Multimed Tools Appl (2015) 74:10177–10194
DOI 10.1007/s11042-014-2158-0

A holistic model of mining product aspects and associated sentiments from online reviews

Yan Li · Zhen Qin · Weiran Xu · Jun Guo

Received: 19 December 2013 / Revised: 18 April 2014 / Accepted: 23 June 2014 /
Published online: 20 July 2014

Abstract Online product reviews are considered a significant information resource useful for both potential customers and product manufacturers. In order to extract the fundamental product aspects and their associated sentiments from those reviews of plain texts, aspect-based sentiment analysis has emerged and has been regarded as a promising technology. This paper proposes a novel model to realize aspect-based sentiment summarization in an integrative way: composing the system with consistently designed feature extraction and clustering, collocation orientation disambiguation, and sentence sentiment strength calculation. Collocations of product features and opinion words are initially extracted through pattern-based bootstrapping. A novel confidence estimation method considering two measurements, *Prevalence* and *Reliability*, is exploited to assess both patterns and features. The obtained features are further clustered into aspects. Each cluster is assigned a weight based on arithmetic means of feature similarities and confidences. The orientations of dynamic sentiment ambiguous adjectives (DSAAs) are then determined within opinion collocations. Finally, sentiment strengths of opinion clauses for each aspect are computed according to a set of fine-grained and stratified scoring formulae. Experimental results on a benchmark data set validates the effectiveness of the proposed model.

Keywords Aspect-based sentiment summarization · Aspect extraction · Feature clustering · Opinion collocation orientation · Sentiment strength

Y. Li (✉) · Z. Qin · W. Xu · J. Guo
School of Information and Telecommunication Engineering, Beijing University of Posts and
Telecommunications, Xitucheng Road 10, Haidian District, Beijing 100876, People's Republic of China
e-mail: buptliyan@gmail.com

Z. Qin
e-mail: qinzhenbupt@gmail.com

W. Xu
e-mail: xuweiran@bupt.edu.cn

J. Guo
e-mail: guojun@bupt.edu.cn

1 Introduction

With e-commerce growing in popularity, online reviews are increasingly effective for customers to assess products, as well as merchants to grasp market sentiment on their products. Demand has thus been growing for opinion mining techniques that can automatically analyze user reviews from large quantities of written data and extract the most desired information for users. However, the unstructured review text brings difficulties to automatic analysis, which makes the development of the technology challenging.

Early approaches to this problem have focused on determining either the overall sentiment orientation (i.e., positive or negative) or the sentiment rating (e.g., one-to-five stars) of a review [2, 15, 16, 19]. However, only considering coarse overall ratings fails to adequately represent the multiple potential dimensions on which a product can be reviewed. As illustration in Fig. 1, while the cell phone review might express an overall sentiment rating of 3-stars, it additionally expresses positive sentiments toward the features *voice quality*, *screen* and *button design*, as well as negative sentiments toward the features *battery* and *price*.

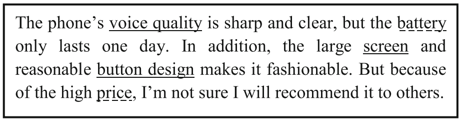
In contrast to determining an overall sentiment score for each review, many research efforts try to discover associated sentiments with specific product features [8, 10, 13]. A typical feature-based sentiment analysis algorithm works in two stages: (1) identifying feature mentions in reviews for each product; (2) identifying review sentences that give positive or negative opinions for each feature.

In this research, we study the problem of generating structured sentiment summaries of online reviews on the basis of product aspects. Here, an aspect, also known as facet, is defined as a product attribute in which customers are mainly interested. One aspect may be represented by multiple features. For instance, features may include *cost*, *payment* and *money*, all of which describe a *price* aspect. Since the number of features normally runs into hundreds, features are grouped into product aspects, and a structured sentiment summary is provided for each aspect. Figure 2 illustrates the summary about a particular product type *cell phone*, where its aspects are exhibited on the top layer. Under the hierarchy of each aspect, opinionated review sentences accompanied with the related product names are ranked in a descending order of sentiment strength over each sentiment orientation. So it is convenient for potential consumers to make a purchase decision when they are only concerned about some aspects of a product type.

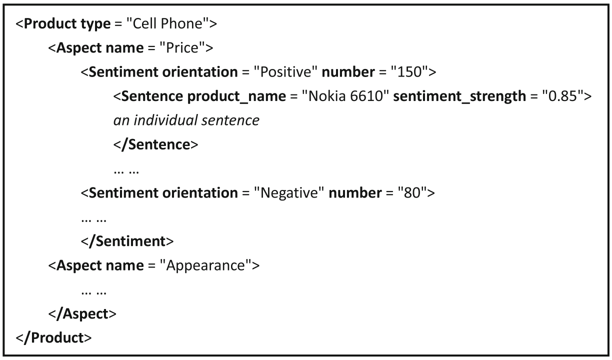
For realizing the feature-based sentiment analysis of online product reviews and acquiring the above summary, we implement a holistic model called SSPA (Sentiment Summarization on Product Aspects), which integrates the following techniques in a mutually consistent way:

- Calculating prior sentiment score for each word using generic opinion lexica.
- Extracting opinion collocations through bootstrapping dependency patterns. The opinion collocations define the pairs of product features and opinion words in reviews. We novelly proposed two measurements, *Prevalence* and *Reliability*, to estimate mutually both the newly generated patterns and features in each iteration.

Fig. 1 A sample review with multiple product features and opinions



The phone's voice quality is sharp and clear, but the battery only lasts one day. In addition, the large screen and reasonable button design makes it fashionable. But because of the high price, I'm not sure I will recommend it to others.



```
<Product type = "Cell Phone">
    <Aspect name = "Price">
        <Sentiment orientation = "Positive" number = "150">
            <Sentence product_name = "Nokia 6610" sentiment_strength = "0.85">
            an individual sentence
            </Sentence>
            … …
        <Sentiment orientation = "Negative" number = "80">
        … …
        </Sentiment>
    <Aspect name = "Appearance">
        … …
    </Aspect>
</Product>
```

Fig. 2 An example of structured sentiment summary

- Clustering product features into aspects based on word semantic similarities. Another two factors, *Sim-bar* and *Conf-bar*, are novelly defined to weight each cluster. And the light ones will be filtered out. The features with small scores in the previous step may be preserved if they have been clustered with the high-confident ones. So both the extraction precision and recall can be guaranteed.
- Disambiguating sentiment orientations of opinion collocations for each aspect. Since the orientations of sentiment words may shift according to different opinion targets, it is necessary to modify the prior sentiment scores of opinion words within their collocations. In this paper, we focus on disambiguating the orientations of dynamic sentiment ambiguous adjectives (DSAAs). DSAAs (e.g., *low*, *small*, *high*) are neutral out of context, but when they co-occur with some target features, positive or negative emotion will be evoked.
- Extracting aspect opinion clauses and analyzing their sentiment strengths for each aspect. A set of fine-grained and stratified scoring formulae is novelly designed making use of part-of-speech tagging, grammatical dependencies and word sentiment scores.

The rest of this paper is organized as follows. We introduce the related work in Section 2 and detail the SSPA model in Section 3. The evaluation results are shown and discussed in Section 4, and finally, Section 5 concludes the paper and points out future plans.

2 Related work

2.1 Pattern-based bootstrapping

Our proposed SSPA model exploits a pattern-based bootstrapping algorithm to extract candidate product features. Pattern-based bootstrapping algorithms have been used in various information extraction tasks, where patterns that express a particular semantic type are used to recognize new terms, and in turn these new terms help identify new patterns iteratively [1, 17, 20, 27]. For estimation of confidence values of the new terms and patterns, most of the approaches [7, 21, 29] follow the so-called "Duality Principle" as mentioned by Brin [4] and Yangarber [29], namely, the confidence values of learned terms and patterns is dependent on the confidence values of their origins. Agichtein [1] considered frequency information and included some heuristics for validation. All of these methods aimed at detecting patterns for a specific domain, and it is not clear whether they can be adapted to new domains. In [27], they made use of domain relevance values of terms occurring in rules to evaluate specific patterns, which is not applicable to general ones. Xu et al. [28] improved the precision of relation extraction by adding some limited closed-world knowledge for confidence estimation of learned rules to the usual seed data. Different from these previous work, we design a novel domain independent estimation method which can be generalized into all learned patterns and doesn't require any prior constraint knowledge.

2.2 Feature-based sentiment summarization

Sentiment summarization is essentially a particular multi-document summarization task. The idea of sentiment summarization is to use "aspects" of products as the basis of generating summary. The induction of sentiment summarization may be traced back to [3], which regarded the task as supervised sentence classification. However, the authors detected summary sentences using Naïve Bayes classifier without considering "aspects". The early mature sentiment summarization system may be the Feature-based Summarization system (FBS) proposed by Hu and Liu [8]. FBS applied association mining to extract frequent product features. And the infrequent ones were found simply using word position information. In contrast, our bootstrapping method extracts simultaneously more precise frequent and infrequent features exploiting grammatical dependency rules. In addition, FBS didn't make any attempt to cluster the acquired product features into appropriate aspects. Carenini et al. [5] incorporated Hu's features and mapped them into a taxonomy of aspects, but the taxonomy has to be predefined manually. A. Popescu and O.Etzioni [18] introduced an unsupervised information extraction system, OPINE, which utilized a fixed set of syntactic dependencies to identify product features and their associated opinion phrases. Ding et al. [6] extended Hu's research and implemented Opinion Observer to handle the task of predicting orientations of context dependent opinion words. They used three global conjunction rules exploiting external information in other sentences and reviews. However, the contexts surrounding these opinion words are limited. S. Moghaddam and M. Ester [14] designed Opinion Digger to handle the task of aspect extraction. They used some existing aspects to generate Part-of-Speech patterns and expanded them with Generalized Sequential Pattern Mining. The aspect mentions were removed just according to their frequencies while we estimate their confidences based on a more comprehensive metric considering both *Prevalence* and *Reliability*.

2.3 Multi-aspect rating prediction

The goal of multi-aspect rating prediction is to assign a review document multiple sentiment ratings over some existing aspects. Recent work has begun to investigate multi-aspect rating prediction using probabilistic generative models such as topic models. Titov and McDonald [24] proposed MG-LDA on the basis of traditional topic model LDA to discover global topics and local topics (aspects) simultaneously. They further extended MG-LDA into a new model MAS to infer an explicit mapping between local topics and aspects with the assistance from aspect-specific ratings [23]. Lu et al. [12] tried to apply structured PLSA to generate a rated aspect summary of short comments, which is a decomposed view of the overall ratings for major aspects. Jo and Oh [9] proposed an aspect and sentiment unification model to discover a pair of aspect and sentiment label for each sentence on the basic assumption that one sentence tends to represent one aspect and one sentiment. Lakkaraju et al. [11] designed a joint modeling CFACTS-R to identify latent facets and sentiments,

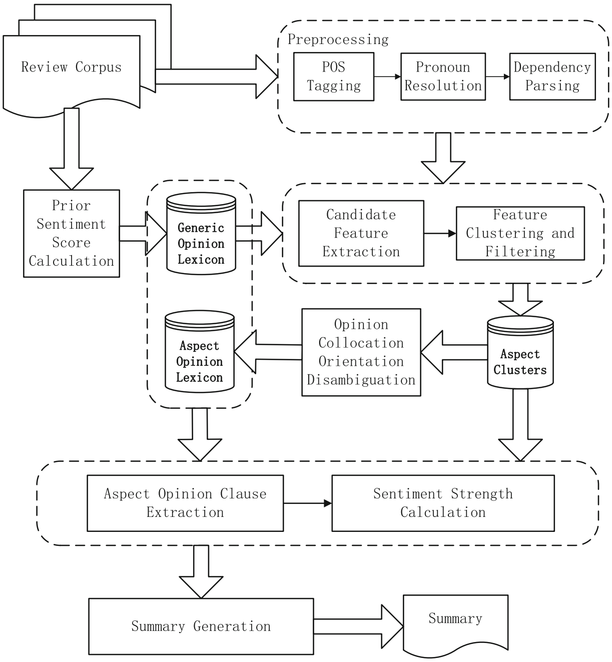

Review Corpus
Preprocessing
POS Tagging
Pronoun Resolution
Dependency Parsing
Prior Sentiment Score Calculation
Generic Opinion Lexicon
Candidate Feature Extraction
Feature Clustering and Filtering
Aspect Opinion Lexicon
Opinion Collocation Orientation Disambiguation
Aspect Clusters
Aspect Opinion Clause Extraction
Sentiment Strength Calculation
Summary Generation
Summary

Fig. 3 The SSPA framework

exploit their coherence, and infer facet-level sentiment ratings. Instead of rating aspects in individual reviews, SSPA computes sentiment strengths (real numbers) of aspects involved in each sentence using a set of stratified formulae.

3 The proposed SSPA system

Figure 3 depicts the architectural overview of our SSPA system. The input is a collection of online reviews about either a specific product name or a product type and the output is the structured summary as the one shown in the introduction section. SSPA performs sentiment summarization mainly in six steps: (1) preprocessing; (2) calculating word prior sentiment scores; (3) extracting candidate product features; (4) clustering and filtering features to obtain product aspects; (5) disambiguating sentiment orientations of opinion collocations for each aspect; (6) extracting aspect opinion clauses and analyzing their sentiment strengths.

The preprocessing mainly involves part-of-speech tagging, pronoun resolution and dependency parsing, all of which can be performed by the prevalent natural language analysis toolkit Stanford CoreNLP.[1] The rest of the procedures will be explained in detail in the following subsections.

[1] http://nlp.stanford.edu/software/corenlp.shtml

Table 1 Seed dependency patterns for extracting candidate features

Type	Pattern	Example
Direct	amod (f, o)	*short* **battery life**
	nsubj (o, f)	The **screen** is *large*.
	dobj (o, f)	I *hate* its **color**.
Indirect	nsubj (verb, f) + advmod (verb, o)	The **battery** works *well*.
	nsubj (verb, f) + acomp (verb, o)	The **phone shell** looks very *solid*.
	nsubj (verb, f) + dobj (verb, o)	The **screen** has a *spot*.

3.1 Calculating prior sentiment scores

The prior sentiment scores of individual words, ranging from -1 to 1, indicate their opinion orientations (negative, positive or neutral) as well as sentiment strengths which will be used to calculate the contextual sentiment scores later. Inspired by [22], two prevalent opinion lexica SentiWordNet[2] and OpinionFinder Subjectivity Lexicon[3] are utilized. The former contains approximately 200,000 entries describing sentiment scores for multiple senses of words and phrases. And the latter records over 8,000 words which were extracted from [25] and were annotated with both sentiment orientation and subjective strength (strong or weak). The multi-sense sentiment scores and the subjectivity clues in these two lexica provide sufficient information to calculate a prior score for each word in our corpus. The detailed scoring schemes are referred to [22]. As a result, the neutral words are ignored and the remaining opinionated words constitute our final generic opinion lexicon which contains not only adjectives but also nouns, verbs and adverbs.

3.2 Extracting candidate features

This sub-step extracts candidate product features on which customers have expressed their opinions. As Hu and Liu [8] have mentioned, implicit features are hard to find (e.g., *The phone will not easily fit in pockets.*). Similar to Hu's work and many others [5, 6, 30], we focus on finding explicit features which are nouns or noun phrases in the reviews.

In understanding of natural languages, there are normally grammatical relations between sentiment targets and opinion terms. According to this observation, we define a set of seed dependency patterns based on the parser of Stanford CoreNLP[4] and bootstrap them to match candidate features and generate more patterns. All the seed patterns are shown in Table 1. There are 3 direct dependency relations and 3 indirect ones which contain only one connective word. We follow the dependency annotations (*nsubj*, *amod*, *dobj*, etc.) used in CoreNLP. In the column 2 of Table 1, each pattern is formatted as *dependency (governor, dependent)*, and *f* and *o* stand for feature and opinion term respectively. The last column gives sentence examples where features and opinion terms are written in boldface and italic respectively.

These seed patterns along with the prior sentiment knowledge are applied to extract candidate features, and in turn these features can generate new dependency patterns.

[2] http://sentiwordnet.isti.cnr.it/

[3] http://mpqa.cs.pitt.edu/lexicons/subj_lexicon/

[4] http://nlp.stanford.edu/software/corenlp.shtml

Confidence estimation of learned patterns and features is essential to prevent "dangerous" or plainly wrong information during the bootstrapping process. To tackle this issue, we defined two new measurements, *Prevalence* and *Reliability*. Specifically, in the *l*-th iteration, the *Prevalence* of feature *i* is formulated as follows:

$$Prev(i^l) = \frac{PattExtr^l(i)}{N^l_{Patt}}, \tag{1}$$

where $PattExtr^l(i)$ is the number of patterns that can extract feature *i* in the *l*-th iteration, and N^l_{Patt} is the total number of patterns in this iteration. And the following equation calculates feature's *Reliability*:

$$Reli(i^l) = \sum_{j \in PattSet^l} Conf^l_j \cdot Prob^l_j(i) = \sum_{j \in PattSet^l} Conf^l_j \cdot \left(\frac{Count^l_j(i)}{\sum_{w \in V} Count^l_j(w)}\right), \tag{2}$$

where $PattSet^l$ is the pattern collection in the *l*-th iteration, $Conf^l_j$ is the confidence value of pattern *j*, and $Prob^l_j(i)$ calculates the probability of *i* being extracted by *j* according to the count ratio between *i* and all other words in the word set *V*. Considering the above two equations, the *Prevalence* measures features' abilities activating source patterns in each iteration while the *Reliability* prefers features extracted with larger probabilities by more confident patterns. The final confidence of feature *i* in the *l*-th iteration is the weighted sum of these two measurements:

$$Conf^l_i = w_1 \cdot Prev(i^l) + w_2 \cdot Reli(i^l). \tag{3}$$

The confidence value for pattern *j*, $Conf^l_j$, is computed in a similar way just exchanging *i* and *j* and substituting *Feat* for *Patt* in (1) and (2).

3.3 Feature clustering and filtering

The infrequent features extracted from the previous step may have a low confidence. On the other hand, one aspect may be represented by multiple features. For example, features may include *cost*, *payment* and *money*, all of which describe a *price* aspect. To retain more infrequent features and form a more compact aspect structure, it is necessary to group all the candidate features into several clusters, each of which represents either a particular aspect or a general noun group that should be pruned (e.g., the cluster with *user*, *customer*, *client*).

Analyzing semantic similarities between the candidate features is crucial for this task. WordNet[5] is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are linked by a complex network of lexical relations. Each synset has one or more hypernym paths that link it to a root hypernym. Based on the WordNet, the similarity between two words w_1 and w_2 can be calculated as follows:

$$sim(w_1, w_2) = \frac{1}{|sw_1| + |sw_2|} \cdot \{\sum_i \underset{j}{MAX}[ps(sw_{1i}, sw_{2j})] + \sum_m \underset{n}{MAX}[ps(sw_{2m}, sw_{1n})]\} \tag{4}$$

where sw_{1i} stands for the i-th sense of w_1, and the $|sw_1|$ is the sense number of w_1. The path similarity measure *ps* equals the inverse of the shortest path connecting the two senses

[5] http://wordnet.princeton.edu/

in the *is-a* taxonomy. As for our two candidate features cf_1 and cf_2, which may involve several words, the similarity is the arithmetic mean over all of the word pairs:

$$Sim(cf_1, cf_2) = \frac{1}{|cf_1| + |cf_2|} \cdot \sum_{w_1 \in cf_1} \sum_{w_2 \in cf_2} sim(w_1, w_2) \tag{5}$$

where the $|cf_1|$ represents the word number in cf_1.

Our clustering algorithm is shown from line 1 to line 6 in Algorithm 1. For a candidate feature cf_i, we find its most similar feature cf_j and group them together if their similarity is larger than a threshold t_1.

The feature set generated in the bootstrapping procedure may contain some false features (e.g., *home*, *anything*, *review*). Instead of pruning individual features themselves, we assign each generated cluster a weight and remove the *light* ones. The weight of cluster c_i is calculated based on its *Sim-bar* and *Conf-bar*, which are defined as follows:

$$Sim\text{-}bar(c_i) = \frac{1}{N_i^{pair}} \cdot \sum_{cf_1, cf_2 \in c_i} Sim(cf_1, cf_2) \tag{6}$$

$$Conf\text{-}bar(c_i) = \frac{1}{N_i} \cdot \sum_{cf \in c_i} Conf(cf) \tag{7}$$

where N_i^{pair} and N_i are the number of feature pairs and individual features in the cluster c_i respectively. Please note that the self-similarity equals 1. The *Sim-bar* evaluates the average similarity between cluster's members, and the *Conf-bar* takes into account the quality of

Algorithm 1 Feature Clustering and Filtering

Input: the candidate feature set CF; the clustering threshold t_1; the filtering threshold t_2
Output: the aspect cluster set AC

```
1:  for each candidate feature $cf_i$ in $CF$ do
2:      Find another feature $cf_j$ that has largest similarity with $cf_i$ according to (5).
3:      if $Sim(cf_i, cf_j) > t_1$ then
4:          Assign $cf_i$ and $cf_j$ into one cluster.
5:      else
6:          $cf_i$ itself form a new cluster.
7:      end if
8:  end for

9:  Construct an empty set $AC$.
10: for each generated cluster $c_i$ do
11:     Calculate $c_i$'s weight $Weig(c_i)$ according to (8)
12:     if $Weig(c_i) < t_2$ then
13:         Remove $c_i$.
14:     else
15:         Append $c_i$ to $AC$.
16:     end if
17: end for
```

each member. Inspired by the mass formula in physics, we regard *Sim-bar* and *Conf-bar* as "volume" and "density" respectively and calculate the weight of c_i as the product of them:

$$Weig(c_i) = \mathit{Sim\text{-}bar}(c_i) \cdot \mathit{Conf\text{-}bar}(c_i). \tag{8}$$

The filtering procedures are also explained from line 7 to line 13 in Algorithm 1. In this step of estimation, the low-confident features in the previous step may be preserved if they have been clustered with the high-confident ones, ensuring performances of both precision and recall in aspect extraction.

3.4 Disambiguating orientations of opinion collocations

With the aspect clusters, we proceed to analyze the sentiment orientations of their opinion collocations. We believe that the generic opinion lexicon obtained in Section 3.1 covers most opinionated words. However, due to the diversity of language expression, the sentiment orientation of a word may shift according to its modified target. For example:

"It takes low quality of outdoor photos."
"I prefer the phone's low price."

The opinion word *low* appears in both sentences, but it exhibits negative in the first sentence while positive in the second one.

Wu and Wen [26] defined these context dependent opinion words (e.g., *low*, *small*, *high*) as dynamic sentiment ambiguous adjectives (DSAAs). They manually divided 14 Chinese DSAAs into two categories: positive-like adjectives (PAs) and negative-like adjectives (NAs). Then the task of identifying sentiment orientations of collocations with DSAAs has been simplified to sentiment classification of target nouns, which is referred to sentiment expectation. Using a Web search engine with some sentiment syntactic patterns as queries, the sentiment expectation of a noun can be inferred by calculating its statistical association with positive and negative hits. Sentiment syntactic patterns are those people frequently use when they express their opinions about something. We applied their methods to English language and listed the 9 PAs and 7 NAs as well as 2 sentiment syntactic patterns in Table 2.

As the sentiment syntactic patterns usually express negative opinions, we take "*n + SSP + NAs*" and "*n + SSP + PAs*" as positive and negative queries about noun *n* respectively. The following two equations calculate the numbers of positive hits $Hit^{+}(n)$ and negative hits $Hit^{-}(n)$ of *n*:

$$Hit^{+}(n) = \sum_{b \in NAs} \sum_{i=1}^{2} HitSSP_i(n, b) \tag{9}$$

$$Hit^{-}(n) = \sum_{a \in PAs} \sum_{i=1}^{2} HitSSP_i(n, a) \tag{10}$$

Table 2 English DSAAs and sentiment syntactic patterns

DSAAs	PAs	heavy, fast, large, high, quick, long, loud, hard, big
	NAs	light, slow, small, low, short, soft, little
Sentiment syntactic patterns	SSP$_1$	<noun> is a little <DSAAs>
	SSP$_2$	<noun> is too <DSAAs>

where $HitSSP_i(n, b)$ is the hit number of the query with sentiment syntactic pattern SSP_i, noun n and DSSA b. The sentiment expectation of n can then be inferred as follows:

$$SE(n) = sgn(Hit^{+}(n) - Hit^{-}(n)) \tag{11}$$

where $sgn()$ is the sign function. The sentiment orientations of opinion collocations $< n, PAs >$ are the same as $SE(n)$ while $< n, NAs >$ invert $SE(n)$. Both the sentiment strengths of $< n, PAs >$ and $< n, NAs >$ can be further computed as follows:

$$SS(n) = \frac{max(Hit^{+}(n), Hit^{-}(n))}{Hit^{+}(n) + Hit^{-}(n)} \tag{12}$$

According to the above strategy, we analyzed sentiment strengths of all the opinion collocations with DSAAs and features in each aspect cluster, and finally constructed the aspect opinion lexicon.

3.5 Calculating sentiment strengths of aspect opinion clauses

In this paper, sentiment analysis for each product aspect is performed at clause level. Similar to Hu's definition, an aspect opinion clause must contain one or more product aspects and opinion words. Based on this definition, all the review sentences are segmented into several clauses initially and the aspect opinion ones are extracted. Their sentiment strengths can be calculated according to Table 3, where the dependency annotations are in accordance with

Table 3 A stratified scoring scheme for calculating clause sentiment strength

Priority	Dependency pattern	Example	Formula
1	neg (o, neg)	not nifty; rarely fail	F1
	advmod (o, da)	pretty good; a bit small; highly recommend	F2
2	conj_and (o_1, o_2)	beautiful and charming	F3
	conj_but (o_1, o_2)	simple but practical	F4
3	amod (NN, JJ)	amazing advantage; great failure; awkward smile	F5
	advmod (VB, RB)	praise happily; worked wrongly; worship blindly	
4	dobj (VB, NN)	pursue excellence; lose trust; suffers pain	F6
	acomp (VB, JJ)	looks beautiful; acts badly	
5	nsubj (VB, NN)	The rechargeable battery performs poorly.	
6	conj_and (f_1, f_2)	button and screen	F7
	conj_negcc (f_1, f_2)	I love the phone cover but not the headset.	F8

F1: $(-1) * sgn(A) * (1 - |A|)$

F2: if $sgn(B) > 0 \rightarrow sgn(A) * [|A| + (1 - |A|) * |B|]$; else$\rightarrow$ $sgn(A) * [|A| * (1 - |B|)]$

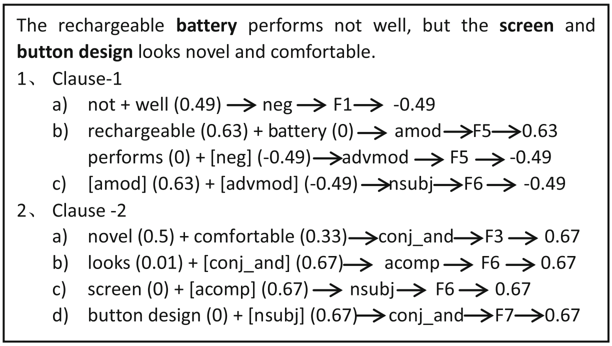
F3: $sgn(A) * [|A| + (1 - |A|) * |B|]$ F4: $sgn(B) * [|B| * (1 - |A|)]$

F5: if $sgn(A) > 0\ and\ sgn(B) < 0 \rightarrow B$; else$\rightarrow$ $(sgn(A) \wedge sgn(B)) * [|A| + (1 - |A|) * |B|]$

F6: if $sgn(A) < 0\ and\ sgn(B) > 0 \rightarrow A$; else$\rightarrow$ $(sgn(A) \wedge sgn(B)) * [|A| + (1 - |A|) * |B|]$

F7: B's score equals A's F8: B's score inverses A's

(o: opinion word; neg: negation; da: degree adverb; f: feature; sgn: the sign function; A: governor; B: dependent)


The rechargeable **battery** performs not well, but the **screen** and **button design** looks novel and comfortable.

1、 Clause-1
   a) not + well (0.49) → neg → F1 → -0.49
   b) rechargeable (0.63) + battery (0) → amod → F5 → 0.63
      performs (0) + [neg] (-0.49) → advmod → F5 → -0.49
   c) [amod] (0.63) + [advmod] (-0.49) → nsubj → F6 → -0.49

2、 Clause -2
   a) novel (0.5) + comfortable (0.33) → conj_and → F3 → 0.67
   b) looks (0.01) + [conj_and] (0.67) → acomp → F6 → 0.67
   c) screen (0) + [acomp] (0.67) → nsubj → F6 → 0.67
   d) button design (0) + [nsubj] (0.67) → conj_and → F7 → 0.67

Fig. 4 An illustration of clause sentiment analysis

the ones in CoreNLP. For this task, we introduce a method for hierarchically calculating the clause sentiment strength. The *Priority* column indicates the order of analyzing each sentiment context and the last column presents its scoring formula. For simplicity, we use A and B in the formulae to represent the corresponding signed sentiment values of the governor and dependent in dependency patterns respectively.

Figure 4 illustrates how the algorithm performs aspect sentiment analysis. The sentence comprises two clauses involving different aspects. The clauses are analyzed separately. The numbers are sentiment strengths and the brackets indicate sentiment contexts that must be analyzed integrally. For example, in the first clause, the negation constituent, *"not well"*, is primarily detected and scored. Then it acts as a whole sentiment context *"[neg]"* in the following steps. Finally, the sentiment strength of the aspect *battery* equals to the first clause's score, i.e., -0.49. As for the second clause, there are two aspects *screen* and *button design* with a coordinative relation, so they share the same sentiment strength, 0.67, according to F.7.

4 Evaluation results

4.1 Data set

We conducted our experiments on a benchmark data set,[6] which was constructed by [6]. It contains customer reviews about 8 electronic products: two digital cameras (DC 1 and DC 2), two cellular phones (Phone 1 and Phone 2), one MP3 player (MP3), one DVD player (DVD), one router (Router) and one anti-virus software (Antivirus). The characteristics of the data set are listed in Table 4. The second and third columns indicate the number of reviews and features for each subset respectively. The last column shows the numbers of opinion words in the collocations with each feature. For each review sentence in this data set, the involved product features and their corresponding sentiment levels ranging from -3 to +3 have been already annotated. We evaluate the effectiveness of each module in SSPA and compare it to some previous mentioned systems (i.e., FBS, OPINE, Opinion Digger and Opinion Observer) in the following subsections. As the feature extraction

[6] http://www.cs.uic.edu/~liub/FBS/Reviews-9-products.rar

Table 4 Characteristics of the data set

Review	#. reviews	#. features	#. opinion words
DC 1	45	103	316
DC 2	34	100	210
Phone 1	49	107	267
MP3	95	146	515
DVD	99	105	263
Phone 2	41	161	240
Router	31	94	185
Antivirus	51	136	185

performances of FBS, OPINE and Opinion Digger were only reported on the first five products, we show the evaluation results of SSPA on feature extraction (Section 4.2) and clustering (Section 4.3) on these review subsets as well. The performances of predicting sentiment orientations and strengths for aspect opinion clauses are evaluated on the entire dataset.

4.2 Evaluating feature extraction

Table 5 shows the performances on precision (P), recall (R) and F1 of SSPA in aspect extraction, where $\text{SSPA}_{\text{boot}}$ denotes our model only conducting the pattern bootstrapping procedure. The iteration number was fixed to 10. We weighed *Prevalence* and *Reliability* equally in (3) and set w_1 and w_2 to 0.5. The reported evaluation results of FBS, OPINE and Opinion Digger are also listed in the table.

According to Table 5, our bootstrapping algorithm ($\text{SSPA}_{\text{boot}}$) performs best in terms of recall on all of the review subsets, demonstrating that the generated dependency patterns are effective to extract most of the candidate features. To retain low frequent features, we don't filter any terms in this step, so they contain some errors, which are mainly derived from exceptions in pattern matching (e.g., *a good chance* will match the pattern *amod(f, o)*), the errors of POS tagging and dependency parsing in CoreNLP, and misspellings of online texts (e.g., *conector*). Using this step alone gives low precision scores. The columns in SSPA show the results after clustering is performed. We can see that the precision is improved dramatically with a little decline in recall, guaranteeing the best F1 performances of SSPA on all of the product reviews. Actually, the clustering procedure not only filters out the "light" clusters but it also preserves the potential product features with low confidences. For instance, *lcd* was scored small by (3), but it can be assigned into a "heavy" aspect cluster owning a frequent feature *screen*.

Regarding the other three models, FBS falls far behind the rest, especially in precision. The main reason is that both association mining and word position information would induce a lot of noises. Although the precision in OPINE is relatively high, benefiting from the feature assessment mechanism incorporating Web PMI statistics, its average recall is even 3 % lower than FBS. Opinion Digger's recall is compatible to SSPA, but the precision is significantly lower. In addition, the F1 score of Opinion Digger is not stable (from 79 to 89 %) across the product domains. In conclusion, our proposed SSPA system is vital in aspect extraction.

Table 5 Evaluation results of feature extraction and clustering

		DC 1	DC 2	Phone 1	MP3	DVD	Average
FBS	P	0.75	0.71	0.74	0.72	0.69	0.72
	R	0.82	0.79	0.80	0.76	0.82	0.80
	F1	0.78	0.75	0.77	0.74	0.75	0.76
OPINE	P	**0.94**	0.93	**0.95**	**0.95**	**0.94**	**0.94**
	R	0.80	0.73	0.78	0.73	0.79	0.77
	F1	0.86	0.82	0.86	0.83	**0.86**	0.85
Opinion Digger	P	0.77	0.79	0.86	0.81	0.70	0.80
	R	0.82	0.87	0.92	0.91	0.90	0.87
	F1	0.79	0.83	**0.89**	0.86	0.79	0.83
$\text{SSPA}_{\text{boot}}$	P	0.67	0.73	0.65	0.67	0.63	0.67
	R	**0.96**	**0.96**	**0.93**	**0.94**	**0.93**	**0.94**
	F1	0.79	0.83	0.77	0.78	0.75	0.78
SSPA	P	0.87	**0.96**	0.92	0.90	0.85	0.90
	R	0.90	0.87	0.87	0.89	0.88	0.88
	F1	**0.89**	**0.91**	**0.89**	**0.89**	**0.86**	**0.89**

The best results of the corresponding criterion are shown in boldface

4.3 Evaluating feature clustering

To further illustrate the effectiveness of our clustering algorithm, Table 6 shows the evaluation results on two commonly used metrics, Purity and Rand Index (RI). The product features in our data set have been already labeled and the gold standard aspect clusters were constructed manually. Here, we ignored the clusters containing none of the gold standard features, because it is costly to annotate cluster labels for all terms. The clustering threshold t_1 and the filtering threshold t_2 were set to 0.25 and 0.2 respectively for all review sets.

It can be concluded in Table 6 that our feature clustering approach performs relatively well. Actually, through the control of a fairly large threshold t_1, the clustering performed so strictly that the two clustered features are likely to share similar semantics, which yields a high Purity. As for RI, it measures the percentage of correct decisions (true positive or true negative). In our experiments, the true negative cases appeared frequently indicating the fact that two features are grouped together only if they are highly relevant (e.g., *sound* and *voice*, *headset* and *earpiece*, *picture* and *photo*).

Table 6 Evaluation results of feature clustering

Review set	#. aspect cluster	Purity	RI
DC 1	23	0.882	0.952
DC 2	24	0.923	0.970
Phone 1	29	0.919	0.941
MP3	49	0.926	0.979
DVD	31	0.936	0.950
average	34.4	0.917	0.958

4.4 Evaluating aspect opinion clause extraction and orientation prediction

In this section, we took the gold standard aspect clusters in each review subset as input to the module of aspect opinion clause extraction and orientation prediction in SSPA. The results of FBS, Opinion Observer and SSPA are shown in Table 7, where the precision and recall are macro-average values over aspects.

In general, the precision results of all the three systems are promising. However, benefiting from the comprehensive generic opinion lexicon, the reasonable confidence estimation in bootstrapping and clustering and the well-designed aspect opinion lexicon, SSPA outperforms on average FBS and Opinion Observer.

As for recall, FBS falls far behind the rest. The main reason may be that FBS only considers adjective opinion words. So it leaves out the sentences such as *"I love its picture quality"* and *"The phone's radio is really my favorite"*. The Opinion Observer's performance is marginally better than SSPA. By observing the missing cases in SSPA, we found that some cases are not covered by any patterns in Table 3. In addition, CoreNLP fails to parse dependency relations when features and opinion words appear in some complex sentences. For instance,

"The price makes it a good buy."
"The first thing that hits me is how good the screen is."

SSPA identifies the above two sentences as aspect opinion ones correctly while regards them neutral mistakenly. In the first sentence, {*price - makes - buy - good*} forms a two-order indirect relation which goes beyond the scope of our pattern definition. The second one contains an attributive clause and a predicative clause, and the latter is even expressed in an inverted format. This makes it very challenging for CoreNLP to understand.

4.5 Evaluating sentiment strength prediction

Besides orientations, we also evaluate the results of predicting sentiment strengths of aspect opinion clauses. In our data set, features' sentiment strengths in each sentence have been

Table 7 Evaluation results of aspect opinion clause extraction and orientation prediction

Review	FBS			Opinion observer			SSPA		
set	precision	recall	F1	precision	recall	F1	precision	recall	F1
DC 1	**0.93**	0.80	0.86	**0.93**	**0.92**	**0.93**	0.90	0.90	0.90
DC 2	**0.98**	0.87	0.92	0.96	**0.96**	**0.96**	0.94	0.91	0.92
Phone 1	0.94	0.70	0.80	0.93	0.90	0.91	**0.96**	**0.91**	**0.93**
MP3	0.91	0.69	0.78	0.87	**0.86**	0.87	**0.98**	0.84	**0.90**
DVD	0.91	0.72	0.80	0.89	0.88	0.89	**0.95**	**0.91**	**0.93**
Phone 2	0.95	0.82	0.88	0.95	**0.95**	**0.95**	**0.96**	0.89	0.92
Router	0.83	0.67	0.74	0.84	0.82	0.83	**0.93**	**0.84**	**0.88**
Antivirus	0.94	0.64	0.76	0.90	**0.87**	0.88	**0.97**	0.84	**0.90**
average	0.92	0.74	0.82	0.91	**0.90**	0.90	**0.95**	0.88	**0.91**

The best results of the corresponding criterion are shown in boldface

Table 8 Evaluation results of sentiment strength prediction

Review set	Accuracy	Review set	Accuracy
DC 1	0.789	DVD	0.768
DC 2	0.783	Phone 2	0.801
Phone 1	0.804	Router	0.775
MP3	0.764	Antivirus	0.780
average accuracy: 0.783			

rated in six levels ranging from -3 to +3. Although the strength annotation is quite subjective, it indeed provides clues in evaluating the effectiveness of our sentiment strength prediction method. We simply scaled the scores of aspect opinion clauses into the annotated six levels according to the following scheme:

$$r_c = \begin{cases} 1, \quad if \;\; abs(s_c) \leq 0.2; \\ 1 \; or \; 2, \quad if \;\; 0.2 < abs(s_c) < 0.4; \\ 2, \quad if \;\; 0.4 \leq abs(s_c) \leq 0.6; \\ 2 \; or \; 3, \quad if \;\; 0.6 < abs(s_c) < 0.8; \\ 3, \quad if \;\; abs(s_c) \geq 0.8. \end{cases}$$

where s_c is the calculated clause score, $abs()$ is the absolute value function and r_c is the predicted rating. To weaken subjective influence, r_c allows double choices when s_c locates medium intensities.

The accuracy results are shown in Table 8, where it stays stable (from 76.4 to 80.4 %) over all the review sets. The promising average accuracy (78.3 %) reveals that it is practical for SSPA to predict sentiment strength for each aspect clause.

5 Conclusion and future work

This paper proposed a holistic model SSPA, which systematically integrates all tasks of feature-based sentiment analysis, including extracting product features, grouping features into aspects, disambiguating orientations of opinion collocations, and analyzing sentiment strengths for individual words and sentences. Through experiments over real-world review data, we have demonstrated that each component in SSPA performs well. It is thus indeed practical for SSPA to generate the structured sentiment summary for product reviews.

In the future, we plan to deal with more types of features including verbs, adjectives and implicit features. And the automatic determination of parameters, i.e., iterations in bootstrapping, the weighting coefficients and the clustering thresholds, is also a crucial issue. Finally, we will also try to analyze sentiment strengths of complex review sentences.

Acknowledgments This work was supported by 111 Project of China under Grant No. B08004, key project of ministry of science and technology of China under Grant No. 2011ZX03002-005-01, National Natural Science Foundation of China (61273217) and the Ph.D. Programs Foundation of Ministry of Education of China (20130005110004).

References

1. Agichtein E (2000) Confidence estimation methods for partially supervised information extraction. In: Proceedings 6th SIAM international conference on data mining, pp 539–543
2. Baccianella S, Esuli A, Sebastiani F (2009) Multi-facet rating of product reviews. Adv Inf Retr 5478:461–472
3. Beineke P, Hastie T, Manning C, Vaithyanathan S (2004) Exploring sentiment summarization. In: AAAI spring symposium on exploring attitude and affect in text: theories and applications
4. Brin S (1998) Extracting patterns and relations from the world wide web. In: Proceedings WebDB workshop at 6th international conference on extending database technology, pp 172–183
5. Carenini G, Ng RT, Zwart E (2005) Extracting knowledge from evaluative text. In: Proceedings 3rd international conference on knowledge capture, pp 11–18
6. Ding X, Liu B, Yu PS (2008) A holistic lexicon-based approach to opinion mining. In: Proceedings 1st international conference on web search and web data mining, pp 231–239
7. Greenwood MA, Stevenson M (2006) Improving semi-supervised acquisition of relation extraction patterns. In: Proceedings workshop on information extraction beyond the document, pp 29–35
8. Hu M, Liu B (2004) Mining and summarizing customer reviews. In: Proceedings 10th international conference on knowledge discovery and data minning, pp 168–177
9. Jo Y, Oh A (2010) Aspect and sentiment unification model for online review analysis. In: Proceedings 4th ACM international conference on web search and data mining, pp 815–824
10. Lakkaraju H, Bhattacharyya C, Bhattacharya I (2011a) Exploiting coherence for the simultaneous discovery of latent facets and associated sentiments. In: Proceedings international conference on data mining, pp 498–509
11. Lakkaraju H, Bhattacharyya C, Bhattacharya I (2011b) Exploiting coherence for the simultaneous discovery of latent facets and associated sentiments. In: Proceedings 2011 SIAM international conference on data mining, pp 498–509
12. Lu Y, Zhai C, Sundaresan N (2009) Rated aspect summarization of short comments. In: Proceedings 18th international conference on world wide web, pp 131–140
13. Mei Q, Ling X, Wondra M, Su H, Zhai C (2007) Topic sentiment mixture: modeling facets and opinions in weblogs. In: Proceedings 16th international conference on world wide web, pp 171–180
14. Moghaddam S, Ester M (2010) Opinion digger: an unsupervised opinion miner from unstructured product reviews. In: Proceedings 19th international conference on information and knowledge management, pp 1825–1828
15. Pang B, Lee L (2005) Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales. In: Proceedings 43rd annual meeting of the association for computational linguistics, pp 115–124
16. Pang B, Lee L, Vaithyanathan S (2002) Thumbs up?: sentiment classification using machine learning techniques. In: Proceedings ACL-02 conference on empirical methods in natural language processing-volume, vol 10, pp 79–86
17. Pantel P, Pennacchiotti M (2006) Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In: Proceedings 46th annual meeting of the association for computational linguistics, pp 113–120
18. Popescu A, Etzioni O (2005) Extracting product features and opinions from reviews. In: Proceedings human language technology conference and conference on empirical methods in natural language processing, pp 339–346
19. Qu L, Ifrim G, Weikum G (2010) The bag-of-opinions method for review rating prediction from sparse text patterns. In: Proceedings 23rd international conference on computational linguistics, pp 913–921
20. Riloff E, Jones R (1999) Learning dictionaries for information extraction by multi-level bootstrapping. In: Proceedings 16th national conference on artificial intelligence, pp 474–479
21. Sudo SKS, Grishman R (2003) An improved extraction pattern representation model for automatic ie pattern acquisition. In: Proceedings 43rd annual meeting of the association for computational linguistics, pp 224–231
22. Thet TT, Na JC, Khoo CS (2010) Aspect-based sentiment analysis of movie reviews on discussion boards. J Inf Sci 36:823–848
23. Titov I, McDonald R (2008a) A joint model of text and aspect ratings for sentiment summarization. In: Proceedings 46th annual meeting of the association for computational linguistics, pp 308–316
24. Titov I, McDonald R (2008b) Modeling online reviews with multi-grain topic models. In: Proceedings 17th international conference on world wide web, pp 111–120

25. Wilson T, Wiebe J, Hoffmann P (2005) Recognizing contextual polarity in phrase-level sentiment analysis. In: Proceedings human language technology conference and conference on empirical methods in natural language processing, pp 347–354
26. Wu Y, Wen M (2010) Disambiguating dynamic sentiment ambiguous adjectives. In: Proceedings 23rd international conference on computational linguistics, pp 1191–1199
27. Xu F, Uszkoreit H, Li H (2007) A seed-driven bottom-up machine learning framework for extracting relations of various complexity. In: Proceedings 47th annual meeting of the association for computational linguistics, pp 584–591
28. Xu F, Uszkoreit H, Krause S, Li H (2010) Boosting relation extraction with limited closed-world knowledge. In: Proceedings 23rd international conference on computational linguistics, pp 1354–1362
29. Yangarber R (2001) Scenarion customization for information extraction. PhD thesis, Department of Computer Science, Graduate School of Arts and Science. New York University, New York
30. Zhuang L, Jing F, Zhu X (2006) Movie review mining and summarization. In: Proceedings 15th international conference on information and knowledge management, pp 43–50

Yan Li received a M.S. degree from Beijing University of Posts and Telecommunications in 2009. He is currently a Ph.D. student in Beijing University of Posts and Telecommunications. Currently, his main research interests cover opinion mining and sentiment analysis.

Zhen Qin received her M.E. and B.E. degrees in automation from University of Science and Technology Beijing, China in 2009 and 2012, respectively. She is currently a Ph.D. student in Beijing University of Posts and Telecommunications.

Weiran Xu received his Ph.D. degree from Beijing University of Posts and Telecommunications in 2003. He is currently an associate professor in Web Searching Teaching and Research Center, Beijing University of Posts and Telecommunications. His current research fields include information retrieval, pattern recognition and machine learning.

Jun Guo received B.E. and M.E. degrees from Beijing University of Posts and Telecommunications (BUPT), China in 1982 and 1985, respectively, Ph.D. degree from the Tohuku- Gakuin University, Japan in 1993. At present he is a professor and a vice president of BUPT. His research interests include pattern recognition theory and application, information retrieval, content based information security, and network management. He has published over 200 papers, some of them are on world-wide famous journals or conferences including SCIENCE, Nature Scientific Reports, IEEE Trans. on PAMI, IEICE, ICPR, ICCV, SIGIR, etc. His book “Network management” was awarded by the government of Beijing city as a finest textbook for higher education in 2004.