

# AspNet: Aspect Extraction by Bootstrapping Generalization and Propagation Using an Aspect Network

Yunqing Xia · Erik Cambria · Amir Hussain

Received: 8 July 2014 / Accepted: 4 September 2014 / Published online: 19 September 2014  
© Springer Science+Business Media New York 2014

**Abstract** Aspect-level opinion mining systems suffer from concept coverage problem due to the richness and ambiguity of natural language opinions. Aspects mentioned by review authors can be expressed in various forms, resulting in a potentially large number of missing or incomplete aspects. This work proposes a novel unsupervised method to extract aspects from raw reviews with a broader coverage. Previous research has shown that unsupervised methods based on dependency relations are promising for opinion target extraction (OTE). In this work, we introduce Aspect Network (AspNet), an AspNet that further improves existing OTE methods by providing a new framework for modeling aspects. AspNet represents the general indecomposable atom aspects and their dependency relations in a two-layered, directed, weighted graph, based on which the specific decomposable compound aspects in reviews can be effectively extracted. AspNet is constructed through an unsupervised learning method that starts from a small number of human-defined, domain-dependent aspects, and bootstraps generalization and propagation in a large volume of raw reviews. In summary, the major contributions of this work are twofold:

Firstly, the proposed AspNet is a new framework in modeling aspects; secondly, an unsupervised method is proposed to construct AspNet in a bootstrapping manner within raw reviews to learn aspects automatically. Experimental results demonstrate that our proposed OTE method, based on AspNet, can achieve significant gains over baseline methods.

**Keywords** Aspect extraction · Opinion mining · Aspect Network · Unsupervised learning

## Introduction

Online shopping Web sites have accumulated billions of product reviews, which are usually very useful for agile manufacturers, smart retailers and picky consumers. Manufacturers go through such reviews to discover major complaints, dominating demands and new market trends. Retailers refer to such reviews to find products that may generate bigger profits, whereas consumers leverage on reviews to make shopping decisions. A popular approach to locate relevant reviews is to use search engines using the product/service name and/or aspect as a query. Unfortunately, current search engines fail to provide the required level of service in review retrieval. A large proportion of returned documents are not product reviews, whereas, even in relevant reviews, one has to exhaustively read through the reviews in order to locate the expected opinions. As a result, product reviews are only partially consulted, leading to many biased decisions.

Targeting the discovery of relevant opinions from a large collection of reviews automatically, opinion mining (OM) systems show great potential to address the aforementioned dilemma. A typical OM system involves two

---

Y. Xia (✉)  
Department of Computer Science, TNLlist, Tsinghua University,  
Beijing 100084, China  
e-mail: yqxia@tsinghua.edu.cn

E. Cambria  
School of Computer Engineering, Nanyang Technological  
University, Singapore 639798, Singapore  
e-mail: cambria@ntu.edu.sg

A. Hussain  
Division of Computing Science and Maths, School of Natural  
Science, University of Stirling, Stirling FK9 4LA, Scotland, UK  
e-mail: ahu@cs.stir.ac.uk

**Table 1** Review examples on *appearance* of digital camera

No.	Review
E1	片子外观漂亮 ( <i>appearance of the camera is beautiful</i> )
E2	外形笨重 ( <i>profile is cumbersome</i> )
E3	机身的线条流畅 ( <i>outline of the camera is smooth</i> )
E4	机器样子好看 ( <i>shape of the camera is beautiful</i> )
E5	身材娇小 ( <i>figure is delicate</i> )

tasks: (1) opinion extraction, which aims to recognize opinion elements (i.e., holder, target and expression) within reviews and (2) polarity judgment, which seeks to detect opinion polarity [7]. OM systems focusing on such tasks are usually referred to as aspect-level OM system. For the aforementioned scenario, aspect extraction is a key function to search results presentation or mining. In this work, we are keen on improving the aspect extraction methods.

### Problem Definition

Aspect-level opinion mining systems usually suffer from concept coverage because of the richness and the ambiguity of natural language opinions. Aspects mentioned by review authors can be expressed in various forms, resulting in a great number of missing or incomplete aspects. For example, the concept of appearance can be expressed through (at least) five different aliases, e.g., 外观 (*appearance*), 外形 (*profile*), 线条 (*outline*), 样子 (*shape*) and 身材 (*figure*), as shown in Table 1. For practical reasons, it is useful to represent such a set of synonyms (synset) through a unique concept, namely 外观 (*appearance*). This ascribes mainly to differences in personal habit, education background, consumption experience, etc.

This work proposes an unsupervised method that can extract aspects, also referred to as opinion targets, from raw reviews. Growing number of research efforts are being made to improve opinion target extraction (OTE). Review of the literature shows that unsupervised methods based on dependency relation are quite promising in OTE [11, 14]. Recently, Qiu et al. [17] reported encouraging results by bootstrapping propagation with a few aspects as seeds. However, this method faces three main issues:

1. **Aspects are used directly as seeds in bootstrapping.** If an aspect is too specific (e.g., *motion image quality*), the propagation capability is limited. Consequently, this approach has to assign manpower to select general aspects (e.g., *quality*) carefully [17]. In fact, general aspects can be obtained with a generalization treatment on the specific aspects.
2. **Propagation is conducted directly on the test data.** However, the test data are usually very small, which

further reduces propagation capability. Since there is a large volume of freely available raw reviews on the Web, the propagation process can be made much more effective through big data retrieval and analysis.

3. **Dependency rules are very important to the propagation process.** However, it is laborious and difficult to compile the rules. The patterns that combine the general aspects are also interesting, and we can assign a strict condition to control quality of the patterns.

### Our Proposal and Contributions

Based on thorough observations of review data, we make the following assumption:

**Assumption** Aspects in reviews are formed with general, indecomposable aspects according to some patterns.

Though obvious, the assumption plays a key role in this work. We design a propagation process as follows (see details in Sect. 3):

1. The real aspects in reviews are first generalized so as to obtain the general, indecomposable aspects, referred to as atom aspects, and the patterns.
2. Each atom aspect is classified to be a component or an attribute since they play different roles in forming aspects.
3. All the atom aspects including the newly discovered ones are put into one bag and clustered so that each discovered atom aspect is assigned to a synset of atom aspects.
4. The atom aspects in the synsets combine with each other with the patterns to generate some compound aspect candidates.
5. The compound aspect candidates are deemed valid if they appear in the review development data.

After one-round propagation, more aspects can be extracted. Next, we bootstrap the propagation process on the review development data, the method can extract a great number of aspects, resulting in a much bigger recall. To control the quality of the extracted aspects (i.e., accuracy), we exclude the patterns which occur in review development data fewer than 10 times.

We note that many variables are used in the propagation process. An effective structure is thus very important to our work, which is the motivation behind our proposal for Aspect Network (AspNet), a two-layered, directed, weighted graph.

The major contributions of this work are twofold: First, the proposed AspNet is a powerful framework in modeling aspects. In our view, aspect is a complicated data structure that involves atom aspects, components, attributes, synsets,

patterns and weights. AspNet makes it possible to share patterns between synonymous atom aspects, share edges between patterns, order atom aspects in patterns, to separate components and attributes, model aspect patterns, and to exclude less trustful patterns. Second, an unsupervised method is proposed to construct AspNet by bootstrapping generalization and propagation within raw reviews to learn aspects automatically. The whole method is based on atom aspect, making generalization and propagation more productive. Experimental results show that our method outperforms the baseline systems significantly on  $F$  measure and recall without significant loss in precision.

## Paper Organization

The rest of this paper is organized as follows. Related work is discussed in Sect. 2. In Sect. 3, we present theory of AspNet with examples. In Sect. 4, we describe the AspNet-based aspect extraction method. Next, we present experimental results and discussions in Sect. 5 and finally conclude the paper in Sect. 6.

## Related Work

Opinion mining research started to attract attention in the early 2000s. In the past decade, numerous research efforts have been made to detect feature-based opinions within news articles or reviews [2–4, 19]. This work focuses on aspect extraction, which seeks to determine which aspect an opinion is ascribed to. Existing research in aspect extraction is mainly carried out on online reviews. The traditional solution is to adopt the human-compiled aspect lexicon in pattern matching techniques. However, lexicon suffers from coverage issue, and out-of-vocabulary (OOV) aspects are constantly encountered, which results in low recall in aspect extraction. A great number of research efforts have been made to expand the lexicons automatically [16, 17], thus the rule-based approaches are applicable [15]. But it is always late in online opinion mining system when OOV aspects appear in reviews. Thus online extraction of aspects attracts much research interests recently.

Methods in aspect extraction can be classified into two groups: supervised and unsupervised. The difference lies in whether labeled training data are required. The supervised methods view the aspect extraction task as a special case of the general information extraction problem. The dominant methods are based on sequential labeling algorithms. For example, a lexicalized HMM model is designed by Jin and Ho [10] to learn patterns for extracting aspects. Conditional random fields (CRF) is used by Jakob and Gurevych [9] for developing a domain independent extraction system.

Li et al. [12] integrated Skip-CRF and Tree-CRF to extract aspects. Researchers also explore collocation of aspect and opinion word in aspect extraction. For example, Kobayashi et al. [11] employ a tree-structured classifier to predict the input pairs of aspect and opinion word. Yu et al. [21] use the one-class support vector machines (SVM) algorithm to extract aspects. Two common drawbacks of supervised methods are notable. Firstly, training data are often difficult to obtain, which usually involve laborious manpower. Secondly, aspects are domain-dependent and domain adaption is a formidable challenge.

On the other hand, the unsupervised methods are popular as they require no training data, and the large-scale raw reviews retrieved by the crawlers can be used without any compilation. The general assumption of unsupervised methods is word dependency relation in either a statistical or syntactical manner. For example, Hu and Liu [8] proposed to extract aspects with association miner considering dependency between aspects and opinion keywords. Popescu and Etzioni [14] proposed to calculate a pointwise mutual information (PMI) score between a phrase and some meronymy discriminators to extract the aspects. Ghani et al. [6] proposed to infer aspects using the iterative co-EM algorithm, in which generic and domain-specific aspects are considered as seeds and applied to locate product aspects within reviews with attribute-value pairs. Kobayashi et al. [11] proposed to extract aspect-of relation from a blog corpus with machine learning techniques and used the relations as statistical patterns for aspect extraction. Zhu et al. [22] proposed a method based on the  $C$ -value measure for extracting multi-word aspects. Qiu et al. [17] view a few finely selected aspects as seeds and proposed a bootstrapping (i.e., double-propagation) method to extract new aspects using predefined dependency rules. Though they require no training data, unsupervised methods usually start from a knowledge resource, say *WordNet*, meronymy discriminators and lexicon.

This work adopts the bootstrapping solution in aspect extraction with assistance of dependency rules [17]. However, our work differs from Qiu et al. [17] in three aspects. Firstly, the general aspects are used directly as seeds in bootstrapping in Qiu et al. [17]. Differently, the aspects are first decomposed into atom aspects in our work, and the atom aspects are used as seeds in bootstrapping. Secondly, Qiu et al. [17] conduct propagation on the small test data directly. But our method first makes use of a large volume of development raw reviews to construct an AspNet, which is in turn applied on the test data to extract opinion targets. Dependency rule plays a key role in Qiu et al. [17] which is laborious and difficult to compile. As a comparison, we discover statistical patterns and assign a strict condition to control the quality of the patterns. The advantage is that patterns can be obtained automatically.

**Table 2** The aspects in the example reviews

No.	Aspects
E1	相机外观, ( <i>appearance of the camera</i> )
E2	外形 ( <i>profile</i> )
E3	机身的线条 ( <i>outline of the body</i> )
E4	机器样子 ( <i>shape of the camera</i> )
E5	身材 ( <i>figure</i> )

**Table 3** The atom aspects in the aspects

No.	Atom aspect(s)
E1	相机 ( <i>camera</i> ), 外观 ( <i>appearance</i> )
E2	外形 ( <i>profile</i> )
E3	机身 ( <i>body</i> ) 线条 ( <i>outline</i> )
E4	机器 ( <i>camera</i> ) 样子 ( <i>shape</i> )
E5	身材 ( <i>figure</i> )

**Table 4** The compound aspects in the reviews

No.	Compound aspect
E1	相机外观, ( <i>camera appearance</i> )
E2	–
E3	机身线条 ( <i>body outline</i> )
E4	机器样子 ( <i>camera shape</i> )
E5	–

## Aspect Network

### Definitions

In a large growing of literature on opinion mining, terms are used inconsistently by researchers inconsistently. For example, Kobayashi et al. [11] use aspect, but Ghani et al. [6] use attribute, and Hu and Liu [8] use feature. In NTCIR MOAT tasks, opinion target is used by Seki et al. [18]. We follow the definition in [11] in presenting the AspNet. Note that we still use aspect in the AspNet. As it presents knowledge discovered from reviews, the network is actually about aspect rather than opinion target. We argue that an aspect will be called opinion target only if it is used to form an opinion in a review.

For the review examples in Table 1, the contained aspects are given in Table 2.

**Definition 1** (*Atom aspect*) An atom aspect is a general, indecomposable aspect.

In practice, atom aspect is usually smaller in size and can be obtained by decomposing to aspects in reviews. For

**Table 5** Aspect patterns in the reviews

No.	Aspect pattern
E1	{ <i>Synset_camera</i> }{ <i>camera_appearance</i> }
E2	–
E3	{ <i>Synset_camera</i> }的{ <i>Synset_appearance</i> }
E4	{ <i>Synset_camera</i> }{ <i>Synset_appearance</i> }
E5	–

the aspects in Table 2, the corresponding atom aspects are given in Table 3.

**Definition 2** (*Compound aspect*) A compound aspect is an aspect which combines more than one atom aspects.

Compound aspect becomes a real aspect once it is verified with real reviews. For the aspects in Table 2, the corresponding compound aspects are given in Table 4.

**Definition 3** (*Component*) A component is an atom aspect which is a part, a function or a phenomenon of an object.

Physically, the component can be seen or experienced by users. For the atom aspects in Table 3, 相机 (*camera*), 机身 (*body*) and 机器 (*camera*) are components

**Definition 4** (*Attribute*) An attribute is an atom aspect which is an abstract property of an object, which happens when human feels the object.

For the aspects in Table 3, 外观, (*appearance*), 外形 (*profile*), 线条 (*outline*), 样子 (*shape*) and 身材 (*figure*) are attributes.

**Definition 5** (*Atom aspect synset*) An atom aspect synset is the set of atom aspects that carry the same concept.

For example, 相机 (*camera*), 相机 (*camera*) and 机器 (*camera*) in fact all refer to camera thus they appear within one atom aspect synset, which is labeled by *Synset\_camera*. Similarly, 外观 (*appearance*), 外形 (*profile*), 线条 (*outline*), 样子 (*shape*) and 身材 (*figure*) also appear under one atom aspect synset, which is labeled by *Synset\_appearance*.

**Definition 6** (*Aspect pattern*) An aspect pattern is the pattern that combines a few atom aspect synsets to form compound aspects.

In practice, aspect pattern holds the following form.

$$\{A_c\}^* \{string\{B_c\}^*\}^*, \quad (1)$$

where  $A_c$  and  $B_c$  represent two atom aspect synsets, and string is a constant. For the aspects in Table 2, the corresponding aspect patterns are given in Table 5.

Aspect pattern exhibits two characteristics: (1) Each aspect pattern involves at least one compound(s) explicitly or implicitly and (2) each aspect pattern involves only one attribute. In practice, compound aspect is generated with certain compound(s), attribute and aspect pattern.

Formalism

The AspNet is a two-layer directed, weighted graph  $G^{AspNet}$  defined as follows:

$$G^{AspNet} = \langle V, E; W^E \rangle : V = V^{COM} \cup V^{ATT}, E = E^{COM} \cup E^\theta. \tag{2}$$

Notations are outlined below:

- $V$ : The set of vertexes, each represents an atom aspect synset;
- $E$ : The set of edges, each of which represents a connection between two atom aspect synsets;
- $V^{COM}$ : The set of vertexes, each of which represents a component synset;
- $V^{ATT}$ : The set of vertexes, each of which represents an attribute synset;
- $E^{COM}$ : The set of edges, each of which represents an edge between two component synsets;
- $E^\theta$ : The set of edges, each of which represents an edge between component synset and attribute synset;
- $W^E$ : The set of weights, each of which is assigned to an edge in  $E$ .

An illustrative AspNet graph is given in Fig. 1. Six characteristics of AspNet are worth noting:

1. Each node of AspNet represents a synset of the atom aspects. That is, the atom aspects in a synset are synonymous, e.g., 外观 (appearance), 外形 (profile), 线条 (outline), 样子 (shape) and 身材 (figure)}. This makes it possible that synonyms can share patterns.

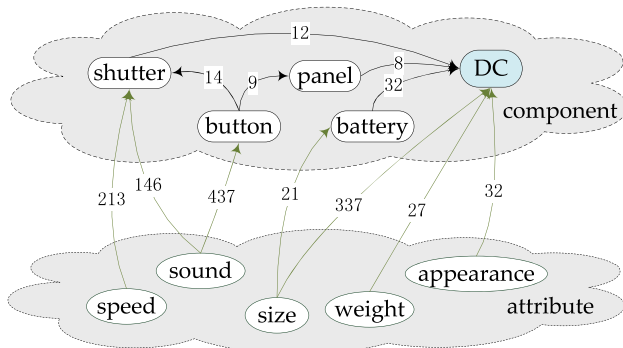


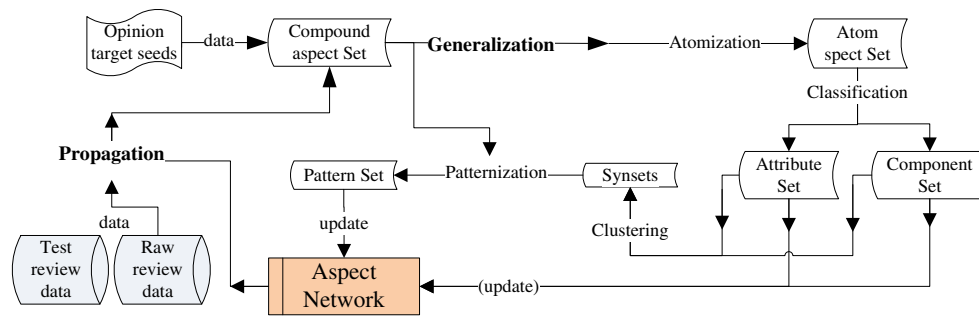
Fig. 1 An illustrative AspNet graph for the digital camera domain

2. Each edge of AspNet represents a step on a pattern. The graph framework makes it possible that patterns can share edges.
3. Every edge is directed. That is to say, the two atom aspects on the edge must appear in the related patterns in a determined order. For example, edge  $\langle \text{相机} (camera) \rightarrow \text{外观} (appearance) \rangle$  indicates that 相机 (camera) must appear before 外观 (appearance) in this pattern.
4. AspNet is a two-layered graph, in which the upper layer represents components and the lower attributes. This is because component and attribute play different roles in forming aspects.
5. Each edge is assigned a weight, which reflects how many times the two atom aspects appear in such an order within the development data. As depicted before, the weights are useful for excluding the less trustful patterns.
6. A few connected edges form a path, which represents a pattern used to generate aspect candidates.

Theoretical Analysis

Advantages of AspNet are summarized as follows: Firstly, the atom aspect is general and indecomposable, which brings AspNet enormous propagation capability in recognition of flexibly given aspects. As synset is involved, one real aspect is able to generate a group of aspect candidates with AspNet. For example, if the aspect 相机外观 (camera figure) is extracted from reviews, we can safely infer the following aspect candidates with the pattern  $\{Synset\_camera\}\{Synset\_appearance\}$  based on the synset  $\{Synset\_camera=\{\text{外观,外貌,轮廓,外形}\}; \text{相机外貌} (camera\ appearance), \text{相机轮廓} (camera\ profile), \text{相机外形} (camera\ shape)\}$ . For another example, 相机 (camera) is also an atom aspect and corresponds to atom aspect synset  $\{Synset\_camera=\{\text{相机,机器,机子}\}\}$ . Following the same propagation rule, the aspect 相机外观 (camera figure) can suggest eleven reasonable candidates. Thanks to the propagation process, AspNet finally becomes very powerful to suggest thousands of reasonable candidates, resulting in broader coverage in aspect extraction.

Secondly, AspNet provides sufficient restrictions to avoid false candidates. It can be seen from Fig. 1, edges within AspNet are all directed, resulting in strict word order in patterns. Word order is crucial for certain languages such as Chinese. For noun+noun combinations, the word order is fixed. For example, 相机 (camera) always appears before 外观 (figure) in reviews, while the inverse combination is invalid, namely, cannot be found within any



**Fig. 2** Workflow for the aspect extraction method

reviews. For other combinations such as adjective+noun and adverb+verb, the word order is not flexible. For example, 漂亮 (*beautiful*) can appear before 外观 (*figure*) in the comment 它有一个漂亮外观 (*It has a beautiful figure*). Meanwhile, 漂亮 (*beautiful*) can also appear after 外观 (*figure*) in the comment 它的外观漂亮 (*Its figure is beautiful*). We rely on the reviews to determine direction of the edges, and the direction is in turn used to restrict propagation.

The AspNet created by our aspect extraction method reveals some interesting information. The AspNet shows that, in real reviews, attributes and components must work with each other in order to form aspects. Another finding is that the core part of opinion is the attribute, which contacts opinion keyword explicitly or implicitly. These findings help further our understanding of how opinion is formed.

## Aspect Extraction Method

### Workflow

In our work, the AspNet is constructed with our aspect extraction method. We put the test data into the development data. Once the AspNet is constructed, aspects in the test data are naturally extracted. Overall, our OTE method comprises two major parts, i.e., generalization and propagation. It executes in a bootstrapping manner (see Fig. 2).

In the workflow, the aspect seeds are input as the initial compound aspects for generalization and propagation. After the first cycle, more compound aspects are extracted from the raw review data. After another few cycles, the size of the compound aspect set starts to stabilize and the aspect extraction task can be concluded.

In the *generalization* part, compound aspects are first resolved into atom aspects, referred to as the *atomization* module. Following this, atom aspects are classified into components and attributes, referred to as the *classification* module, and are further assigned appropriate synsets, referred

to as the *clustering* module. Finally, aspect patterns are generalized from the compound aspects, referred to as the *patternization* module. In the *propagation* part, synsets and patterns are applied on raw review data to find compound aspects.

### Generalization

The generalization part seeks to extract atom aspects from compound aspects, in order to: classify atom aspects into components and attributes; find the synset that every new atom aspect carries; and finally, generate aspect patterns from the compound aspects.

### Atomization

Atom aspects are extracted based on the degree of cohesion and flexibility. The cohesion degree reflects how the parts are cohesive to each other. Let  $w_1$  and  $w_2$  be two neighboring words, the cohesion degree is obtained by calculating the point-wise mutual information [11] as follows:

$$MI(w_1, w_2) = \ln \frac{P(w_1, w_2)}{P(w_1)P(w_2)}, \quad (3)$$

in which  $P(w_1, w_2)$  is the joint probability of  $w_1$  and  $w_2$ , and  $P(w_1)$  is the probability of  $w_1$  and  $P(w_2)$  the probability of  $w_2$ .

The flexibility degree of word  $w$  is calculated as follows:

$$FD(w) = \frac{1}{2} \left( \frac{\sum_{w_i \in N^L} \frac{1}{N^R(w_i)}}{N^L(w)} + \frac{\sum_{w_i \in N^R} \frac{1}{N^L(w_i)}}{N^R(w)} \right), \quad (4)$$

in which  $N^L$  denotes the set of neighboring words to the left, and  $N^R$  is the set of neighboring words to the right. The function  $N^L(x)$  returns the number of unique left-neighboring words, and  $N^R(x)$  returns those of right-neighboring words. It can be seen from Eq. (2) that the flexibility degree reflects flexibility from both sides.

We select the words as atom aspects if cohesion degree and flexibility degree both satisfy empirical thresholds, which are obtained in our experiments.

### Classification

The classification module aims at differentiating components from attributes with a probabilistic classifier by considering the following two features.

#### (1) Average Edit Distance ( $d^{AVG}$ )

Average edit distance measures string similarity which is calculated as follows:

$$d^{AVG}(t|X) = \frac{1}{|X|} \sum_{x_i \in X} d(t, x_i), \quad (5)$$

where  $t$  denotes the to-be-classified atom aspects,  $X = x_i$  represents an atom aspect set of component or attribute,  $|X|$  represents the size of set  $X$  and  $d(t, x_i)$  is the function to measure edit distance between  $t$  and  $x_i$ . With Eq. 5, we are now able to calculate how likely it is for an atom aspect to be from component set  $C$  or attribute set  $A$ .

#### (2) Overall Position Tendency ( $t^{OVA}$ )

Overall position tendency measures how likely it is for an atom aspect to be a component or an attribute according to position heuristics. In certain language, the attributes tend to appear at the end of compound aspects. So, the overall position tendency is calculated as follows:

$$t^{OVA}(t) = \frac{\text{count}(t, A)}{\text{count}(C, t)}, \quad (6)$$

where  $\text{count}(t, A)$  returns the number of compound aspects in which  $t$  appears before the attributes, and  $\text{count}(C, t)$  returns the number of compound aspects in which  $t$  appears after the components.

Note that the initial component and attribute sets are extracted from the seed aspects. We notice that attribute words are common in different domains, say price and weight. To improve coverage, we extract human-compiled attribute words from *WordNet* and *HowNet* [5]. Finally, each atom aspect is classified as component or attribute by simply comparing  $d^{AVG}$  and  $d^{OVA}$ .

We realize that the imbalanced data may influence the classification quality. We will adopt algorithms in [1] to handle the imbalanced issue.

### Clustering

The clustering module seeks to assign synset labels to every new atom aspect. We first apply the k-means clustering

algorithm to group atom aspects into a few clusters. We then adjust parameters to find a cluster that satisfies the following two conditions:

1. The cluster contains more than three atom aspects carrying the same synset label;
2. The cluster contains at least one new atom aspects.

Once such a cluster is found, the atom aspects with an unknown synset label are considered synonyms of the other atom aspects. This updates the atom aspect set. We repeat the clustering process until no unknown atom aspect can be assigned any synset label. The following two features are considered in atom aspect clustering.

1. Opinion words neighboring the atom aspects in the raw review sentences.
2. Edit distance between unknown atom aspects and known ones.

There must be some unknown atom aspects that cannot be assigned any synset label. We run the clustering algorithm merely on these unknown atom aspects and attempt to find new synsets. A new synset is created if one cluster is found satisfying the following two conditions.

Now a synset label should be selected for the new synset. We choose the atom aspect with the most occurrences.

After the above procedure, there might still be some atom aspects with an unknown synset label. These atoms could be catered for in the next iteration.

### Patternization

The patternization module seeks to discover patterns that form aspects in reviews. The aspect patterns follow regular expressions of the form in Eq. 1.

For example, we have a pattern  $\langle \text{color of image} \rangle$ . In this pattern, *image* is the synset label named after *image*, *color* the synset label named after *color*, and *of* the pattern string.

To control quality of the patterns, we assign a strict condition on the discovered patterns. That is, we exclude the patterns that occur in review development data for five times.

### AspNet Update

Once new atom aspects and new aspect patterns are found, we create nodes and/or edges in the AspNet. Note that no node will be created if a new atom aspect carries a known synset label. New edges should be drawn when new aspect patterns are found.

### Propagation

The propagation algorithm aims to extract unknown aspects. To achieve this, the algorithm makes use of AspNet and dependency to infer the aspect candidates.

#### (1) Propagation with AspNet

AspNet allows synonymous atom aspects to share synset and the pattern. This makes it possible to infer new aspects with synonymous atom aspects and aspect patterns with AspNet. In other words, if an edge exists between atom aspect synset A and synset B, combinations between atom aspects in synset A and in synset B very probably exist. Based on this assumption, a large number of aspect candidates can be inferred.

We note that the assumption might lead to invalid aspect candidates. Thus we apply two processes to exclude them. The first process is to apply the *sequence confidence measure* on the raw review development data to estimate how likely atom aspect A appears before atom aspect B. To be formal, given that an aspect candidate X contains N atom aspects, i.e.,  $A_i, i = 1, \dots, N$ , the *sequence confidence* (SC) is calculated as follows:

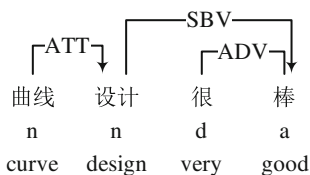
$$SC(X) = \sum_{i < j} \frac{count(A_i, A_j)}{C_N^2}, \tag{7}$$

where  $count(A_i, A_j)$  denotes number of occurrences of  $A_i$  appearing before  $A_j$ , and  $C_N^2$  is the number of binary combinations. In our experiments, an empirical threshold for sequence confidence is set to exclude the invalid aspect candidates.

#### (2) Propagation with dependency

AspNet is powerful enough to find new combinations of atom aspects, but not new atom aspects. Thus, a new process should be designed to discover unknown atom aspects. We make use of dependency to achieve this goal. Figure 3 presents a typical output of the dependency parser.

In Fig. 3, a 3-tuple {曲线 (*curve*), 设计 (*design*), ATT} exhibits an ATT dependency from word 曲线 (*curve*) to



**Fig. 3** An illustrative dependency parsing output. *ATT* represent a modification dependency, *SBV* a subjective one, and *ADV* an adverbial one. The letter *n* denotes a nouns, *d* denotes an adverb, and *a* denotes an adjective

word 设计 (*design*). Suppose that the word 曲线 (*curve*) is an atom aspect but 设计 (*design*) is not, we can make good use of the specific dependency to predict a new atom aspect. We carefully select candidates with the following strict conditions: To be formal, we use 3-tuples  $\langle A, B, dp \rangle$  to represent the dependency from word A to word B, where A and B represent two words, and *dp* denotes a dependency. If word A is an atom aspect in AspNet, B is considered a new atom aspect if:

1. Word A is a noun or verb;
2. Word B is a noun or verb;
3. *dp* is one of five dependency types: ATT(attribute), COO(coordinate), APP(appositive), DE(的, A-of-B structure);
4. The 3-tuple  $\langle A, B, dp \rangle$  appears for more than 5 times in the review development data.

The fourth condition is set to control the quality of the discovered new atom aspect in an empirical way.

### Bootstrapping

The bootstrapping algorithm sequentially executes the generalization and the propagation algorithm and updates AspNet in each cycle. The incremental learning process terminates when the AspNet is stabilized, namely a new atom aspect or pattern cannot be extracted any more. This goal is usually achieved in a few cycles.

## Evaluation

### Setup

#### Test Set

We use the Opinmine corpus [20] as test set in this evaluation. Four domains are involved in the second version: digital camera, mobile phone, hotel and restaurant. Statistics of the Opinmine corpus v2 are given in Table 6.

As no annotation data is required in this work, we use the whole Opinmine corpus as test set.

**Table 6** Statistics of Opinmine corpus v2

Domain	# of reviews	# of opinions	# of unique sentiment words
Mobile phone	1,200	6,034	1,437
Digital camera	1,200	9,706	1,689
Hotel	1,200	5,566	1,500
Restaurant	1,200	4,869	1,192



**Table 7** Statistics of the seed aspects and examples

Domain	Count	Aspect example
Digital camera	84	电池价格 ( <i>battery price</i> )
Mobile phone	96	屏幕分辨率 ( <i>screen resolution</i> )
Hotel	46	房间布局 ( <i>room layout</i> )
Restaurant	53	服务态度 ( <i>service attitude</i> )

### Seed Aspects

A few high-frequent aspects are selected as seeds for the four domains, separately. Statistics of the seed aspects as well as examples are given in Table 7.

It is worth noting that the seed aspects are not selected by human experts. We extract the aspects of digital camera and mobile phone from a digital user manual, respectively. For the hotel domain and the restaurant domain, we extract aspects from the inspection guidelines, which are available on the Internet. The data source can guarantee that the seed aspects are correct and professional.

### Development Set

The review development data are downloaded with crawlers from e-commerce Web sites. For each domain, we make use of 200K raw reviews in this work.

### Evaluation Metrics

We use precision ( $p$ ), recall ( $r$ ) and  $f - 1$  score ( $f$ ) as our evaluation metrics.

### Experiment 1: Methods for Aspect Extraction

This experiment aims at comparing our method against the baseline methods.

The following baseline methods are implemented in this work:

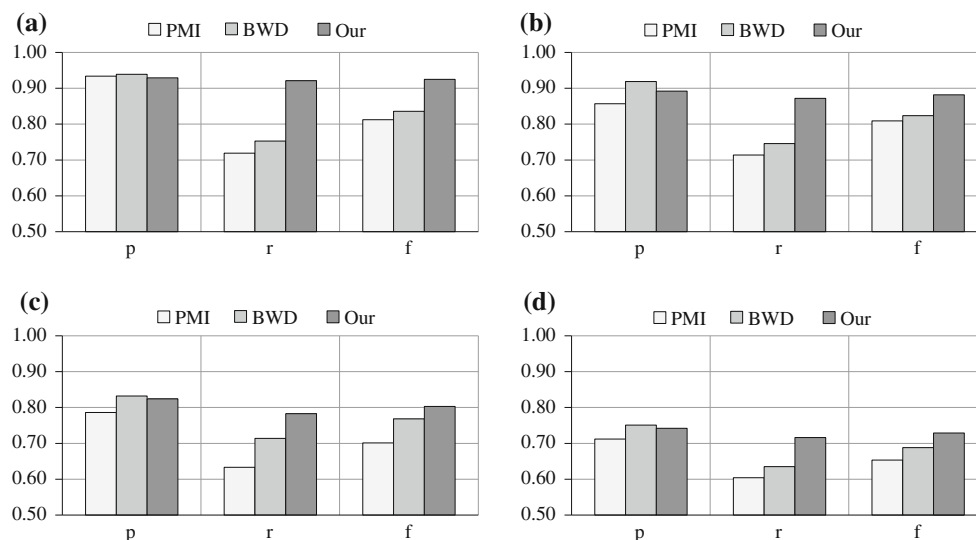
- *Point-wise Mutual Information* (PMI). Starting from the human-compiled aspects, the PMI-based method explores statistical information (i.e., co-occurrences) to extract aspects in reviews [14].
- *Bootstrapping with Dependency* (BWD). Using the input aspects as seeds and bootstrapping with dependency rules [17].

Note that we do not compare our method with any supervised learning methods as our method is an unsupervised one.

The proposed method in this work implements all modules in Fig. 2. The thresholds in our method are configured empirically, e.g., threshold for cohesion degree is set as 0.001, 0.333 for flexibility degree, and 0.8 for sequence confidence.

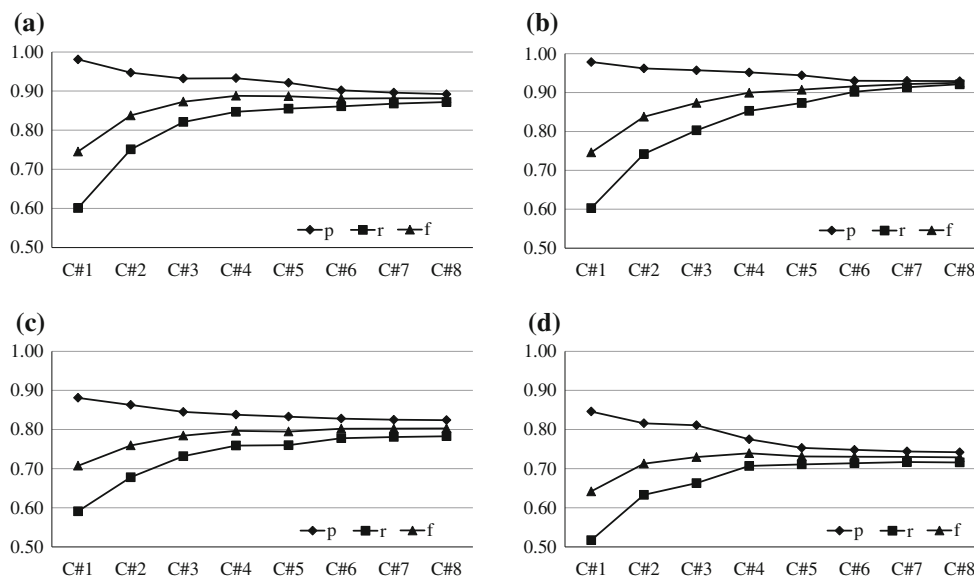
In all the aforementioned methods, we employ the HIT Language Technology Platform [13] for Chinese word segmentation and dependency parsing. Experimental results are shown in Fig. 4.

Experimental results in Fig. 4 consistently show that in all domains, our method improves recall significantly over the baseline methods, and the maximum improvement is 0.168 in the digital camera domain. This is because our method can discover more aspects from reviews by generalization and propagation in a bootstrapping manner. It



**Fig. 4** Experimental results: Precision ( $p$ ), recall ( $r$ ) and  $f - 1$  measure ( $f$ ) of the baseline methods and our method in the four domains. **a** Mobile phone, **b** digital camera, **c** hotel, **d** restaurant

**Fig. 5** Experimental results: Precision ( $p$ ), recall ( $r$ ) and  $f - 1$  measure ( $f$ ) of our method for eight cycles in the four domains. **a** Mobile phone, **b** digital camera, **c** hotel, **d** restaurant



also ascribes to the AspNet which organizes aspects with synsets of components and attributes with patterns.

It can also be seen from Fig. 4 that precision drops slightly in the four domains, and the maximum drop is 0.017 in the mobile phone domain. This indicates that the bootstrapping mechanism accepts some false candidates. However, the precision loss is very little. As shown in Fig. 4, the  $f - 1$  measure is improved in our method across all domains, and the maximum improvement is 0.065 in the digital camera domain. This is largely due to the maximum improvement in recall.

We also find that our method performs better in the product domains (i.e., *digital camera* and *mobile phone*) than in the service domains (i.e., *hotel* and *restaurant*). The reason is that aspects in the product domains are less flexible than in the service domains. For example, when they complain relating to noise, the service reviewers may use different aspects such as noise, cleaning, quietness, sound, horn and loud speaker. As can be seen from Fig. 4, the  $f - 1$  measure of all methods in the service domains is much lower than in the product domains. This indicates that our method is more suitable to the product domains.

## Experiment 2: Aspect Extraction Cycles

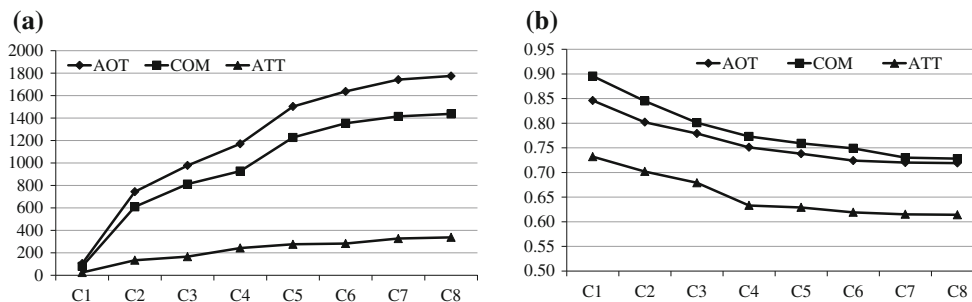
This experiment aims at observing how our method performs at different cycles. In this work, we execute the method for eight cycles. Experimental results of our method for the eight cycles are presented in Fig. 5 across the four domains.

Figure 5 discloses a consistent trend that the  $f - 1$  measure of our method improves gradually and starts to converge at cycle C#7. Comparing cycle C#1 and C#8,

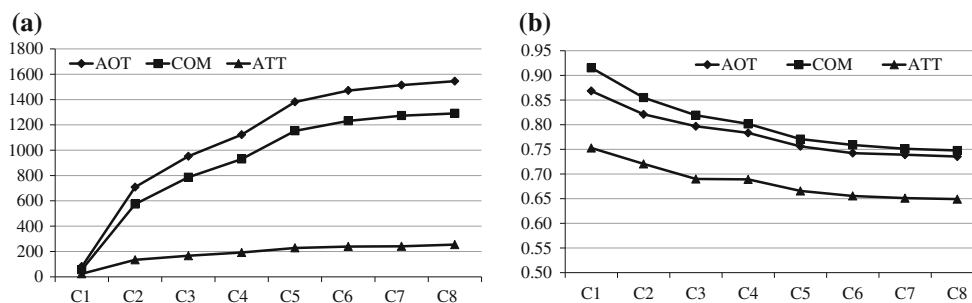
the method improves  $f - 1$  measure significantly by 0.179 in the digital camera domain, 0.137 in the mobile phone domain, 0.096 in the hotel domain and 0.087 in the restaurant domain, respectively. This indicates that the bootstrapping mechanism is indeed helpful. Although a slight precision loss occurs in later cycles, recall is improved by our method significantly, leading to big gain on the  $f - 1$  measure. The precision loss is caused by false aspect candidates accepted by our method. However, the strict conditions used in our method are important to exclude a large proportion of false candidates. This proves that our method is robust in handling large-scale raw reviews.

In order to understand how much the bootstrapping mechanism contributes, we count the atom aspects that are extracted at every cycle and calculate the extraction accuracy, respectively. Results are given in Figs. 6, 7, 8, 9 for the four domains.

A general trend can be seen from Figs. 6, 7, 8, 9 that the bootstrapping cycles indeed make significant contribution to aspect extraction. In the earlier cycles, accuracy drops significantly, which is due to the fact that a great number of aspects (including false ones) are extracted from reviews. The newly extracted aspects greatly help to improve the quality of the aspect patterns, which in turn help to find more atom aspects. In the later cycles, the patterns are made very precise, which lead to aspects being extracted from reviews more accurately. Resultantly, fewer atom aspects are discovered in the later cycles. In other words, the method starts to converge to a stable state. The above observations reveal that the cycles can be limited and the bootstrapping process may finally produce an optimal performance of our proposed method.

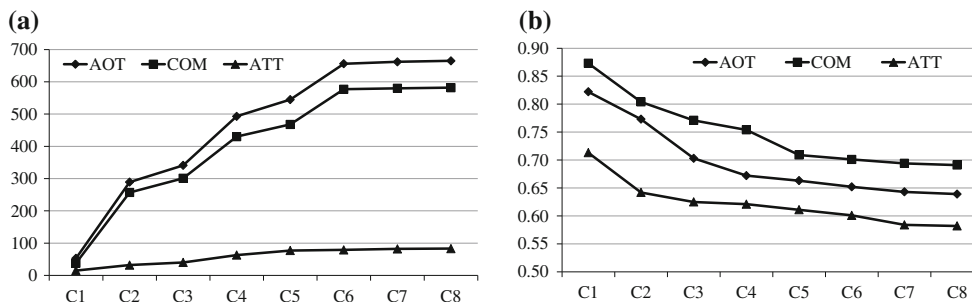
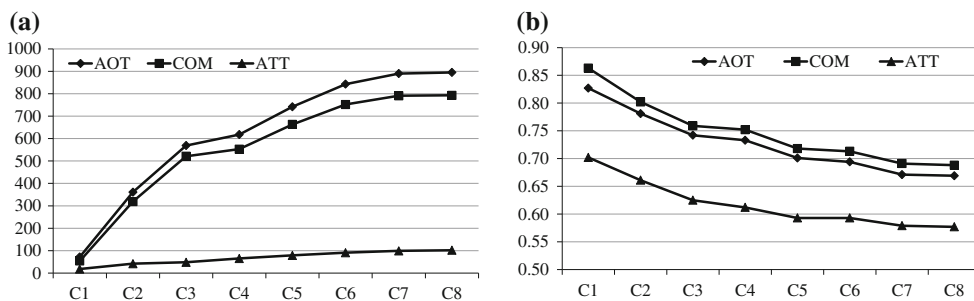


**Fig. 6** Results showing *count* and *accuracy* of components (COM), attributes (ATT) and atom aspects (AOT) extracted at every cycle in the *mobile phone* domain. **a** Count, **b** accuracy

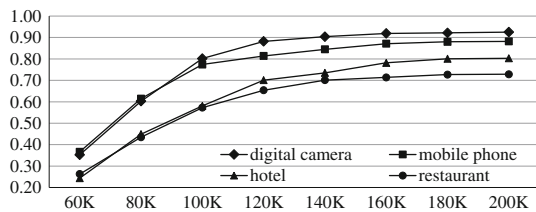


**Fig. 7** Results showing *count* and *accuracy* of components (COM), attributes (ATT) and atom aspects (AOT) extracted at every cycle in the *digital camera* domain. **a** Count, **b** accuracy

**Fig. 8** Results showing *count* and *accuracy* of components (COM), attributes (ATT) and atom aspects (AOT) extracted at every cycle in the *hotel* domain. **a** Count, **b** accuracy



**Fig. 9** Results showing *count* and *accuracy* of components (COM), attributes (ATT) and atom aspects (AOT) extracted at every cycle in the *restaurant* domain. **a** Count, **b** accuracy



**Fig. 10** Experimental results of our method using the  $f-1$  measure in the four domains (with various volumes of development data)

### Experiment 3: Volume of Development Data

This experiment seeks to observe how volume of development data influences the performance of the method. For each domain, we have 200K raw reviews. We generate smaller development data sets by randomly selecting some reviews from the whole collection. Finally, we obtain 8 development data sets for each domain, which comprise 60K, 80K, 100K, 120K, 140K, 160K, 180K and 200K reviews, respectively. We run our method on the 6 development data sets in each domain and present results using the  $f-1$  measure in Fig. 10.

It can be observed from Fig. 10 that more development data brings more gains. Our method performs poorly on the size-60K development set, but with more development data, the  $f-1$  measure of the method climbs quickly to above 0.9. This is because the AspNet is being gradually expanded, covering more atom aspects and effective patterns.

It is also implied in Fig. 10 that a development set bigger than 200K does not bring more gains. This is because the AspNet can be well constructed with only about 16K reviews. When more reviews are input, very few atom aspects will be discovered. That is, the AspNet remains relatively stable after 16K reviews are input, resulting in performance convergence.

### Conclusion and Future Work

Extraction of aspects from reviews is a core task in opinion mining. To address the coverage issue, we develop AspNet to model aspects with atom aspects and patterns. We also develop a new aspect extraction method by bootstrapping generalization and propagation. Experimental results confirm the significance and potential of our work. The novelty of this work is twofold: First, we present an effective framework, i.e., AspNet, to model the intrinsic structure of product/service aspects, that is, real aspects are composed of atom aspect synsets with patterns. This equips AspNet with tremendous power to form real aspects. Second, the

generalization process and propagation process are designed to put the power into practice. We further adopt the bootstrapping mechanism, by which the power is thoroughly exploited.

A number of challenging future work directions are currently being pursued. First, we attempt to further refine the aspect extraction method by developing new structure patterns. In this strand of work, we plan to use regular expressions to represent the patterns, which can also be further developed into rules. Second, we plan to make use of opinion keywords in aspect propagation. The collocations between aspects and opinion keywords can lead to the discovery of more heuristics in aspect detection.

**Acknowledgments** Research work in this paper is partially supported by NSFC (No. 61272233). We thank the anonymous reviewers for their valuable comments.

### References

1. Cai Q, He H, Man H. Imbalanced evolving self-organizing learning. *Neurocomputing*. 2014;133:258–70.
2. Cambria E, Hussain A. *Sentic computing: techniques, tools, and applications*. Dordrecht: Springer; 2012.
3. Cambria E, Schuller B, Xia Y, Havasi C. New avenues in opinion mining and sentiment analysis. *IEEE Intell. Syst.* 2013;28(2): 15–21.
4. Das D, Bandyopadhyay S. Sentence-level emotion and valence tagging. *Cogn. Comput.* 2012;4(4):420–35.
5. Dong Z, Dong Q. *Hownet and the computation of meaning*. River Edge: World Scientific Publishing Co., Inc.; 2006.
6. Ghani R, Probst K, Liu Y, Krema M, Fano A. Text mining for product attribute extraction. *SIGKDD Explor. Newsl.* 2006;8(1): 41–8.
7. Hu M, Liu B. Mining and summarizing customer reviews. In: *Proceedings of the tenth ACM SIGKDD international conference on knowledge discovery and data mining., KDD '04*New York, NY, USA: ACM; 2004. p. 168–77.
8. Hu M, Liu B. Mining opinion features in customer reviews. In: *Proceedings of the 19th national conference on artificial intelligence, AAAI'04*. AAAI Press; 2004. p. 755–60
9. Jakob N, Gurevych I. Extracting opinion targets in a single- and cross-domain setting with conditional random fields. In: *Proceedings of the 2010 conference on empirical methods in natural language processing., EMNLP '10*Stroudsburg, PA, USA: Association for Computational Linguistics; 2010. p. 1035–45.
10. Jin W, Ho HH. A novel lexicalized hmm-based learning framework for web opinion mining. *note from acm: a joint acm conference committee has been determined that the authors of this article violated acm's publication policy on simultaneous submissions. Therefore acm has shut off access to this paper.* In: *Proceedings of the 26th annual international conference on machine learning, ICML'09*New York, NY, USA: ACM; 2009. p. 465–72.
11. Kobayashi N, Inui K, Matsumoto Y. Extracting aspect-evaluation and aspect-of relations in opinion mining. In: *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL, EMNLP-CoNLL'07)*. Association for Computational Linguistics; 2007. p. 1065–74

12. Li F, Huang M, Zhu X. Sentiment analysis with global topics and local dependency. In: Fox M, Poole D, editors. Proceedings of the twenty-fourth conference on artificial intelligence, AAAI '10. AAAI Press; 2010
13. Ma J, Zhang Y, Liu T, Li S. A statistical dependency parser of chinese under small training data. In: Proceedings of workshop: beyond shallow analyses-formalisms and statistical modeling for deep analyses, IJCNLP'04; 2004. p. 1–5.
14. Popescu AM, Etzioni O. Extracting product features and opinions from reviews. In: Proceedings of the conference on human language technology and empirical methods in natural language processing, HLT '05Stroudsburg, PA, USA: Association for Computational Linguistics; 2005. p. 339–46.
15. Poria S, Cambria E, Ku L-WCG, Gelbukh A. In: Proceedings of the 25th international conference on computational linguistics.
16. Poria S, Cambria E, Winterstein G, Huang GB. Sentic patterns: dependency-based rules for concept-level sentiment analysis. Knowledge-Based Systems 2014.
17. Qiu G, Liu B, Bu J, Chen C. Opinion word expansion and target extraction through double propagation. *Comput. Linguist.* 2011;37(1):9–27. doi:[10.1162/coli\\_a\\_00034](https://doi.org/10.1162/coli_a_00034).
18. Seki Y, Evans DK, Ku LW, Chen HH, Kando N, Lin CY. Overview of opinion analysis pilot task at NTCIR-6. In: Proceedings of the workshop meeting of the National Institute of Informatics (NII) test collection for information retrieval systems (NTCIR), 2007. p. 265–78.
19. Sun R. Moral judgment, human motivation, and neural networks. *Cogn. Comput.* 2013;5(4):566–79.
20. Xu R, Xia Y, Wong KF, Li W. Opinion annotation in on-line chinese product reviews. In: LREC 2008.
21. Yu J, Zha ZJ, Wang M, Chua TS. Aspect ranking: identifying important product aspects from online consumer reviews. In: Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies, vol. 1, HLT'11Stroudsburg, PA, USA: Association for Computational Linguistics; 2011. p. 1496–505.
22. Zhu J, Wang H, Tsou BK, Zhu M. Multi-aspect opinion polling from textual reviews. In: Proceedings of the 18th ACM conference on information and knowledge management, CIKM '09New York, NY, USA: ACM; 2009. p. 1799–802.