Product Review Summarization
from a Deeper Perspective

By
LY Duy Khang

Department of Computer Science
School of Computing
National University of Singapore

2009/10
Product Review Summarization
from a Deeper Perspective

By
LY Duy Khang

Department of Computer Science
School of Computing
National University of Singapore

2009/10

Project No: U036090
Advisor: Associate professor KAN Min Yen
Deliverables:
   Report: 1 Volume
   Source Code: 1 CD
Abstract

Product review nowadays has become an important source of information, not only for customers to find opinions about products easily and share them with their peers, but also for producers to get certain degrees of feedback. Unfortunately, the number of reviews is often too large, making it very difficult to utilize this resource. Therefore, in this project, we build a product review summarization system that can automatically process a large collection of reviews and aggregate information in a concise summary. More importantly, we identified a severe limitation in existing product summarization systems of not being able to provide the underlying reasons to justify users’ opinions. Therefore, we initiated a clustering-based solution to this subtopic problem, and obtained a promising result of 75% in F measure.

Subject Descriptors:
- H.3.1 Content Analysis and Indexing
- H.3.3 Information Search and Retrieval
- I.2.7 Natural Language Processing

Keywords:
- Summarization, Semantic Analysis, Clustering

Implementation Software and Hardware:
- Linux, Stanford POS Tagger, Stanford Dependency Parser, Perl, Eclipse Java 3.4
Acknowledgement

I would like to thank my supervisor Associate professor KAN Min-Yen for leading me through the full process of doing research, teaching me what I should do, and giving me critical comments. I have learned a lot from his way of observing things and targeting the problem. I also would like to thank SUGIYAMA Kazunari and LIN Ziheng, who have followed me closely in my project, gave me hints and helpful feedbacks as well as encouragement. Without them, I would not have been able to complete this project.
List of Figures

3.1 Example sentences showing explicit/implicit product facets .......................... 9
3.2 Architecture diagram for component Product Facet Identification .................. 11
3.3 Example sentences together with their syntactic roles ................................. 12
3.4 Example sentences labeled with multiple facets ......................................... 15
3.5 Performance of the component without syntactic role ................................. 16
3.6 Performance of the component with syntactic role ..................................... 17
4.1 A reproduced snapshot of the output summary generated by existing systems .... 19
4.2 A snapshot of the output summary produced by our proposed system ............. 20
4.3 Architecture diagram for component Summarization ................................. 22
4.4 Example of sentences together with their vector representation ..................... 26
4.5 Performance of the Subtopic Clustering module ........................................ 31
# Table of Contents

Title i
Abstract ii
Acknowledgement iii
List of Figures iv

1 Introduction 1

2 Related work 3
   2.1 Sentiment Analysis ...................................................... 3
   2.2 Summarization ............................................................ 5

3 Product Facet Identification 8
   3.1 Preliminary ................................................................. 9
   3.2 Methodology ............................................................... 10
      3.2.1 Architecture Overview ........................................... 10
      3.2.2 Preprocessing ....................................................... 11
      3.2.3 Association Mining ............................................... 12
      3.2.4 Post Processing ................................................... 13
      3.2.5 Sentence Extraction .............................................. 14
   3.3 Experiments ............................................................... 15
      3.3.1 Experimental Data ................................................. 15
      3.3.2 Evaluation Measure .............................................. 16
      3.3.3 Experimental Results ............................................ 16

4 Summarization 18
   4.1 Methodology ............................................................... 21
      4.1.1 Architecture Overview ........................................... 21
      4.1.2 Preprocessing ....................................................... 22
      4.1.3 Sentence Representation and Similarity Measurement ........... 24
      4.1.4 Sentence Clustering .............................................. 26
      4.1.5 Compact Presentation ............................................ 28
   4.2 Experiments ............................................................... 28
      4.2.1 Experiments Data .................................................. 29
      4.2.2 Evaluation Measure .............................................. 29
      4.2.3 Experiments Results ............................................ 31

5 Conclusion and Future Work 35
Chapter 1

Introduction

With the rapid expansion of e-commerce, more and more products are sold on the Web, and more and more people are also buying products online. In order to enhance customer satisfaction and shopping experience, it has become a common practice for online merchants to enable their customers to review or to express opinions on the products that they have purchased. With more and more common users becoming comfortable with the Web, an increasing number of people are writing reviews. As a result, the number of reviews that a product receives grows rapidly. Some popular products can get hundreds of reviews at some large merchant sites.

Many reviews are long and have only a few sentences containing opinions on the product. This makes it hard for a potential customer to read them and make an informed decision on whether to purchase the product. If he/she only reads a few reviews, he/she may get a biased view. The large number of reviews also makes it hard for product manufacturers to keep track of customer opinions of their products. Therefore, we want to build a product review summarization system that can automatically process a large collection of reviews and aggregate information in a summary. The system shall achieve two important goals: 1. Adopt an efficient way to identify topics and subtopics discussed about the product in the reviews. 2. Summarize the correspondent opinions and present a coherent summary to users.
Contributions

In this project, we design a complete end-to-end summarization system that offers a well-organized structure of the information from a large collection of reviews. The final summary is able to capture opinions from different dimensions of the product. More importantly, it allows a potential customer to quickly see how the existing customers feel about the product, yet equips him/her with sufficiently detailed information. We build a system with two main components, namely Product Facet Identification and Summarization. For the first component that identifies frequent product dimensions being discussed in the reviews, our algorithm achieves results close to the current state-of-the-art system in the same task. The second component is a unique construct that we introduce into the domain of product review summarization: we initiate and implement a clustering algorithm to identify group of sentences sharing the same topic, before analyze their sentiment. On top of that, this system is one of the few summarization system used for product review summarization.

Organization of thesis

The rest of this thesis is organized as follow: Chapter 2 gives the overview of existing sentiment analysis and automatic text summarization methods. Chapter 3 describes the implementation of our product facet identification component. Chapter 4 discusses our discovery of the subtopic problem and presents our solutions to the summarization problem. Finally, Chapter 5 concludes the thesis with some discussions about some important aspects of our system and suggests potential future works.
Chapter 2

Related work

In general, the task of summarizing product reviews can be divided into two subproblems: discover the user opinions expressed in the reviews, and then summarize the opinions. Both of them belong to two broad research topics in the field of Natural Language Processing, namely Sentiment Analysis and Summarization respectively. Therefore, it is very important to understand the existing related techniques in each area before applying the knowledge to solve our problem. In the next section, we first mention related works done in Sentiment Analysis. After that we discuss the methods used in Summarization in section 2.2.

2.1 Sentiment Analysis

Sentiment Analysis refers to the computational treatment of subjectivity (whether there exists sentiment), the sentiment polarity (positive, negative, neutral or a scale of sentiment intensity) and the opinion content information (opinion holder, topic of opinion, etc.), that underlies a text span. The increasing granularity of the text span starts at the level of word, then phrase/sentence and finally the entire document, which also offers a natural way of looking at the techniques developed in Sentiment Analysis. However, we will not discuss works at the document level in this section, as the target of our project is not to examine the overall sentiment of the review, but the detailed opinions within the review.

At the word level, it is worth mentioning the early work by (Hatzivassiloglou & McKeown,
on predicting the binary semantic orientation of adjectives. They utilized textual con-
junctions (e.g. and, but) available in a large training corpus between the target adjective and
a seed list of adjectives with manual annotated polarity, achieving 82% accuracy in average.
Later work by (Turney et al., 2002) offered comparable results; however, they allowed the target
words to be not only adjectives, but also nouns, verbs and adverbs. Moreover, their system did
not require a corpus as training data. Instead, they approximated the Pointwise Mutual Infor-
mation (Church & Hanks, 1990) between the target word with the positive word “excellent”,
and with the negative word “poor” respectively, by counting the number of results returned by
the web searches on queries that join each pair of words by a NEAR operator. Since the scores
correspond to the similarity between the target adjective with each positive/negative extreme,
the polarity of that adjective can be determined by taking the label that results in the promi-
nent score. More recently, (Hu & Liu, 2004b) utilized WordNet (Miller et al., 1990) – a large
lexical database of English with synonym and antonym pointers between the synsets – to grow
a initial seed list of known orientation adjectives into a larger list that covers all the remaining
adjectives in WordNet. His system achieved higher results (84% accuracy in average) than the
two aforementioned systems, due to the WordNet’s better information organization compared
to those large corpus or web pool used in that two systems.

The initial success of Sentiment Analysis at the word level provides the necessary building
blocks for studying higher text unit granularity, as demonstrated in (Wiebe et al., 1999) and
(Bruce & Wiebe, 2000). Both papers established a positive and statistically significant cor-
relation with the presence of adjectives on determining the subjectivity of sentences (as well
as documents). Furthermore, for determining sentence sentiment orientation, many approaches
(Yu & Hatzivassiloglou, 2003),(Kim & Hovy, 2004) aggregated the polarity of each individual ad-
djective or sentimental word appeared within the sentence itself. Later approaches by (Wiebe &
Riloff, 2005),(Wilson et al., 2005),(Kim & Hovy, 2006) introduced additional sentence-surface
features (e.g. counts of positive/negative adjectives in current sentence, or in a window of pre-
vious and next sentence; binary feature whether the sentence contains a pronoun, etc.) in a
supervised setting to achieve fairly good results (up to 70% accuracy) in the same task.
Nevertheless, in the domain of product reviews, knowing the orientation of the sentence as a whole is not enough. In fact, it is necessary to identify the semantic of the opinion holder in the sentence, as this opinion holder may correspond to a particular facet of the subject in review that users are interested in. Examples of facets that belong to a camera product would typically be: battery life, lens, flash system, price, etc. In case of a music player, the facets are: sound system, battery life, weight/size, storage capability, etc. Previous work by (Hu & Liu, 2004b) tackle this problem by first applying data mining techniques to extract facets of the product, then classifying the orientation for each of the sentences that the facets appear as positive or negative, using the WordNet framework discussed earlier. The performance of this system was very promising with 72% accuracy in identifying product facets, and 84% in facet orientation prediction. Subsequently, (Popescu & Etzioni, 2005) introduced the use of relaxation labeling technique (Hummel & Zucker, 1987) in their OPINE system to help determine facet orientation, and achieved a better result of 78% accuracy. The improvement is obtained due to the consideration of neighboring facets that appear in the same sentence as the target facet, based on surface linguistic connective cues such as conjunction and disjunction. More recently, the work by (Ding et al., 2008) further incorporated a set of complex carefully-built grammar rules between adjacent sentence constructs as well as neighboring facets, together with a collection of comprehensive polarity-annotated lists of idioms, nouns, verbs, adjectives and adverbs, to solve the same problem. The system achieved significant results of 92% accuracy that closely matched with the standard of human perception, and is currently the state of the art.

Having discussed many works in Sentiment Analysis, it is important to re-emphasize that these techniques only enable us to discover the user opinions in the review, but do not attempt to aggregate these pieces of information together. The latter goal is achieved by summarization techniques, which are described in the next section below.

2.2 Summarization

To some extent, many works discussed in previous section are also capable of producing some forms of summarization: Turney’s system (Turney et al., 2002) produced a thumbs-up/thumbs-
down indication for movie reviews as the output of its orientation classification component. The systems by (Hu & Liu, 2004b) and (Popescu & Etzioni, 2005) generated facet-driven summary, supplied with sentence-level statistics, i.e. the number of positive/negative sentences that the facet belongs to. Subsequent work by (Liu et al., 2005) extends the single facet-driven summary into comparative-based summary between many products, where the orientation of all shared facets are plotted together with their number of supporting sentences for visualization. While users may prefer these systems for an at-a-glance presentation of products, we argue that they actually provide too limited information. To illustrate, users can learn that more people prefer/hate a particular product facet; however, they cannot figure out the underlying reasons for such preference/dislike unless they actually read the full reviews. As a result, our aim of building a system that is capable of summarizing not only sentiments but also the textual supporting reasons would help clarify the aggregated sentiment rating for many users.

Summarizing product reviews is more relevant to techniques in multi-document summarization, since we do not consider a single review alone but a set of reviews about a particular product. The major problem that multi-document summarization technique needs to take into consideration is the redundancy of data, where similar information may appear across different sources. Early works by (FRUMP), (Radev & McKeown, 1998) applied Information Extraction techniques to gather information from different inputs, and generated summary by filling those extracted information into some predefined sentence templates. However, this framework requires significant background knowledge in order to create the template at a suitable level of details, and hence is domain dependent. Later work by (Barzilay et al., 1999) suggests a novel approach that does not depend on domain knowledge. In their system, each sentence is first transformed into a predicate-argument structure called DSYNT tree (Kittredge & Mel’Cuk, 1983) with the nodes being the sentence constituents. Under this representation, grammar dependencies between sentence constituents (subject-verb relation, adjective-noun relation, etc.) are captured and essentially abstracted from their ordering in the sentence. Therefore, with the assistance of a set of paraphrasing rules that are capable of recognizing identical or similar predicates, they were able to derive rules to combine similar DSYNT trees of sentences from
different sources together. The resulting tree is fed to a final sentence generation component to formulate a new sentence. However, all aforementioned approaches are not suitable to apply to our domain of product reviews directly. The reasons being that these techniques were previously experimented on the news domain, where almost every sentences when reporting news events are well structured and grammatically correct. Obviously users writing product reviews do not need to conform to any sentence standardizations, and thus give rise to many ill-formed sentence structures that may need special treatments. Furthermore, there is no sentiment information involved in the news domain as opposed to product reviews.

Another widely used technique for multi-document summarization is developed from the concept of Maximum Marginal Relevance in (Carbonell & Goldstein, 1998a). Many systems (Carbonell & Goldstein, 1998b),(Ye et al., 2005) have leveraged MMR to solve their summarization problem in general news domain and obtained reasonable results. In details, MMR is an iterative algorithm, which selects one sentence from the collection per round to insert into the final summary based on the following two criteria: 1. the selected sentence covers the most information mentioned by the remaining unselected sentences. 2. It also has minimum similarity with all previous selected sentences in the summary. The algorithm either terminates when a fixed number of sentences is selected, or when the content overlapping between any candidate sentence and the summary at that iteration exceeds a predefined threshold. Obviously the algorithm requires a metric to compute the content similarity between any two sentences, but when come to our domain of product reviews that exhibits both content and sentiment information, it is unclear on how to define such a good metric.

In short, to the best of our knowledge, none of the existing systems has really combined sentiment analysis with summarization techniques to generate product review summary. Therefore, it is time for us to investigate this problem and contribute a system that would incorporate the results from both research areas, aiming to produce reasonably good summary.
Chapter 3

Product Facet Identification

An immediate question that one may ask is whether we should tackle this task of identifying product facet automatically, as it seems intuitive to ask the seller or the manufacturer of the product to provide this list of facets directly. While this is a possible approach, it presents a number of problems:

- It is hard or even impossible to obtain a complete list of facets, as users may comment on facets that the manufacturers have never thought about. Consider an example where iPhone user comments about the usefulness of using the device as an alarm clock, but this function is never introduced on Apple official website.

- Even for those more common facets, users may use different set of words to describe them. For instance, the manufacturers or the sellers often mention the price of their product, while users tend to use the term value to mean the same thing.

- The manufacturers may not want to include those weak facets of their product (e.g., iPhone cannot view flash content on the Web, or the plastic construct of the camera body, etc.), while these information obviously are of high interest to the users.

There are existing systems that have tackled this product facet identification problem. In (Hu & Liu, 2004a), they applied data mining technique and achieved an accuracy of 72% on a set of electronics reviews. Later in (Popescu & Etzioni, 2005), their OPINE system incorporated
web mining technique similar to that in (Turney et al., 2002), and achieved a 22% improvement in accuracy on the same data set as in Hu & Liu’s system. However, it is not clear from their paper how they constructed queries that combine a set of cue words associated with the product class (e.g., “of camera”, “camera has”, “camera comes with”, etc.) and the candidate facet together. Our early experiments with different query combinations also do not show consistent results. Therefore, we decide to only re-implement a system following Hu & Liu’s approach. Nevertheless, we manage to introduce the use of sentence syntactic role as a feature that helps improve the system and achieve result close to that of Popescu & Etzioni’s system.

3.1 Preliminary

Product facets can be expressed explicitly or implicitly. Consider the following two sentences from a camera review:

1. The pictures of this camera are very clear.
2. The camera fits nicely into my palm.

Figure 3.1: Example sentences showing explicit/implicit product facets

In the first sentence, the user expresses his/her satisfaction about the picture quality of the camera, and we can say that the word picture is a facet of the camera. On the other hand, the second sentence discusses about the size of the camera. However, the word size does not appear explicitly in the sentence. In order to identify implicit product facets, deep semantic understanding of the domain is needed, which implies a more sophisticated algorithm is required. Fortunately, explicit facets appear much more frequently in the review than implicit ones. Therefore, we follow the same assumption as in (Hu & Liu, 2004a) as well as (Popescu & Etzioni, 2005) mentioned earlier, so that only product facets that appear as nouns or noun phrases will be considered. Note that for noun phrase, the right most word must be a noun and is called the head noun, while the rest of the words can be either nouns or adjectives (e.g., battery life, external flash). Since we barely observe too long noun phrases in the review, in this work, we only capture noun phrases with no greater than 3 adjacent words to the left including
the head noun. The goal of our product facet identification component is thus to capture these explicit product facets.

3.2 Methodology

3.2.1 Architecture Overview

Figure 3.2 provides an overview of our product facet identification component. The inputs to this component are sentences from a set of product reviews. An review typically includes a text body and a title, together with additional information such as date, time, author name, star-based ratings. In this project, we do not retain any of these information, neither do we capture which review the sentence is originated from nor the relative positions between sentences within the same review. We first preprocess these sentences with a Part-of-Speech (POS) tagger to obtain the POS label of each word. In the next step, only those words that received the label as Noun or Adjective are collected and fed to the association mining module, which would generate a list of candidate frequent product facets. The pipeline is followed by some post processing operations in order to remove some redundant results. Finally, opinionated sentences that contain product facet are extracted.
3.2.2 Preprocessing

Part-of-Speech tagging

In this work, we utilize Stanford POSTagger\(^1\) to process each input sentence and yield the part-of-speech (POS) label for each word. We observe that the tagger performs fairly well on getting the correct label for nouns/noun phrases, even though there are a number of odd-structured sentences presented in the reviews. We do not consider stopwords in the tagging results, while the remaining noun/noun phrases are also converted to their stemmed version using an implementation of Porter stemming algorithm (Porter, 1997). The following shows a sentence with the POS tag (*NN, *JJ* are labels for noun and adjective respectively):

*I/PRP recommend/VB this/DT camera/NN for/IN excellent/JJ picture/NN quality/NN .*

\(^1\)http://nlp.stanford.edu/software/pos tagging.shtml
Syntactic roles

However, while inspecting the candidate noun/noun phrases, we observe many noisy results such as: “light”, “hand”, “time”, “month”, “hour” etc. These are nouns that do not get filtered away from our minimal stopword list, and yet are likely to appear quite a number of times in the reviews. Therefore, we introduce syntactic role within a sentence as feature that help distinguish a genuine product facet from those noisy results. Consider the following motivated sentences that have been parsed by Stanford Dependency Parser:\footnote{http://nlp.stanford.edu/software/lex-parser.shtml}:

1. The larger lens of the g3 gives better picture quality in low light.
   \[\ldots, \text{nsubj}(\text{gives}-7, \text{lens}-3), \ldots, \text{dobj}(\text{gives}-7, \text{quality}-10), \ldots\]
2. When I took outdoor photos with plenty of light, the photos were awesome.
   \[\ldots, \text{dobj}(\text{took}-3, \text{photos}-5), \ldots, \text{nsubj}(\text{awesome}-14, \text{photos}-12), \ldots\]
3. My fiance just did not like the size, it is so small in her hand.
   \[\ldots, \text{dobj}(\text{like}-6, \text{size}-8), \ldots\]

Figure 3.3: Example sentences together with their syntactic roles

From the examples, we observe that genuine facets tend to appear as either subjects or objects. In fact, our analysis on a subset of camera reviews (more than 300 sentences that contain some facets over 24 reviews) shows that more than 90% of the instances agree with the above observation. This is not too surprising since users often structure their review and guide the readers using product facets as the main topical words. More importantly, this suggests that we can remove those nouns/noun phrases that do not appear as subject/object during the preprocessing step and only deliver those legitimate ones to the association mining module.

3.2.3 Association Mining

In this component, association rule mining technique proposed by (Agrawal & Srikant, 1994) is used to statistically identify all the frequent explicit product facets. Before we draw the relation between association rule mining and our domain of interest, we outline the general
descriptions of this technique as follow:

**Items**: item is the smallest entity being considered in a particular domain of interest. An itemset is a set of items, and the set of all items is denoted as $I$.

**Transaction**: transaction $t$ contains itemset $X$ if $X \subseteq t$. The set of all transactions is denoted as $D$.

**Association rule**: $X \Rightarrow Y$ where $X \subseteq I$, $Y \subseteq I$ and $X \cap Y = \emptyset$.

**Support**: $\text{supp}(X)$ is the number of transactions in $D$ that contain itemset $X$. If applied to a rule, $\text{supp}(X \Rightarrow Y) = \text{supp}(X \cup Y)$.

**Confidence**: $\text{cond}(X \Rightarrow Y)$ is the number of transactions in $D$ that contain itemset $X$ if only contain itemset $Y$.

The problem of mining association rules is then stated as generating all possible rules that have support and confidence greater than the user-specified minimum values. The Apriori algorithm (Agrawal & Srikant, 1994) solves this problem in two phases: (a) Identify all frequent itemsets that satisfy the minimum support, (b) Generate rules from those discovered frequent itemsets that satisfy the minimum confidence.

In applying to our work, items are nouns/noun phrases extracted from the previous step and transactions are sentences containing those nouns/noun phrases. We only need to run the first phase of the Apriori solution in order to obtain the set of frequent itemsets, or equivalently the set of candidate frequent product facets. At the same time, we also conveniently obtain the ranking for this set of candidate frequent product facets simply based on their support values. This ranking would be of high importance when we come to the summarization module on how to present the information to the users.

### 3.2.4 Post Processing

As we are considering a large portion of possible nouns/noun phrases appeared in the review, not all of them are genuine facets; i.e. some of them are not interesting or redundant. Therefore, this step of post processing aims to remove those incorrect facets by applying the following rules:
Usefulness pruning: This criterion focuses on removing single-word facets that are likely to be meaningless. For example, in the context of camera reviews, *life* by itself is not a useful facet, while *battery life* is a meaningful facet. We can resolve this problem by computing the pure support of a facet $f$, which is defined as the number of sentences that $f$ appears alone without being subsumed by any other facets. If this number is below a certain threshold, there is a strong evidence that we can just keep the superset of $f$ as the useful facet.

Compactness pruning: This criterion in turns targets redundant facet phrases - noun phrases that are discovered as facets. For example, *photo pixel, sample image* are not as compact as *pixel* and *image*. For each individual words that the phrase contains, we compute the ratio between the support of the phrase and the support of that individual word. If any of these ratios falls below a threshold, we will prune away the facet phrase.

Hu & Liu’s also applied an additional heuristic rule to identify infrequent explicit facets, i.e. genuine facets that are not mentioned a lot in the reviews.

Infrequent facet discovery: Our association mining module would not be able to discover these infrequent facets, as they have fairly low support value. However, when comment about product facets, users tend to put similar opinion words. To illustrate, consider the following two sentences: (1) The camera gives absolutely amazing pictures. (2) The accompanied software is amazing. In the first sentence, *Picture* is a frequent facet that has been indentified by our association mining module, while *software* in the latter sentence is an infrequent one. Nevertheless, we observe that they share the same adjective ‘amazing’. Hence the heuristic works in two steps: first gather all opinion words that modify frequent facets. Then for each sentence, if it contains no frequent facet but one or more opinion words, the nearest noun/noun phrase being modified is included as a facet.

3.2.5 Sentence Extraction

Sentences that contain any of the product facets that we have discovered are labeled with that corresponding facet. A sentence can receive more than one label, as that sentence may
discuss a relation between many facets. The following instances show sentences being labeled with one and two product facets respectively:

1. The **lens** blocks the **viewfinder** when the lens is set to wide angle.

2. The 10 **megapixels** produces really sharp **pictures**.

Figure 3.4: Example sentences labeled with multiple facets

However, not all labeled sentences are sent down to the summarization component. Only those opinionated sentences are chosen, since we are more concerning with summarizing users opinions in this work. We build a module that applies sentiment analysis technique to filter sentences as follows:

**Sentiment analysis**: As mentioned in the related work section, (Ding, Liu, & Yu, 2008) currently provides the best result for classifying the sentence orientation with respect to a facet as opinion holder. Unfortunately, their system has not been made available to the research communities. Reimplementing this system is also very difficult, due to its complicated hand-made grammar rules and sentiment word lists. As a result, we only follow the approach in (Hu & Liu, 2004b) to grow a seed list of known-polarity adjectives using synonym/antonym pointers in WordNet and cover the other unknown adjectives. The sentence polarity is then determined as the summation of all subjectivity scores of those adjectives in the sentence. If the resulting summation score is positive (negative), the sentence is classified as the corresponding orientation.

### 3.3 Experiments

#### 3.3.1 Experimental Data

We obtain product reviews for three electronics products: 1 digital camera (Canon G3), 1 DVD player (Apex) and 1 cellular phone (Nokia 6610) using the same data set as in (Hu & Liu, 2004a). All of their reviews are collected from Amazon.com, a merchant site that has a large number of reviews for many products. The data set also conveniently comes with human
annotation – a manual facet list (both explicit and implicit) for each product. We evaluate the output of our system against this manual facet list.

### 3.3.2 Evaluation Measure

We use the standard precision and recall measures to evaluate the performance of our system. Precision can be seen as a measure of exactness, whereas recall is a measure of completeness. In our context, they are defined as follow:

\[
\text{precision} = \frac{|\{\text{Manual facets}\} \cap \{\text{Extracted facets}\}|}{|\{\text{Extracted facets}\}|} \quad (3.1)
\]

\[
\text{recall} = \frac{|\{\text{Manual facets}\} \cap \{\text{Extracted facets}\}|}{|\{\text{Manual facets extraction}\}|} \quad (3.2)
\]

### 3.3.3 Experimental Results

<table>
<thead>
<tr>
<th>Product</th>
<th>No. of manual facet</th>
<th>Association mining</th>
<th>Usefulness pruning</th>
<th>Compactness pruning</th>
<th>Infrequent facet discovery</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Recall</td>
<td>Precision</td>
<td>Recall</td>
<td>Precision</td>
<td>Recall</td>
</tr>
<tr>
<td>Camera</td>
<td>79</td>
<td>0.671</td>
<td>0.552</td>
<td>0.658</td>
<td>0.634</td>
</tr>
<tr>
<td>Phone</td>
<td>67</td>
<td>0.731</td>
<td>0.563</td>
<td>0.716</td>
<td>0.676</td>
</tr>
<tr>
<td>DVD</td>
<td>49</td>
<td>0.754</td>
<td>0.531</td>
<td>0.754</td>
<td>0.634</td>
</tr>
<tr>
<td>Average</td>
<td>65</td>
<td>0.719</td>
<td>0.549</td>
<td>0.709</td>
<td>0.648</td>
</tr>
</tbody>
</table>

The figure shows the performance at each snapshot of the system. We purposely leave out the use of syntactic role during the preprocessing step in this first evaluation, as we want to show the effectiveness of our proposed feature in improving the original system. Our system (without syntactic role) follows closely all the steps in Hu & Liu’s original paper, hence we obtain results similar to those reported by them. In general, the system performs quite well in identifying product facet. As we look at the results, we can actually identify most of the more common facets such as: battery, picture, lens for camera, signal, headset for phone and remote control, format for DVD player.

Next, we evaluate our system again, including the syntactic role preprocessing step.
We observe that both systems share the same recall, while the system using syntactic role has a higher precision. This result justifies that the syntactic role information has successfully prevent some noisy nouns/noun phrases from being considered as facets. In fact, it never made a mistake from our three data set. During the last step, when we consider new infrequent facets, the system also does not include the closest noun (to the opinion word) if that noun is not subject/object of the sentence, hence further improves the overall accuracy. Nevertheless, this approach would not perform well if we only have a small set of reviews, since it relies purely on the statistical information exist the input.
Chapter 4

Summarization

Taking the collection of sentences that belong to different individual facets from the Product Facet Identification component as input, this component aims to construct a facet-based summary. This way of structuring the output summary is similar to the approach taken by (Hu & Liu, 2004a), and in fact identical to those commercial systems like Bing Shopping\(^1\) and Google Products\(^2\). A snapshot of these systems is reproduced as follow:

\(^1\)http://www.bing.com/shopping
\(^2\)http://www.google.com/products
Figure 4.1: A reproduced snapshot of the output summary generated by existing systems

In this example, we observe that the system is capable of aggregating positive and negative opinions for each product facet. However, we also notice that not all sentences under each polarity cluster are mentioning about the same topic. Consider the positive cluster under the facet *lens*: the first sentence (“The lens feels very solid!”) comments about the quality of the material that made up the lens, while the second sentence (“I have taken a whole bunch of excellent pictures with this lens.”) mentions about the capability of the lens that produces excellent photos. Similarity, in the negative cluster of *lens*, the first sentence (“I do not satisfy with the included lens kit”) expresses user’s dissatisfaction with the lens kit, while the second sentence (“The lens cap is very loose and can come off very easily!”) talks about the cap of the lens being very loose. In the rest of this chapter, we will refer these different topics within the set of sentences belong to a particular facet as subtopics. As a result, users who prefer more detailed information behind the positive/negative sentiment will need to go through every sentences in the correspondent cluster. Since a cluster can potentially include a few hundred sentences for those large reviews, this approach just consumes too much time and effort.
Therefore, we propose a system that enhances the existing facet-based summarization system with the following two additional features:

(a) **Subtopic clustering:** this feature groups together sentences within each product facet that mention about the same underlying subtopics. It is important to note that subtopic is not equivalent to cluster of positive and negative sentiment of the facet. In fact, it is possible for a subtopic to contain a subset of sentences from both sentiments.

(b) **Compact presentation:** Having captured the subtopics, we need a way to present the information to the users. This feature suggests an output similar to figure 4.2 below. To achieve this, the system first determines the sentiment information for each sentence belong to the subtopic. It then shows only one representative sentence (together with the aggregated count of the rest) for each polarity under that subtopic.

```
a/ Lens
   (+) The lens is excellent in terms of value returned! (and 10 similar sentences)
   (-) I think the lens does not worth it. (and 2 similar sentences)

   (+) The lens is very easy to handle. (and 7 similar sentences)
   (-) I cannot really adjust the lens to what I want. (and 0 similar sentences)

   ...

b/ Battery
   (+) ...
   (-) ...
```

Figure 4.2: A snapshot of the output summary produced by our proposed system

In the next section, we discuss our methods for solving the two tasks corresponding to the two above features.
4.1 Methodology

4.1.1 Architecture Overview

Figure 4.3 provides an overview of our summarization component. The inputs to this component are clusters of sentences belong to each of the product facets obtained from the previous product facet identification component. We preprocess this list of facets to remove some facets that do not worth summarizing. After this step, we start considering sentences under each facet independently from others. The sentences will be sent to a sentence clustering module, which is developed to tackle the subtopic clustering problem. The algorithm proceeds in two steps: first define a sentence representation for the similarity score computation between any two sentences, then combine similar sentences to form clusters based on some criteria. The output from this module is fed to the second module, which applies sentiment analysis and summarization techniques to generate the desired summary.
4.1.2 Preprocessing

General entity pruning

Refer back to the data set that we adopted from (Hu & Liu, 2004a), we note that the words “camera”, “phone” and “DVD” are all annotated as genuine product facets. While this is fine in the context of identifying product facet, it is not desirable for summarization. Consider the facet “camera”, a summary for this facet essentially means summarizing the whole camera product. The reason being that the word “camera” is always used as a general entity to accompany or to describe other genuine facets. (e.g., (1) The camera has a very long battery life. (2) It is a really powerful camera.)

Similarly, different brand names that belong to a particular product class (e.g., Nikon, Canon (Camera); Pioneer (DVD); iPod (Music Player), etc.), or product/manufacturer names of the
accessories that go together with the main product (e.g., Kingston (compact flash card for camera), Nvidia (graphic card for computer, etc.), are all treated as genuine facets in the annotation from the data set. However, in most cases, they appear together with some other facets when comparison is made between that product and its competitors (“My Canon camera has longer battery life than Nikon”), or just as general references to express general opinion (“Canon is amazing !!!”). As a result, we argue that these general/proper entities are not very useful for summarization and should be excluded.

Because of the time constraint, we do not specifically build a module to recognize these proper names automatically. However, we argue that it is not difficult to manually look up a large portion of this list of entities if we know the product class in advance. We indeed observe this assumption used by Popescu & Etzioni in their OPINE system, for the construction of the queries used in their web mining technique.

**Similarity pruning**

Users may employ different synonymous words to mention the same facet. For example: *picture* versus *image*, *photo*; or *screen* versus *monitor*. However, they are treated as different genuine facets in both Hu & Liu’s system. As a result, different pieces of summary for the same facet will be produced, which is not desirable. To resolve this, we apply word semantic similarity measurement as proposed in (Kong, Liu, Zhou, & Zheng, 2007) to compute the similarity score between any two candidate facets. If the score is above a threshold, the two words (and hence their correspondent sentences) will be combined together.

In (Kong et al., 2007), they constructed a edge-counting based model that considers the depth of least common subsumer and the shortest path length between any two words in WordNet (Miller et al., 1990). Formally, given two words $w_1$ and $w_2$, the semantic similarity $s_w(w_1, w_2)$ is calculated as:

$$s_w(w_1, w_2) = \frac{f(d)}{f(d) + f(l)}$$  \hspace{1cm} (4.1)$$

where $l$ is the shortest path length between $w_1$ and $w_2$, $d$ the depth of the least common subsumer in the WordNet hierarchical semantic net, and $f(x)$ the transfer function for $d$ and
l. For \( s_w(w_1, w_2) \), the interval of similarity is \([0, 1]\), 1 for the maximum similarity and 0 for no similarity at all. We follow the experimental results in (Kong et al., 2007) and choose \( f(x) = e^x - 1 \). The resulting formula is:

\[
 s_w(w_1, w_2) = \frac{e^{\alpha d} - 1}{e^{\alpha d} + e^{\beta l} - 2} \quad (0 < \alpha, \beta < 1)
\]

where \( \alpha, \beta \) are smoothing factors. As reported in (Hammerton, Osborne, Armstrong, & Daelemans, 2002), the optimal values are: \( \alpha = 0.25 \) and \( \beta = 0.25 \).

It is important to note that only nouns and verbs organize their synsets in a hierarchical manner, with the network of nouns far deeper than verbs. Adjectives are organized into many distinct clusters, while adverbs are defined in terms of the adjectives they are derived from, and thus inherit their structure from that of the adjectives. As a result, the above formulas will only give reasonable score for input as a pair of noun. Nevertheless, it can still apply well in our setting, as explicit facets are nouns under our definitions.

### 4.1.3 Sentence Representation and Similarity Measurement

In this work, we decide to adopt a simple yet novel sentence representation, together with a sentence similarity measurement scheme as proposed in (Li et al., 2006), which yielded state-of-the-art results. In a high-level view, the algorithm utilizes a dynamic vector representation that adapts well with the size of the sentence, and computes the similarity score as the cosine distance between two sentence vectors.

The algorithm starts with identifying "concepts" in the sentence. Concepts are defined as those open class words (nouns, verbs, adjectives and adverbs, excluding stopwords) in the sentence. We also adopt the observation in Section 3.2.2 so that we only include those words that hold important syntactic roles in the sentence. In details, we consider important nouns being subject or object, main verbs/adjectives associating with those important nouns, adverb modifying the main verb/adjectives. Then given two sentences that need to compute similarity, \( s_1 \) with the set of concepts \( C_1 \), and \( s_2 \) with the set of concepts \( C_2 \), a joint concept vector is defined as \( C = C_1 \cup C_2 \). In the next step, \( V_i \) – the vector representation for \( s_{i, i=1,2} \) – is created, with size equal to that of \( C \), and values determined by the following rule:
At index $k$,

- If $s_i$ contains $C[k]$ – concept at index $k^{th}$ in the joint vector, $V_1[k]$ is set to 1.0.

- If $s_i$ does not contain $C[k]$, a semantic similarity score is computed between $C[k]$ with all concepts in that sentence. $V_i[k]$ is then set to the highest similarity score. However, as mentioned in section 4.1.2, the method by (Kong et al., 2007) only works for nouns in WordNet. Moreover, there is also no other existing work that helps us compute the rest of the scores reliably. Therefore, In our work, we decided to manually build a small dictionary that helps us obtain the score for any two synonyms or antonyms, regardless of the part-of-speech. While this manual step is obviously not desirable, we argue that this dictionary is an one-off task, and thus worth the effort for benefiting the current as well as future system.

At this point, it is important to note that the vector representation for a particular sentence is not fixed, as it depends on which sentence we need to compute similarity score with. Therefore, the vectors scale effectively for each pair of sentences. The semantic similarity between two sentences can now be measured by the cosine distance between the two representative vectors, which results in a score within the range $[0, 1]$ inclusively, with 0 means zero similarity and 1 means maximum similarity.

$$sim(s_1, s_2) = \frac{V_1 \cdot V_2}{\|V_1\| \|V_2\|}$$

(4.3)

In the following, we reproduce an example for clarifying the above steps:
Consider the following two sentences with the contained concepts underlined:

\[ s_1 = \text{The battery of this camera is very impressive.} \]
\[ s_1 = \text{Canon camera always has a long battery life.} \]

Therefore, the joint vector is:

\[ C = \{ \text{battery, camera, impressive, has, long, life} \} \]

The resulting sentence vectors \( V_1, V_2 \) are:

\[ V_1 = \{1.0, 1.0, 1.0, 0.0, 0.3, 0.15\} \]
\[ V_2 = \{1.0, 1.0, 0.3, 1.0, 1.0, 1.0\} \]

The semantic similarity between two sentences is thus:

\[ \text{sim}(s_1, s_2) = 0.69 \]

Figure 4.4: Example of sentences together with their vector representation

### 4.1.4 Sentence Clustering

Once similarities between any two sentences have been calculated, we feed them to the sentence clustering module. Since there is no literature in this new problem, we implement both hierarchical as well as non-hierarchical algorithms to compare their performances.

#### 1/ Hierarchical clustering

We apply hierarchical clustering in an agglomerative (bottom-up) manner. Sentences are initially treated as singleton cluster, and at each iteration, the algorithm would successively merge clusters with the minimum pairwise distance together, until a terminating criterion is reached.

There are three variations where the pairwise cluster distance can be computed, namely Complete-link, Single-link and Groupwise-average. However, our early experiment shows that Groupwise-average distance performs more consistently. Given \( c_i \) and \( c_j \), being two different clusters at the time of consideration:

\[
\text{Groupwise-average} \quad \text{sim}(c_i, c_j) = \frac{1}{|c_i \cup c_j| - |c_i \cap c_j| - 1} \sum_{x \in c_i \cup c_j} \sum_{y \in c_i \cup c_j ; y \neq x} \text{sim}(x, y)
\]
Many small clusters would result in an overly detailed summary and an over-estimation of the actual subtopics, while a few large clusters would result in a summary that omits important information. Therefore, it is very important to determine the terminating criterion. In this work, we adopt the algorithm proposed in (Hatzivassiloglou et al., 2001) to estimate the final number of clusters. The clustering process will terminate as soon as the number of clusters exceeds this value. In (Hatzivassiloglou et al., 2001), they first defined the notion of links: if the semantic similarity score between any two sentences are above a certain threshold, a link exists and joins that two sentences. Therefore, if we compute the similarity score for every two sentences in the collection and apply the notion of link, a graph with the vertex being sentences, and edges representing those links will be created. Finally, the number of estimated clusters $c$ given the input of $n$ sentences corresponding a graph with $m$ connected components is determined as

$$c = m + \left( \frac{n}{2} - m \right) \left( 1 - \frac{\log(L)}{\log(P)} \right)$$

(4.4)

where $L$ is the observed number of links and $P = n(n-1)/2$ is the maximum possible number of links.

2/ Non-hierarchical clustering

We also implement a non-hierarchical clustering technique – the exchange method (Spath, 1985), which casts the clustering problem as an optimizing task. The algorithm seeks to minimize an objective function $\Phi$ that measures the intra-cluster dissimilarity between a partition $P = C_1, C_2, \ldots, C_k$:

$$\Phi(P) = \sum_{i=1}^{k} \left( \frac{1}{|C_i|} \sum_{x,y \in C_i, x \neq y} (1 - \text{sim}(x, y)) \right)$$

(4.5)

The same estimation on the number of final clusters mentioned earlier is first applied to determine the size of the partition $P$. The algorithm then proceeds by creating an initial assignment of the sentences into the partition, and looking for locally optimal moves or swaps of sentences between clusters that improve $\Phi$ in each iteration, until convergence is achieved. Since this is a hill-climbing method, it is necessary to call the algorithm multiple times, with random partition of sentences into the clusters each time. The best overall configuration will be selected as the
4.1.5 Compact Presentation

This step aims to generate and present to the users a summary similar to that in figure 4.2. It takes in sentence clusters from all facets, with each of them delivered by the previous component after subtopic clustering. By applying the same sentiment analysis module as discussed in Section 3.2.5, we are able to determine the orientation for every sentences in a particular subtopic. With this information, we are able to partition the sentences in each subtopic based on their polarity. The remaining task is to select the most representative sentence for each partition. The selected sentence must gather most of the information of the other sentences, or in other words, that sentence is most similar to all the remaining sentences. Thus we define a metric to compute the representative power of a sentence as follows:

For each sentence $s_i$ in the correspondent positive/negative partition $P$, we define its representative power $Rep(s_i)$ as follow:

$$Rep(s_i) = \sum_{s_j \in P - s_i} sim(s_i, s_j)$$

(4.6)

The sentence with the highest representative power will be selected as the displayed sentence.

Last but not least, regarding the outermost structure in the final output summary, we organize facets based on their supporting values as obtained in Section 3.2.3.

4.2 Experiments

We have described a complete two-module solution in the summarization component, as well as the initial preprocessing step. However, in this section, we will only evaluate the effectiveness of the first module. The reasons being that: (a) Subtopic clustering targets a unique new problem that we discovered in summarizing product reviews. (b) The second module on compact presentation only serves as a post processing step for the output from the first module. (c) We do not have enough resources to conduct a human evaluation, in order to assess the preference in the way we present our final summary against those produced by other product review
summarization systems.

With our focus to the subtopic clustering module, we essentially want to study the following questions through the experiments:

1. We have seen examples showing the existence of subtopics in the introduction section of this chapter. But in general, does subtopic really exist? If yes, what is the number of subtopics in average?
2. Are the two proposed clustering algorithms able to solve this problem with a good word semantic similarity score? If yes, which algorithm gives a better overall performance?
3. How does the number of subtopics affect the clustering algorithms?

4.2.1 Experiments Data

From the output obtained from previous component on product facet identification, we extract 22 sentence clusters that belong to those facets being discussed a lot by users. Then we manually discover the subtopics within each cluster, as well as label the sentences with their corresponding subtopics. These annotated data will serve as the gold standard for evaluating the output produced by the each of the two proposed clustering algorithms.

4.2.2 Evaluation Measure

We use the standard purity and inverse purity clustering measures to evaluate the performance of the two algorithms. Purity is related to the precision measure, well known in Information Retrieval. This measure focuses on the frequency of the most common category in each cluster, and rewards the clustering algorithm that introduce less noise in each cluster. Being $C$ the set of automatic clusters to be evaluated, $L$ the set of manual annotated clusters and $n$ the number of sentences to be clustered, purity is computed by taking the weighted average of maximum precision values:

$$
purity = \sum_i \frac{|C_i|}{n} \max \text{Precision}(C_i, L_j) \quad (4.7)$$
where the precision of an automatic cluster $C_i$ for a given manual subtopic $L_j$ is defined as:

$$\text{Precision}(C_i, L_j) = \frac{|C_i \cap L_j|}{C_i}$$

(4.8)

Inverse Purity focuses on the cluster with maximum recall for each category, rewarding the clustering solutions that gather more elements of each category in a corresponding single cluster. Inverse Purity is defined as:

$$\text{Inverse Purity} = \sum_i \frac{|L_i|}{n} \max \text{Precision}(L_i, C_j)$$

(4.9)

For the final ranking of systems we used the harmonic mean of purity and inverse purity $F_{\alpha=0.5}$. The F measure is defined as follows:

$$F_{\alpha} = \frac{1}{\alpha \text{Purity} + (1 - \alpha) \text{Inverse Purity}}$$

(4.10)

$F_{\alpha=0.2}$ is included as additional measure given more importance to the inverse purity aspect. The rationale is that, for a user to our summarization system, it should be easier to discard a few incorrect results in a cluster containing all the information needed, than having to collect the relevant information across many different clusters. Therefore, achieving a high inverse purity should be rewarded more than having high purity.
4.2.3 Experiments Results

In the rest of this section, we will address each of our earlier questions based on the obtained experiment results.

**Subtopic existence**

As illustrated in the 3rd column in each of the above tables, our human annotation shows the existence of facet subtopics, with an average of 3.5 subtopics per facet among all three products. It confirms the motivation behind our methods in solving the subtopic clustering problem.
However, different facets contain different number of subtopics. Consider the facet \textit{Price} belong to the \textit{DVD} product, it essentially has no subtopic. The reason being that users only express their opinions toward two extremes on whether the DVD player is expensive or affordable (recall that subtopic is independent with sentiment information). Similar to the facet \textit{Format} on the same \textit{DVD} product, where users only discuss whether the DVD player can play all video formats or not. On the other hand, those facets having a lot of subtopics (Camera’s \textit{lens}, Camera’s \textit{LCD}, etc.) are due to the fact that they exhibit many different properties (the size, ease of use, price, etc. of the \textit{lens}, or the resolution, material, color, etc. of \textit{LCD}). Therefore, users have more freedom to discuss on any of these subtopics. We also observe that the same facet \textit{Customer Service} mentioned in \textit{Phone} experiences more subtopics compared to that mentioned in \textit{DVD}. This is because \textit{Phone} users tend to compare the customer service between many providers, while \textit{DVD} users only complain about the service of the player’s producer in the review.

Interestingly, the number of subtopics not only varies from facet to facet, but also from product to product. In our data, the product \textit{Camera} exhibits the greatest number of subtopics (an average of 5 subtopics per facet), while \textit{DVD} only contains an average of 2 subtopics per facet. This again can be explained from the above observation: the facets that belong to the \textit{Camera} usually have richer properties to be commented on compared to those belong to \textit{DVD}. However, this actually creates an impact to the performance of our clustering algorithms, as we discuss later in section 4.2.3.

\textbf{Clustering performance}

We compare the performance of our two algorithms against a baseline, which randomly assigns sentences to clusters. Recall that the number of clusters for all clustering solutions is predetermined by the estimation in formula (4.4). In that formula, we use the average sentence similarity score within the whole sentence cluster as the threshold for link creation.

The random clustering baseline is executed 200 times, then we record the average performance. For the non-hierarchical clustering approach, we also execute the algorithm 200 times,
in order to prevent it from trapping in the local peak solution. We record the run that minimizes the objective function in equation (4.5) the best. Finally, we need to execute the hierarchical clustering algorithm only once, as it is a deterministic algorithm given the estimated number of final clusters.

The last row in all three tables shows the relative performance of the proposed algorithms with respect to the random baseline. From all tables, it is clear that our two proposed clustering algorithms always outperform the random baseline by a large margin.

On the other hand, we notice no subtle difference in average performance between the hierarchical approach and the non-hierarchical one; although non-hierarchical approach tends to perform better when there are more subtopics, and obtains worse results when the number of subtopics reduces. The explanation for this phenomenon might be that when we have more subtopics, the non-hierarchical approach would have a better chance to reach the global solution as every move/swap operation it suggests does affect the objective function. However, when we have just too few subtopics, its move/swap operation may not be that effective, and the algorithm also terminates quickly; while the hierarchical approach using average-link distance maintains a better balance between the size of the clusters.

Effect of the number of subtopics

We have showed that both of our proposed algorithms outperform the baseline random clustering in all three product instances. Nevertheless, the marginal percentage in performance between them decreases as the number of subtopics reduces. This phenomenon is interesting, but not really surprising. In most of the cases, with a reliable sentence similarity measurement, the estimated number of final clusters is indeed very close to the annotated subtopics. Therefore, when we have too few topics, the estimated number of final clusters is also very small. Under this condition, each sentence assignment of the random clustering algorithm would have a higher chance of getting the correct cluster. As a result, we do not observe a sharp improvement for our two proposed clustering algorithms over the random algorithm. On the other hand, if we have more topics, the estimated number of final clusters is also larger. This is when the random
assignment gets little success in assigning sentences to their correct clusters. On the other hand, both of our proposed algorithms perform relatively stable in either scenarios.

Conclusion

In summary, with an average F measure ($F_{\alpha=0.2}$) of 75%, we believe that both hierarchical and non-hierarchical clustering algorithms are quite capable of tackling the tasks of subtopic clustering, and can be used in practical settings.
Chapter 5

Conclusion and Future Work

In this project, we build an end-to-end system that is capable of summarizing product reviews. The existing systems related to product reviews summarization normally constructed a facet-based summary, which is capable of aggregating sentiment information that belongs to each facet. We implemented this similar method as the first component in our system. We subsequently contributed an improvement to the system’s performance by applying syntactic role information within a sentence.

More importantly, as we proved the existence of underlying subtopics in the facet clusters, we introduced a second task that actually summarizes the review from a deeper perspective. Our summarization component proceeded by grouping together sentences mentioning about the same subtopics, and provided a compact summary to the users after adding in the sentiment information. We initiated a clustering approach in solving the subtopic problem, which achieved the average performance of 75% on F measure and ultimately outperformed the simple random clustering baseline. Nevertheless, the approach is highly dependent on the semantic similarity between words as well as sentences, which is not a complete solvable problem at the moment without some forms of manual input processing. In addition, we do not utilize deep semantics information in determining the similarities between sentences, which probably prevents us from achieving higher performance.

Several extensions from our current system are possible. Comparative-based summarization system would benefit directly from our systems, as it is now able to compare product facets
at a deeper level. Alternatively, as our summarization system only generates extractive-based summary, it might be more desirable to have a system that can reformulates the output sentences from our subtopic clustering and generates content back to users. Last but not least, more useful information about the product can also be augmented to the summarization system.
References


