

Mining User Reviews: from Specification to Summarization

Xinfan Meng

Key Laboratory of
Computational Linguistics
(Peking University)
Ministry of Education, China
mxf@pku.edu.cn

Houfeng Wang

Key Laboratory of
Computational Linguistics
(Peking University)
Ministry of Education, China
wanghf@pku.edu.cn

Abstract

This paper proposes a method to extract product features from user reviews and generate a review summary. This method only relies on product specifications, which usually are easy to obtain. Other resources like segmenter, POS tagger or parser are not required. At feature extraction stage, multiple specifications are clustered to extend the vocabulary of product features. Hierarchy structure information and unit of measurement information are mined from the specification to improve the accuracy of feature extraction. At summary generation stage, hierarchy information in specifications is used to provide a natural conceptual view of product features.

1 Introduction

Review mining and summarization aims to extract users' opinions towards specific products from reviews and provide an easy-to-understand summary of those opinions for potential buyers or manufacture companies. The task of mining reviews usually comprises two subtasks: product features extraction and summary generation.

Hu and Liu (2004a) use association mining methods to find frequent product features and use opinion words to predict infrequent product features. A.M. Popescu and O. Etzioni (2005) proposes OPINE, an unsupervised information extraction system, which is built on top of the KonwItAll Web information-extraction system. In order to reduce the features redundancy and provide a conceptual view of extracted features, G. Carenini et al. (2006a) enhances the earlier work of Hu and Liu (2004a) by mapping the extracted features into a hierarchy of features which describes the entity of interest. M. Gamon et al.

(2005) clusters sentences in reviews, then label each cluster with a keyword and finally provide a tree map visualization for each product model. Qi Su et al. (2008) describes a system that clusters product features and opinion words simultaneously and iteratively.

2 Our Approach

To generate an accurate review summary for a specific product, product features must be identified accurately. Since product features are often domain-dependent, it is desirable that the features extraction system is as flexible as possible. Our approach are unsupervised and relies only on product specifications.

2.1 Specification Mining

Product specifications can usually be fetched from web sites like Amazon automatically. Those materials have several characteristics that are very helpful to review mining:

1. Nicely structured, provide a natural conceptual view of products;
2. Include only relevant information of the product and contain few noise words;
3. Except for the product feature itself, usually also provide a unit to measure this feature.

A typical mobile phone specification is partially given below:

- Physical features
 - Form: Mono block with full keyboard
 - Dimensions: 4.49 x 2.24 x 0.39 inch
 - Weight: 4.47 oz
- Display and 3D
 - Size: 2.36 inch
 - Resolution: 320 x 240 pixels (QVGA)

2.2 Architecture

The architecture of our approach is depicted in Figure 1. We first retrieve multiple specifications from various sources like websites, user manuals etc. Then we run clustering algorithms on the specifications and generate a specification tree. And then we use this specification tree to extract features from product reviews. Finally the extracted features are presented in a tree form.

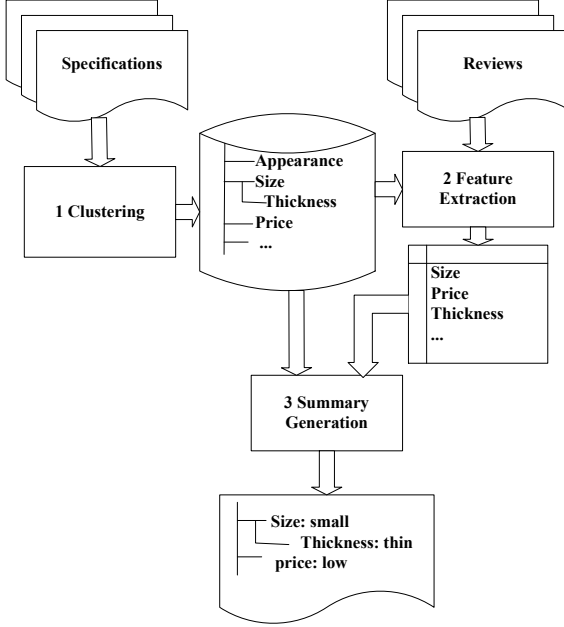


Figure 1: Architecture Overview

2.3 Specification Clustering

Usually, each product specification describes a particular product model. Some features are present in every product specification. But there are cases that some features are not available in all specifications. For instance, “WiFi” features are only available in a few mobile phones specifications. Also, different specifications might express the same features with different words or terms. So it is necessary to combine multiple specifications to include all possible features. Clustering algorithm can be used to combine specifications.

We propose an approach that takes following inherent information of specifications into account:

- **Hierarchy structure:** Positions of features in hierarchy reflect relationships between features. For example, “length”, “width” feature are often placed under “size” feature.

- **Unit of measurement:** Similar features are usually measured in similar units. Though different specification might refer the same feature with different terms, the units of measurement used to describe those terms are usually the same. For example, “dimension” and “size” are different terms, but they share the same unit “mm” or “inch”.

Naturally, a product can be viewed as a tree of features. The root is the product itself. Each node in the tree represents a feature in the product. A complex feature might be conceptually split into several simple features. In this case, the complex feature is represented as a parent and the simple features are represented as its children.

To construct such a product feature tree, we adopt the following algorithm:

- **Parse specifications:** We first build a dictionary for common units of measurement. Then for every specification, we use regular expression and unit dictionary to parse it to a tree of (feature, unit) pairs.
- **Cluster specification trees:** Given multiple specification trees, we cluster them into a single tree. Similarities between features are a combination of their lexical similarity, unit similarity and positions in hierarchy:

$$\begin{aligned}
 Sim(f1, f2) = & Sim_{lex}(f1, f2) \\
 & + Sim_{unit}(f1, f2) \\
 & + \alpha * Sim_{parent}(f1, f2) \\
 & + (1 - \alpha) * Sim_{children}(f1, f2)
 \end{aligned}$$

The parameter α is set to 0.7 empirically. If $Sim(f1, f2)$ is larger than 5, we merge features $f1$ and $f2$ together.

After clustering, we can get a specification tree resembles the one in subsection 2.1. However, this specification tree contains much more features than any single specification.

2.4 Features Extraction

Features described in reviews can be classified into two categories: explicit features and implicit features (Hu and Liu, 2004a). In the following sections, we describe methods to extract features in Chinese product reviews. However, these methods are designed to be flexible so that they can be easily adapted to other languages.

2.4.1 Explicit Feature Extraction

We generate bi-grams in character level for every feature in the specification tree, and then match them to every sentence in the reviews. There might be cases that some bi-grams would overlap or concatenated. In these cases, we join those bi-grams together to form a longer expression.

2.4.2 Implicit Feature Extraction

Some features are not mentioned directly but can be inferred from the text. Qi Su et al. (2008) investigates the problem of extracting those kinds of features. Their approach utilizes the association between features and opinion words to find implicit features when opinion words are present in the text. Our methods consider another kind of association: the association between features and units of measurement. For example, in the sentence “A mobile phone with 8 mega-pixel, not very common in the market.” feature name is absent in the sentence, but the unit of measurement “mega pixel” indicates that this sentence is describing the feature “camera resolution”.

We use regular expression and dictionary of unit to extract those features.

2.5 Summary Generation

There are many ways to provide a summary. Hu and Liu (2004b) count the number of positive and negative review items towards individual feature and present these statistics to users. G. Carenini et al. (2006b) and M. Gamon et al. (2005) both adopt a tree map visualization to display features and sentiments associated with features.

We adopt a relatively simple method to generate a summary. We do not predict the polarities of the user’s overall attitudes towards product features. Predicting polarities might entail the construction of a sentiment dictionary, which is domain dependent. Also, we believe that text descriptions of features are more helpful to users. For example, for feature “size”, descriptions like “small” and “thin” are more readable than “positive”.

Usually, the words used to describe a product feature are short. For each product feature, we report several most frequently occurring uni-grams and bi-grams as the summary of this feature. In Figure 2, we present a snippet of a sample summary output.

- mobile phone: not bad, expensive
 - appearance: cool
 - color: white
 - size: small, thin
 - camera functionality: so-so, acceptable
 - picture quality: good
 - picture resolution: not high
 - entertainment functionality: powerful
 - game: fun, simple

Figure 2: A Summary Snippet

3 Experiments

In this paper, we mainly focus on Chinese product reviews. The experimental data are retrieved from ZOL websites (www.zol.com.cn). We collected user reviews on 2 mobile phones, 1 digital camera and 2 notebook computers. To evaluate performance of our algorithm on real-world data, we do not perform noise word filtering on these data. Then we have a human tagger to tag features in the user reviews. Both explicit features and implicit features are tagged.

No. of Clustering Specifications	Mobile Phone	Digital Camera	Notebook Computer
1	153	101	102
5	436	312	211
10	520	508	312

Table 1: No. of Features in Specification Trees.

The specifications for all 3 kinds of products are retrieved from ZOL, PConline and IT168 websites. We run the clustering algorithm on the specifications and generate a specification tree for each kind of product. Table 1 shows that our clustering method is effective in collecting product features. The number of features increases rapidly with the number of specifications input into clustering algorithm. When we use 10 specifications as input, the clustering methods can collect several hundred features.

Then we run our algorithm on the data and evaluate the precision and recall. We also run the algorithms described in Hu and Liu (2004a) on the same data as the baseline.

From Table 2, we can see the precision of baseline system is much lower than its recall. Examining the features extracted by baseline system, we find that many mistakenly recognized features are high-frequency words. Some of those words appear many times in text. They are related to prod-

Product Model	No. of Features	Hu and Liu's Approach			the Proposed Approach		
		Precision	Recall	F-measure	Precision	Recall	F-measure
Mobile Phone 1	507	0.58	0.74	0.65	0.69	0.78	0.73
Mobile Phone 2	477	0.59	0.65	0.62	0.71	0.77	0.74
Digital camera	86	0.56	0.68	0.61	0.69	0.78	0.73
Notebook Computer 1	139	0.41	0.63	0.50	0.70	0.74	0.72
Notebook Computer 2	95	0.71	0.88	0.79	0.76	0.88	0.82

Table 2: Precision and Recall of Product Extraction.

uct but are not considered to be features. Some examples of these words are “advantages”, “disadvantages” and “good points” etc. And many other high-frequency words are completely irrelevant to product reviews. Those words include “user”, “review” and “comment” etc. In contrast, our approach recognizes features by matching bigrams to the specification tree. Because those high-frequency words usually are not present in specifications. They are ignored by our approach. Thus from Table 2, we can conclude that our approach could achieve a relatively high precision while keep a high recall.

Product Model	Precision
Mobile Phone 1	0.78
Mobile Phone 2	0.72
Digital camera	0.81
Notebook Computer 1	0.73
Notebook Computer 2	0.74

Table 3: Precision of Summary.

After the summary is given, for each word in summary, we ask one person to decide whether this word correctly describe the feature. Table 3 gives the summary precision for each product model. In general, on-line reviews have several characteristics in common. The sentences are usually short. Also, words describing features usually co-occur with features in the same sentence. Thus, when the features in a sentence are correctly recognized, Words describing those features are likely to be identified by our methods.

4 Conclusion

In this paper, we describe a simple but effective way to extract product features from user reviews and provide an easy-to-understand summary. The proposed approach is based only on product specifications. The experimental results indicate that our approach is promising.

In future works, we will try to introduce other resources and tools into our system. We will also explore different ways of presenting and visualizing the summary to improve user experience.

Acknowledgments

This research is supported by National Natural Science Foundation of Chinese (No.60675035) and Beijing Natural Science Foundation (No.4072012).

References

- M. Hu and B. Liu. 2004a. *Mining and Summarizing Customer Reviews*. In Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining, pages 168-177. ACM Press New York, NY, USA.
- M. Hu and B. Liu. 2004b. *Mining Opinion Features in Customer Reviews*. In Proceedings of Nineteenth National Conference on Artificial Intelligence.
- M. Gamon, A. Aue, S. Corston-Oliver, and E. Ringger. 2005. *Pulse: Mining Customer Opinions from Free Text*. In Proceedings of the 6th International Symposium on Intelligent Data Analysis.
- A.M. Popescu and O. Etzioni. 2005. *Extracting Product Features and Opinions from reviews*. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).
- Giuseppe Carenini, Raymond T. Ng, and Adam Pauls. 2006a. *Multi-Document Summarization of Evaluative Text*. In Proceedings of the conference of the European Chapter of the Association for Computational Linguistics.
- Giuseppe Carenini, Raymond T. Ng, and Adam Pauls. 2006b. *Interactive multimedia summaries of evaluative text*. In Proceedings of Intelligent User Interfaces (IUI), pages 124-131. ACM Press, 2006.
- Qi Su, Xinying Xu, Honglei Guo, Zhili Guo, Xian Wu, Xiaoxun Zhang, Bin Swen. 2008. *Hidden Sentiment Association In Chinese Web Opinion Mining*. In Proceedings of the 17th International Conference on the World Wide Web, pages 959-968.