Generating Templates of Entity Summaries with an Entity-Aspect Model and Pattern Mining

Peng Li[1] and **Jing Jiang**[2] and **Yinglin Wang**[1]
[1]Department of Computer Science and Engineering, Shanghai Jiao Tong University
[2]School of Information Systems, Singapore Management University
{lipeng,ylwang}@sjtu.edu.cn jingjiang@smu.edu.sg

Abstract

In this paper, we propose a novel approach to automatic generation of summary templates from given collections of summary articles. This kind of summary templates can be useful in various applications. We first develop an entity-aspect LDA model to simultaneously cluster both sentences and words into aspects. We then apply frequent subtree pattern mining on the dependency parse trees of the clustered and labeled sentences to discover sentence patterns that well represent the aspects. Key features of our method include automatic grouping of semantically related sentence patterns and automatic identification of template slots that need to be filled in. We apply our method on five Wikipedia entity categories and compare our method with two baseline methods. Both quantitative evaluation based on human judgment and qualitative comparison demonstrate the effectiveness and advantages of our method.

1 Introduction

In this paper, we study the task of automatically generating templates for entity summaries. An entity summary is a short document that gives the most important facts about an entity. In Wikipedia, for instance, most articles have an introduction section that summarizes the subject entity before the table of contents and other elaborate sections. These introduction sections are examples of entity summaries we consider. Summaries of entities from the same category usually share some common structure. For example, biographies of physicists usually contain facts about the nationality, educational background, affiliation and major contributions of the physicist, whereas introductions of companies usually list information such as the industry, founder and headquarter of the company. Our goal is to automatically construct a summary template that outlines the most salient types of facts for an entity category, given a collection of entity summaries from this category.

Such kind of summary templates can be very useful in many applications. First of all, they can uncover the underlying structures of summary articles and help better organize the information units, much in the same way as infoboxes do in Wikipedia. In fact, automatic template generation provides a solution to induction of infobox structures, which are still highly incomplete in Wikipedia (Wu and Weld, 2007). A template can also serve as a starting point for human editors to create new summary articles. Furthermore, with summary templates, we can potentially apply information retrieval and extraction techniques to construct summaries for new entities automatically on the fly, improving the user experience for search engine and question answering systems.

Despite its usefulness, the problem has not been well studied. The most relevant work is by Filatova et al. (2006) on automatic creation of domain templates, where the defintion of a domain is similar to our notion of an entity category. Filatova et al. (2006) first identify the important verbs for a domain using corpus statistics, and then find frequent parse tree patterns from sentences containing these verbs to construct a domain template. There are two major limitations of their approach. First, the focus on verbs restricts the template patterns that can be found. Second, redundant or related patterns using different verbs to express the same or similar facts cannot be grouped together. For example, "*won X award*" and "*received X prize*" are considered two different patterns by this approach. We propose a method that can overcome these two limitations. Automatic template generation is also related to a number of other problems that have been studied before, in-

Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pages 640–649,
Uppsala, Sweden, 11-16 July 2010.

cluding unsupervised IE pattern discovery (Sudo et al., 2003; Shinyama and Sekine, 2006; Sekine, 2006; Yan et al., 2009) and automatic generation of Wikipedia articles (Sauper and Barzilay, 2009). We discuss the differences of our work from existing related work in Section 6.

In this paper we propose a novel approach to the task of automatically generating entity summary templates. We first develop an entity-aspect model that extends standard LDA to identify clusters of words that can represent different aspects of facts that are salient in a given summary collection (Section 3). For example, the words "received," "award," "won" and "Nobel" may be clustered together from biographies of physicists to represent one aspect, even though they may appear in different sentences from different biographies. Simultaneously, the entity-aspect model separates words in each sentence into background words, document words and aspect words, and sentences likely about the same aspect are naturally clustered together. After this aspect identification step, we mine frequent subtree patterns from the dependency parse trees of the clustered sentences (Section 4). Different from previous work, we leverage the word labels assigned by the entity-aspect model to prune the patterns and to locate template slots to be filled in.

We evaluate our method on five entity categories using Wikipedia articles (Section 5). Because the task is new and thus there is no standard evaluation criteria, we conduct both quantitative evaluation using our own human judgment and qualitative comparison. Our evaluation shows that our method can obtain better sentence patterns in terms of f1 measure compared with two baseline methods, and it can also achieve reasonably good quality of aspect clusters in terms of purity. Compared with standard LDA and K-means sentence clustering, the aspects identified by our method are also more meaningful.

2 The Task

Given a collection of entity summaries from the same entity category, our task is to automatically construct a summary template that outlines the most important information one should include in a summary for this entity category. For example, given a collection of biographies of physicists, ideally the summary template should indicate that important facts about a physicist include his/her educational background, affiliation, major contributions, awards received, etc.

Aspect	Pattern
1	*ENT* received his phd from ? university *ENT* studied ? under ? *ENT* earned his ? in physics from university of ?
2	*ENT* was awarded the medal in ? *ENT* won the ? award *ENT* received the nobel prize in physics in ?
3	*ENT* was ? director *ENT* was the head of ? *ENT* worked for ?
4	*ENT* made contributions to ? *ENT* is best known for work on ? *ENT* is noted for ?

Table 1: Examples of some good template patterns and their aspects generated by our method.

However, it is not clear what is the best representation of such templates. Should a template comprise a list of subtopic labels (e.g. "education" and "affiliation") or a set of explicit questions? Here we define a template format based on the usage of the templates as well as our observations from Wikipedia entity summaries. First, since we expect that the templates can be used by human editors for creating new summaries, we use sentence patterns that are human readable as basic units of the templates. For example, we may have a sentence pattern "*ENT* graduated from ? University" for the entity category "physicist," where *ENT* is a placeholder for the entity that the summary is about, and '?' is a slot to be filled in. Second, we observe that information about entities of the same category can be grouped into subtopics. For example, the sentences "Bohr is a Nobel laureate" and "Einstein received the Nobel Prize" are paraphrases of the same type of facts, while the sentences "Taub earned his doctorate at Princeton University" and "he graduated from MIT" are slightly different but both describe a person's educational background. Therefore, it makes sense to group sentence patterns based on the subtopics they pertain to. Here we call these subtopics the *aspects* of a summary template.

Formally, we define a summary template to be a set of sentence patterns grouped into aspects. Each sentence pattern has a placeholder for the entity to be summarized and possibly one or more template slots to be filled in. Table 1 shows some sentence patterns our method has generated for the "physicist" category.

2.1 Overview of Our Method

Our automatic template generation method consists of two steps:

Aspect Identification: In this step, our goal is to automatically identify the different aspects or subtopics of the given summary collection. We simultaneously cluster sentences and words into aspects, using an entity-aspect model extended from the standard LDA model that is widely used in text mining (Blei et al., 2003). The output of this step are sentences clustered into aspects, with each word labeled as a stop word, a background word, a document word or an aspect word.

Sentence Pattern Generation: In this step, we generate human-readable sentence patterns to represent each aspect. We use frequent subtree pattern mining to find the most representative sentence structures for each aspect. The fixed structure of a sentence pattern consists of aspect words, background words and stop words, while document words become template slots whose values can vary from summary to summary.

3 Aspect Identification

At the aspect identification step, our goal is to discover the most salient aspects or subtopics contained in a summary collection. Here we propose a principled method based on a modified LDA model to simultaneously cluster both sentences and words to discover aspects.

We first make the following observation. In entity summaries such as the introduction sections of Wikipedia articles, most sentences are talking about a single fact of the entity. If we look closely, there are a few different kinds of words in these sentences. First of all, there are stop words that occur frequently in any document collection. Second, for a given entity category, some words are generally used in all aspects of the collection. Third, some words are clearly associated with the aspects of the sentences they occur in. And finally, there are also words that are document or entity specific. For example, in Table 2 we show two sentences related to the "affiliation" aspect from the "physicist" summary collection. Stop words such as "is" and "the" are labeled with "S." The word "physics" can be regarded as a background word for this collection. "Professor" and "university" are clearly related to the "affiliation" aspect. Finally words such as "Modena" and "Chicago" are specifically associated with the subject entities being discussed, that is, they are specific to the summary documents.

To capture background words and document-specific words, Chemudugunta et al. (2007) proposed to introduce a background topic and document-specific topics. Here we borrow their idea and also include a background topic as well as document-specific topics. To discover aspects that are local to one or a few adjacent sentences but may occur in many documents, Titov and McDonald (2008) proposed a multi-grain topic model, which relies on word co-occurrences within short paragraphs rather than documents in order to discover aspects. Inspired by their model, we rely on word co-occurrences within single sentences to identify aspects.

3.1 Entity-Aspect Model

We now formally present our entity-aspect model. First, we assume that stop words can be identified using a standard stop word list. We then assume that for a given entity category there are three kinds of unigram language models (i.e. multinomial word distributions). There is a background model $\phi^{\mathcal{B}}$ that generates words commonly used in all documents and all aspects. There are D document models ψ^d ($1 \leq d \leq D$), where D is the number of documents in the given summary collection, and there are A aspect models ϕ^a ($1 \leq a \leq A$), where A is the number of aspects. We assume that these word distributions have a uniform Dirichlet prior with parameter β.

Since not all aspects are discussed equally frequently, we assume that there is a global aspect distribution θ that controls how often each aspect occurs in the collection. 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 $z_{d,s}$ for each sentence to indicate the aspect a sentence belongs to. We also 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. Figure 1 shows the process of

```
    Venturi/D is/S a/S professor/A of/S physics/B at/S the/S University/A of/S
Modena/D ./S
    He/S was/S a/S professor/A of/S physics/B at/S the/S University/A of/S
Chicago/D until/S 1982/D ./S
```

Table 2: Two sentences on "affiliation" from the "physicist" entity category. S: stop word. B: background word. A: aspect word. D: document word.

1. Draw $\theta \sim \text{Dir}(\alpha), \phi^{\mathcal{B}} \sim \text{Dir}(\beta), \pi \sim \text{Dir}(\gamma)$
2. For each aspect $a = 1, \ldots, A$,
   (a) draw $\phi^a \sim \text{Dir}(\beta)$
3. For each document $d = 1, \ldots, D$,
   (a) draw $\psi^d \sim \text{Dir}(\beta)$
   (b) for each sentence $s = 1, \ldots, S_d$
      i. draw $z_{d,s} \sim \text{Multi}(\theta)$
      ii. for each word $n = 1, \ldots, N_{d,s}$
         A. draw $y_{d,s,n} \sim \text{Multi}(\pi)$
         B. draw $w_{d,s,n} \sim \text{Multi}(\phi^{\mathcal{B}})$ if $y_{d,s,n} = 1$, $w_{d,s,n} \sim \text{Multi}(\psi^d)$ if $y_{d,s,n} = 2$, or $w_{d,s,n} \sim \text{Multi}(\phi^{z_{d,s}})$ if $y_{d,s,n} = 3$

Figure 1: The document generation process.

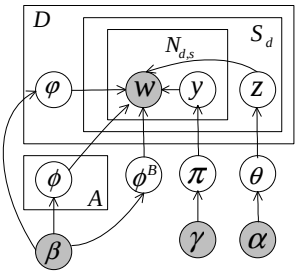

Figure 2: The entity-aspect model.

generating the whole document collection. The plate notation of the model is shown in Figure 2. Note that the values of α, β and γ are fixed. The number of aspects A is also manually set.

3.2 Inference

Given a summary collection, i.e. the set of all $w_{d,s,n}$, our goal is to find the most likely assignment of $z_{d,s}$ and $y_{d,s,n}$, that is, the assignment that maximizes $p(\boldsymbol{z}, \boldsymbol{y}|\boldsymbol{w}; \alpha, \beta, \gamma)$, where $\boldsymbol{z}$, $\boldsymbol{y}$ and $\boldsymbol{w}$ represent the set of all z, y 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, or an aspect word.

We approximate $p(\boldsymbol{y}, \boldsymbol{z}|\boldsymbol{w}; \alpha, \beta, \gamma)$ by $p(\boldsymbol{y}, \boldsymbol{z}|\boldsymbol{w}; \hat{\phi}^{\mathcal{B}}, \{\hat{\psi}^d\}_{d=1}^D, \{\hat{\phi}^a\}_{a=1}^A, \hat{\theta}, \hat{\pi})$, where $\hat{\phi}^{\mathcal{B}}$, $\{\hat{\psi}^d\}_{d=1}^D$, $\{\hat{\phi}^a\}_{a=1}^A$, $\hat{\theta}$ and $\hat{\pi}$ are estimated using Gibbs sampling, which is commonly used for inference for LDA models (Griffiths and Steyvers, 2004). Due to space limit, we give the formulas for the Gibbs sampler below without derivation.

First, given sentence s in document d, we sample a value for $z_{d,s}$ given the values of all other z and y variables using the following formula:

$$p(z_{d,s} = a|\boldsymbol{z}_{\neg\{d,s\}}, \boldsymbol{y}, \boldsymbol{w}) \propto \frac{C^{\mathcal{A}}_{(a)} + \alpha}{C^{\mathcal{A}}_{(\cdot)} + A\alpha} \cdot \frac{\prod_{v=1}^{V} \prod_{i=0}^{E_{(v)}} (C^a_{(v)} + i + \beta)}{\prod_{i=0}^{E_{(\cdot)}} (C^a_{(\cdot)} + i + V\beta)}.$$

In the formula above, $\boldsymbol{z}_{\neg\{d,s\}}$ is the current aspect assignment of all sentences excluding the current sentence. $C^{\mathcal{A}}_{(a)}$ is the number of sentences assigned to aspect a, and $C^{\mathcal{A}}_{(\cdot)}$ is the total number of sentences. V is the vocabulary size. $C^a_{(v)}$ is the number of times word v has been assigned to aspect a. $C^a_{(\cdot)}$ is the total number of words assigned to aspect a. All the counts above exclude the current sentence. $E_{(v)}$ is the number of times word v occurs in the current sentence and is assigned to be an aspect word, as indicated by $\boldsymbol{y}$, and $E_{(\cdot)}$ is the total number of words in the current sentence that are assigned to be an aspect word.

We then sample a value for $y_{d,s,n}$ for each word in the current sentence using the following formulas:

$$p(y_{d,s,n} = 1|\boldsymbol{z}, \boldsymbol{y}_{\neg\{d,s,n\}}) \propto \frac{C^{\pi}_{(1)} + \gamma}{C^{\pi}_{(\cdot)} + 3\gamma} \cdot \frac{C^{\mathcal{B}}_{(w_{d,s,n})} + \beta}{C^{\mathcal{B}}_{(\cdot)} + V\beta},$$

$$p(y_{d,s,n} = 2|\boldsymbol{z}, \boldsymbol{y}_{\neg\{d,s,n\}}) \propto \frac{C^{\pi}_{(2)} + \gamma}{C^{\pi}_{(\cdot)} + 3\gamma} \cdot \frac{C^{d}_{(w_{d,s,n})} + \beta}{C^{d}_{(\cdot)} + V\beta},$$

$$p(y_{d,s,n} = 3|\boldsymbol{z}, \boldsymbol{y}_{\neg\{d,s,n\}}) \propto \frac{C^{\pi}_{(3)} + \gamma}{C^{\pi}_{(\cdot)} + 3\gamma} \cdot \frac{C^{a}_{(w_{d,s,n})} + \beta}{C^{a}_{(\cdot)} + V\beta}.$$

In the formulas above, $\boldsymbol{y}_{\neg\{d,s,n\}}$ is the set of all y variables excluding $y_{d,s,n}$. $C^{\pi}_{(1)}$, $C^{\pi}_{(2)}$ and $C^{\pi}_{(3)}$ are the numbers of words assigned to be a background word, a document word, or an aspect word, respectively, and $C^{\pi}_{(\cdot)}$ is the total number of words. $C^{\mathcal{B}}$ and C^d are counters similar to C^a but are for the background model and the document models. In all these counts, the current word is excluded.

With one Gibbs sample, we can make the following estimation:

$$\hat{\phi}_v^B = \frac{C^{\mathcal{B}}_{(v)} + \beta}{C^{\mathcal{B}}_{(\cdot)} + V\beta}, \hat{\psi}_v^d = \frac{C^d_{(v)} + \beta}{C^d_{(\cdot)} + V\beta}, \hat{\phi}_v^a = \frac{C^a_{(v)} + \beta}{C^a_{(\cdot)} + V\beta},$$

$$\hat{\theta}_a = \frac{C^{\mathcal{A}}_{(a)} + \alpha}{C^{\mathcal{A}}_{(\cdot)} + A\alpha}, \hat{\pi}_t = \frac{C^{\pi}_{(t)} + \gamma}{C^{\pi}_{(\cdot)} + 3\gamma} (1 \leq t \leq 3).$$

Here the counts include all sentences and all words.

In our experiments, we set $\alpha = 5$, $\beta = 0.01$ and $\gamma = 20$. 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 found that empirically 100 burn-in iterations were sufficient for our data set. 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 $\hat{\phi}^{\mathcal{B}}$, $\{\hat{\psi}^d\}_{d=1}^D$, $\{\hat{\phi}^a\}_{a=1}^A$, $\hat{\theta}$ and $\hat{\pi}$, we find the values of each $z_{d,s}$ and $y_{d,s,n}$ that maximize $p(\boldsymbol{y}, \boldsymbol{z}|\boldsymbol{w}; \hat{\phi}^{\mathcal{B}}, \{\hat{\psi}^d\}_{d=1}^D, \{\hat{\phi}^a\}_{a=1}^A, \hat{\theta}, \hat{\pi})$. This assignment, together with the standard stop word list we use, gives us sentences clustered into A aspects, where each word is labeled as either a stop word, a background word, a document word or an aspect word.

3.3 Comparison with Other Models

A major difference of our entity-aspect model from standard LDA model is that we assume each sentence belongs to a single aspect while in LDA words in the same sentence can be assigned to different topics. Our one-aspect-per-sentence assumption is important because our goal is to cluster sentences into aspects so that we can mine common sentence patterns for each aspect.

To cluster sentences, we could have used a straightforward solution similar to document clustering, where sentences are represented as feature vectors using the vector space model, and a standard clustering algorithm such as K-means can be applied to group sentences together. However, there are some potential problems with directly applying this typical document clustering method. First, unlike documents, sentences are short, and the number of words in a sentence that imply its aspect is even smaller. Besides, we do not know the aspect-related words in advance. As a result, the cosine similarity between two sentences may not reflect whether they are about the same aspect. We can perform heuristic term weighting, but the method becomes less robust. Second, after sentence clustering, we may still want to identify the the aspect words in each sentence, which are useful in the next pattern mining step. Directly taking the most frequent words from each sentence cluster as aspect words may not work well even after stop word removal, because there can be background words commonly used in all aspects.

4 Sentence Pattern Generation

At the pattern generation step, we want to identify human-readable sentence patterns that best represent each cluster. Following the basic idea from (Filatova et al., 2006), we start with the parse trees of sentences in each cluster, and apply a frequent subtree pattern mining algorithm to find sentence structures that have occurred at least K times in the cluster. Here we use dependency parse trees.

However, different from (Filatova et al., 2006), the word labels (S, B, D and A) assigned by the entity-aspect model give us some advantages. Intuitively, a representative sentence pattern for an aspect should contain at least one aspect word. On the other hand, document words are entity-specific and therefore should not appear in the generic template patterns; instead, they correspond to template slots that need to be filled in. Furthermore, since we work on entity summaries, in each sentence there is usually a word or phrase that refers to the subject entity, and we should have a placeholder for the subject entity in each pattern.

Based on the intuitions above, we have the following sentence pattern generation process.

1. ***Locate subject entities***: In each sentence, we want to locate the word or phrase that refers to the subject entity. For example, in a biography, usually a pronoun "he" or "she" is used to refer to the subject person. We use the following heuristic to locate the subject entities: For each summary document, we first find the top 3 frequent base noun phrases that are subjects of sentences. For example, in a company introduction, the phrase "the company" is probably used frequently as a sentence subject. Then for each sentence, we first look for the title of the Wikipedia article. If it occurs, it is tagged as the subject entity. Otherwise, we check whether one of the top 3 subject base noun phrases occurs, and if so, it is tagged as the subject entity. Otherwise, we tag the subject of the sentence as the subject entity. Finally, for the identified subject entity word or phrase, we replace the label assigned by the entity-aspect model with a

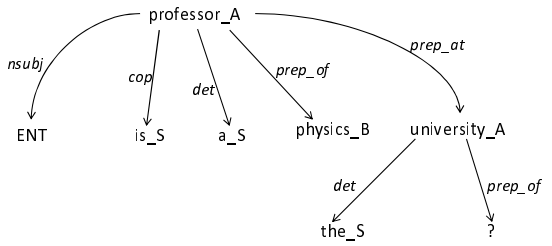

Figure 3: An example labeled dependency parse tree.

new label E.

2. ***Generate labeled parse trees***: We parse each sentence using the Stanford Parser[1]. After parsing, for each sentence we obtain a dependency parse tree where each node is a single word and each edge is labeled with a dependency relation. Each word is also labeled with one of $\{E, S, B, D, A\}$. We replace words labeled with E by a placeholder *ENT*, and replace words labeled with D by a question mark to indicate that these correspond to template slots. For the other words, we attach their labels to the tree nodes. Figure 3 shows an example labeled dependency parse tree.

3. ***Mine frequent subtree patterns***: For the set of parse trees in each cluster, we use FREQT[2], a software that implements the frequent subtree pattern mining algorithm proposed in (Zaki, 2002), to find all subtrees with a minimum support of K.

4. ***Prune patterns***: We remove subtree patterns found by FREQT that do not contain *ENT* or any aspect word. We also remove small patterns that are contained in some other larger pattern in the same cluster.

5. ***Covert subtree patterns to sentence patterns***: The remaining patterns are still represented as subtrees. To covert them back to human-readable sentence patterns, we map each pattern back to one of the sentences that contain the pattern to order the tree nodes according to their original order in the sentence.

In the end, for each summary collection, we obtain A clusters of sentence patterns, where each cluster presumably corresponds to a single aspect or subtopic.

[1] http://nlp.stanford.edu/software/lex-parser.shtml

[2] http://chasen.org/~taku/software/freqt/

Category	D	S	S_d		
			min	max	avg
US Actress	407	1721	1	21	4
Physicist	697	4238	1	49	6
US CEO	179	1040	1	24	5
US Company	375	2477	1	36	6
Restaurant	152	1195	1	37	7

Table 3: The number of documents (D), total number of sentences (S) and minimum, maximum and average numbers of sentences per document (S_d) of the data set.

5 Evaluation

Because we study a non-standard task, there is no existing annotated data set. We therefore created a small data set and made our own human judgment for quantitative evaluation purpose.

5.1 Data

We downloaded five collections of Wikipedia articles from different entity categories. We took only the introduction sections of each article (before the tables of contents) as entity summaries. Some statistics of the data set are given in Table 3.

5.2 Quantitative Evaluation

To quantitatively evaluate the summary templates, we want to check (1) whether our sentence patterns are meaningful and can represent the corresponding entity categories well, and (2) whether semantically related sentence patterns are grouped into the same aspect. It is hard to evaluate both together. We therefore separate these two criteria.

5.2.1 Quality of sentence patterns

To judge the quality of sentence patterns without looking at aspect clusters, ideally we want to compute the precision and recall of our patterns, that is, the percentage of our sentence patterns that are meaningful, and the percentage of true meaningful sentence patterns of each category that our method can capture. The former is relatively easy to obtain because we can ask humans to judge the quality of our patterns. The latter is much harder to compute because we need human judges to find the set of true sentence patterns for each entity category, which can be very subjective.

We adopt the following pooling strategy borrowed from information retrieval. Assume we want to compare a number of methods that each can generate a set of sentence patterns from a summary collection. We take the union of these sets

of patterns generated by the different methods and order them randomly. We then ask a human judge to decide whether each sentence pattern is meaningful for the given category. We can then treat the set of meaningful sentence patterns found by the human judge this way as the ground truth, and precision and recall of each method can be computed. If our goal is only to compare the different methods, this pooling strategy should suffice.

We compare our method with the following two baseline methods.

Baseline 1: In this baseline, we use the same subtree pattern mining algorithm to find sentence patterns from each summary collection. We also locate the subject entities and replace them with *ENT*. However, we do not have aspect words or document words in this case. Therefore we do not prune any pattern except to merge small patterns with the large ones that contain them. The patterns generated by this method do not have template slots.

Baseline 2: In the second baseline, we apply a verb-based pruning on the patterns generated by the first baseline, similar to (Filatova et al., 2006). We first find the top-20 verbs using the scoring function below that is taken from (Filatova et al., 2006), and then prune patterns that do not contain any of the top-20 verbs.

$$s(v_i) = \frac{N(v_i)}{\sum_{v_j \in \mathcal{V}} N(v_j)} \cdot \frac{M(v_i)}{D},$$

where $N(v_i)$ is the frequency of verb v_i in the collection, $\mathcal{V}$ is the set of all verbs, D is the total number of documents in the collection, and $M(v_i)$ is the number of documents in the collection that contains v_i.

In Table 4, we show the precision, recall and f1 of the sentence patterns generated by our method and the two baseline methods for the five categories. For our method, we set the support of the subtree patterns K to 2, that is, each pattern has occurred in at least two sentences in the corresponding aspect cluster. For the two baseline methods, because sentences are not clustered, we use a larger support K of 3; otherwise, we find that there can be too many patterns. We can see that overall our method gives better f1 measures than the two baseline methods for most categories. Our method achieves a good balance between precision and recall. For BL-1, the precision is high but recall is low. Intuitively BL-1 should have a higher recall than our method because our method does more pattern pruning than BL-1 using aspect words. Here it is not the case mainly because we used a higher frequency threshold (K = 3) to select frequent patterns in BL-1, giving overall fewer patterns than in our method. For BL-2, the precision is higher than BL-1 but recall is lower. It is expected because the patterns of BL-2 is a subset of that of BL-1.

Category	B	Purity
US Actress	4	0.626
Physicist	6	0.714
US CEO	4	0.674
US Company	4	0.614
Restaurant	3	0.587

Table 5: The true numbers of aspects as judged by the human annotator (B), and the purity of the clusters.

There are some advantages of our method that are not reflected in Table 4. First, many of our patterns contain template slots, which make the pattern more meaningful. In contrast the baseline patterns do not contain template slots. Because the human judge did not give preference over patterns with slots, both "*ENT* won the award" and "*ENT* won the ? award" were judged to be meaningful without any distinction, although the former one generated by our method is more meaningful. Second, compared with BL-2, our method can obtain patterns that do not contain a non-auxiliary verb, such as "*ENT* was ? director."

5.2.2 Quality of aspect clusters

We also want to judge the quality of the aspect clusters. To do so, we ask the human judge to group the ground truth sentence patterns of each category based on semantic relatedness. We then compute the purity of the automatically generated clusters against the human judged clusters using purity. The results are shown in Table 5. In our experiments, we set the number of clusters A used in the entity-aspect model to be 10. We can see from Table 5 that our generated aspect clusters can achieve reasonably good performance.

5.3 Qualitative evaluation

We also conducted qualitative comparison between our entity-aspect model and standard LDA model as well as a K-means sentence clustering method. In Table 6, we show the top 5 frequent words of three sample aspects as found by our method, standard LDA, and K-means. Note that although we try to align the aspects, there is

Method		Category				
		US Actress	Physicist	US CEO	US Company	Restaurant
BL-1	precision	0.714	0.695	0.778	0.622	0.706
	recall	0.545	0.300	0.367	0.425	0.361
	f1	0.618	0.419	0.499	0.505	0.478
BL-2	precision	0.845	0.767	0.829	0.809	1.000
	recall	0.260	0.096	0.127	0.167	0.188
	f1	0.397	0.17	0.220	0.276	0.316
Ours	precision	0.544	0.607	0.586	0.450	0.560
	recall	0.710	0.785	0.712	0.618	0.701
	f1	0.616	0.684	0.643	0.520	0.624

Table 4: Quality of sentence patterns in terms of precision, recall and f1.

Method	Sample Aspects		
	1	2	3
Our entity-aspect model	university received ph.d. college degree	prize nobel physics awarded medal	academy sciences member national society
Standard LDA	physics american professor received university	nobel prize physicist awarded john	physics institute research member sciences
K-means	physics university institute work research	physicist american physics university nobel	physics academy sciences university new

Table 6: Comparison of the top 5 words of three sample aspects using different methods.

no correspondence between clusters numbered the same but generated by different methods.

We can see that our method gives very meaningful aspect clusters. Standard LDA also gives meaningful words, but background words such as "physics" and "physicist" are mixed with aspect words. Entity-specific words such as "john" also appear mixed with aspect words. K-means clusters are much less meaningful, with too many background words mixed with aspect words.

6 Related Work

The most related existing work is on domain template generation by Filatova et al. (2006). There are several differences between our work and theirs. First, their template patterns must contain a non-auxiliary verb whereas ours do not have this restriction. Second, their verb-centered patterns are independent of each other, whereas we group semantically related patterns into aspects, giving more meaningful templates. Third, in their work, named entities, numbers and general nouns are treated as template slots. In our method, we apply the entity-aspect model to automatically identify words that are document-specific, and treat these words as template slots, which can be potentially more robust as we do not rely on the quality of named entity recognition. Last but not least, their documents are event-centered while ours are entity-centered. Therefore we can use heuristics to anchor our patterns on the subject entities.

Sauper and Barzilay (2009) proposed a framework to learn to automatically generate Wikipedia articles. There is a fundamental difference between their task and ours. The articles they generate are long, comprehensive documents consisting of several sections on different subtopics of the subject entity, and they focus on learning the topical structures from complete Wikipedia articles. We focus on learning sentence patterns of the short, concise introduction sections of Wikipedia articles.

Our entity-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 than sets of sentences as in our model. Titov and McDonald (2008) exploited the idea that a short paragraph within a document is likely to be about the same aspect. Our one-aspect-per-sentence assumption is a stricter than theirs, but it is required in our model for the purpose of mining sentence patterns. 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), but their task is multi-document summarization while ours is to induce summary templates.

7 Conclusions and Future Work

In this paper, we studied the task of automatically generating templates for entity summaries. We proposed an entity-aspect model that can automatically cluster sentences and words into aspects. The model also labels words in sentences as either a stop word, a background word, a document word or an aspect word. We then applied frequent subtree pattern mining to generate sentence patterns that can represent the aspects. We took advantage of the labels generated by the entity-aspect model to prune patterns and to locate template slots. We conducted both quantitative and qualitative evaluation using five collections of Wikipedia entity summaries. We found that our method gave overall better template patterns than two baseline methods, and the aspect clusters generated by our method are reasonably good.

There are a number of directions we plan to pursue in the future in order to improve our method. First, we can possibly apply linguistic knowledge to improve the quality of sentence patterns. Currently the method may generate similar sentence patterns that differ only slightly, e.g. change of a preposition. Also, the sentence patterns may not form complete, meaningful sentences. For example, a sentence pattern may contain an adjective but not the noun it modifies. We plan to study how to use linguistic knowledge to guide the construction of sentence patterns and make them more meaningful. Second, we have not quantitatively evaluated the quality of the template slots, because our judgment is only at the whole sentence pattern level. We plan to get more human judges and more rigorously judge the relevance and usefulness of both the sentence patterns and the template slots. It is also possible to introduce certain rules or constraints to selectively form template slots rather than treating all words labeled with D as template slots.

Acknowledgments

This work was done during Peng Li's visit to the Singapore Management University. This work was partially supported by the National High-tech Research and Development Project of China (863) under the grant number 2009AA04Z106 and the National Science Foundation of China (NSFC) under the grant number 60773088. We thank the anonymous reviewers for their helpful comments.

References

David Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

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 44th Annual Meeting of the Association for Computational Linguistics*, pages 305–312.

Elena Filatova, Vasileios Hatzivassiloglou, and Kathleen McKeown. 2006. Automatic creation of domain templates. In *Proceedings of 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, pages 207–214.

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.

Aria Haghighi and Lucy Vanderwende. 2009. Exploring content models for multi-document summarization. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 362–370.

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.

Satoshi Sekine. 2006. On-demand information extraction. In *Proceedings of 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, pages 731–738.

Yusuke Shinyama and Satoshi Sekine. 2006. Preemptive information extraction using unrestricted relation discovery. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 304–311.

Kiyoshi Sudo, Satoshi Sekine, and Ralph Grishman. 2003. An improved extraction pattern representation model for automatic IE pattern acquisition. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 224–231.

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.

Fei Wu and Daniel S. Weld. 2007. Autonomously semantifying Wikipedia. In *Proceedings of the 16th ACM Conference on Information and Knowledge Management*, pages 41–50.

Yulan Yan, Naoaki Okazaki, Yutaka Matsuo, Zhenglu Yang, and Mitsuru Ishizuka. 2009. Unsupervised relation extraction by mining Wikipedia texts using information from the Web. 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 1021–1029.

Mohammed J. Zaki. 2002. Efficiently mining frequent trees in a forest. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 71–80.