08286	0.08055	0.22976
BL-2	greedy 1	0.27802	0.04872	0.08302	0.08488	0.24426
	greedy 2	0.27898	0.04723	0.08275	0.08500	0.24430
BL-3	KL-Div	0.29286	0.05369	0.09117	0.08827	0.25100
BL-4	HIERSUM	0.28736	0.05502	0.08932	0.08923	0.25285
Without compression		0.30563	0.05983	0.09513	0.09468	0.25487
Our Method		0.32641	0.06508	0.10146	0.09998	0.28610

Table 3: ROUGE evaluation results on TAC2010 Summarization data sets

Category	A	Purity
Accidents and Natural Disasters	7	0.613
Attacks	8	0.658
Health and Safety	5	0.724
Endangered Resources	4	0.716
Investigations and Trials	6	0.669

Table 4: The true numbers of aspects as judged by the human annotator (A), and the purity of the clusters.

Category	Average Score
Accidents and Natural Disasters	2.4
Attacks	2.3
Health and Safety	2.6
Endangered Resources	2.5
Investigations and Trials	2.4

Table 5: The average score of each event topic.

6.4 Quality of sentence compression

To judge the quality of the dependency tree based sentence compression algorithm, we ask the human judges to choose 20 sentences from each event topic then score them. The judges follow 3-point scale to score each compressed sentence: 1 means poor, 2 means barely acceptable, and 3 means good. We then compute the average scores. The results are shown in Table 5. To evaluate the effectiveness of sentence compression component, we conduct the system without sentence compression component, then compare it with our system. In Table 3, we can see that sentence compression can improve the system performance.

7 Related Work

Our event-aspect model is related to a number of previous extensions of LDA models. Chemudugunta et al. (2007) proposed to introduce a background topic and document-specific topics. Our background and document language models are similar to theirs. However, they still treat documents as bags of words rather then sets of sentences as in our models. Titov and McDonald (2008) exploited the idea that a short paragraph within a document is likely to be about the same aspect. The way we separate words into stop words, background words, document words and aspect words bears similarity to that used in (Daumé III and Marcu, 2006; Haghighi and Vanderwende, 2009). Paul and Girju (2010) proposed a topic-aspect model for simultaneously finding topics and aspects. The most related extension is entity-aspect model proposed by Li et al. (2010). The main difference between event-aspect model and entity-aspect model is our model further consider aspect granularity and add a layer to model topic-related events.

Filippova and Strube (2008) proposed a dependency tree based sentence compression algorithm. Their approach need a large corpus to build language model for compression, whereas we prune dependency tree using grammatical rules.

Paul et al. (2010) proposed to modify LexRank algorithm using their topic-aspect model. But their task is to summarize contrastive viewpoints in opinionated text. Furthermore, they use a simple greedy approach for constructing summary.

McDonald (2007) proposed to use Integer Linear Programming framework in multi-document sum-

marization. And Sauper and Barzilay (2009) use integer linear programming framework to automatically generate Wikipedia articles. There is a fundamental difference between their method and ours. They used trained perceptron algorithm for ranking excerpts, whereas we give an extended LexRank with integer linear programming to optimize sentence selection for our aspect-oriented multi-document summarization.

8 Conclusions and Future Work

In this paper, we study the task of automatically generating aspect-oriented summary from multiple documents. We proposed an event-aspect model that can automatically cluster sentences into aspects. We then use an extension of the LexRank algorithm to rank sentences. We took advantage of the output generated by the event-aspect model to modify jumping probabilities so as to favor visiting representative sentence. We also proposed dependency tree compression algorithm to prune sentence for improving linguistic quality of the summaries. Finally we use Integer Linear Programming Framework to select aspect relevant sentences. We conducted quantitative evaluation using standard test data sets. We found that our method gave overall better ROUGE scores than four baseline methods, and the new sentence clustering and compression algorithm are robust.

There are a number of directions we plan to pursue in the future in order to improve our method. First, we can possibly apply more linguistic knowledge to improve the quality of sentence compression. Currently the sentence compression algorithm may generate meaningless subtrees. It is relatively hard to decide which clause is redundant in terms of summarization. Second, we may explore more domain knowledge to improve the quality of aspect-oriented summaries. For example, we know that the "who-affected" aspect is related to person, and "when, where" are related to Time and Location. we can import Name Entity Recognition to annotate these phrases and then help locate relevant sentences. Third, we want to extend our event-aspect model to simultaneously find topics and aspects.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (NSFC No. 60773088), the National High-tech R&D Program of China (863 Program No. 2009AA04Z106), and the Key Program of Basic Research of Shanghai Municipal S&T Commission (No. 08JC1411700).

References

Chaitanya Chemudugunta, Padhraic Smyth, and Mark Steyvers. 2007. Modeling general and specific aspects of documents with a probabilistic topic model. In *Advances in Neural Information Processing Systems 19*, pages 241–248.

Hal. Daumé III and Daniel. Marcu. 2006. Bayesian query-focused summarization. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 305–312. Association for Computational Linguistics.

Günes. Erkan and Dragomir Radev. 2004. LexRank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22(1):457–479.

K. Filippova and M. Strube. 2008. Dependency tree based sentence compression. In *Proceedings of the Fifth International Natural Language Generation Conference*, pages 25–32. Association for Computational Linguistics.

Dan Gillick and Benoit Favre. 2009. A scalable global model for summarization. In *Proceedings of the Workshop on Integer Linear Programming for Natural Langauge Processing*, pages 10–18.

Dan Gillick, Benoit Favre, D. Hakkani-Tur, B. Bohnet, Y. Liu, and S. Xie. 2010. The icsi/utd summarization system at tac 2009. In *Proceedings of the Second Text Analysis Conference, Gaithersburg, Maryland, USA: National Institute of Standards and Technology*.

Thomas L. Griffiths and Mark Steyvers. 2004. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101(Suppl. 1):5228–5235.

A. Haghighi and L. Vanderwende. 2009. Exploring content models for multi-document summarization. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics on ZZZ*, pages 362–370. Association for Computational Linguistics.

Peng Li, Jing Jiang, and Yinglin Wang. 2010. Generating templates of entity summaries with an entity-aspect model and pattern mining. In *Proceedings of the Joint Conference of the 48th Annual Meeting of the ACL*. Association for Computational Linguistics.

C.Y. Lin and E. Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 71–78. Association for Computational Linguistics.

Ryan McDonald. 2007. A study of global inference algorithms in multi-document summarization. *Advances in Information Retrieval*, pages 557–564.

A. Nenkova and L. Vanderwende. 2005. The impact of frequency on summarization. *Microsoft Research, Redmond, Washington, Tech. Rep. MSR-TR-2005-101*.

Michael J. Paul and Roxana Girju. 2010. A two-dimensional topic-aspect model for discovering multi-faceted topics. In *In AAAI-2010: Twenty-Fourth Conference on Artificial Intelligence*.

Michael J. Paul, ChengXiang Zhai, and Roxana Girju. 2010. Summarizing contrastive viewpoints in opinionated text. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 66–76, Morristown, NJ, USA. Association for Computational Linguistics.

Christina Sauper and Regina Barzilay. 2009. Automatically generating wikipedia articles: A structure-aware approach. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 208–216, Suntec, Singapore, August. Association for Computational Linguistics.

Ivan Titov and Ryan McDonald. 2008. Modeling online reviews with multi-grain topic models. In *Proceeding of the 17th International Conference on World Wide Web*, pages 111–120.

D. Zajic, B.J. Dorr, J. Lin, and R. Schwartz. 2007. Multi-candidate reduction: Sentence compression as a tool for document summarization tasks. *Information Processing & Management*, 43(6):1549–1570.

Jin. Zhang, Xueqi. Cheng, and Hongbo. Xu. 2008. GSP-Summary: a graph-based sub-topic partition algorithm for summarization. In *Proceedings of the 4th Asia information retrieval conference on Information retrieval technology*, pages 321–334. Springer-Verlag.