Aspect and Entity Extraction for Opinion Mining

Lei Zhang and Bing Liu

Abstract: Opinion mining or sentiment analysis is the computational study of people's opinions, appraisals, attitudes, and emotions toward entities such as products, services, organizations, individuals, events, and their different aspects. It has been an active research area in natural language processing and Web mining in recent years. Researchers have studied opinion mining at the document, sentence and aspect levels. Aspect-level (called *aspect-based opinion mining*) is often desired in practical applications as it provides the detailed opinions or sentiments about different aspects of entities and entities themselves, which are usually required for action. Aspect extraction and entity extraction are thus two core tasks of aspect-based opinion mining. In this chapter, we provide a broad overview of the tasks and the current state-of-the-art extraction techniques.

1   Introduction

Opinion mining or sentiment analysis is the computational study of people's opinions, appraisals, attitudes, and emotions toward entities and their aspects. The entities usually refer to products, services, organizations, in-

Lei Zhang
Department of Computer Science, University of Illinois at Chicago
Chicago, United States
e-mail: lzhang32@gmail.com

Bing Liu
Department of Computer Science, University of Illinois at Chicago
Chicago, United States
e-mail: liub@cs.uic.edu

dividuals, events, etc and the aspects are attributes or components of the entities (Liu, 2006). With the growth of *social media* (i.e., reviews, forum discussions, and blogs) on the Web, individuals and organizations are increasingly using the opinions in these media for decision making. However, people have difficulty, owing to their mental and physical limitations, producing consistent results when the amount of such information to be processed is large. Automated opinion mining is thus needed, as subjective biases and mental limitations can be overcome with an objective opinion mining system.

In the past decade, opinion mining has become a popular research topic due to its wide range of applications and many challenging research problems. The topic has been studied in many fields, including natural language processing, data mining, Web mining, and information retrieval. The survey books of Pang and Lee (2008) and Liu (2012) provide a comprehensive coverage of the research in the area. Basically, researchers have studied opinion mining at three levels of granularity, namely, document level, sentence level, and aspect level. Document level *sentiment classification* is perhaps the most widely studied problem (Pang, Lee and Vaithyanathan, 2002; Turney, 2002). It classifies an opinionated document (e.g., a product review) as expressing an overall positive or negative opinion. It considers the whole document as a basic information unit and it assumes that the document is known to be opinionated. At the sentence level, sentiment classification is applied to individual sentences in a document (Wiebe and Riloff, 2005; Wiebe et al., 2004; Wilson et al., 2005). However, each sentence cannot be assumed to be opinionated. Therefore, one often first classifies a sentence as opinionated or not opinioned, which is called *subjectivity classification*. The resulting opinionated sentences are then classified as expressing positive or negative opinions.

Although opinion mining at the document level and the sentence level is useful in many cases, it still leaves much to be desired. A positive evaluative text on a particular entity does not mean that the author has positive opinions on every aspect of the entity. Likewise, a negative evaluative text for an entity does not mean that the author dislikes everything about the entity. For example, in a product review, the reviewer usually writes both positive and negative aspects of the product, although the general sentiment on the product could be positive or negative. To obtain more fine-grained opinion analysis, we need to delve into the aspect level. This idea leads to *aspect-based opinion mining,* which was first called the *feature-based opinion mining* in Hu and Liu (2004b). Its basic task is to extract and summarize people's opinions expressed on entities and aspects of entities. It consists of three core sub-tasks.

(1) identifying and extracting entities in evaluative texts

(2) identifying and extracting aspects of the entities

(3) determining sentiment polarities on entities and aspects of entities

For example, in the sentence "*I brought a Sony camera yesterday, and its picture quality is great*," the aspect-based opinion mining system should identify the author expressed a positive opinion about the picture quality of the Sony camera. Here *picture quality* is an aspect and *Sony camera* is the entity. We focus on studying the first two tasks here. For the third task, please see (Liu, 2012). Note that some researchers use the term *feature* to mean aspect and the term *object* to mean entity (Hu and Liu, 2004a). Some others do not distinguish aspects and entities and call both of them *opinion targets* (Qiu et al., 2011; Jakob and Gurevych, 2010; Liu et al., 2012)**,** *topics* (Li et al., 2012a) or simply *attributes* (Putthividhya and Hu, 2011) that opinions have been expressed on.

2   Aspect-Based Opinion Mining Model

In this section, we give an introduction to the aspect-based opinion mining model, and discuss the aspect-based opinion summary commonly used in opinion mining (or sentiment analysis) applications.

2.1 Model Concepts

Opinions can be expressed about anything such as a product, a service, or a person by any person or organization. We use the term *entity* to denote the target object that has been evaluated. An entity can have a set of components (or parts) and a set of attributes. Each component may have its own sub-components and its set of attributes, and so on. Thus, an entity can be hierarchically decomposed based on the *part-of* relation (Liu, 2006).

Definition (**entity**)**:** An *entity e* is a product, service, person, event, organization, or topic. It is associated with a pair, *e*: (*T, W*), where *T* is a hierarchy of *components* (or *parts*), *sub-components*, and so on, and *W* is a set of *attributes* of *e*. Each component or sub-component also has its own set of attributes.

Example: A particular brand of cellular phone is an entity, e.g., *iPhone*. It has a set of components, e.g., *battery* and *screen*, and also a set of attrib-

utes, e.g., *voice quality*, *size*, and *weight*. The battery component also has its own set of attributes, e.g., *battery life*, and *battery size*.

Based on this definition, an entity can be represented as a tree or hierarchy. The root of the tree is the name of the entity. Each non-root node is a component or sub-component of the entity. Each link is a *part-of* relation. Each node is associated with a set of attributes. An opinion can be expressed on any node and any attribute of the node.

Example: One can express an opinion about the iPhone itself (the root node), e.g., "*I do not like iPhone*", or on any one of its attributes, e.g., "*The voice quality of iPhone is lousy*". Likewise, one can also express an opinion on any one of the iPhone's components or any attribute of the component.

In practice, it is often useful to simplify this definition due to two reasons: First, natural language processing is difficult. To effectively study the text at an arbitrary level of detail as described in the definition is very hard. Second, for an ordinary user, it is too complex to use a hierarchical representation. Thus, we simplify and flatten the tree to two levels and use the term *aspects* to denote both components and attributes. In the simplified tree, the root level node is still the entity itself, while the second level nodes are the different aspects of the entity.

Definition (**aspect and aspect expression**)**:** The *aspects* of an entity *e* are the components and attributes of *e*. An *aspect expression* is an actual word or phrase that has appeared in text indicating an aspect.

Example: In the cellular phone domain, an aspect could be named *voice quality*. There are many expressions that can indicate the aspect, e.g., "*sound*," "*voice*," and "*voice quality*."

Aspect expressions are usually nouns and noun phrases, but can also be verbs, verb phrases, adjectives, and adverbs. We call aspect expressions in a sentence that are nouns and noun phrases *explicit aspect expressions*. For example, "*sound*" in "*The sound of this phone is clear*" is an explicit aspect expression. We call aspect expressions of the other types, *implicit aspect expressions*, as they often imply some aspects. For example, "*large*" is an implicit aspect expression in "*This phone is too large*". It implies the aspect *size*. Many implicit aspect expressions are adjectives and adverbs, which imply some specific aspects, e.g., *expensive* (price), and *reliably* (reliability). Implicit aspect expressions are not just adjectives and adverbs. They can be quite complex, for example, "*This phone will not easily fit in pockets*". Here, "*fit in pockets*" indicates the aspect *size* (and/or *shape*).

Like aspects, an entity also has a name and many expressions that indicate the entity. For example, the brand *Motorola* (entity name) can be expressed in several ways, e.g., "*Moto*", "*Mot*" and "*Motorola*" itself.

Definition (**entity expression**)**:** An *entity expression* is an actual word or phrase that has appeared in text indicating a particular entity.

Definition (**opinion holder**)**:** The *holder* of an opinion is the person or organization that expresses the opinion.

For product reviews and blogs, opinion holders are usually the authors of the postings. Opinion holders are more important in news articles as they often explicitly state the person or organization that holds an opinion. Opinion holders are also called *opinion sources*. Some research has been done on identifying and extracting opinion holders from opinion documents (Bethard et al., 2004; Choi et al., 2005; Kim and Hovy, 2006; Stoyanov and Cardie, 2008).

We now turn to opinions. There are two main types of opinions: *regular opinions* and *comparative opinions* (Liu, 2010; Liu, 2012). Regular opinions are often referred to simply as opinions in the research literature. A comparative opinion is a relation of similarity or difference between two or more entities, which is often expressed using the comparative or superlative form of an adjective or adverb (Jindal and Liu, 2006a and 2006b).

An *opinion* (or regular opinion) is simply a positive or negative view, attitude, emotion or appraisal about an entity or an aspect of the entity from an opinion holder. Positive, negative and neutral are called *opinion orientations*. Other names for opinion orientation are *sentiment orientation*, *semantic orientation*, or *polarity*. In practice, neutral is often interpreted as no opinion. We are now ready to formally define an opinion.

Definition (opinion): An *opinion* (or *regular opinion*) is a quintuple,

$$(e_i, a_{ij}, oo_{ijkl}, h_k, t_l),$$

where e_i is the name of an entity, a_{ij} is an aspect of e_i, oo_{ijkl} is the orientation of the opinion about aspect a_{ij} of entity e_i, h_k is the opinion holder, and t_l is the time when the opinion is expressed by h_k. The opinion orientation oo_{ijkl} can be positive, negative or neutral, or be expressed with different strength/intensity levels. When an opinion is on the entity itself as a whole, we use the special aspect GENERAL to denote it.

We now put everything together to define a model of entity, a model of opinionated document, and the mining objective, which are collectively called the *aspect-based opinion mining*.

Model of entity: An entity e_i is represented by itself as a whole and a finite set of aspects, $A_i = \{a_{i1}, a_{i2}, \ldots, a_{in}\}$. The entity itself can be expressed with any one of a final set of entity expressions $OE_i = \{oe_{i1}, oe_{i2}, \ldots, oe_{is}\}$. Each aspect $a_{ij} \in A_i$ of the entity can be expressed by any one of a finite set of aspect expressions $AE_{ij} = \{ae_{ij1}, ae_{ij2}, \ldots, ae_{ijm}\}$.

Model of opinionated document: An opinionated document d contains opinions on a set of entities $\{e_1, e_2, \ldots, e_r\}$ from a set of opinion holders $\{h_1, h_2, \ldots, h_p\}$. The opinions on each entity e_i are expressed on the entity itself and a subset A_{id} of its aspects.

Objective of opinion mining: Given a collection of opinionated documents D, discover all opinion quintuples $(e_i, a_{ij}, oo_{ijkl}, h_k, t_l)$ in D.

2.2 Aspect-Based Opinion Summary

Most opinion mining applications need to study opinions from a large number of opinion holders. One opinion from a single person is usually not sufficient for action. This indicates that some form of summary of opinions is desired. *Aspect-Based opinion summary* is a common form of opinion summary based on aspects, which is widely used in industry (see Figure 1). In fact, the discovered opinion quintuples can be stored in database tables. Then a whole suite of database and visualization tools can be applied to visualize the results in all kinds of ways for the user to gain insights of the opinions in structured forms as bar charts and/or pie charts. Researchers have also studied opinion summarization in the tradition fashion, e.g., producing a short *text summary* (Carenini et al, 2006). Such a summary gives the reader a quick overview of what people think about a product or service. A weakness of such a text-based summary is that it is not quantitative but only qualitative, which is usually not suitable for analytical purposes. For example, a traditional text summary may say "*Most people do not like this product*". However, a quantitative summary may say that 60% of the people do not like this product and 40% of them like it. In most applications, the quantitative side is crucial just like in the traditional survey research. Instead of generating a text summary directly from input reviews, we can also generate a text summary based on the mining results from bar charts and/or pie charts (see (Liu, 2012)).

Reviews

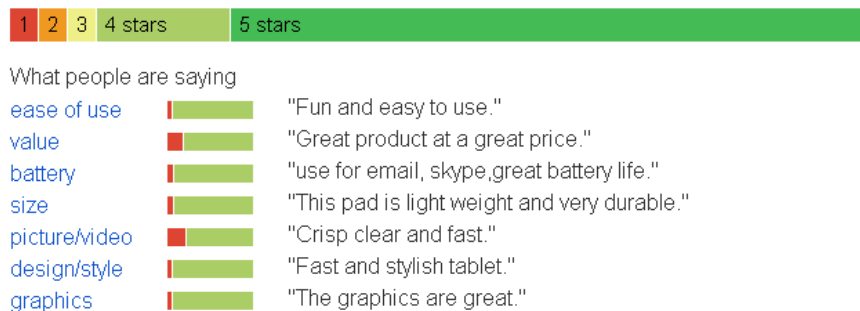**Summary** - Based on 1,668 reviews

| 1 | 2 | 3 | 4 stars | 5 stars |

What people are saying

ease of use		"Fun and easy to use."
value		"Great product at a great price."
battery		"use for email, skype,great battery life."
size		"This pad is light weight and very durable."
picture/video		"Crisp clear and fast."
design/style		"Fast and stylish tablet."
graphics		"The graphics are great."

Figure 1. Opinion summary based on product aspects of iPad (from Google Product[1])

3   Aspect Extraction

Both aspect extraction and entity extraction fall into the broad class of information extraction (Sarawagi, 2008), whose goal is to automatically extract structured information (e.g., names of persons, organizations and locations) from unstructured sources. However, traditional information extraction techniques are often developed for formal genre (e.g., news, scientific papers), which have some difficulties to be applied effectively to opinion mining applications. We aim to extract fine-grained information from opinion documents (e.g., reviews, blogs and forum discussions), which are often very noisy and also have some distinct characteristics that can be exploited for extraction. Therefore, it is beneficial to design extraction methods that are specific to opinion documents. In this section, we focus on the task of aspect extraction. Since aspect extraction and entity extraction are closely related, some ideas or methods proposed for aspect extraction can be applied to the task of entity extraction as well. In Section 4, we will discuss a special problem of entity extraction for opinion mining and some approaches for solving the problem.

Existing research on aspect extraction is mainly carried out on online reviews. We thus focus on reviews here. There are two common review formats on the Web.

---

[1] http://www.google.com/shopping

Format 1 – Pros, Cons and the detailed review: The reviewer is asked to describe some brief Pros and Cons separately and also write a detailed/full review.

Format 2 – Free format: The reviewer can write freely, i.e., no separation of pros and cons.

To extract aspects from Pros and Cons in reviews of Format 1 (not the detailed review, which is the same as Format 2), many information extraction techniques can be applied. An important observation about Pros and Cons is that they are usually very brief, consisting of short phrases or sentence segments. Each sentence segment typically contains only one aspect, and sentence segments are separated by commas, periods, semi-colons, hyphens, *&*, *and*, *but*, etc. This observation helps the extraction algorithm to perform more accurately (Liu, Hu and Cheng, 2005). Since aspect extraction from Pros and Cons is relatively simple, we will not discuss it further.

We now focus on the more general case, i.e., extracting aspects from reviews of Format 2, which usually consist of full sentences.

3.1 Extraction Approaches

We introduce only the main extraction approaches for aspects (or aspect expressions) proposed in recent years. As discussed in Section 2.1, there are two types of aspect expressions in opinion documents: *explicit aspect expression* and *implicit aspect expression*. We will discuss implicit aspects in Section 3.4. In this section, we focus on explicit aspect extraction. We categorize the existing extraction approaches into three main categories: language rules, sequence models and topic models.

3.1.1 Exploiting Language Rules

Language rule-based systems have a long history of usage in information extraction. The rules are based on contextual patterns, which capture various properties of one or more terms and their relations in the text. In reviews, we can utilize the grammatical relations between aspects and opinion words or other terms to induce extraction rules.

Hu and Liu (2004a) first proposed a method to extract product aspects based on association rules. The idea can be summarized briefly by two points: (1) finding frequent nouns and noun phrases as frequent aspects. (2)

using relations between aspects and opinion words to identify infrequent aspects. The basic steps of the approach are as follows.

Step 1: Find frequent nouns and noun phrases. Nouns and noun phrases are identified by a part-of-speech (POS) tagger. Their occurrence frequencies are counted, and only the frequent ones are kept. A frequency threshold is decided experimentally. The reason for using this approach is that when people comment on different aspects of a product, the vocabulary that they use usually converges. Thus, those nouns and noun phrases that are frequently talked about are usually genuine and important aspects. Irrelevant contents in reviews are often diverse, i.e., they are quite different in different reviews. Hence, those infrequent nouns are likely to be non-aspects or less important aspects.

Step 2: Find infrequent aspects by exploiting the relationships between aspects and opinion words (words that expressing positive or negative opinion, e.g., "*great*" and "*bad*"). The step 1 may miss many aspect expressions which are infrequent. This step tries to find some of them. The idea is as follows: The same opinion word can be used to describe or modify different aspects. Opinion words that modify frequent aspects can also modify infrequent aspects, and thus can be used to extract infrequent aspects. For example, "*picture*" has been found to be a frequent aspect, and we have the sentence,

> "*The pictures are absolutely amazing.*"

If we know that "*amazing*" is an opinion word, then "*software*" can also be extracted as an aspect from the following sentence,

> "*The software is amazing.*"

because the two sentences follow the same dependency pattern and "*software*" in the sentence is also a noun.

The idea of extracting frequent nouns and noun phrases as aspects is simple but effective. Blair-Goldensohn et al. (2008) refined the approach by considering mainly those noun phrases that are in sentiment-bearing sentences or in some syntactic patterns which indicate sentiments. Several filters were applied to remove unlikely aspects, for example, dropping aspects which do not have sufficient mentions along-side known sentiment words. The frequency-based idea was also utilized in (Popescu and Etzioni, 2005; Ku et al., 2006; Moghaddam and Ester, 2010; Zhu et al., 2009; Long et al., 2010).
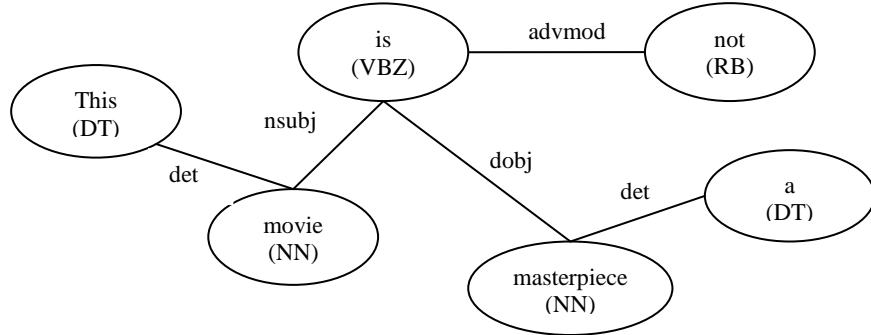
Figure 2. Dependency grammar graph (Zhuang et al., 2006)

The idea of using the modifying relationship of opinion words and aspects to extract aspects can be generalized to using dependency relation. Zhuang et al. (2006) employed the dependency relation to extract aspect-opinion pairs from movie reviews. After parsed by a dependency parser (e.g., MINIPAR[2] (Lin, 1998)), words in a sentence are linked to each other by a certain dependency relation. Figure 2 shows the dependency grammar graph of an example sentence, "*This movie is not a masterpiece*", where "*movie*" and "*masterpiece*" have been labeled as aspect and opinion word respectively. A dependency relation template can be found as the sequence *"NN - nsubj - VB - dobj - NN"*. NN and VB are POS tags. *nsubj* and *dobj* are dependency tags. Zhuang et al. (2006) first identified reliable dependency relation templates from training data, and then used them to identify valid aspect-opinion pairs in test data.

In Wu et al. (2009), a phrase dependency parser was used for extracting noun phrases and verb phrases as aspect candidates. Unlike a normal dependency parser that identifies dependency of individual words only, a phrase dependency parser identifies dependency of phrases. Dependency relations have also been exploited by Kessler and Nicolov (2009).

Wang and Wang (2008) proposed a method to identify product aspects and opinion words simultaneously. Given a list of seed opinion words, a bootstrapping method is employed to identify product aspects and opinion words in an alternation fashion. Mutual information is utilized to measure association between potential aspects and opinion words and vice versa. In addition, linguistic rules are extracted to identify infrequent aspects and

---

[2] http://webdocs.cs.ualberta.ca/~lindek/minipar.htm

opinion words. The similar bootstrapping idea is also utilized in (Hai et al., 2012).

Double propagation (Qiu et al., 2011) further developed aforementioned ideas. Similar to Wang and Wang (2008), the method needs only an initial set of opinion word seeds as the input. It observed that opinions almost always have targets, and there are natural relations connecting opinion words and targets in a sentence due to the fact that opinion words are used to modify targets. Furthermore, it found that opinion words have relations among themselves and so do targets among themselves too. The opinion targets are usually aspects. Thus, opinion words can be recognized by identified aspects, and aspects can be identified by known opinion words. The extracted opinion words and aspects are utilized to identify new opinion words and new aspects, which are used again to extract more opinion words and aspects. This propagation process ends when no more opinion words or aspects can be found. As the process involves propagation through both opinion words and aspects, the method is called *double propagation*. Extraction rules are designed based on different relations between opinion words and aspects, and also opinion words and aspects themselves. Dependency grammar was adopted to describe these relations.

The method only uses a simple type of dependencies called *direct dependencies* to model useful relations. A direct dependency indicates that one word depends on the other word without any additional words in their dependency path or they both depend on a third word directly. Some constraints are also imposed. Opinion words are considered to be adjectives and aspects are nouns or noun phrases. Table 1 shows the rules for aspect and opinion word extraction. It uses *OA-Rel* to denote the relations between opinion words and aspects, *OO-Rel* between opinion words themselves and *AA-Rel* between aspects. Each relation in OA-Rel, OO-Rel or AA-Rel can be formulated as a triple $\langle POS(w_i), R, POS(w_j) \rangle$, where $POS(w_i)$ is the POS tag of word w_i, and R is the relation. For example, in an opinion sentence "*Canon G3 produces great pictures*", the adjective "*great*" is parsed as directly depending on the noun "*pictures*" through *mod*, formulated as an OA-Rel $\langle JJ, mod, NNS \rangle$. If we know "*great*" is an opinion word and are given the rule 'a noun on which an opinion word directly depends through *mod* is taken as an aspect', we can extract "*pictures*" as an aspect. Similarly, if we know "*pictures*" is an aspect, we can extract "*great*" as an opinion word using a similar rule. In a nut shell, the propagation performs four subtasks: (1) extracting aspects using opinion words, (2) extracting aspects using extracted aspects, (3) extracting opinion words using the extracted aspects, and (4) extracting opinion words using both the given and the extracted opinion words.

	Observations	Output	Examples
R1$_1$ (OA-Rel)	$O{\rightarrow}O\text{-}Dep{\rightarrow}A$ **s.t.** $O{\in}\{O\}$, $O\text{-}Dep{\in}\{MR\}$, $POS(A){\in}\{NN\}$	$a = A$	*The phone has a <u>good</u> "screen".* *good${\rightarrow}$**mod**${\rightarrow}$screen*
R1$_2$ (OA-Rel)	$O{\rightarrow}O\text{-}Dep{\rightarrow}H{\leftarrow}A\text{-}Dep{\leftarrow}A$ **s.t.** $O{\in}\{O\}$, $O/A\text{-}Dep{\in}\{MR\}$, $POS(A){\in}\{NN\}$	$a = A$	*"iPod" is the <u>best</u> mp3 player.* *best${\rightarrow}$**mod**${\rightarrow}$player${\leftarrow}$**subj**${\leftarrow}$iPod*
R2$_1$ (OA-Rel)	$O{\rightarrow}O\text{-}Dep{\rightarrow}A$ **s.t.** $A{\in}\{A\}$, $O\text{-}Dep{\in}\{MR\}$, $POS(O){\in}\{JJ\}$	$o = O$	same as R1$_1$ with *screen* as the known word and *good* as the extracted word
R2$_2$ (OA-Rel)	$O{\rightarrow}O\text{-}Dep{\rightarrow}H{\leftarrow}A\text{-}Dep{\leftarrow}A$ **s.t.** $A{\in}\{A\}$, $O/A\text{-}Dep{\in}\{MR\}$, $POS(O){\in}\{JJ\}$	$o = O$	same as R1$_2$ with *iPod* is the known word and *best* as the extract word.
R3$_1$ (AA-Rel)	$A_{i(j)}{\rightarrow}A_{i(j)}\text{-}Dep{\rightarrow}A_{j(i)}$ **s.t.** $A_{j(i)}{\in}\{A\}$, $A_{i(j)}\text{-}Dep{\in}\{CONJ\}$, $POS(A_{i(j)}){\in}\{NN\}$	$a = A_{i(j)}$	*Does the player play dvd with <u>audio</u> and "video"?* *video${\rightarrow}$**conj**${\rightarrow}$audio*
R3$_2$ (AA-Rel)	$A_i{\rightarrow}A_i\text{-}Dep{\rightarrow}H{\leftarrow}A_j\text{-}Dep{\leftarrow}A_j$ **s.t.** $A_i{\in}\{A\}$, $A_i\text{-}Dep{=}A_j\text{-}Dep$ OR ($A_i\text{-}Dep = subj$ AND $A_j\text{-}Dep = obj$), $POS(A_j){\in}\{NN\}$	$a = A_j$	*Canon "G3" has a great <u>len</u>.* *len${\rightarrow}$**obj**${\rightarrow}$has${\leftarrow}$**subj**${\leftarrow}$G3*
R4$_1$ (OO-Rel)	$O_{i(j)}{\rightarrow}O_{i(j)}\text{-}Dep{\rightarrow}O_{j(i)}$ **s.t.** $O_{j(i)}{\in}\{O\}$, $O_{i(j)}\text{-}Dep{\in}\{CONJ\}$, $POS(O_{i(j)}){\in}\{JJ\}$	$o = O_{i(j)}$	*The camera is <u>amazing</u> and "easy" to use.* *easy${\rightarrow}$**conj**${\rightarrow}$amazing*
R4$_2$ (OO-Rel)	$O_i{\rightarrow}O_i\text{-}Dep{\rightarrow}H{\leftarrow}O_j\text{-}Dep{\leftarrow}O_j$ **s.t.** $O_i{\in}\{O\}$, $O_i\text{-}Dep{=}O_j\text{-}Dep$ OR ($O_i/O_j\text{-}Dep {\in}\{pnmod, mod\}$), $POS(O_j){\in}\{JJ\}$	$o = O_j$	*If you want to buy a <u>sexy</u>, "cool", accessory-available mp3 player, you can choose iPod.* *sexy${\rightarrow}$**mod**${\rightarrow}$player${\leftarrow}$**mod**${\leftarrow}$cool*

Table 1. Rules for aspect and opinion word extraction.

Column 1 is the rule ID, column 2 is the observed relation and the constraints that it must satisfy, column 3 is the output, and column 4 is an example. In each example, the underlined word is the known word and the word with double quotes is the extracted word. The corresponding instantiated relation is given right below the example.

    OA-Rels are used for tasks (1) and (3), *AA-Rels* are used for task (2) and *OO-Rels* are used for task (4). Four types of rules are defined respectively for these four subtasks and the details are given in Table 1. In the table, *o* (or *a*) stands for the output (or extracted) opinion word (or aspect). $\{O\}$ (or $\{A\}$) is the set of known opinion words (or the set of aspects) either given or extracted. *H* means any word. *POS(O*(or *A*)) and *O*(or *A*)-*Dep* stand for the POS tag and dependency relation of the word *O* (or *A*) respectively. $\{JJ\}$ and $\{NN\}$ are sets of POS tags of potential opinion words and aspects respectively. $\{JJ\}$ contains *JJ*, *JJR* and *JJS*; $\{NN\}$ contains *NN* and *NNS*. $\{MR\}$ consists of dependency relations describing relations be-

tween opinion words and aspects (*mod*, *pnmod*, *subj*, *s*, *obj*, *obj2* and *desc*). {*CONJ*} contains *conj* only. The arrows mean dependency. For example, $O \rightarrow O\text{-}Dep \rightarrow A$ means O depends on A through a syntactic relation *O-Dep*. Specifically, it employs $R1_i$ to extract aspects (*a*) using opinion words (*O*), $R2_i$ to extract opinion words (*o*) using aspects (*A*), $R3_i$ to extract aspects (*a*) using extracted aspects (*A_i*) and $R4_i$ to extract opinion words (*o*) using known opinion words (*O_i*). Take $R1_1$ as an example. Given the opinion word O, the word with the POS tag *NN* and satisfying the relation *O-Dep* is extracted as an aspect.

The double propagation method works well for medium-sized corpuses, but for large and small corpora, it may result in low precision and low recall. The reason is that the patterns based on *direct dependencies* have a large chance of introducing noises for large corpora and such patterns are limited for small corpora. To overcome the weaknesses, Zhang et al. (2010) proposed an approach to extend double propagation. It consists of two steps: *aspect extraction* and *aspect ranking*. For aspect extraction, it still adopts double propagation to populate aspect candidates. However, some new linguistic patterns (e.g., *part-whole* relation patterns) are introduced to increase recall. After extraction, it ranks aspect candidates by aspect importance. That is, if an aspect candidate is genuine and important, it will be ranked high. For an unimportant aspect or noise, it will be ranked low. It observed that there are two major factors affecting the aspect importance: *aspect relevance* and *aspect frequency*. The former describes how likely an aspect candidate is a genuine aspect. There are three clues to indicate aspect relevance in reviews. The first clue is that an aspect is often modified by multiple opinion words. For example, in the mattress domain, "*delivery*" is modified by "*quick*" "*cumbersome*" and "*timely*". It shows that reviewers put emphasis on the word "*delivery*". Thus, "*delivery*" is a likely aspect. The second clue is that an aspect can be extracted by multiple part-whole patterns. For example, in car domain, if we find following two sentences, "*the engine of the car*" and "*the car has a big engine*", we can infer that "*engine*" is an aspect for car, because both sentences contain part-whole relations to indicate "*engine*" is a part of "*car*". The third clue is that an aspect can be extracted by a combination of opinion word modification relation, part-whole pattern or other linguistic patterns. If an aspect candidate is not only modified by opinion words but also extracted by part-whole pattern, we can infer that it is a genuine aspect with high confidence. For example, for sentence "*there is a bad hole in the mattress*", it strongly indicates that "*hole*" is an aspect for a mattress because it is modified by opinion word "*bad*" and also in the part-whole pattern. What is more, there are mutual enforcement relations between opinion words, linguistic patterns, and aspects. If an adjective modifies many genuine aspects, it is

highly possible to be a good opinion word. Likewise, if an aspect candidate can be extracted by many opinion words and linguistic patterns, it is also highly likely to be a genuine aspect. Thus, Zhang et al. utilized the HITS algorithm (Klernberg, 1999) to measure aspect relevance. Aspect frequency is another important factor affecting aspect ranking. It is desirable to rank those frequent aspects higher than infrequent aspects. The final ranking score for a candidate aspect is the score of aspect relevancy multiplied by the log of aspect frequency.

Liu et al. (2012) also utilized the relation between opinion word and aspect to perform extraction. However, they formulated the opinion relation identification between aspects and opinion words as a word alignment task. They employed the word-based translation model (Brown et al., 1993) to perform monolingual word alignment. Basically, the associations between aspects and opinion words are measured by translation probabilities, which can capture opinion relations between opinion words and aspects more precisely and effectively than linguistic rules or patterns.

Li et al., (2012a) proposed a domain adaption method to extract opinion words and aspects together across domains. In some cases, it has no labeled data in the target domain but a plenty of labeled data in the source domain. The basic idea is to leverage the knowledge extracted from the source domain to help identify aspects and opinion words in the target domain. The approach consists of two main steps: (1) identify some common opinion words as seeds in the target domain (e.g., "*good*", "*bad*"). Then, high-quality opinion aspect seeds for the target domain are generated by mining some general syntactic relation patterns between the opinion words and aspects from the source domain. (2) a bootstrapping method called *Relational Adaptive bootstrapping* is employed to expand the seeds. First, a cross-domain classifier is trained iteratively on labeled data from the source domain and newly labeled data from the target domain, and then used to predict the labels of the target unlabeled data. Second, top predicted aspects and opinion words are selected as candidates based on confidence. Third, with the extracted syntactic patterns in the previous iterations, it constructs a bipartite graph between opinion words and aspects extracted from the target domain. A graph-based score refinement algorithm is performed on the graph, and the top candidates are added into aspect list and opinion words list respectively.

Besides exploiting relations between aspect and opinion words discussed above, Popescu and Etzioni (2005) proposed a method to extract product aspects by utilizing a *discriminator* relation in context, i.e., the relation between aspects and product class. They first extract noun phrases with high frequency from reviews as candidate product aspects. Then they evaluate each candidate by computing a pointwise mutual information

(PMI) score between the candidate and some *meronymy discriminators* associated with the product class. For example, for "*scanner*", the meronymy discriminators for the scanner class are patterns such as "*of scanner*", "*scanner has*", "*scanner comes with*", etc. The PMI measure is calculated by searching the Web. The equation is as follows.

$$PMI(a,d) = \frac{hits(a \wedge d)}{hits(a)hits(d)} \qquad (1)$$

where *a* is a candidate aspect and *d* is a discriminator. Web search is used to find the number of hits of individual terms and also their co-occurrences. The idea of this approach is clear. If the PMI value of a candidate aspect is too low, it may not be a component or aspect of the product because *a* and *d* do not co-occur frequently. The algorithm also distinguishes components/parts from attributes using WordNet[3]'s *is-a* hierarchy (which enumerates different kinds of properties) and morphological cues (e.g., "*-iness*", "*-ity*" suffixes).

Kobayashi et al. (2007) proposed an approach to extract aspect-evaluation (aspect-opinion expression) and aspect-of relations from blogs, which also makes use of association between aspect, opinion expression and product class. For example, in aspect-evaluation pair extraction, evaluation expression is first determined by a dictionary look-up. Then, syntactic patterns are employed to find its corresponding aspect to form the candidate pair. The candidate pairs are tested and validated by a classifier, which is trained by incorporating two kinds of information: contextual and statistical clues in corpus. The contextual clues are syntactic relations between words in a sentence, which can be determined by the dependency grammar, and the statistical clues are normal co-occurrences between aspects and evaluations.

3.1.2   Sequence Models

Sequence models have been widely used in information extraction tasks and can be applied to aspect extraction as well. We can deem aspect extraction as a sequence labeling task, because product aspects, entities and opinion expressions are often interdependent and occur at a sequence in a sentence. In this section, we will introduce two sequence models: Hidden

[3] http://wordnet.princeton,edu

Markov Model (Rabiner, 1989) and Conditional Random Fields (Lafferty et al., 2001).

Hidden Markov Model

Hidden Markov Model (HMM) is a directed sequence model for a wide range of state series data. It has been applied successfully to many sequence labeling problems such as named entity recognition (NER) in information extraction and POS tagging in natural language processing. A generic HMM model is illustrated in Figure 3.
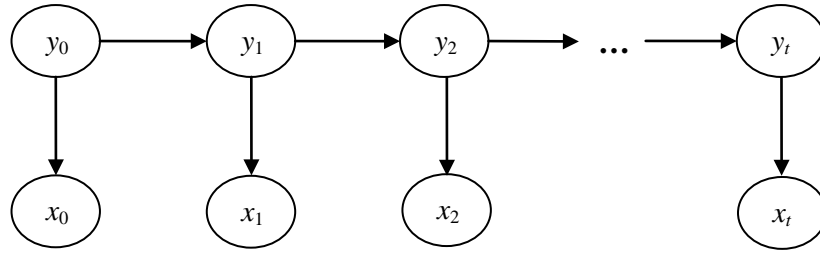

Figure 3.   Hidden Markov model

We have

$$Y \ = \ <y_0, y_1, \ldots y_t> \ = \ \text{hidden state sequence}$$
$$X \ = \ <x_0, x_1, \ldots x_t> \ = \ \text{observation sequence}$$

HMM models a sequence of observations X by assuming that there is a *hidden* sequence of states Y. Observations are dependent on states. Each state has a probability distribution over the possible observations. To model the joint distribution $p(y, x)$ tractably, two independence assumptions are made. First, it assumes that state y_t only depends on its immediate predecessor state y_{t-1}. y_t is independent of all its ancestor $y_1, y_2, y_3, \ldots, y_{t-2}$. This is also called the *Markov* property. Second, the observation x_t only depends on the current state y_t. With these assumptions, we can specify HMM using three probability distributions: $p(y_0)$ over initial state, state transition distribution $p(y_t | y_{t-1})$ and observation distribution $p(x_t | y_t)$. That is, the joint probability of a state sequence Y and an observation sequence X factorizes as follows.

$$p(Y,X) = \prod_{t=1}^{t} p(y_t \mid y_{t-1}) p(x_t \mid y_t) \qquad (2)$$

where we write the initial state distribution $p(y_1)$ as $p(y_1|y_0)$.

Given some observation sequences, we can learn the model parameter of HMM that maximizes the observation probability. That is, the learning of HMM can be done by building a model to best fit the training data. With the learned model, we can find an optimal state sequence for new observation sequences.

In aspect extraction, we can regard words or phrases in a review as observations and aspects or opinion expressions as underlying states. Jin et al. (2009a and 2009b) utilized lexicalized HMM to extract product aspects and opinion expressions from reviews. Different from traditional HMM, they integrate linguistic features such as part-of-speech and lexical patterns into HMM. For example, an observable state for the lexicalized HMM is represented by a pair $(word_i, POS(word_i))$, where $POS(word_i)$ represents the part-of-speech of $word_i$.

Conditional Random Fields

One limitation of HMM is that its assumptions may not be adequate for real-life problems, which leads to reduced performance. To address the limitation, linear-chain Conditional Random fields (CRF) (Lafferty et al., 2001; Sutton and McCallum, 2006) is proposed as an undirected sequence model, which models a conditional probability $p(Y|X)$ over hidden sequence Y given observation sequence X. That is, the conditional model is trained to label an unknown observation sequence X by selecting the hidden sequence Y which maximizes $p(Y|X)$. Thereby, the model allows relaxation of the strong independence assumptions made by HMM. The linear-chain CRF model is illustrated in Figure 4.
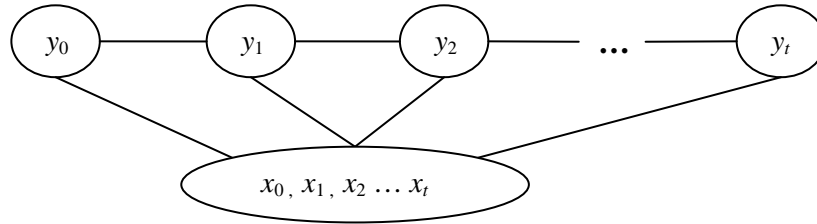

Figure 4.   Linear chain Conditional Random fields

We have

$$Y \; = \; <y_0, y_1, \ldots y_t> \; = \; \text{hidden state sequence}$$
$$X \; = \; <x_0, x_1, \ldots x_t> \; = \; \text{observation sequence}$$

The conditional distribution $p(Y|X)$ takes the form

$$p(Y \mid X) = \frac{1}{Z(X)} \exp\{\sum_{k=1}^{k} \lambda_k f_k(y_t, y_{t-1}, x_t)\} \qquad (3)$$

where $Z(X)$ is a normalization function

$$Z(X) = \sum_{y} \exp\{\sum_{k=1}^{k} \lambda_k f_k(y_t, y_{t-1}, x_t)\} \qquad (4)$$

CRF introduces the concept of *feature function*. Each feature function has the form $f_k(y_t, y_{t-1}, x_t)$ and λ_k is its corresponding weight. Figure 4 indicates that CRF makes independence assumption among Y, but not among X. Note that one argument for feature function f_k is the vector x_t which means each feature function can depend on observation X from any step. That is, all the components of the global observations X are needed in computing feature function f_k at step t. Thus, CRF can introduce more features than HMM at each step.

Jakob and Gurevych (2010) utilzied CRF to extract opinion targets (or aspects) from sentences which contain an opinion expression. They emplyed the following features as input for the CRF-based approach.

Token: This feature represents the string of the current token.
Part of Speech: This feature represents the POS tag of the current token. It can provide some means of lexical disambiguation.
Short Dependency Path: Direct dependency realtions show accurate connections between a target and an opinion expression. Thus, all tokens which have a direct dependency relation to an opinion expression in a sentence are labelled.
Word Distance: Noun phrases are good candidates for opinion targets in product reviews. Thus token(s) in the closest noun phrase regarding word distance to each opinion expression in a sentence are labelled.

Jakob and Gurevych represented the possible labels following the Inside-Outside-Begin (IOB) labelling schema: *B-Target*, identifying the beginning of an opinion target; *I-Target*, identifying the continuation of a target, and *O* for other (non-target) tokens.

Similar work has been done in (Li et al., 2010a). In order to model the long distance dependency with conjunctions (e.g., "*and*", "*or*", "*but*") at

the sentence level and deep syntactic dependencies for aspects, positive opinions and negative opinions, they used the skip-tree CRF models to detect product aspects and opinions.

3.1.3 Topic Models

Topic models are widely applied in natural language processing and text mining. They are based on the idea that documents are mixtures of topics, and each topic is a probability distribution of words. A topic model is a *generative* model for documents. Generally, it specifies a probabilistic procedure by which documents can be generated. Assuming constructing a new document, one chooses a distribution D_i over topics. Then, for each word in that document, one chooses a topic randomly according to D_i and draws a word from the topic. Standard statistical techniques can be used to invert the procedure and infer the set of topics that were responsible for generating a collection of documents. Naturally, topic models can be applied to aspect extraction. We can deem that each aspect is a unigram language model, i.e., a multinomial distribution over words. Although such a representation is not as easy to interpret as aspects, its advantage is that different words expressing the same or related aspects (more precisely aspect expressions) can be automatically grouped together under the same aspect. Currently, a great deal of research has been done on aspect extraction using topic models. They basically adapted and extended the Probabilistic Latent Semantic Analysis (pLSA) model (Hofmann, 2001) and the Latent Dirichlet Allocation (LDA) model (Blei et al., 2003).

Probabilistic Latent Semantic Analysis

pLSA is also known as Probabilistic Latent Semantic Indexing (PLSI). It is proposed in (Hofmann, 2001), which uses a generative latent class model to perform a probabilistic mixture decomposition.

Figure 5(a) illustrate graphical model of pLSA. In the figure, d represents a document, z_i represents a latent topic (assuming K topics overall), and w_j represents a word, which are modeled by the parameters ρ, θ, φ respectively, where ρ is the probability of choosing document d, θ is the distribution $p(z_i|d)$ of topics in document d and φ is the distribution $p(w_j|z_i)$ of the word w_j in latent topic z_i. The ρ and φ are observable variables and the topic variable θ is a latent variable.
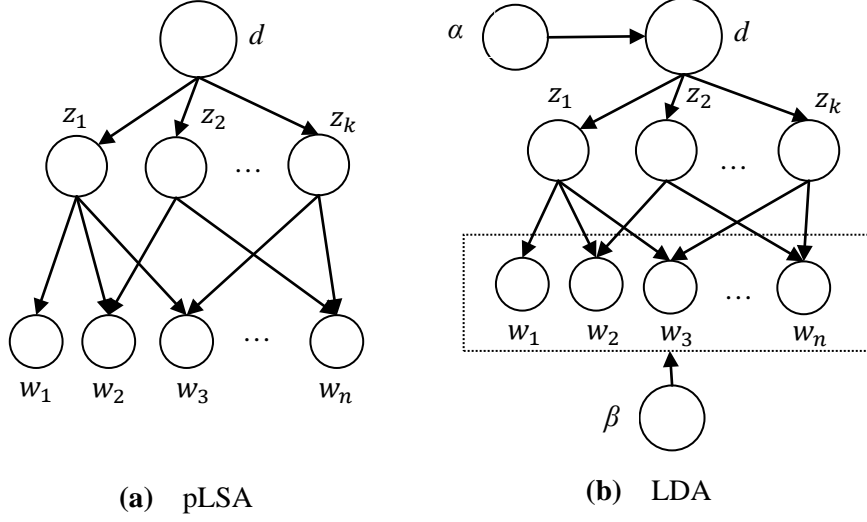
(a) pLSA   **(b)** LDA

Figure 5. PLSA and LDA topic models

The generation of a word by pLSA is defined as follows.

    (1) choose document $d \sim \rho$

    (2) choose topic $z_i \sim \theta$

    (3) choose word $w_j \sim \varphi$

The probability of observed word w_j in a document d is then defined by the mixture of equation (5):

$$p(w_j \mid d) = \sum_{z=1}^{k} \theta(z)\varphi(w_j) \qquad (5)$$

The joint probability of observing all words in document d is as follows:

$$p(d, w_j) = \rho(d)\prod_{j=1}^{n} p(w_j \mid d)^{c_j} \qquad (6)$$

where c_j is the count of word w_j occur in document d.

And the joint probability of observing the document collection is given by the following equation (assuming m documents overall).

$$p(D) = \prod_{i=1}^{m} p(d_i) \qquad (7)$$

Obviously, the main parameters of the model are θ and φ. They can be estimated by Expectation Maximization (EM) algorithm (Dempster et al., 1977), which is used to calculate maximum likelihood estimates of the parameters.

For aspect extraction task, we can regard product aspects as latent topics in opinion documents. Lu et al. (2009) proposed a method for aspect discovery and grouping in short comments. They assume that each review can be parsed into opinion phrases of the format $<$ *head term*, *modifier* $>$ and incorporate such structure of phrases into the pLSA model, using the co-occurrence information of head terms and their modifiers. Generally, the head term is an aspect, and the modifier is opinion word, which expresses some opinion towards the aspect. The proposed approach defines k unigram language models: $\Theta = \{\theta_1, \theta_2, \ldots, \theta_k\}$ as k topic models, each is a multinomial distribution of head terms, capturing one aspect. Note that each modifier could be represented by a set of head terms that it modifies as the following equations:

$$d(w_m) = \{w_h \mid (w_m, w_h) \in T\} \qquad (8)$$

where w_h is the head term and w_m is the modifier.

Actually, a modifier can be regarded as a sample of the following mixture model.

$$p_{d(w_m)}(w_h) = \sum_{j=1}^{k} [\pi_{d(w_m),j} p(w_h \mid \theta_j)] \qquad (9)$$

where $\pi_{d(w_m),j}$ is a modifier-specific mixing weight for the j-th aspect, which sums to one. The log-likelihood of the collection of modifiers V_m is

$$\log p(V_m \mid \Delta) =$$

$$\sum_{w_m \in v_m} \sum_{w_h \in v_h} \{c(w_h, d(w_m)) \times \log \sum_{j=1}^{k} [\pi_{d(w_m),j} p(w_h \mid \theta_j)]\} \qquad (10)$$

where $c(w_h, d(w_m))$ is the number of co-occurrences of head term w_h with modifiers w_m, and Δ is the set of all model parameters.

Using the EM algorithm, k topic models can be estimated and aspect expressions can be grouped. In addition, Lu et al. use conjugate prior to in-

corporate human knowledge to guide the clustering of aspects. Since the proposed method models the co-occurrence of head terms at the level of the modifiers they use, it can use more meaningful syntactic relations.

Moghaddam and Ester (2011) extended the above pLSA model by incorporating latent rating information for reviews into the model to extract aspects and their corresponding ratings.

However, the main drawback of the pLSA method is that it is inherently transductive, i.e., there is no direct way to apply the learned model to new documents. In pLSA, each document d in the collection is represented as a mixture coefficients θ, but it does not define such representation for documents outside the collection.

Latent Dirichlet Allocation (LDA)

To address the limitation of pLSA, the Bayesian LDA model is proposed in (Blei et al., 2003). It extends pLSA by adding priors to the parameters θ and φ. In LDA, a prior Dirichlet distribution *Dir* (α) is added for θ and a prior Dirichlet distribution *Dir* (β) is added for φ. The generation of a document collection is started by sampling a word distribution φ from *Dir* (β) for each latent topic. Then each document d in LDA is assumed to be generated as follows.

(1) choose distribution of topics $\theta \sim Dir(\alpha)$

(2) choose distribution of words $\varphi \sim Dir(\beta)$

(3) for each word w_j in document d
- choose topic $z_i \sim \theta$
- choose word $w_j \sim \varphi$

The model is represented in Figure 5 (b). LDA has only two parameters: α and β, which prevent it from overfitting. Exact inference in such a model is intractable and various approximations have been considered, such as the variational EM method and the Markov Chain Monte Carlo (MCMC) algorithm (Gilks et al.,1996). Note that, compared with pLSA, LDA has a stronger generative power, as it describes how to generate topic distribution θ for an unseen document d.

LDA based topic models have been used for aspect extraction by several researchers. Titov and McDonald (2008a) pointed that global topic models such as pLSA and LDA might not be suitable for detecting aspects. Both pLSA and LDA use the bag-of-words representation of documents, which depends on topic distribution differences and word co-occurrence among documents to identify topics and word probability distribution in

each topic. However, for opinion documents such as reviews about a particular type of products, they are quite homogenous. That is, every document talks about the same aspects, which makes global topic models ineffective and are only effective for discovering entities (e.g., brands or product names). In order to tackle this problem, they proposed Multi-grain LDA (MG-LDA) to discover aspects, which models two distinct types of topics: global topics and local topics. As in pLSA and LDA, the distribution of global topics is fixed for a document (review). However, the distribution of local topics is allowed to vary across documents. A word in a document is sampled either from the mixture of global topics or from the mixture of local topics specific for the local context of the word. It is assumed that aspects will be captured by local topics and global topics will capture properties of reviewed items. For example, a review of a London hotel: "… *public transport in London is straightforward, the tube station is about an 8 minute walk … or you can get a bus for £1.50*". The review can be regarded as a mixture of global topic *London* (words: "*London*", "*tube*", "*£*") and the local topic (aspect) *location* (words: "*transport*", "*walk*", "*bus*").

MG-LDA can distinguish local topics. But due to the many-to-one mapping between local topics and ratable aspects, the correspondence is not explicit. It lacks direct assignment from topics to aspects. To resolve the issue, Titov and McDonald (2008b) extended the MG-LDA model and constructed a joint model of text and aspect ratings, which is called the Multi-Aspect Sentiment model (MAS). It consists of two parts. The first part is based on MG-LDA to build topics what are representative of ratable aspects. The second part is a set of classifiers (sentiment predictors) for each aspect, which attempt to infer the mapping between local topics and aspects with the help of aspect-specific ratings provided along with the review text. Their goal is to use the rating information to identity more coherent aspects.

The idea of LDA has also been applied and extended in (Branavan et al., 2008; Lin and He, 2009; Brody and Elhadad, 2010; Zhao et al., 2010; Wang et al., 2010; Jo and Oh, 2011; Sauper et al., 2011; Moghaddam and Ester, 2011; Mukajeee and Liu, 2012). Branavan used the aspect descriptions as keyphrases in Pros and Cons of review *Format 1* to help finding aspects in the detailed review text. Keyphrases are clustered based on their distributional and orthographic properties, and a hidden topic model is applied to the review text. Then, a final graphical model integrates both of them. Lin and He (2009) proposed a joint topic-sentiment model (JST), which extends LDA by adding a sentiment layer. It can detect aspect and sentiment simultaneously from text. Brody and Elhadad (2010) proposed to identify aspects using a local version of LDA, which operates on sen-

tences, rather than documents and employs a small number of topics that correspond directly to aspects. Zhao et al. (2010) proposed a MaxEnt-LDA hybrid model to jointly discover both aspect words and aspect-specific opinion words, which can leverage syntactic features to help separate aspects and opinion words. Wang et al. (2010) proposed a regression model to infer both aspect ratings and aspect weights at the level of individual reviews based on learned latent aspects. Jo and Oh (2011) proposed an *Aspect and Sentiment Unification* Model (ASUM) to model sentiments toward different aspects. Sauper et al. (2011) proposed a joint model, which worked only on short snippets already extracted from reviews. It combined topic modeling with a HMM, where the HMM models the sequence of words with types (aspect, opinion word, or background word). Moghaddam and Ester (2011) proposed a model called ILDA, which is based on LDA and jointly models latent aspects and rating. ILDA can be viewed as a generative process that first generates an aspect and subsequently generates its rating. In particular, for generating each opinion phrase, ILDA first generates an aspect a_m from an LDA model. Then it generates a rating r_m conditioned on the sampled aspect a_m. Finally, a head term t_m and a sentiment s_m are drawn conditioned on a_m and r_m, respectively. Mukajeee and Liu (2012) proposed two models (SAS and ME-SAS) to jointly model both aspects and aspect specific sentiments by using seeds to discover aspects in an opinion corpus. The seeds reflect the user needs to discover specific aspects.

Other closely related work with topic model is the topic-sentiment model (TSM). Mei et al. (2007) proposed it to perform joint topic and sentiment modeling for blogs, which uses a positive sentiment model and a negative sentiment model in additional to aspect models. They do sentiment analysis on documents level and not on aspect level. In (Su et al., 2008), the authors also proposed a clustering based method with mutual reinforcement to identify aspects. Similar work has been done in (Scaffidi et al., 2007), they proposed a language model approach for product aspect extraction with the assumption that product aspects are mentioned more often in a product review than they are mentioned in general English text. However, statistics may not be reliable when the corpus is small.

In summary, topic modeling is a powerful and flexible modeling tool. It is also very nice conceptually and mathematically. However, it is only able to find some general/rough aspects, and has difficulty in finding fine-grained or precise aspects. We think it is too statistics centric and come with its limitations. It could be fruitful if we can shift more toward natural language and knowledge centric for a more balanced approach.

3.1.4   Miscellaneous Methods

Yi et al. (2003) proposed a method for aspect extraction based on the like-lihood-ratio test. Bloom et al. (2007) manually built a taxonomy for aspects, which indicates aspect type. They also constructed an aspect list by starting with a sample of reviews that the list would apply to. They examined the seed list manually and used WordNet to suggest additional terms to add to the list. Lu et al. (2010) exploited the online ontology Freebase[4] to obtain aspects to a topic and used them to organize scattered opinions to generate structured opinion summaries. Ma and Wan (2010) exploited Centering theory (Grosz et al., 1995) to extract opinion targets from news comments. The approach uses global information in news articles as well as contextual information in adjacent sentences of comments. Ghani et al. (2006) formulated aspect extraction as a classification problem and used both traditional supervised learning and semi-supervised learning methods to extract product aspects. Yu et al. (2011) used a partially supervised learning method called one-class SVM to extract aspects. Using one-class SVM, one only needs to label some positive examples, which are aspects. In their case, they only extracted aspects from Pros and Cons of the reviews. Li et al. (2012b) formulated aspect extraction as a shallow semantic parsing problem. A parse tree is built for each sentence and structured syntactic information within the tree is used to identify aspects.

3.2 Aspect Grouping and Hierarchy

It is common that people use different words and expressions to describe the same aspect. For example, *photo* and *picture* refer to the same aspect in digital camera reviews. Although topic models (discussed in Section 3.1.3) can identify and group aspects to some extent, the results are not fine-grained because such models are based on word co-occurrences rather than word semantic meanings. As a result, a topic is often a list of related words about a general topic rather than a set of words referring to the same aspect. For example, a topic about *battery* may contain words like *life*, *battery*, *charger*, *long*, and *short*. We can clearly see that these words do not mean the same thing, although they may co-occur frequently. Alternative-

---

[4] http://www.freebase.com

ly, we can extract aspect expressions first and then group them into different aspect categories.

Grouping aspect expressions indicating the same aspect are essential for opinion applications. Although WordNet and other thesaurus dictionaries can help, they are far from sufficient due to the fact that many synonyms are domain dependent. For example, *picture* and *movie* are synonyms in movie reviews, but they are not synonyms in digital camera reviews as *picture* is more related to *photo* while *movie* refers to *video*. It is also important to note that although most aspect expressions of an aspect are domain synonyms, they are not always synonyms. For example, "*expensive*" and "*cheap*" can both indicate the aspect *price* but they are not synonyms of *price*.

Liu, Hu and Cheng (2005) attempted to solve the problem by using the WordNet synonym sets, but the results were not satisfactory because WordNet is not sufficient for dealing with domain dependent synonyms. Carenini et al. (2005) also proposed a method to solve this problem in the context of opinion mining. Their method is based on several similarity metrics defined using string similarity, synonyms and distances measured using WordNet. However, it requires a taxonomy of aspects to be given beforehand for a particular domain. The algorithm merges each discovered aspect expression to an aspect node in the taxonomy.

Guo et al. (2009) proposed a multilevel latent semantic association technique (called mLSA) to group product aspect expressions. At the first level, all the words in product aspect expressions are grouped into a set of concepts/topics using LDA. The results are used to build some latent topic structures for product aspect expressions. At the second level, aspect expressions are grouped by LDA again according to their latent topic structures produced from level 1 and context snippets in reviews.

Zhai et al. (2010) proposed a semi-supervised learning method to group aspect expressions into the user specified aspect groups or categories. Each group represents a specific aspect. To reflect the user needs, they first manually label a small number of seeds for each group. The system then assigns the rest of the discovered aspect expressions to suitable groups using semi-supervised learning based on labeled seeds and unlabeled examples. The method used the Expectation-Maximization (EM) algorithm. Two pieces of prior knowledge were used to provide a better initialization for EM, i.e., (1) aspect expressions sharing some common words are likely to belong to the same group, and (2) aspect expressions that are synonyms in a dictionary are likely to belong to the same group. Zhai et al. (2011) further proposed an unsupervised method, which does not need any pre-labeled examples. Besides, it is further enhanced by lexical (or WordNet) similarity. The algorithm also exploited a piece of natu-

ral language knowledge to extract more discriminative distributional context to help grouping.

Mauge et al. (2012) used a maximum entropy based clustering algorithm to group aspects in a product category. It first trains a maximum-entropy classifier to determine the probability p that two aspects are synonyms. Then, an undirected weighted graph is constructed. Each vertex represents an aspect. Each edge weight is proportional to the probability p between two vertices. Finally, approximate graph partitioning methods are employed to group product aspects.

Closely related to aspect grouping, aspect hierarchy is to present product aspects as a tree or hierarchy. The root of the tree is the name of the entity. Each non-root node is a component or sub-component of the entity. Each link is a *part-of* relation. Each node is associated with a set of product aspects. Yu et al. (2011b) proposed a method to create aspect hierarchy. The method starts from an initial hierarchy and inserts the aspects into it one-by-one until all the aspects are allocated. Each aspect is inserted to the optimal position by semantic distance learning. Wei and Gulla (2010) studied the sentiment analysis based on aspect hierarchy trees.

3.3 Aspect Ranking

A product may have hundreds of aspects. Sometimes, we need to identify important one from reviews, which are more influential for people's decision making. Zhang et al. (2010) proposed a method to rank product aspects. They rank candidate aspects based on aspect importance which consists of two factors: aspect relevancy and aspect frequency. Aspect relevance indicates the aspect's correctness and aspect frequency is the occurrence frequency of an aspect in reviews. As discussed in Section 3.1.1, Zhang et al. modeled mutual enforcement relation between aspects and aspect indictors (e.g., opinion words and relation patterns) in a bipartite graph utilizing Web page ranking algorithm HITS. Aspects only have authority scores and aspect indicators only have hub scores. If an aspect candidate has a high authority score, it is considered as a highly relevant aspect. Likewise, if an aspect indicator has a high hub score, it is considered as a good aspect indicator. The final ranking score of a candidate aspect is the multiplication of the aspect relevancy score (authority score) and logarithm of aspect frequency.

Yu et al. (2011a) showed the important aspects are identified according to two observations: the important aspects of a product are usually commented by a large number of consumers and consumers' opinions on the

important aspects greatly influence their overall ratings on the product. Given reviews of a product, they first identify product aspects by a shallow dependency parser and determine opinions on these aspects via a sentiment classifier. They then develop an aspect ranking algorithm to identify the important aspects by considering the aspect frequency and the influence of opinions given to each aspect on their overall opinions.

Liu et al. (2012) proposed a graph-based algorithm to compute the confidence of each opinion target and its ranking. They argued that the ranking of a candidate is determined by two factors: *opinion relevancy* and *candidate importance*. To model these two factors, a bipartite graph (similar to that in Zhang et al., 2010) is constructed. An iterative algorithm based on the graph is proposed to compute candidate confidences. Then the candidates with high confidence scores are extracted as opinion targets. Similar work has also been reported in (Li et al., 2012a).

3.4 Mapping Implicit Aspect Expressions

There are many types of implicit aspect expressions. Adjectives are perhaps the most common type. Many adjectives modify or describe some specific attributes or properties of entities. For example, the adjective "*heavy*" usually describes the aspect *weight* of an entity. "*Beautiful*" is normally used to describe (positively) the aspect *look* or *appearance* of an entity. By no means, however, does this say that these adjectives only describe such aspects. Their exact meanings can be domain dependent. For example, "*heavy*" in the sentence "*the traffic is heavy*" does not describe the *weight* of the traffic. Note that some implicit aspect expressions are very difficult to extract and to map, e.g., "*fit in pockets*" in the sentence "*This phone will not easily fit in pockets*".

Limited research has been done on mapping implicit aspects to their explicit aspects. In Su et al. (2008), a clustering method was proposed to map implicit aspect expressions, which were assumed to be sentiment words, to their corresponding explicit aspects. The method exploits the mutual reinforcement relationship between an explicit aspect and a sentiment word forming a co-occurring pair in a sentence. Such a pair may indicate that the sentiment word describes the aspect, or the aspect is associated with the sentiment word. The algorithm finds the mapping by iteratively clustering the set of explicit aspects and the set of sentiment words separately. In each iteration, before clustering one set, the clustering results of the other set is used to update the pairwise similarity of the set. The pairwise similarity in a set is determined by a linear combination of

intra-set similarity and inter-set similarity. The intra-set similarity of two items is the traditional similarity. The inter-set similarity of two items is computed based on the degree of association between aspects and sentiment words. The association (or mutual reinforcement relationship) is modeled using a bipartite graph. An aspect and an opinion word are linked if they have co-occurred in a sentence. The links are also weighted based on the co-occurrence frequency. After the iterative clustering, the strong links between aspects and sentiment word groups form the mapping.

In Hai et al. (2011), a two-phase co-occurrence association rule mining approach was proposed to match implicit aspects (which are also assumed to be sentiment words) with explicit aspects. In the first phase, the approach generates association rules involving each sentiment word as the condition and an explicit aspect as the consequence, which co-occur frequently in sentences of a corpus. In the second phase, it clusters the rule consequents (explicit aspects) to generate more robust rules for each sentiment word mentioned above. For application or testing, given a sentiment word with no explicit aspect, it finds the best rule cluster and then assigns the representative word of the cluster as the final identified aspect.

Fei et al. (2012) focused on finding implicit aspects (mainly nouns) indicated by opinion adjectives, e.g., to identify *price*, *cost*, etc., for adjective *expensive*. A dictionary-based method was proposed, which tries to identify attribute nouns from the dictionary gloss of the adjective. They formulated the problem as a collective classification problem, which can exploit lexical relations of words (e.g., synonyms, antonyms, hyponym and hypernym) for classification.

Some other related work for implicit aspect mapping includes those in (Wang and Wang, 2008; Yu et al., 2011b).

3.5 Identifying Aspects that Imply Opinions

Zhang and Liu (2011a) found that in some domains nouns and noun phrases that indicate product aspects may also imply opinions. In many such cases, these nouns are not subjective but objective. Their involved sentences are also objective sentences but imply positive or negative opinions. For example, the sentence in a mattress review "*Within a month, a valley formed in the middle of the mattress.*" Here "*valley*" indicates the quality of the mattress (a product aspect) and also implies a negative opinion. Identifying such aspects and their polarities is very challenging but critical for effective opinion mining in these domains.

Zhang and Liu observed that for a product aspect with an implied opinion, there is either no adjective opinion word that modifies it directly or the opinion words that modify it have the same opinion orientation.

Observation: No opinion adjective word modifies the opinionated product aspect ("*valley*"):

"*Within a month, a valley formed in the middle of the mattress.*"

Observation: An opinion adjective modifies the opinionated product aspect:

"*Within a month, a bad valley formed in the middle of the mattress.*"

Here, the adjective "*bad*" modifies "*valley*". It is unlikely that a positive opinion word will also modify "*valley*" in another sentence, e.g., "*good valley*" in this context. Thus, if a product aspect is modified by both positive and negative opinion adjectives, it is unlikely to be an opinionated product aspect.

Based on these observations, they designed the following two steps to identify noun product aspects which imply positive or negative opinions:

Step 1: *Candidate Identification*: This step determines the surrounding sentiment context of each noun aspect. The intuition is that if an aspect occurs in negative (respectively positive) opinion contexts significantly more frequently than in positive (or negative) opinion contexts, we can infer that its polarity is negative (or positive). A statistical test (test for population proportion) is used to test the significance. This step thus produces a list of candidate aspects with positive opinions and a list of candidate aspects with negative opinions.

Step 2: *Pruning*: This step prunes the two lists. The idea is that when a noun product aspect is directly modified by both positive and negative opinion words, it is unlikely to be an opinionated product aspect.

3.6 Identifying Resource Noun

Liu (2010) point out that there are some types of words or phrases that do not bear sentiments on their own, but when they appear in some particular contexts, they imply positive or negative opinions. All these expressions have to be extracted and associated problems solved before sentiment analysis can achieve the next level of accuracy.

1. Positive ← consume no or little resource
2. | consume less resource
3. Negative ← consume a large quantity of resource
4. | consume more resource

Figure 6. Sentiment polarity of statements involving resources

One such type of expressions involves resources, which occur frequently in many application domains. For example, *money* is a resource in probably every domain ("*this phone costs a lot of money*"), *gas* is a resource in the car domain, and *ink* is a resource in the printer domain. If a device consumes a large quantity of resource, it is undesirable (negative). If a device consumes little resource, it is desirable (positive). For example, the sentences, "*This laptop needs a lot of battery power*" and "*This car eats a lot of gas*" imply negative sentiments on the laptop and the car. Here, "*gas*" and "*battery power*" are resources, and we call these words *resource terms* (which cover both *words* and *phrases*). They are a kind of special product aspects.

In terms of sentiments involving resources, the rules in Figure 6 are applicable (Liu, 2010). Rules 1 and 3 represent normal sentences that involve resources and imply sentiments, while rules 2 and 4 represent comparative sentences that involve resources and also imply sentiments, e.g., "*this washer uses much less water than my old GE washer*".

Zhang and Liu (2011a) formulated the problem based on a bipartite graph and proposed an iterative algorithm to solve the problem. The algorithm was based on the following observation:

Observation: The sentiment or opinion expressed in a sentence about resource usage is often determined by the flowing triple,

(verb, quantifier, noun_term),

where *noun_term* is a noun or a noun phrase representing a resource.

The proposed method used such triples to help identify resources in a domain corpus. The model used a circular definition to reflect a special reinforcement relationship between *resource usage verbs* (e.g., *consume*) and resource terms (e.g., *water*) based on the bipartite graph. The quantifier was not used in computation but was employed to identify candidate verbs and resource terms. The algorithm assumes that a list of quantifiers is given, which is not numerous and can be manually compiled. Based on the circular definition, the problem is solved using an iterative algorithm similar to the HITS algorithm in (Kleinberg, 1999). To start the iterative computation, some global *seed resources* are employed to find and to

score some strong resource usage verbs. These scores are then applied as the initialization for the iterative computation for any application domain. When the algorithm converges, a ranked list of candidate resource terms is identified.

4 Entity Extraction

The task of entity extraction belongs to the traditional named entity recognition (NER) problem, which has been studied extensively. Many supervised information extraction approaches (e.g., HMM and CRF) can be adopted directly (Putthividhya and Hu, 2011). However, opinion mining also presents some special problems. One of them is the following: in a typical opinion mining application, the user wants to find opinions about some competing entities, e.g., competing products or brands (e.g., Canon, Sony, Samsung and many more). However, the user often can only provide a few names because there are so many different brands and models. Web users also write the names of the same product in various ways in forums and blogs. It is thus important for a system to automatically discover them from relevant corpora. The key requirement of this discovery is that the discovered entities must be relevant, i.e., they must be of the same class/type as the user provided entities, e.g., same brands or models.

Essentially, this is a *PU learning problem* (Positive and Unlabeled Learning), which is also called *learning from positive and unlabeled examples* (Liu et al., 2002). Formally, the problem is stated as follows: given a set of examples P of a particular class, called the *positive class*, and a set of unlabeled examples U, we wish to determine which of the unlabeled examples in U belong to the positive class represented by P. This gives us a two-class classification problem. Many algorithms are available in the literature for solving this problem (see the references in (Liu, 2006-2011).

A specialization of the PU learning problem for named entity extraction is called the *set expansion problem* (Ghahramani and Heller, 2005). The problem is stated similarly: Given a set Q of seed entities of a particular class C, and a set D of candidate entities, we wish to determine which of the entities in D belong to C. That is, we "*grow*" the class C based on the set of seed examples Q. As a specialization of PU learning, this is also a two-class classification problem which needs a binary decision for each entity in D (belonging to C or not belonging to C). However, in practice, the problem may be solved as a ranking problem, i.e., to rank the entities in D based on their likelihoods of belonging to C. In our scenario, the user-

given entities are the set of initial seeds. The opinion mining system needs to expand the set using a text corpus.

4.1 Extraction Methods

The classic methods for solving set expansion problem are based on distributional similarity (Lee, 1999; Pantel et al., 2009). This approach works by comparing the similarity of the word distribution of the surrounding words of a candidate entity and the seed entities, and then ranking the candidate entities based on their similarity values. However, Li et al. (2010b) pointed out that this approach is inaccurate. In this section, we will discuss two machine learning approaches: *Positive and Unlabeled Learning* (PU Learning) and *Bayesian Sets*, which show better results than traditional methods.

4.1.1 PU Learning

In machine learning, there is a class of semi-supervised learning algorithms that learns from *positive* and *unlabeled* examples (PU learning). Its key characteristic (Liu et al., 2002) is that there is no negative training example available for learning. As stated above, PU learning is a two-class classification model. Its objective is to build a classifier using P and U to classifying the data in U or future test cases. The results can be either binary decisions (whether each test case belongs to the positive class or not), or a ranking based on how likely each test case belongs to the positive class represented by P. Clearly, the set expansion problem is a special case of PU learning, where the set Q is P here and the set D is U here.

There are several PU learning algorithms (Liu et al., 2002; Li and Liu, 2003; Li et al., 2007; Yu et al., 2002). Li et al. (2010b) used the S-EM algorithm proposed in (Liu et al., 2002) for entity extraction in opinion documents. The main idea of S-EM is to use a *spy* technique to identify some *reliable negatives* (*RN*) from the unlabeled set U, and then use an EM algorithm to learn from P, RN and U–RN. To apply S-EM algorithm, Li et al. (2010b) takes following basic steps.

Generating candidate entities: It selects single words or phrases as candidate entities based on their part-of-speech (POS) tags. In particular, it chooses the following POS tags as entity indicators — NNP (proper noun), NNPS (plural proper noun), and CD (cardinal number).

Generating positive and unlabeled sets: For each seed, each occurrence in the corpus forms a vector as a positive example in *P*. The vector is formed based on the surrounding word context of the seed mention. Similarly, for each candidate $d \in D$ (*D* denotes the set of all candidates), each occurrence also forms a vector as an unlabeled example in *U*. Thus, each unique seed or candidate entity may produce multiple feature vectors, depending on the number of times that the seed appears in the corpus. The components in the feature vectors are term frequencies.

Ranking entity candidates: With positive and unlabeled data, S-EM applied. At convergence, S-EM produces a Bayesian classifier *C*, which is used to classify each vector $u \in U$ and to assign a probability $p(+|u)$ to indicate the likelihood that *u* belongs to the positive class. Note that each unique candidate entity may generate multiple feature vectors, depending on the number of times that the candidate entity occurs in the corpus. As such, the rankings produced by S-EM are not the rankings of the entities, but rather the rankings of the entities' occurrences. Since different vectors representing the same candidate entity can have very different probabilities, Li et al. (2010b) compute a single score for each unique candidate entity for ranking based on Equation (11).

Let the probabilities (or scores) of a candidate entity $d \in D$ be $V_d = \{v_1, v_2 \ldots, v_n\}$ obtained from the feature vectors representing the entity. Let M_d be the median of V_d. The final score *f* for *d* is defined as following:

$$f(d) = M_d \times \log(1 + n) \qquad (11)$$

The use of the median of V_d can be justified based on the statistical *skewness* (Neter et al, 1993). Note that here *n* is the frequency count of candidate entity *d* in the corpus. The constant 1 is added to smooth the value. The idea is to push the frequent candidate entities up by multiplying the logarithm of frequency. *log* is taken in order to reduce the effect of big frequency counts.

The final score *f(d)* indicates candidate *d*'s overall likelihood to be a relevant entity. A high *f(d)* implies a high likelihood that *d* is in the expanded entity set. The top-ranked candidates are most likely to be relevant entities to the user-provided seeds.

4.1.2 Bayesian Sets

Bayesian Sets is also a semi-supervised learning method, more specifically, a PU learning method, which is based on Bayesian inference and only per-

forms ranking. Let D be a collection of items and Q be a user-given seed set of items, which is a (small) subset of D (i.e., $Q \subseteq D$). The task of Bayesian Sets is to use a model-based probabilistic criterion to give a score to each item e in D ($e \in D$) to gauge how well e fits into Q. In other words, it measures how likely e belongs to the *hidden* class represented/implied by Q. Each item e is represented with a binary feature vector.

The Bayesian criterion score for item e is expressed as follows:

$$score(e) = \frac{p(e \mid Q)}{p(e)} \qquad (12)$$

$p(e \mid Q)$ represents how probable that e belongs to the same class as Q given the examples in Q. $p(e)$ is the prior probability of item e. Using Bayes rule, the equation can be re-written as:

$$score(e) = \frac{p(e, Q)}{p(e)\, p(Q)} \qquad (13)$$

Equation (13) can be interpreted as the ratio of the joint probability of observing e and Q, to the probability of independently observing e and Q. The ratio basically compares the probability that e and Q are generated by the same model with parameters θ, and the probability that e and Q are generated by different models with different parameters θ and $\tilde{\theta}$. Equation (13) says that if the probability that e and Q are generated from the same model with the parameters θ is high, the score of e will be high. On the other hand, if the probability that e and Q come from different models with different parameters θ and $\tilde{\theta}$ is high, the score will be low.

In pseudo code, the Bayesian Sets algorithm is given in Figure 7.

Algorithm: BayesianSets(Q, D)
Input: A small seed set Q of entities
       A set of candidate entities D (= $\{e_1, e_2, e_3 \dots e_n\}$)
Output: A ranked list of entities in D

1. **for** each entity e_i in D
2.   compute: $score(e_i) = \dfrac{p(e_i, Q)}{P(e_i)\, p(Q)}$
3. **end for**
4. Rank the items in D based on their scores;

Figure 7. The Bayesian Sets learning algorithm

If we assume that $q_k \in Q$ is independently and identically distributed (i.i.d.) and Q and e_i come from the same model with the same parameters θ, each of the three terms in Equation (13) are marginal likelihoods and can be written as integrals of the following forms:

$$p(Q) = \int [\prod_{q_k \in Q} p(q_k \mid \theta)] p(\theta) d\theta \qquad (14)$$

$$p(e_i) = \int p(e_i \mid \theta) p(\theta) d\theta \qquad (15)$$

$$p(e_i, Q) = \int [\prod_{q_k \in Q} p(q_k \mid \theta)] p(e_i \mid \theta) p(\theta) d\theta \qquad (16)$$

Let us first compute the integrals of Equation (14). Each seed entity $q_k \in Q$ is represented as a binary feature vector (q_{k1}, q_{k2}, ... q_{kj}). We assume each element of the feature vector has an independent Bernoulli distribution:

$$p(q_k \mid \theta) = \prod_{j=1}^{J} \theta_j^{q_{kj}} (1 - \theta_j)^{1 - q_{kj}} \qquad (17)$$

The conjugate prior for the parameters of a Bernoulli distribution is the Beta distribution:

$$p(\theta \mid \alpha, \beta) = \prod_{j=1}^{J} \frac{\Gamma(\alpha_j + \beta_j)}{\Gamma(a_j)\Gamma(\beta_j)} \theta_j^{\alpha_j - 1} (1 - \theta_j)^{\beta_j - 1} \qquad (18)$$

Where α and β are hyperparameters (which are also vectors). We set α and β empirically from the data, $\alpha_j = km_j$, $\beta_j = k(1 - m_j)$, where m_j is the mean value of j-th components of all possible entities, and k is a scaling factor. The Gamma function is a generalization of the factorial function. For $Q = \{q_1, q_2, ..., q_n\}$, Equation (14) can be represented as follows:

$$p(Q \mid \alpha, \beta) = \prod_{j} \frac{\Gamma(a_j + \beta_j)}{\Gamma(\alpha_j)\Gamma(\beta_j)} \frac{\Gamma(\tilde{\alpha}_j)\Gamma(\tilde{\beta}_j)}{\Gamma(\tilde{\alpha}_j + \tilde{\beta}_j)} \qquad (19)$$

where $\tilde{\alpha}_j = \alpha_j + \sum_{k=1}^{N} q_{kj}$ and $\tilde{\beta}_j = \beta_j + N - \sum_{k=1}^{N} q_{kj}$ With the same idea, we can compute Equation (15) and Equation (16).

Overall, the score of e_i, which is also represented a feature vector, (e_{i1}, e_{i2}, ... e_{ij}) in the data, is computed with:

$$score(e_i) = \prod_j \frac{\alpha_j + \beta_j}{a_j + \beta_j + N} (\frac{\tilde{\alpha}_j}{\alpha_j}) e_{ij} (\frac{\tilde{\beta}_j}{\beta_j})^{1-e_{ij}} \qquad (20)$$

The log of the score is linear in e_i:

$$\log score(e_i) = c + \sum_j w_j e_{ij} \qquad (21)$$

where

$$c = \sum_j \log(\alpha_j + \beta_j) - \log(\alpha_j + \beta_j + N) + \log \tilde{\beta}_j - \log \beta_j$$

and $w_j = \log \tilde{\alpha}_j - \log \alpha_j - \log \tilde{\beta}_j + \log \beta_j \qquad (22)$

All possible entities e_i will be assigned a similarity score by Equation (21). Then we can rank them accordingly. The top ranked entities should be highly related to the seed set Q according to the Bayesian Sets algorithm.

However, Zhang and Liu (2011c) found that this direct application of Bayesian Sets produces poor results. They believe there are two main reasons. First, since Bayesian Sets uses binary features, multiple occurrences of an entity in the corpus, which give rich contextual information, is not fully exploited. Second, since the number of seeds is very small, the learned results from Bayesian Sets can be quite unreliable.

They proposed a method to improve Bayesian Sets, which produces much better results. The main improvements are as follows.

Raising Feature Weights: From Equation (21), we can see that the score of an entity e_i is determined only by its corresponding feature vector and the weight vector $w = (w_1, w_2, ..., w_j)$. Equation (22) shows a value of the weight vector w. They rewrite Equation (22) as follows,

$$w_j = \log \frac{\tilde{\alpha}_j}{\alpha_j} - \log \frac{\tilde{\beta}_j}{\beta_j} = \log(1 + \frac{\sum_{i=1}^{N} q_{ij}}{km_j}) - \log(1 + \frac{N - \sum_{i=1}^{N} q_{ij}}{k(1 - m_j)}) \qquad (23)$$

In Equation (23), N is the number of items in the seed set. As mentioned before, m_j is the mean of feature j of all possible entities and k is a scaling factor. m_j can be regarded as the prior information empirically set from the data.

In order to make a positive contribution to the final score of entity e, w_j must be greater than zero. Under this circumstance, it can obtain the following inequality based on Equation (23).

$$\sum_{i=1}^{N} q_{ij} > Nm_j \qquad (24)$$

Equation (24) shows that if feature j is effective ($w_j > 0$), the seed data mean must be greater than the candidate data mean on feature j. Only such kind of features can be regarded as high-quality features in Bayesian Sets. Unfortunately, it is not always the case due to the idiosyncrasy of the data. There are many high-quality features, whose seed data mean may be even less than the candidate data mean. For example, in drug data set, "*prescribe*" can be a left first verb for an entity. It is a very good entity feature. "*Prescribe EN/NNP*" (EN represents an entity, NNP is its POS tag) strongly suggests that EN is a drug. However, the problem is that the mean of this feature in the seed set is 0.024 which is less than its candidate set mean 0.025. So if we stick with Equation (24), the feature will have negative contribution, which means that it is worse than no feature at all. The fact that all pattern features are from sentences containing seeds, a candidate entity associated with a feature should be better than no feature.

Zhang and Liu tackled this problem by fully utilizing all features found in corpus. They changed original m_j to $\widetilde{m}_j$ by multiplying a scaling factor t to force all feature weights $w_j > 0$:

$$\widetilde{m}_j = tm_j \qquad (0 < t < 1) \qquad (25)$$

The idea is that they lower the candidate data mean intentionally so that all the features found from the seed data can be utilized.

Identifying High-Quality Features: Equation (23) shows that besides m_j value, w_j value is also affected by the sum $\sum_{i=1}^{N} q_{ij}$. It means that if the feature occurs more times in the seed data, its corresponding w_j will also be high. However, Equation (23) may not be sufficient since it only considers the feature occurrence but does not take feature quality into consideration. For example, two different features A and B, which have the same

feature occurrence in the seed data and thus the same mean, According to Equation (23), they should have the same feature weight *w*. However, for feature A, all feature counts may come from only one entity in the seed set, but for feature B, the feature counts are from four different entities in the seed set. Obviously, feature B is a better feature than feature A simply because the feature is shared by or associated with more entities. To detect such high-quality features to increase their weights, Zhang and Liu used the following formula to change the original w_j to $\tilde{w}_j$.

$$\tilde{w}_j = rw_j \qquad (26)$$

$$r = (1 + \frac{\log h}{T}) \qquad (27)$$

In Equation (26), *r* is used to represent feature quality for feature *j*. *h* is the number of unique entities that have *j*-th feature. *T* is the total number of entities in the seed set.

In Zhang and Liu (2011c), different vectors representing the same candidate entity are produced as in (Li et al., 2010b). Thus, the same ranking algorithm is adopted, which is the multiplication of the median of the score vector obtained from feature vectors representing the entity and the logarithm of entity frequency.

5   Summary

With the explosive growth of social media on the Web, organizations are increasingly relying on opinion mining methods to analyze the content of these media for their decision making. Aspect-based opinion mining, which aims to obtain detailed information about opinions, has attracted a great of deal of attention from both the research community and industry. Aspect extraction and entity extraction are two of its core tasks. In this chapter, we reviewed some representative works for aspect extraction and entity extraction from opinion documents.

For aspect extraction, existing solutions can be grouped into three main categories:

(1) using language dependency rules, e.g., *double propagation* (Qiu et al., 2011). These methods utilize the relationships between aspects and opinion words or other terms to perform aspect extraction. The approaches are unsupervised and domain-independent. Thus, they can be applied to any domain.

(2) using sequence learning algorithms such as HMM and CRF (Jin et al., 2009a; Jakob and Gurevych, 2010). These supervised methods are the dominating techniques for traditional information extraction. But they need a great deal of manual labeling effort.

(3) using topic models, e.g., MG-LDA (Titov and McDonald, 2008a). This is a popular research area for aspect extraction recently. The advantages of topic models are that they can group similar aspect expressions together and that they are unsupervised. However, their limitation is that the extracted aspects are not fine-grained.

For entity extraction, supervised learning has also been the dominating approach. However, semi-supervised methods have drawn attention recently. As in opinion mining, users often want to find competing entities for opinion analysis, they can provide some knowledge (e.g., entity instances) as seeds for semi-supervised learning. In this chapter, we introduced *PU learning* and *Bayesian Sets* based semi-supervised extraction methods.

For evaluation, the commonly used measures for information extraction such as precision, recall and F-1 scores are also often used in aspect and entity extraction. The current F-1 score results range from 0.60 to 0.85 depending on domains and datasets. Thus, the problems, especially aspect extraction, remain to be highly challenging. We expect that the future work will improve the accuracy significantly. We also believe that semi-supervised and unsupervised methods will play a larger role in these tasks.

References

Bethard, S., H. Yu, A. Thornton, V. Hatzivassiloglou, and D. Jurafsky. Automatic extraction of opinion propositions and their holders. In *Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text,* 2004.

Blair-Goldensohn, S., K. Hannan, R. McDonald, T. Neylon, G. A. Reis, and J. Reyna. Building a sentiment summarizer for local service reviews. In *Proceedings of International Conference on World Wide Web Workshop of NLPIX* (*WWW-NLPIX-2008*), 2008.

Blei, D., A. Ng, and M. Jordan. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 2003.

Bloom, K., N. Grag and S. Argamon. Extracting apprasial expressions. In *Proceedings of The 2007 Annual Conference of the North American Chapter of the ACL*(*NAACL-2007*), 2007.

Branavan, S.R.K., H. Chen, J. Eisenstein, and R. Barzilay. Learning document-level semantic properties from free-text annotations. In *Proceedings of Annual Meeting of the Association for Computational Linguistics* (*ACL-2008*), 2008.

Brown, F. P., S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer. The mathematics of statitical machine translation: parameter estimation. *Computational Linguistics*, 1993.

Brody, S. and S. Elhadad. An unsupervised aspect-sentiment model for online reviews. In *Proceedings of The 2010 Annual Conference of the North American Chapter of the ACL*(*NAACL-2010*), 2010.

Carenini, G., R. Ng, and A. Pauls. Multi-Document summarization of evaluative text. In *Proceeding of Conference of the European Chapter of the ACL(EACL-2006), 2006*.

Carenini, G,. R. Ng, and E. Zwart. Extracting knowledge from evaluative text. In *Proceedings of Third International Conference on Knowledge Capture (K-CAP-2005)*, 2005.

Choi, Y., C. Cardie, E. Riloff, and S. Patwardhan. Identifying sources of opinions with conditional random fields and extraction patterns. In *Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing* (*HLT/EMNLP-2005*), 2005.

Dempster, P., A. M. N. Laird, and B. D. Rubin. Maximum likelihood from incomplete data via the EM algorithms. *Journal of the Royal Statistical Society*, Series B, 1977.

Fei, G., B. Liu, M. Hsu, M. Castellanos, R. Ghosh. A dictionary-based approach to identifying aspects implied by adjectives for opinion mining. In *Proceedings of International Conference on Computational Linguistics (COLING-2012)*, 2012.

Ghahramani, Z. and K.A. Heller. Bayesian sets. In *Proceeding of Annual Neural Information Processing Systems (NIPS-2005)*, 2005.

Ghani, R., K. Probst, Y. Liu, M. Krema, A. Fano. Text mining for product attribute extraction. *ACM SIGKDD Explorations Newsletter*, 2006. 8(1).

Gilks, R. W., S. Richardson, and D. Spiegelhalter. Markov Chain Monte Carlo in practice. Chapman and Hall, 1996.

Grosz, J. B., S. Winstein and A. K. Joshi. Centering: a framework for modeling the local coherence of discourse. *Computational Linguistics*, 21(2), 1995.

Guo H., H. Zhu, Z. Guo, X. Zhang and Z. Su. Product feature categorization with multilevel latent semantic association. In *Proceedings of ACM International Conference on Information and Knowledge Management (CIKM-2009)*, 2009.

Hai, Z., K. Chang., and J. Kim. Implicit feature identification via co-occurrence association rule mining. *Computational Linguistic and Intelligent Text Processing*, 2011.

Hai, Z., K. Chang., and G. Cong. One seed to find them all: mining opinion features via association. In *Proceedings of ACM International conference on Information and Knowledge Management (CIKM-2012)*, 2012.

Hofmann, T. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 2001.

Hu, M. and B. Liu. Mining opinion features in customer reviews. In *Proceedings of National Conference on Artificial Intelligence (AAAI-2004)*, 2004a.

Hu, M. and B. Liu. Mining and summarizing customer reviews. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2004)*, 2004b.

Jakob, N. and I. Gurevych. Extracting opinion targets in a single and cross-domain setting with conditional random fields. In *Proceedings of*

Conference on Empirical Methods in Natural Language Processing (*EMNLP-2010*), 2010.

Jin, W. and H. Ho. A novel lexicalized HMM-based learning framework for web opinion mining. In *Proceedings of International Conference on Machine Learning* (*ICML-2009*), 2009a.

Jin, W., H. Ho, and R. K. Srihari. OpinionMiner: a novel machine learning system for web opinion mining and extraction. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (*KDD-2009*), 2009b.

Jindal, N. and B. Liu. Mining comparative sentences and relations. In *Proceedings of National Conference on Artificial Intelligence* (*AAAI-2006*), 2006a.

Jindal, N. and B. Liu. Identifying comparative sentences in text documents. In *Proceedings of ACM SIGIR International Conference on Information Retrieval* (*SIGIR-2006*), 2006b.

Jo, Y. and A. Oh. Aspect and sentiment unification model for online review analysis. In *Proceedings of the Conference on Web Search and Web Data Mining* (*WSDM-2011*), 2011.

Kessler, J. and N. Nicolov. Targeting sentiment expressions through supervised ranking of linguistic configurations. In *Proceedings of the International AAAI Conference on Weblogs and Social Media* (*ICWSM-2009*), 2009.

Kim, S. M. and E. Hovy. Extracting opinions, opinion holders, and topics expressed in online news media text. In *Proceedings of the ACL Workshop on Sentiment and Subjectivity in Text*, 2006.

Kleinberg, J. Authoritative sources in hyper-linked environment. Journal of the ACM 46 (5): 604-632 1999.

Kobayashi, N., K. Inui, and Y. Matsumoto. 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-2007*), 2007.

Ku, L., Y. Liang, and H. Chen. Opinion extraction, summarization and tracking in news and blog corpora. In *Proceedings of AAAI-CAAW-2006*, 2006.

Lafferty, J., A. McCallum, and F, Pereira. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *Pro-*

ceedings of International Conference on Machine Learning (*ICML-2001*), 2001.

Lee, L. Measures of distributional similarity. In *Proceedings of Annual Meeting of the Association for Computational Linguistics* (*ACL-1999*), 1999.

Li, F., C. Han, M. Huang, X. Zhu, Y. Xia, S. Zhang and H. Yu. Structure-aware review mining and summarization. In *Proceedings of International Conference on Computational Linguistics* (*COLING-2010*), 2010a.

Li, F., S. J. Pan, Q. Jin, Q. Yang and X. Zhu. Cross-Domain co-extraction of sentiment and topic lexicons. In *Proceedings of Annual Meeting of the Association for Computational Linguistics* (*ACL-2012*), 2012a.

Li, S., R. Wang, and G. Zhou. Opinion target extraction using a shallow semantic parsing framework. In *Proceedings of National Conference on Artificial Intelligence* (*AAAI-2012*), 2012b.

Li, X., and B. Liu. Learning to classify texts using positive and unlabeled data. In *Proceedings of International Joint Conferences on Artificial Intelligence* (*IJCAI-2003*), 2003.

Li, X., B. Liu, S. Ng. Learning to identify unexpected instances in the test set. In *Proceedings of International Joint Conferences on Artificial Intelligence* (*IJCAI-2007*), 2007.

Li, X., L. Zhang, B. Liu, and S. Ng. Distributional similarity vs. PU learning for entity set expansion. In *Proceedings of Annual Meeting of the Association for Computational Linguistics* (*ACL-2010*), 2010b.

Lin, C. and Y. He. Joint sentiment/topic model for sentiment analysis. In *Proceedings of ACM International Conference on Information and Knowledge Management* (*CIKM-2009*), 2009.

Lin, D. Dependency-based evaluation of MINIPAR. In *Proceedings of the Workshop on Evaluation of Parsing System* (*ICLRE-1998*), 1998.

Liu, B. Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data. Springer, First Edition, 2006 and Second Edition, 2011.

Liu, B. Sentiment analysis and subjectivity. Handbook of Natural Language Processing, Second Edition, 2010.

Liu, B. Sentiment Analysis and Opinion Mining. Morgan & Claypool Publishers, 2012.

Liu, B., M. Hu, and J. Cheng. Opinion observer: analyzing and comparing opinions on the web. In *Proceedings of International Conference on World Wide Web* (*WWW-2005*), 2005.

Liu, B., W-S. Lee, P. S. Yu and X. Li. Partially supervised text classification. In *Proceedings of International Conference on Machine Learning* (*ICML-2002*), 2002.

Liu, K., L. Xu and J. Zhao. Opinion target extraction using word-based translation model. In *Proceeding of Conference on Empirical Methods in Natural Language Processing* (*EMNLP-2012*), 2012.

Long, C., J. Zhang, and X. Zhu. A review selection approach for accurate feature rating estimation. In *Proceedings of International Conference on Computational Linguistics* (*COLING-2010*), 2010.

Lu, Y., H. Duan, H. Wang, and C. Zhai. Exploiting structured ontology to organize scattered online opinions. In *Proceedings of International Conference on Computational Linguistics* (*COLING-2010*), 2010.

Lu, Y., C. Zhai, and N. Sundaresan. Rated aspect summarization of short comments. In *Proceedings of International Conference on World Wide Web* (*WWW-2009*), 2009.

Ma, T., and X. Wan. Opinion target extraction in Chinese news comments. In *Proceedings of International Conference on Computational Linguistics*(*COLING-2010*), 2010.

Mauge, K., K. Rohanimanesh and J. D. Ruvini. Structuring e-commerce inventory. In *Proceedings of Annual Meeting of the Association for Computational Linguistics* (*ACL-2012*), 2012.

Mukherjee, A. and B. Liu. Aspect extraction through semi-supervised modeling. In *Proceedings of Annual Meeting of the Association for Computational Linguistics* (*ACL-2012*), 2012.

Mei, Q., X. Ling, M. Wondra, H. Su, and C. Zhai. Topic sentiment mixture: modeling facets and opinions in weblogs. In *Proceedings of International Conference on World Wide Web* (*WWW-2007*), 2007.

Moghaddam, S. and M. Ester. Opinion digger: an unsupervised opinion miner from unstructured product reviews. In *Proceedings of ACM International conference on Information and Knowledge Management (CIKM-2010)*, 2010.

Moghaddam, S. and M. Ester. ILDA: interdependent LDA model for learning latent aspects and their ratings from online product reviews. In

Proceedings of ACM SIGIR International Conference on Information Retrieval (*SIGIR-2011*), 2011.

Neter, J., W. Wasserman, and G. A, Whitmore. Applied Statistics. Allyn and Bacon 1993.

Pang, B. and L. Lee. Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval, 2008. 2(1-2): p. 1-135

Pang, B., L. Lee, and S. Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of Conference on Empirical Methods in Natural Language Processing* (*EMNLP-2002*), 2002.

Pantel, P., E. Crestan, A. Borkovsky, A. Popescu and V, Vyas. Web-Scale distributional similarity and entity set expansion. In *Proceedings of the 2009 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning* (*EMNLP-2009*), 2009.

Popescu, A. and O. Etzioni. Extracting product features and opinions from reviews. In *Proceedings of Conference on Empirical Methods in Natural Language Processing* (*EMNLP-2005*), 2005.

Putthividhya, D. and J. Hu. Bootstrapped name entity recognition for product attribute extraction. In *Proceedings of Conference on Empirical Methods in Natural Language Processing* (*EMNLP-2011*), 2011.

Qiu, G., B. Liu, J. Bu, and C. Chen. Opinion word expansion and target extraction through double propagation. *Computational Linguistics*, 2011.

Rabiner, R. L. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of IEEE* ,1989. 77(2).

Sarawagi, S. Information Extraction. *Foundations and Trends in Databases*, 2008.

Sauper, C., A. Haghighi, R. Barzilay. Content models with attribute. In *Proceedings of Annual Meeting of the Association for Computational Linguistics* (*ACL-2011*), 2011.

Scaffidi, C., K. Bierhoff, E. Chang, M. Felker, H. Ng, and C. Jin. Red opal: product-feature scoring from reviews. In *Proceedings of the 9th International Conference on Electronic Commerce* (*EC-2007*). 2007.

Stoyanov, V. and C. Cardie. Topic identification for fine-grained opinion analysis. In *Proceedings of International Conference on Computational Linguistics* (*COLING-2008*), 2008.

Su, Q., X. Xu, H. Guo, Z. Guo, X. Wu, X. Zhang, B. Swen, and Z. Su. Hidden sentiment association in Chinese web opinion mining. In *Proceedings of International Conference on World Wide Web* (*WWW-2008*), 2008.

Sutton, C. and A. McCallum. An introduction to conditional random fields for relational learning. *Introduction to Statistical Relational Learning*. MIT Press, 2006.

Titov, I. and R. McDonald. Modeling online reviews with multi-grain topic models. In *Proceedings of International Conference on World Wide Web* (*WWW-2008*), 2008a.

Titov, I. and R. McDonald. A joint model of text and aspect ratings for sentiment summarization. In *Proceedings of Annual Meeting of the Association for Computational Linguistics* (*ACL-2008*), 2008b.

Turney, P. D. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of Annual Meeting of the Association for Computational Linguistics* (*ACL-2002*). 2002.

Wang, B. and H, Wang. Bootstrapping both product features and opinion words from Chinese customer reviews with cross-inducing. In *Proceedings of the International Joint Conference on Natural Language Processing* (*IJCNLP-2008*), 2008.

Wang, H., Y. Lu, and C. Zhai. Latent aspect rating analysis on review text data: a rating regression approach. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (*KDD-2010*), 2010.

Wei, W. and J. A. Gulla. Sentiment learning on product reviews via sentiment ontology tree. In *Proceedings of Annual Meeting of the Association for Computational Linguistics* (*ACL-2010*).

Wiebe, J. and E. Riloff. Creating subjective and objective sentence classifiers from unannotated texts. In *Proceedings of Computational Linguistics and Intelligent Text Processing* (*CICLing-2005*), 2005..

Wiebe, J., T. Wilson, R. Bruce, M. Bell, and M. Martin. Learning subjective language. *Computational Linguistics*, 2004, 30(3): p. 277-308.

Wilson, T., J. Wiebe, and P. Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing* (*HLT/EMNLP-2005*), 2005.

Wu, Y., Q. Zhang, X. Huang, and L. Wu. Phrase dependency parsing for opinion mining. In *Proceedings of Conference on Empirical Methods in Natural Language Processing* (*EMNLP-2009*), 2009.

Yi, J., T. Nasukawa, R. Bunescu, and W. Niblack. Sentiment analyzer: extracting sentiments about a given topic using natural language processing techniques. In *Proceedings of International Conference on Data Mining* (*ICDM-2003*), 2003.

Yu, H., J. Han, K. Chang. PEBL: Positive example based learning for Web page classification using SVM. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (*KDD-2002*), 2002.

Yu, J., Z. Zha, M. Wang, and T. Chua. Aspect ranking: identifying important product aspects from online consumer reviews. In *Proceedings of Annual Meeting of the Association for Computational Linguistics* (*ACL-2011*), 2011a.

Yu, J., Z. Zha, M. Wang, K. Wang and T. Chua. Domain-Assisted product aspect hierarchy generation: towards hierarchical organization of unstructured consumer reviews. In *Proceedings of Conference on Empirical Methods in Natural Language Processing* (*EMNLP-2011*), 2011b.

Zhai, Z., B. Liu, H. Xu, and P. Jia. Clustering product features for opinion mining. In *Proceedings of ACM International Conference on Web Search and Data Mining* (*WSDM-2011*), 2011.

Zhai, Z., B. Liu, H. Xu, and P. Jia. Grouping product features using semi-supervised learning with soft-constraints. In *Proceedings of International Conference on Computational Linguistics* (*COLING-2010*), 2010.

Zhang, L., B. Liu, S. Lim, and E. O'Brien-Strain. Extracting and ranking product features in opinion documents. In *Proceedings of International Conference on Computational Linguistics* (*COLING-2010*), 2010.

Zhang, L and B. Liu. Identifying noun product features that imply opinions. In *Proceedings of Annual Meeting of the Association for Computational Linguistics* (*ACL-2011*), 2011a.

Zhang, L. and B. Liu. Extracting resource terms for sentiment analysis. In *Proceedings of the International Joint Conference on Natural Language Processing* (*IJCNLP-2011*), 2011b.

Zhang, L. and B. Liu. Entity set expansion in opinion documents. In *Proceedings of ACM Conference on Hypertext and Hypermedia* (*HT-2011*), 2011c.

Zhao, W., J. Jiang, H. Yan, and X. Li. Jointly modeling aspects and opinions with a MaxEnt-LDA hybrid. In *Proceedings of Conference on Empirical Methods in Natural Language Processing* (*EMNLP-2010*), 2010.

Zhu, J., H. Wang, B. K. Tsou, and M. Zhu. Multi-aspect opinion polling from textual reviews. In *Proceedings of ACM International conference on Information and Knowledge Management* (*CIKM-2009*), 2009.

Zhuang, L., F. Jing, and X. Zhu. Movie review mining and summarization. In *Proceedings of ACM International Conference on Information and Knowledge Management* (*CIKM-2006*), 2006.