Product Review Summarization by Exploiting Phrase Properties

Naitong Yu, Minlie Huang, Yuanyuan Shi*, Xiaoyan Zhu
State Key Laboratory of Intelligent Technology and Systems
Tsinghua National Laboratory for Information Science and Technology
Dept. of Computer Science and Technology, Tsinghua University, Beijing 100084, PR China
*Samsung R&D Institute China - Beijing
ynt12@mails.tsinghua.edu.cn, aihuang@tsinghua.edu.cn
yy.shi@samsung.com, zxy-dcs@tsinghua.edu.cn

Abstract

We propose a phrase-based approach for generating product review summaries. The main idea of our method is to leverage phrase properties to choose a subset of optimal phrases for generating the final summary. Specifically, we exploit two phrase properties, *popularity* and *specificity*. *Popularity* describes how popular the phrase is in the original reviews. *Specificity* describes how descriptive a phrase is in comparison to generic comments. We formalize the phrase selection procedure as an optimization problem and solve it using integer linear programming (ILP). An aspect-based bigram language model is used for generating the final summary with the selected phrases. Experiments show that our summarizer outperforms the other baselines.

1 Introduction

With the growth of the Internet over the decades, e-commerce is becoming more and more popular. Product reviews are helpful for both merchants and customers. Merchants analyze the reviews to get feedback to improve their products. Customers make use of the reviews to get a better understanding of the product. The opinions in the reviews can help them make the final decision. However, the vast availability of such reviews becomes overwhelming to users when there is just too much to digest. Product review summarization is the task to address this problem. It summarizes the large number of reviews and generates a short readable summary which contains the overall rating of the opinions in the reviews.

Traditional extractive summarization has been studied for a long time, such as (Hovy and Lin, 1999; Kupiec et al., 1995; Paice, 1990). Recently, there are also a number of studies on abstractive summarization, such as (Banerjee et al., 2015; Bing et al., 2015; Liu et al., 2015). However, applying traditional summarization methods directly on product reviews doesn't yield satisfying results. This is due to that product review summarization is quite different from traditional extractive summarization. From the perspective of data size, the number of reviews of a product is often much larger than that of traditional data such as news articles. Another important difference is that sentences in product reviews are usually colloquial and contain lots of noises. Directly extractive summaries may contain a large number of undesired information.

A number of researchers have studied the task of review summarization. (Ganesan et al., 2010) proposed a graph-based method for generating ultra concise opinion summaries of products. They used predefined rules for finding valid sub-paths in the graph and converted those sub-paths into sentences. Since the sentence generation was rule-based, their method didn't provide a well-formed grammatical summary. (Gerani et al., 2014) generated product review summaries by using discourse structure. After simplifying the discourse graph, they used a template-based NLG framework to generate natural language summaries. Their summary produced a statistical overview of the product but lacked detailed information. (Ganesan et al., 2012) proposed some heuristic rules to generate phrases, they used a modified mutual information function and an n-gram language model to ensure the representativeness and readability of the phrases. However, their method didn't consider the descriptiveness of the phrases.

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers,
pages 1113–1124, Osaka, Japan, December 11-17 2016.

We propose a phrase-based approach for generating product review summaries. We provide users with information that cover the most popular opinions in the original reviews targeting at each aspect. We use phrases as the basic unit of our summary, instead of sentences. We adopt the phrase definition in (Lu et al., 2009), that each phrase is composed by a pair of head term and modifier. The head term of a phrase denotes an *aspect* of the product, and the modifier denotes the *opinion* towards the aspect. For example, a phrase about the screen of a cellphone, "stunning [*modifier*] screen [*head*]". Based on the structure of phrases, we define two phrase properties, *popularity* and *specificity*. *Popularity* models how popular the phrase is in the original reviews. *Specificity* models how descriptive the modifier is to the head term. These two properties indicate the most important features of phrases in a good summary. We formalize this problem as an optimization problem and solve it using integer linear programming (ILP). A bigram aspect-based language model is used to order the selected phrases by aspects to form the final summary.

To summarize, our contributions are as follows:

- We propose a phrase-based approach for generating product review summaries. Our method leverages phrase properties, i.e., specificity and popularity, to choose popular and descriptive phrases from the original reviews.
- We formalize the summarization task as an optimization problem, and solve it using integer linear programming (ILP).
- We evaluate our summarization algorithm with both preference evaluation and qualitative evaluation. Our system performs better than other baselines in both evaluations.

The rest of this paper is organized as follows: In Section 2, we will present our phrase-based review summarization algorithm. In Section 3, we will describe the dataset and experiment results. In Section 4, we will describe the related work. In Section 5, we will summarize our work.

2 Summarization Algorithm

Our summarization algorithm takes a set of reviews of one product and a set of aspects as input and generates a summary based on the properties of the phrases extracted from the input reviews. The first step is phrase extraction. Phrases are extracted from the reviews using a given list of aspects. There are various methods for extracting aspects (Hu and Liu, 2004a; Hu and Liu, 2004b; Kim et al., 2011; Lu et al., 2009). In this paper, we do not focus on aspect extraction but consider them as the input. Each of the extracted phrases is tagged with its corresponding sentiment orientation. The second step is optimal phrase selection. We calculate properties of the phrases and select a subset of optimal phrases for constructing the final summary. We formalize this selection problem as an optimization problem and solve it using integer linear programming (ILP). The third step is aspects ordering. We use an aspect-based bigram language model to decide the order of the aspects in the final summary. In the last step, summary generation, phrases are filled into their corresponding aspect placeholders to form the final summary. The summarization framework is shown in Figure 1.

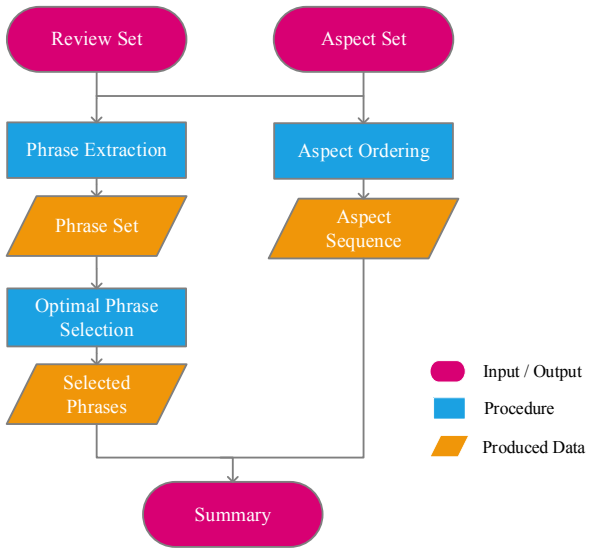

Figure 1: The overall framework of our summarization algorithm.

2.1 Phrase Extraction

In product reviews, most opinions are expressed in concise phrases, such as "camera is excellent" or "stunning screen". We adopt the phrase definition in (Lu et al., 2009), that each phrase can be parsed into a pair of a head term and a modifier. [1]

Aspect. An *aspect* denotes some specific feature of the product. For each aspect, there is a set of *aspect keywords* describing the corresponding aspect.

For example, available aspects of cell phones may include "appearance", "screen", "battery", etc. The aspect keywords set of "appearance" may include "appearance", "design", "surface", etc. Each keyword in the same keywords set is describing the same aspect.

Phrase. A phrase $p = (w_h, w_m)$ is in the form of a pair of a head term w_h and a modifier w_m. The head term is an aspect keyword of the product and the modifier expresses some opinion towards the aspect.

For example, "camera [*head*] is excellent [*modifier*]" and "stunning [*modifier*] screen [*head*]".

Phrase extraction is based on lexical and syntactic rules. First we perform part-of-speech (POS) tagging on the reviews.[2] Then we extract phrases from the reviews with Algorithm 1.

The input of the algorithm is the reviews of one product, denoted as R, and the keywords of all aspects, denoted as K. The output of the algorithm is a list of phrases denoted as P, with the corresponding indexes of the reviews denoted as $Index$, from which each phrase is extracted.

Algorithm 1 Phrase Extraction

Input:
Reviews of one product, $R = \{r_i\}_{i=1}^{n}$
Keywords, $K = \{k_j\}_{j=1}^{m}$
Output:
Phrases P and the corresponding indexes $Index$

```
1:  for each r_i in R do
2:    for each k_j in K do
3:      if ExistKeyword(k_j, r_i) then
4:        w_h ← k_j
5:        pos ← GetPosition(k_j, r_i)
6:        if ExistModifier(r_i, pos) then
7:          w_m ← GetModifier(r_i, pos)
8:          p ← (w_h, w_m)
9:          P ← P + {p}
10:         Index ← Index + {i}
11:       end if
12:     end if
13:   end for
14: end for
```

(1) For each review in R, first we check whether there are any aspect keywords in the review. If any aspect keywords are found (Line 3), then for each keyword found in the review, we set w_h as the aspect word (Line 4).

(2) From the position where we found the aspect word (Line 5), we do a forward and backward search to find the nearest adjective word. If the adjective word is found (Line 6), then set w_m as the adjective word (Line 7). If the adjective word is modified by an adverb, then the adjective word along with the adverb become the modifier w_m. If the adjective word is modified by a negative word, the negative word is also included in the modifier. This is handled by the function $GetModifier(r_i, pos)$. For example, in the phrase "screen/n is/v not/adv very/adv clear/adj", the modifier w_m is "not very clear".

(3) If both the head term w_h and the modifier w_m are found, a phrase $p = (w_h, w_m)$ is extracted, along with the index of the corresponding review (Line 8 - 10).

2.2 Optimal Phrase Selection: Definitions

In this section, we select a subset of optimal phrases from the phrase set. The subset should best represents the overall opinions expressed in the original reviews.

It is intuitive that a phrase is more likely to be included in a summary if it represents most of the users' opinion. For example, if there are 75% of the reviews containing the phrase "camera is excellent" while

[1] We demonstrate our summarization algorithm with running examples in English, but the datasets we use in our experiments are in Chinese.

[2] We use THULAC (`http://thulac.thunlp.org/`) as the POS tagging tool.

there are only 15% of the reviews containing other phrase "photo quality is very bad", then we should choose the former one as a candidate, because it is more popular in the original reviews.

On the other hand, for phrases describing the same aspect, we prefer the one whose modifier describes its head term more descriptive, i.e., the one which is more specific. For example, there are two phrases about the same aspect: "screen is clear" and "screen is good", we prefer "screen is clear" because that the modifier "clear" is more specific and better expresses the characteristic of the aspect "screen" while the modifier "good" is more general and can be used to describe other aspects.

A phrase is considered to be a candidate of the summary if it is popular in the original reviews and its modifier is specific to its head term. To better describe the phrase properties proposed above, we give the definition of *popularity* and *specificity* formally.

Definition 1 (Popularity). For a phrase p, let R_p denote the set of reviews that contain p, let R_{all} denote the set of all reviews. The *popularity* of phrase p is defined as:

$$\text{Popularity}(p) = \frac{|R_p|}{|R_{all}|} \tag{1}$$

where $|R|$ is the size of the review set R.

For a phrase set P, $p_i \in P$, $i = 1, \ldots, n$, let R_{p_i} denote the set of reviews that contain p_i, let R_{all} denote the set of all reviews. $\text{Popularity}(P) = |\cup R_{p_i}|/|R_{all}|$.

Suppose that we want to calculate the popularity of the phrase "long battery". Let's say there are 120 reviews in total and 25 of them contain the phrase "long battery", then the popularity of the phrase is $\text{Popularity}(\text{"long battery"}) = 25/120 = 0.21$.

Definition 2 (Specificity). For a phrase $p = (w_h^p, w_m^p)$, w_h^p denotes the aspect keyword of p, and A_p denotes the aspect that w_h^p belongs to, i.e., $w_h^p \in A_p$. w_m^p denotes the modifier of p. $P_{w_m = w_m^p}$ denotes the set of phrases whose modifier $w_m = w_m^p$, and $P_{w_h \in A_p, w_m = w_m^p}$ denotes the set of phrases whose head term $w_h \in A_p$ and modifier $w_m = w_m^p$. The *specificity* of phrase p is defined as:

$$\text{Specificity}(p) = \frac{|P_{w_h \in A_p, w_m = w_m^p}|}{|P_{w_m = w_m^p}|} \tag{2}$$

For a phrase set P, $p_i \in P, \quad i = 1, \ldots, n$, $\text{Specificity}(P) = \sum_i \text{Specificity}(p_i)$.

For example, suppose that we want to calculate the specificity of the phrase "beautiful design". The head term of this phrase is "design" and it belongs to the aspect *appearance*. The modifier term of this phrase is "beautiful". Let's say that there are 50 phrases whose modifier is "beautiful" in total, and there are 42 phrases whose modifier is "beautiful" and whose head term belong to the aspect *appearance*, then the specificity of the phrase is $\text{Specificity}(\text{"beautiful design"}) = 42/50 = 0.84$.

2.3 Optimal Phrase Selection: Problem Formalization

To select the optimal subset of phrases, we combine popularity and specificity to form an optimization problem. We use an integer linear programming (ILP) library[3] to solve this problem. We maximize $\text{Popularity}(P)$ and $\text{Specificity}(P)$ of a phrase set P together with the following constraints:

- **Length Constraint:** The total length of the summary is no longer than L_s.
- **Aspect Constraint:** For each aspect, the number of phrases in the cluster is no more than L_a.
- **Consistency Constraint:** For phrases in the same aspect, the sentiment orientation of these phrases should agree with each other.

To define the problem formally, let p_i denote the ith phrase in the phrase set P_{all}, and let r_j denote the jth review in the review set R_{all}. Let x_i represent a binary variable, that can take 0 or 1, depending on

[3] `http://sourceforge.net/projects/lpsolve/`

whether the ith phrase is selected for the final summary or not, and let y_j also represent a binary variable, that denotes whether the jth review is selected or not. Let P_{sel} denotes the set of phrases which are selected for the final summary. The objective function can be denoted as:

$$\begin{aligned} F(x_1, \ldots, x_n, y_1, \ldots, y_m) &= \text{Specificity}(P_{sel}) + \text{Popularity}(P_{sel}) \\ &= \sum_i \text{Specificity}(p_i) \cdot x_i + \frac{1}{|R_{all}|} \sum_j y_j \end{aligned} \tag{3}$$

The length constraint can be denoted as:

$$\sum_i l(p_i) \cdot x_i \leq L_s \tag{4}$$

where $l(p_i)$ denotes the length of phrase p_i. This constraint limits the total length of the summary to be no longer than L_s.

The aspect constraint can be denoted as:

$$\sum_{p_i \in P_{A^k}} x_i \leq L_a, \quad \forall A^k \tag{5}$$

where A^k denotes the kth aspect, and P_{A^k} denotes the set of phrases whose head term $w_h \in A^k$. These constraints limit the phrase number of each aspect to be no more than L_a.

The consistency constraint can be denoted as:

$$\left| \sum_{p_i \in P_{A^k}} o(p_i) \cdot x_i \right| = \sum_{p_i \in P_{A^k}} x_i, \quad \forall A^k \tag{6}$$

where $o(p_i)$ denotes the sentiment orientation of phrase p_i. $o(p_i) = 1$ if the sentiment orientation of phrase p_i is positive, and $o(p_i) = -1$ if the sentiment orientation of phrase p_i is negative. These constraints ensure that phrases in the same aspect have the same sentiment orientation.

There are other constraints to ensure the consistency between the phrase set and review set:

$$x_i \cdot Occ_{i,j} \leq y_j, \quad \forall i, j \tag{7}$$

$$\sum_i x_i \cdot Occ_{i,j} \geq y_j, \quad \forall j \tag{8}$$

where $Occ_{i,j}$ is a binary value, $Occ_{i,j} = 1$ if and only if phrase p_i is in review r_j, i.e., $p_i \in r_j$. Equation (7) means that if phrase p_i is selected ($x_i = 1$), then any review r_j that $p_i \in r_j$ is also selected ($y_j = 1$). Equation (8) means that if review r_j is selected ($y_j = 1$), then at least one phrase p_i that $p_i \in r_j$ is selected ($x_i = 1$).

2.4 Aspects Ordering

When writing comments, customers tend to first mention the aspect they care about most. We determine the aspect order by finding the most common aspect sequence in the original reviews. By this means, the generated summary would be more natural and coherent to human-generated reviews. Since we are only interested in the relative order of two adjacent aspects, we use an aspect-based bigram language model that assign probabilities to sequence of aspects.

For each review r, let $\Gamma(r)$ denote a function that maps each review to its corresponding aspects sequence, i.e., $\Gamma(r) = s_r$, where $s_r = [a_1^r, a_2^r, \ldots, a_q^r]$ is a sequence of aspects, and a_i^r is the ith aspects in the review r. Let $S = \{\Gamma(r) | r \in R\}$ denote the set of all aspect sequences for all reviews, let A denote the available aspect set, for any $a_i, a_j \in A$, $s_k \in S$,

$$p(a_i | a_j) = \frac{\sum_k I(a_i a_j \in s_k)}{\sum_k I(a_j \in s_k)} \tag{9}$$

where $I(x) = 1$ if x is true, otherwise $I(x) = 0$. For an arbitrary aspect sequence $s = a_1, a_2, \ldots, a_r$, the probability of the sequence is:

$$p(s) = p(a_1)p(a_2|a_1)\ldots p(a_r|a_{r-1}) \tag{10}$$

The objective sequence is the sequence that has the maximum probability in the language model and each aspect appears and only appears in the sequence once:

$$s^* = \arg\max p(s) \tag{11}$$

where $s = a_1, a_2, \ldots, a_n$, n is the number of aspects, $a_i \neq a_j, \forall i, j, i \neq j$.

2.5 Summary Generation

The summary generation procedure is quite straightforward. We sort the phrases in the optimal phrase set by the order of their corresponding aspects, and the final summary is a sequence of these phrases.

For phrases which have different aspects, the order of them is the same as the order of the corresponding aspects in the aspect sequence. For phrases which have the same aspect, the order is the same as the descending order of the size of the corresponding review set. Specially, if two phrases in the same aspect have the same head term, we merge their modifier term into one. For example, "screen is big" and "screen is clear" can be merged into "screen is big and clear".

Table 1 shows an example summary generated by our system.[4]

屏幕舒服、效果出色，电池耐用、续航长，做工精致，质量很棒，性能优越，速度快，性价比极高，价格适中，像素一般，摄像头很一般，外观漂亮、好看，软件太少、不丰富，界面十分简洁，画质清楚，操作简单、简洁，音效不错，音质特好，内存不够、较小，信号相当不错，通话声音清晰，按钮位置不好，机身重、太大。
Screen is comfortable and display is excellent. Battery is durable and lasts long. Exquisite workmanship and good quality. Performance is superior and fast. Price is cost-effective and affordable. Camera is very general. Appearance is beautiful and nice. Software is not rich. Interface is very simple. Picture is clear and of good quality. Operation is simple. Sound quality is especially good. Memory is not enough. Signal is quite good. Clear voice calls. Button location is not good. Body is heavy and too big.

Table 1: Example summary generated by our system.

3 Experiments

3.1 Data Preparation

Phrase Extraction. We construct our experiment dataset using customer reviews of 10 cellphones. The reviews are collected from `jd.com`, `zol.com` and `weibo.com`. All of the reviews are written in Chinese. We get 33,948 reviews in total. We construct 17 aspects manually, each aspect contains about 6 aspect keywords on average. With these aspect keywords, we perform phrase extraction on the review dataset. Each of the extracted phrases is then tagged with its corresponding sentiment orientation. In total, we have extracted 44,461 phrases, i.e., 1.31 phrases per review on average. 83% of the extracted phrases are positive, while 17% are negative.

Sentiment Detection. We train an SVM classifier to classify the sentiment orientations for the phrases. We use words as features and the corresponding values are binaries which indicate whether the words are present or not. First we drop words with frequency lower than 100, then we rank words by their chi-square values in descending order and choose the first third of all words as the feature words. We use `svmlight`[5] with default parameters to implement our classifier.

We use the review data for cell phones from `jd.com`. Since a single review may contain different sentiment orientations, we split each review into short sentences by a splitting delimiter set, which contains punctuations such as "，。；：！？ ". For each short sentence, we ask two annotators to judge the sentiment

[4]The summary is in Chinese and we translate it into English manually.

[5]`http://svmlight.joachims.org/`

orientation as "positive", "negative" or "neutral". Conflicting results are reviewed by a third annotator. We are only interested in short sentences with opinions, so we drop those tagged with "neutral". We get 182,120 short sentences in total, 72.5% are positive and 27.5% are negative. Since the same opinion word may have different orientations in different aspects, we cluster the short sentences by their aspects and train an SVM classifier for each aspect separately. For each of the 17 aspects, we randomly select 320 short sentences for testing, and the rest are used for training. The overall performance of sentiment classification is shown in Table 2. This result shows that out SVM classifier performs good enough for our review summarization task.

Precision		Recall		F1		Accuracy
Pos	Neg	Pos	Neg	Pos	Neg	All
91.8	91.2	92.3	90.7	92.0	90.9	91.5

Table 2: % of precision, recall, F1 and overall accuracy on sentiment classification

3.2 Baselines

The evaluation of review summarization is a very challenging task. On one hand, to the best of our knowledge, there is no dataset of product reviews with human written summaries. On the other hand, since the amount of product reviews is often large, it is quite difficult to generate human written summaries.

We evaluate the summaries generated by our system (denoted as **ReviewSum**) with a state-of-the-art extractive baseline, a state-of-the-art abstractive baseline and a simplified version of our system. The details of the baselines are described as follows:

1) LexRank: LexRank (Erkan and Radev, 2004) is a graph-based extractive summarization method which computes sentence importance based on the concept of eigenvector centrality in a graph representation of sentences. In the experiment, first we cluster sentences by their aspects. Then for each sentence cluster, LexRank is performed for summary generation. The final summary is generated by putting summaries of different aspects together in the same aspect order of ReviewSum.

2) Opinosis: Opinosis (Ganesan et al., 2010) is a novel graph-based summarization method which generates concise abstractive summaries of highly redundant opinions. In the experiment, for each aspect, we build an Opinosis graph and get the top candidate summaries. The final summary is generated by putting summaries of different aspects together in the same aspect order of ReviewSum.

3) BasicSum: BasicSum is a simplified version of our summarization method. Instead of popularity and specificity, TF-IDF score is used in the objective function. The objective function of BasicSum can be denoted as:

$$F(x_1, \ldots, x_n) = \sum_i \text{tf-idf}(p_i) \cdot x_i \tag{12}$$

where tf-idf(p_i) is the TF-IDF score of phrase p_i, and x_i is a binary value representing whether phrase p_i is selected in the final summary or not.

3.3 Experiments Evaluation

Due to the lack of human written summaries as gold standard, we perform two tasks to evaluate the summaries generated by our system. Task 1 is pairwise user preference evaluation and Task 2 is user scoring evaluation.

In Task 1, we run six pairwise comparisons of four summaries generated by our method and baselines. For each comparison, two summaries of the same product are shown to the annotators in random order. The name of the product and the original reviews are also shown to the annotators. For two summaries S_1 and S_2, annotators need to make a choice in the following three options: 1) Prefer S_1, 2) Prefer S_2, 3) No preference. Note that the exact names of S_1 and S_2 are hidden to annotators.

In order to ensure the quality of the evaluation, annotators are instructed to read the original reviews first before they make their choice. Annotators are specially instructed that their choice should based

on "*overall satisfaction with the information provided by the summary and intuitive feelings about the summary*".

In Task 2, we ask annotators to evaluate four aspects of each summary. The aspects considered during the evaluation include Grammaticality, Non-Redundancy, Consistency and Descriptiveness. Each aspect is rated with a score from 0 (bad) to 10 (excellent). Annotators are instructed to read the summary carefully and rate each aspect with scores matching the quality of the corresponding aspect.

20 annotators participate in Task 1 and Task 2, 10 annotators for each task. All of them are native Chinese speakers with experiences of product review writing. In Task 1, each comparison is evaluated by at least 5 annotators, and more than 300 comparison results are generated. In Task 2, each summary is rated by at least 5 annotators, and more than 160 rated scores are generated.

3.4 Results and Discussions

Sys I	**Sys II**	No pref	Pref Sys I	Pref Sys II	Agreement
BasicSum	LexRank	8%	42%	50%	0.2
BasicSum	Opinosis	2%	22%	76%	0.5
LexRank	Opinosis	12%	32%	56%	0.6
LexRank	ReviewSum	8%	14%	**78%**	0.8
BasicSum	ReviewSum	14%	0%	**86%**	0.8
Opinosis	ReviewSum	8%	18%	**74%**	0.6

Table 3: Results of pairwise comparison preferences. Statistically significant improvements ($p < 0.01$) over the baselines are demonstrated by bold fonts.

Preference Evaluation. Table 3 shows the results of Task 1. The first two columns denote systems compared in each comparison. The following three columns indicate the percentage of preference decisions for each preference category. Statistically significant improvements ($p < 0.01$) of our system over the baselines are demonstrated in bold fonts. The last column indicates the agreement rate of preference comparisons for different systems. Specifically, in our experiments, we treat one pairwise comparison as in agreement if four (out of five) annotators give the same preference decision. Table 4 shows system preference results of each product. System preferences are computed based on the results of pairwise comparison. For each system, the preference is the number of times annotators prefer the system, divided by the total number of comparisons for the system. For example, we have three systems A, B and C. A is preferred over B 10 out of 20 times, and A is preferred over C 25 out of 30 times, then the overall preference of A is $(10 + 25)/(20 + 30) = 70\%$.

Products	BasicSum	LexRank	Opinosis	ReviewSum
Galaxy S5	7%	27%	60%	**87%**
iPhone 5S	20%	40%	53%	**80%**
iPhone 5C	7%	20%	67%	**87%**
Ascend P7	20%	33%	60%	**80%**
Lumia 1320	33%	60%	13%	60%
Sony l36h	33%	20%	40%	**80%**
HTC One	27%	53%	20%	**93%**
LG G2	20%	27%	67%	67%
Galaxy Note 4	13%	33%	67%	**80%**
Galaxy Grand 2	33%	7%	53%	**80%**
Total	21%	32%	50%	**79%**

Table 4: System preference results of each product. Statistically significant improvements ($p < 0.01$) over the baselines are demonstrated by bold fonts.

From Table 3 and Table 4 we can see that our system significantly outperforms BasicSum. The only difference between BasicSum and our system is the objective function. Our system uses popularity and specificity of the phrases while BasicSum uses TF-IDF score. The result shows that popularity and specificity can prominently improve the quality of the summary. In fact, popularity and specificity improve the descriptiveness of the summary significantly, which we will discuss later.

The results in Table 3 and Table 4 show statistically significant improvements in pairwise comparisons of our system over the extractive baseline (LexRank) and the abstractive baseline (Opinosis). Due to the limitations of sentence-based extractive models, summaries produced by LexRank contain long sentences with useless information, while our system produces phrase-based summaries without unwanted information. Opinosis produces much shorter and concise summaries than LexRank, but the grammar of the sentences are not very well. In our method, phrases are generated in a concise manner by joining aspect and opinion together. The generated summaries are clear and well-formatted.

Qualitative Evaluation. Table 5 shows the results of Task 2. In order to avoid scoring varies per person, rating scores are normalized by each annotator, i.e., for each annotator, scores in range $[s_{min}, s_{max}]$ are remapped into range $[0, 10]$. The first column denotes systems in the rating task, the following four columns denote average scores of each system in four different aspects: grammaticality, non-redundancy, consistency and descriptiveness.

Systems	Grammaticality	Non-Redundancy	Consistency	Descriptiveness
BasicSum	6.46	4.89	6.92	3.42
LexRank	4.13	5.31	6.56	5.54
Opinosis	5.83	6.38	7.58	6.17
ReviewSum	**6.87**	6.07	7.41	**8.15**

Table 5: Results of aspects rating scores.

The results in Table 5 show that our system achieves the best score in grammaticality and descriptiveness. This exactly matches what we expect from our method that outputs well-formatted summaries by choosing neat and descriptive phrases. Also, our system is doing better than BasicSum and LexRank in non-redundancy. TF-IDF scores are used in BasicSum for phrase selection, and this will cause phrases with similar opinion words being selected (such as *good*, *very good*, etc.), which results in the increase of redundancy. LexRank extracts sentences directly from reviews, and information redundant may also be included. In consistency, our system performs nearly as good as Opinosis.

4 Related Work

Our work is related to aspect-based opinion summarization, which can be divide into three distinct steps: aspect extraction, sentiment detection and summary generation.

Aspect extraction involves identifying salient aspects within the text to be summarized. (Lu et al., 2009) used shallow parsing to identify aspects for short comments. (Popescu and Etzioni, 2007) used a web-based domain-independent information extraction system to extract aspects from parsed review data. (Hu and Liu, 2004a) and (Hu and Liu, 2004b) used supervised association rule mining-based approach to perform the task of aspect extraction. (Zhuang et al., 2006) used a feature list combining the full cast of all movies and a regular expression set to identify features in movie reviews. (Ku et al., 2006) used paragraph level frequencies as well as document level ones to help identify features.

Sentiment detection is the task of detecting sentiment orientation (positive or negative) on the aspect or feature. (Lu et al., 2009) used a learning-based method for sentiment detection. (Hu and Liu, 2004a; Hu and Liu, 2004b) used an effective method based on WordNet. (Ku et al., 2006) also used a set of positive and negative words from GI and CNSD to predict sentiments of aspects. (Zhuang et al., 2006) used dependency relationships to identify opinions associated with feature words.

Summary generation involves aggregating the results of aspect extraction and sentiment detection and generate the final opinion summary in an effective and easy to understand format. Statistical summary

is the most commonly adopted format, such as (Hu and Liu, 2004a; Hu and Liu, 2004b; Hu and Liu, 2006; Zhuang et al., 2006). (Titov and McDonald, 2008b) used a topic modeling method to provide a word level summary for each topic. (Popescu and Etzioni, 2007) also provided a word level summary by ranking opinion words associated to features and showing the strongest opinionated word for each aspect. (Mei et al., 2007) scored the probability of each sentence using TSM model and generated a sentence level summary. (Ku et al., 2006) used TF-IDF to score sentences and select the most relevant and discriminative sentence to be shown as summary. Besides texts, aggregated ratings can also be shown for summary, such as (Lu et al., 2009). (Ku et al., 2006) and (Mei et al., 2007) showed summary as a timeline with opinion changes over time.

5 Conclusion

In this paper, we propose a phrase-based summarization algorithm for the task of product review summarization. The proposed phrase selection scheme can fully utilize the characteristics of review sentences and capture the main information. We propose two properties of phrases, popularity and specificity, to score the phrase. Integer linear programming (ILP) is used to optimize the objective function. We use an aspect-based bigram language model to determine the aspect order of the candidate phrases to make the generated summaries more fluent and natural. Experimental results show that our system outperforms state-of-the-art systems in most cases. Our proposed summarization algorithm can produce concise, well-formatted and descriptive summaries of product reviews.

Acknowledgements

This work was partly supported by the National Basic Research Program (973 Program) under grant No. 2013CB329403, the National Science Foundation of China under grant No.61272227/61332007. The work was also supported by Tsinghua University Beijing Samsung Telecom R&D Center Joint Laboratory for Intelligent Media Computing.

References

Alexandra Balahur and Andrés Montoyo. 2008. Multilingual feature-driven opinion extraction and summarization from customer reviews. In *NLDB*, volume 5039, pages 345–346. Springer.

Siddhartha Banerjee, Prasenjit Mitra, and Kazunari Sugiyama. 2015. Multi-document abstractive summarization using ilp based multi-sentence compression. In *24th International Joint Conference on Artificial Intelligence (IJCAI). Buenos Aires, Argentina: AAAI press*.

Lidong Bing, Piji Li, Yi Liao, Wai Lam, Weiwei Guo, and Rebecca J Passonneau. 2015. Abstractive multi-document summarization via phrase selection and merging.

Giuseppe Carenini, Jackie Chi Kit Cheung, and Adam Pauls. 2013. Multi-document summarization of evaluative text. *Computational Intelligence*, 29(4):545–576.

Kushal Dave, Steve Lawrence, and David M Pennock. 2003. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th international conference on World Wide Web*, pages 519–528. ACM.

Giuseppe Di Fabbrizio, Amanda J Stent, and Robert Gaizauskas. 2014. A hybrid approach to multi-document summarization of opinions in reviews. *INLG 2014*, page 54.

Yajuan Duan, Zhimin Chen, Furu Wei, Ming Zhou, and Heung-Yeung Shum. 2012. Twitter topic summarization by ranking tweets using social influence and content quality. In *Proceedings of the 24th International Conference on Computational Linguistics*, pages 763–780.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, pages 457–479.

Dimitrios Galanis, Gerasimos Lampouras, and Ion Androutsopoulos. 2012. Extractive multi-document summarization with integer linear programming and support vector regression. In *COLING*, pages 911–926. Citeseer.

Kavita Ganesan, ChengXiang Zhai, and Jiawei Han. 2010. Opinosis: A graph based approach to abstractive summarization of highly redundant opinions. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING '10)*.

Kavita Ganesan, ChengXiang Zhai, and Evelyne Viegas. 2012. Micropinion generation: an unsupervised approach to generating ultra-concise summaries of opinions. In *Proceedings of the 21st international conference on World Wide Web*, pages 869–878. ACM.

Shima Gerani, Yashar Mehdad, Giuseppe Carenini, Raymond T Ng, and Bita Nejat. 2014. Abstractive summarization of product reviews using discourse structure. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1602—1613.

Eduard Hovy and Chin-Yew Lin. 1999. Automated text summarization in summarist. In *Advances in Automatic Text Summarization*. MIT Press.

Minqing Hu and Bing Liu. 2004a. Mining opinion features in customer reviews. In *AAAI*, volume 4, pages 755–760.

Minqing Hu and Bing Liu. 2004b. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM.

Minqing Hu and Bing Liu. 2006. Opinion extraction and summarization on the web. In *AAAI*, volume 7, pages 1621–1624.

Hyun Duk Kim, Kavita Ganesan, Parikshit Sondhi, and ChengXiang Zhai. 2011. Comprehensive review of opinion summarization.

Lun-Wei Ku, Yu-Ting Liang, and Hsin-Hsi Chen. 2006. Opinion extraction, summarization and tracking in news and blog corpora. In *AAAI spring symposium: Computational approaches to analyzing weblogs*, volume 100107.

Julian Kupiec, Jan Pedersen, and Francine Chen. 1995. A trainable document summarizer. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 68–73. ACM.

Kevin Lerman, Sasha Blair-Goldensohn, and Ryan McDonald. 2009. Sentiment summarization: evaluating and learning user preferences. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 514–522. Association for Computational Linguistics.

Fangtao Li, Chao Han, Minlie Huang, Xiaoyan Zhu, Ying-Ju Xia, Shu Zhang, and Hao Yu. 2010. Structure-aware review mining and summarization. In *Proceedings of the 23rd international conference on computational linguistics*, pages 653–661. Association for Computational Linguistics.

Fei Liu, Jeffrey Flanigan, Sam Thomson, Norman Sadeh, and Noah A Smith. 2015. Toward abstractive summarization using semantic representations. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1077–1086.

Yue Lu, ChengXiang Zhai, and Neel Sundaresan. 2009. Rated aspect summarization of short comments. In *Proceedings of the 18th international conference on World wide web*, pages 131–140. ACM.

Rebecca Mason, Benjamin Gaska, Benjamin Van Durme, Pallavi Choudhury, Ted Hart, Bill Dolan, Kristina Toutanova, and Margaret Mitchell. 2016. Microsummarization of online reviews: An experimental study. Association for the Advancement of Artificial Intelligence.

Qiaozhu Mei, Xu Ling, Matthew Wondra, Hang Su, and ChengXiang Zhai. 2007. Topic sentiment mixture: modeling facets and opinions in weblogs. In *Proceedings of the 16th international conference on World Wide Web*, pages 171–180. ACM.

Chris D Paice. 1990. Constructing literature abstracts by computer: techniques and prospects. *Information Processing & Management*, 26(1):171–186.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd annual meeting on Association for Computational Linguistics*, page 271. Association for Computational Linguistics.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: Sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics.

Ana-Maria Popescu and Orena Etzioni. 2007. Extracting product features and opinions from reviews. In *Natural language processing and text mining*, pages 9–28. Springer.

Benjamin Snyder and Regina Barzilay. 2007. Multiple aspect ranking using the good grief algorithm. In *HLT-NAACL*, pages 300–307.

Maosong Sun, Xinxiong Chen, Kaixu Zhang, Zhipeng Guo, and Zhiyuan Liu. 2016. Thulac: An efficient lexical analyzer for chinese.

Ivan Titov and Ryan McDonald. 2008a. Modeling online reviews with multi-grain topic models. In *Proceedings of the 17th international conference on World Wide Web*, pages 111–120. ACM.

Ivan Titov and Ryan T McDonald. 2008b. A joint model of text and aspect ratings for sentiment summarization. In *ACL*, volume 8, pages 308–316. Citeseer.

Lu Wang, Hema Raghavan, Claire Cardie, and Vittorio Castelli. 2014. Query-focused opinion summarization for user-generated content. In *COLING*, pages 1660–1669.

Jin-ge Yao, Xiaojun Wan, and Jianguo Xiao. 2015. Phrase-based compressive cross-language summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 118–127. Association for Computational Linguistics.

Renxian Zhang, Wenjie Li, and Dehong Gao. 2012. Generating coherent summaries with textual aspects. In *AAAI*.

Li Zhuang, Feng Jing, and Xiao-Yan Zhu. 2006. Movie review mining and summarization. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 43–50. ACM.