11 Opinion Mining and Sentiment Analysis

In Chap. 9, we studied the extraction of structured data from Web pages. The Web also contains a huge amount of information in unstructured texts. Analyzing these texts is of great importance as well and perhaps even more important than extracting structured data because of the sheer volume of valuable information of almost any imaginable type contained in text. In this chapter, we only focus on mining opinions which indicate positive or negative sentiments. The task is technically challenging and practically very useful. For example, businesses always want to find public or consumer opinions about their products and services. Potential customers also want to know the opinions of existing users before they use a service or purchase a product.

This area of study is called **opinion mining** or **sentiment analysis**. It analyzes people's opinions, appraisals, attitudes, and emotions toward entities, individuals, issues, events, topics, and their attributes. Opinions are important because they are key influencers of our behaviors. Our beliefs and perceptions of reality, and the choices we make, are to a considerable degree conditioned on how others see and evaluate the world. For this reason, when we need to make a decision we often seek out the opinions of others. This is true not only for individuals but also for organizations.

With the explosive growth of **social media** (i.e., reviews, forum discussions, blogs, and social networks) on the Web, individuals and organizations are increasingly using the content in these media for their decision making. Nowadays, if one wants to buy a consumer product, one is no longer limited to asking one's friends and family for opinions as in the past because there are many user reviews of products on the Web. For an organization, it may no longer be necessary to conduct opinion polls, surveys, and focus groups in order to gather public opinions about its products and services because there is an abundance of such information publicly available. However, finding and monitoring opinion sites on the Web and distilling the information contained in them remains a formidable task because of the proliferation of diverse sites. Each site typically contains a huge volume of **opinionated text** that is not always easily deciphered in long forum postings and blogs. The average human reader will have difficulty identifying relevant sites and accurately summarizing the information

B. Liu, *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*,
Data-Centric Systems and Applications, DOI 10.1007/978-3-642-19460-3_11,

and opinions contained in them. Moreover, it is also known that human analysis and evaluation of text information is subject to considerable biases, e.g., people often pay greater attention to opinions that are consistent with their own preferences. People also have difficulty, owing to their mental and physical limitations, producing consistent results when the amount of information to be processed is large. Automated opinion mining and summarization systems are thus needed, as subjective biases and mental limitations can be overcome with an objective opinion analysis system.

In the past decade, a considerable amount of research has been done in academia [70, 91]. There are also numerous commercial companies that provide opinion mining services. In this chapter, we first define the opinion mining problem. From the definition, we will see the key technical issues that need to be addressed. We then describe various key mining tasks that have been studied in the research literature and their representative techniques. After that, we discuss the related issue of opinion spam detection. Opinion spam refers to dishonest opinions or reviews that try to promote or demote some target products or services. Detecting such spam opinions is critical for practical applications of opinion mining.

11.1 The Problem of Opinion Mining

In this first section, we define an abstraction of the opinion mining problem. It enables us to see a *structure* from the complex and intimidating unstructured text. Moreover, for most opinion-based applications, it is essential to analyze a collection of opinions rather than only one because one opinion represents only the view of a single person, which is usually not sufficient for action. This indicates that some form of **summary of opinions** is needed [37]. The abstraction should facilitate this summarization.

11.1.1 Problem Definitions

We use the following review segment on iPhone to introduce the problem (an id number is associated with each sentence for easy reference):

> "(1) I bought an iPhone a few days ago. (2) It was such a nice phone. (3) The touch screen was really cool. (4) The voice quality was clear too. (5) However, my mother was mad with me as I did not tell her before I bought it. (6) She also thought the phone was too expensive, and wanted me to return it to the shop. … "

The question is: what we want to mine or extract from this review? The first thing that we notice is that there are several opinions in this review.

Sentences (2), (3), and (4) express some positive opinions, while sentences (5) and (6) express negative opinions or emotions. Then we also notice that the opinions all have some targets. The target of the opinion in sentence (2) is the iPhone as a whole, and the targets of the opinions in sentences (3) and (4) are "touch screen" and "voice quality" of the iPhone, respectively. The target of the opinion in sentence (6) is the price of the iPhone, but the target of the opinion/emotion in sentence (5) is "me", not iPhone. Finally, we may also notice the holders of opinions. The holder of the opinions in sentences (2), (3), and (4) is the author of the review ("I"), but in sentences (5) and (6) it is "my mother." With this example in mind, we now formally define the opinion mining problem. We start with the **opinion target**.

In general, opinions can be expressed about anything, e.g., a product, a service, an individual, an organization, an event, or a topic, 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. Formally, we have the following:

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 1: 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 attributes, 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 2: Following Example 1, one can express an opinion on the cellular phone 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 phone'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 a difficult task. 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**)**:** The *aspects* of an entity *e* are the components and attributes of *e*.

Note that in the first edition of this book, we used the term **feature** to mean aspect. Using feature is natural for the product domain as people often say "product features." However, when entities are events and topics, the term feature becomes unnatural. Furthermore, the term feature also confuses with the term feature used in machine learning, where a feature means a data attribute. To avoid confusion, we adopt the term **aspect** in this edition.

Definition (**aspect name and aspect expression**)**:** An *aspect name* is the name of an aspect given by the user, while an *aspect expression* is an actual word or phrase that has appeared in text indicating an aspect.

Example 3: 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 also "voice quality" itself. ■

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 also 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, e.g., "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."

Definition (**entity name and entity expression**)**:** An *entity name* is the name of an entity given by the user, while an *entity expression* is an actual word or phrase that has appeared in text indicating an 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 [4, 13, 55]. Opinion holders are also called **opinion sources** [128].

We now turn to opinions. There are two main types of opinions: **regular opinions** and **comparative opinions**. Regular opinions are often referred to simply as **opinions** in the research literature. A comparative opinion expresses a relation of similarities or differences between two or more entities and/or a preference of the opinion holder based on some of the shared aspects of the entities [42, 43]. A comparative opinion is usually expressed using the *comparative* or *superlative* form of an adjective or adverb, although not always. More detailed definitions will be given in Sect. 11.6. The discussion below focuses only on regular opinions. For simplicity, the terms *regular opinion* and *opinion* are used interchangeably below.

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 [70].

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 while, we use the special aspect GENERAL to denote it.

Some important remarks about this definition are in order:

1. It should be stressed that the five pieces of information in the quintuple must correspond to one another. That is, the opinion oo_{ijkl} must be given by opinion holder h_k about aspect a_{ij} of entity e_i at time t_l. Otherwise, we may assign an opinion to a wrong entity or wrong aspect, etc.
2. These five components are essential. Without any of them, it can be problematic in general. For example, one says "The picture quality is great," but if we do not know whose picture quality, the opinion is of little use. However, we do not mean that every piece of information is

needed in every application. For example, knowing each opinion holder is not necessary if we want to summarize opinions from a large number of people. Similarly, we do not claim that nothing else can be added to the tuple. For example, in some applications (e.g., marketing), the user may want to know the sex and age of each opinion holder.

3. This definition provides a basis for transforming unstructured text into structured data. The quintuple gives us the essential information for a rich set of qualitative and quantitative analysis of opinions. More specifically, the quintuple is basically a schema/relation for a database table. With a large set of opinion quintuples mined from text, the whole suite of database management systems (DBMS) and OLAP tools can be applied to slice and dice the opinions for all kinds of analyses.
4. Opinions (regular opinions) also have sub-types, e.g., **direct opinions** and **indirect opinions**. For direct opinions, opinions are expressed directly on entities or their aspects, e.g., "The voice quality of this phone is great." For indirect opinions, opinions on entities are expressed based on their effects on some other entities. This sub-type often occurs in the medical domain. For example, "After taking this drug, my hand felt much better" describes a desirable effect of the drug on "my hand," which indirectly gives a positive opinion to the drug. For simplicity, we will not distinguish these sub-types in this chapter.
5. In the original definition of an entity, it is a hierarchy/tree of components, sub-components, and so on. Every component can have its set of attributes. Due to simplification by flattening the tree, the quintuple representation can result in information loss. For example, "battery" is a component/part of a digital camera. In a camera review, one wrote "The battery for this camera is expensive." This does not say that the camera is expensive (which indicates the aspect *price*). If one does not care about any attribute of the battery, this sentence just gives a negative opinion to the battery, which is an aspect of the camera entity. However, if one also wants to study opinions about different aspects of the battery, e.g., battery life, price, etc., the battery needs to be treated as a separate entity. The quintuple representation still applies, but the part-of relationship needs to be saved. Of course, conceptually one may also treat the quintuple as a nested relation rather than a flat relation.

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** (or *feature-based opinion mining* as it was called earlier and in the first edition of this book) [37, 71].

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.

To achieve this objective, one needs to perform the following tasks:

Task 1 (entity extraction and grouping): Extract all entity expressions in D, and group synonymous entity expressions into entity clusters. Each entity expression cluster indicates a unique entity e_i.

Task 2 (aspect extraction and grouping): Extract all aspect expressions of the entities, and group aspect expressions into clusters. Each aspect expression cluster of entity e_i indicates a unique aspect a_{ij}.

Task 3 (opinion holder and time extraction): Extract these pieces of information from the text or structured data.

Task 4 (aspect sentiment classification): Determine whether each opinion on an aspect is positive, negative or neutral.

Task 5 (opinion quintuple generation): Produce all opinion quintuples (e_i, a_{ij}, oo_{ijkl}, h_k, t_l) expressed in D based on the results of the above tasks.

The difficulty of opinion mining lies in the fact that none of the above problems or tasks is a solved problem. To make matters worse, a sentence may not explicitly mention some pieces of information, but they are implied due to pronouns, language conventions, and contexts. What is also challenging is to ensure that the five pieces of information in an opinion correspond to one another as we discussed earlier. We now use an example blog to illustrate the tasks (a sentence id is associated with each sentence):

Example 4: *Posted by*: *bigXyz on Nov-4-2010*: (1) I bought a Motorola phone and my girlfriend bought a Nokia phone yesterday. (2) We called each other when we got home. (3) The voice of my Moto phone was unclear, but the camera was good. (4) My girlfriend was quite happy with her phone, and its sound quality. (5) I want a phone with good voice quality. (6) So I probably will not keep it. ■

Task 1 should extract the entity expressions, “Motorola,” “Nokia,” and “Moto,” and group “Motorola” and “Moto” together as they represent the same entity. Task 2 should extract aspect expressions “camera,” “voice,” and “sound” and group “voice” and “sound” together as they are synonyms representing the same aspect. Task 3 should find the holder of the opinions in sentence (3) to be bigXyz (the blog author) and the holder of the opinions in sentence (4) to be bigXyz's girlfriend. It should also find the time when the blog was posted, which is Nov-4-2010. Task 4 should find that sentence (3) gives a negative opinion to the voice quality of the Motorola phone but a positive opinion to its camera. Sentence (4) gives positive opinions to the Nokia phone as a whole and also its sound quality. Sentence (5) seemingly expresses a positive opinion, but it does not. To generate opinion quintuples for sentence (4), we also need to know what "her phone" is and what "its" refers to. All these are challenging problems. Task 5 should finally generate the following four opinion quintuples:

(Motorola, voice_quality, negative, bigXyz, Nov-4-2010)
(Motorola, camera, positive, bigXyz, Nov-4-2010)
(Nokia, GENERAL, positive, bigXyz's girlfriend, Nov-4-2010)
(Nokia, voice_quality, positive, bigXyz's girlfriend, Nov-4-2010)

Before going further, let us discuss two other important concepts related to opinion mining and sentiment analysis, i.e., **subjectivity** and **emotion**.

Definition (sentence subjectivity): An *objective sentence* presents some factual information about the world, while a *subjective sentence* expresses some personal feelings, views, or beliefs.

For example, in Example 4, sentences (1) and (2) are objective sentences, while all other sentences are subjective sentences. Subjective expressions come in many forms, e.g., opinions, allegations, desires, beliefs, suspicions, and speculations [102, 123]. Thus, a subjective sentence may not contain an opinion. For example, sentence (5) in Example 4 is subjective but it does not express a positive or negative opinion about anything. Interestingly, objective sentences can imply opinions [140]. For example,

"The earphone broke in two days."

is an objective sentence but it implies a negative opinion. There is some confusion among researchers to equate subjectivity with opinionated. As we can see, the concepts of subjective sentences and opinion sentences are not the same, although they have a large intersection. The task of determining whether a sentence is subjective or objective is called **subjectivity classification** [126], which we will discuss in Sect. 11.3.

Definition (emotion): Emotions are our subjective feelings and thoughts.

Emotions have been studied in many fields, e.g., psychology, philosophy, and sociology. However, there is still not a set of agreed basic emotions of people among researchers. Based on [96], people have six primary emotions, i.e., love, joy, surprise, anger, sadness, and fear, which can be sub-divided into many secondary and tertiary emotions. Each emotion can also have different intensities. The strengths of opinions are related to the intensities of certain emotions, e.g., joy and anger. However, the concepts of emotions and opinions are not equivalent. Many opinion sentences express no emotion (e.g., "The voice of this phone is clear"), and many emotion sentences express no opinion (e.g., "I am so surprised to see you").

11.1.2 Aspect-Based Opinion Summary

As mentioned at the beginning of this section, most opinion mining applications need to study opinions from a large number of opinion holders. One opinion from a single holder is usually not sufficient for action. This indicates that some form of summary of opinions is needed. Opinion quintuples defined above provide an excellent source of information for generating both qualitative and quantitative summaries. A common form of summary is based on aspects and is called **aspect-based opinion summary** (or *feature-based opinion summary*) [37, 72]. Below, we use an example to illustrate this form of summary, which is widely used in industry.

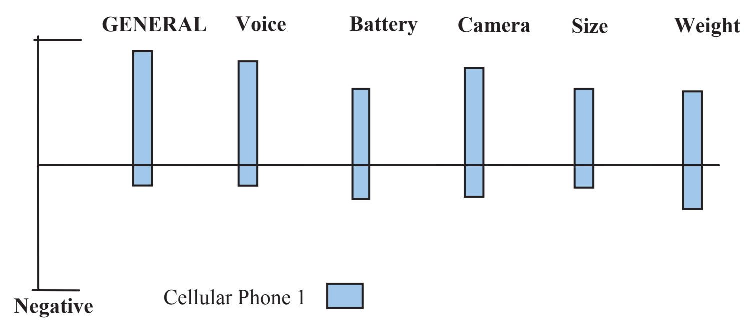
Example 5: Assume we summarize all the reviews of a particular cellular phone, *cellular phone* 1. The summary looks like that in Fig. 11.1, which was proposed in [37] and is called a **structured summary**. In the figure, GENERAL represents the phone itself (the entity). 125 reviews expressed positive opinions about the phone and 7 expressed negative opinions. *Voice quality* and *size* are two product aspects. 120 reviews expressed positive opinions about the voice quality, and only 8 reviews expressed negative opinions. The <individual review sentences> link points to the specific sentences and/or the whole reviews that give the positive or negative opinions. With such a summary, the user can easily see how existing customers feel about the phone. If he/she is interested in a particular aspect, he/she can drill down by following the <individual review sentences> link to see why existing customers like it and/or dislike it. ■

As mentioned earlier, the discovered quintuples can be stored in database tables. Then a whole suite of database and visualization tools can be applied to see the results in all kinds of ways to gain insights of the opinions in structured forms and displayed as bar charts and/or pie charts. For example, the aspect-based summary in Fig. 11.1 can be visualized using the bar chart in Fig. 11.2(A) [72]. In the figure, each bar above the *X*-axis shows the number of positive opinions on the aspect given at the top. The corresponding bar below the *X*-axis shows the number of negative opinions on the same aspect. Obviously, other visualizations are also possible. For example, one may only show the percent of positive opinions.

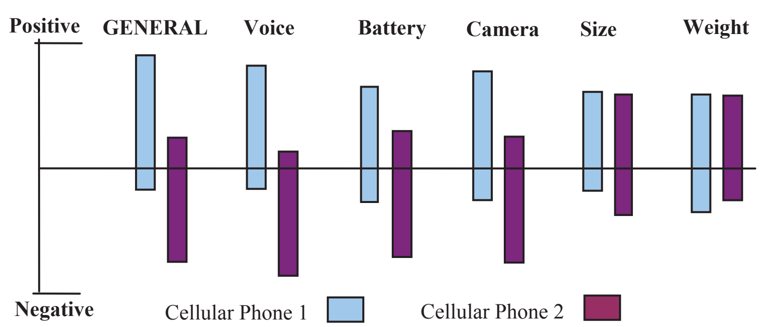
Cellular phone 1:

Aspect: **GENERAL**		
Positive:	125	<individual review sentences>
Negative:	7	<individual review sentences>
Aspect: **Voice quality**		
Positive:	120	<individual review sentences>
Negative:	8	<individual review sentences>
Aspect: **Battery**		
Positive:	80	<individual review sentences>
Negative:	12	<individual review sentences>
…		

Fig. 11.1. An aspect-based opinion summary.



(A) Visualization of aspect-based summary of opinions on a cellular phone



(B) Visual opinion comparison of two cellular phones

Fig. 11.2. Visualization of aspect-based summaries of opinions

Comparing opinion summaries of a few competing products is even more interesting [72]. Fig. 11.2(B) shows the visual opinion comparison of two competing phones. We can clearly see how consumers view each of them along different aspect dimensions.

Researchers have also studied opinion summarization in the tradition fashion, e.g., producing a short **text summary** [3, 10, 61, 106, 109]. 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 opinion mining applications, the quantitative side is crucial just like in the traditional survey research. In survey research, aspect-based summaries displayed as bar charts or pie charts are commonly used because they give the user a concise, quantitative and visual view. Instead of generating a text summary directly from input reviews, it is also possible to generate a text summary based on the mining results as displayed in Figs. 11.1 and 11.2. For example, one can generate natural language sentences based on what are shown in the bar charts using some predefined sentence templates. For instance, the first two bars in Fig. 11.2(B) can be summarized as "Most people are positive about cellular phone 1 and negative about cellular phone 2." Recently, researchers also tried to produce text summaries similar to that in Fig. 11.1 but in a more readable form [89, 97, 113].

11.2 Document Sentiment Classification

We are now ready to discuss some main research topics of opinion mining. This section focuses on **sentiment classification**, which has been studied extensively in the literature (see a survey in [91]). It classifies an opinion document (e.g., a product review) as expressing a positive or negative opinion or sentiment. The task is also commonly known as the **document-level sentiment classification** because it considers the whole document as the basic information unit.

Problem Definition: Given an opinionated document d evaluating an entity e, determine the opinion orientation oo on e, i.e., determine oo on aspect GENERAL in the quintuple (e, *GENERAL*, oo, h, t). e, h, and t are assumed known or irrelevant.

Assumption: Sentiment classification assumes that the opinion document d (e.g., a product review) expresses opinions on a single entity e and the opinions are from a single opinion holder h.

This assumption holds for customer reviews of products and services because each such review usually focuses on a single product and is written by a single reviewer. However, it may not hold for a forum and blog posting because in such a posting the author may express opinions on multiple products and compare them using comparative sentences.

Most existing techniques for document-level sentiment classification are based on supervised learning, although there are also some unsupervised methods. We give an introduction to them below.

11.2.1 Classification Based on Supervised Learning

Sentiment classification obviously can be formulated as a supervised learning problem with three classes, *positive*, *negative*, and *neutral*. Training and testing data used in the existing research are mostly product reviews, which is not surprising due to the above assumption. Since each review already has a reviewer-assigned rating (e.g., 1–5 stars), training and testing data are readily available. For example, a review with 4 or 5 stars is considered a positive review, a review with 1 or 2 stars is considered a negative review and a review with 3 stars is considered a neutral review.

Sentiment classification is similar to but also somewhat different from classic topic-based text classification, which classifies documents into predefined topic classes, e.g., politics, sciences, sports, etc. In topic-based classification, topic-related words are important. However, in sentiment classification, topic-related words are unimportant. Instead, **opinion words** (also called **sentiment words**) that indicate positive or negative opinions are important, e.g., *great*, *excellent*, *amazing*, *horrible*, *bad*, *worst*, etc.

Any existing supervised learning methods can be applied to sentiment classification, e.g., naïve Bayesian classification, and support vector machines (SVM). Pang et al. [94] took this approach to classify movie reviews into two classes, positive and negative. It was shown that using unigrams (a bag of individual words) as features in classification performed well with either naïve Bayesian or SVM.

Subsequent research used many more features and techniques in learning [91]. As most machine learning applications, the main task of sentiment classification is to engineer an effective set of features. Some of the example features used in research and possibly in practice are listed below.

Terms and their frequency. These features are individual words or word n-grams and their frequency counts (they are also commonly used in traditional topic-based text classification). In some cases, word positions may also be considered. The TF-IDF weighting scheme from information retrieval may be applied too. These features have been shown quite effective in sentiment classification.

Part of speech. It was found in many researches that adjectives are important indicators of opinions. Thus, adjectives have been treated as special features.

Opinion words and phrases. **Opinion words** are words that are commonly used to express positive or negative sentiments. For example, *beautiful*, *wonderful*, *good*, and *amazing* are positive opinion words, and *bad*, *poor*, and *terrible* are negative opinion words. Although many opinion words are adjectives and adverbs, nouns (e.g., *rubbish*, *junk*, and *crap*) and verbs (e.g., *hate* and *like*) can also indicate opinions. Apart from individual words, there are also **opinion phrases** and **idioms**, e.g., *cost someone an arm and a leg*. Opinion words and phrases are instrumental to sentiment analysis for obvious reasons.

Rules of opinions. Although opinion words and phrases are important, there are also many other expressions that contain no opinion words or phrases but indicate opinions or sentiments. We will list and discuss some of such expressions in Sect. 11.5.2.

Negations. Clearly negation words are important because their appearances often change the opinion orientation. For example, the sentence "I don't like this camera" is negative. However, negation words must be handled with care because not all occurrences of such words mean negation. For example, "not" in "not only … but also" does not change the orientation direction (see opinion shifters in Sect. 11.5.1).

Syntactic dependency. Words dependency-based features generated from parsing or dependency trees are also tried by several researchers.

Instead of using a standard machine learning method, researchers have also proposed several custom techniques specifically for sentiment classification, e.g., the score function in [15] based on words in positive and negative reviews, and the aggregation method in [117] using manually compiled domain-specific words and phrases.

Apart from classification of positive or negative sentiments, research has also been done on predicting the rating scores (e.g., 1–5 stars) of reviews [92]. In this case, the problem is formulated as regression since the rating scores are ordinal. Another interesting research direction is transfer learning or domain adaptation, as it has been shown that sentiment classification is highly sensitive to the domain from which the training data is extracted. A classifier trained using opinionated documents from one domain often performs poorly when it is applied or tested on opinionated documents from another domain. The reason is that words and even language constructs used in different domains for expressing opinions can be quite different. To make matters worse, the same word in one domain may mean positive but in another domain may mean negative. Thus, domain adaptation is needed. Existing research has used labeled data from one domain and unlabeled data from the target domain and general opinion words as features for adaptation [2, 6, 90, 133].

Table 11.1. Penn Treebank Part-Of-Speech (POS) tags

Tag	Description	Tag	Description
CC	Coordinating conjunction	PRP$	Possessive pronoun
CD	Cardinal number	RB	Adverb
DT	Determiner	RBR	Adverb, comparative
EX	Existential *there*	RBS	Adverb, superlative
FW	Foreign word	RP	Particle
IN	Preposition or subordinating conjunction	SYM	Symbol
JJ	Adjective	TO	*to*
JJR	Adjective, comparative	UH	Interjection
JJS	Adjective, superlative	VB	Verb, base form
LS	List item marker	VBD	Verb, past tense
MD	Modal	VBG	Verb, gerund or present participle
NN	Noun, singular or mass	VBN	Verb, past participle
NNS	Noun, plural	VBP	Verb, non-3rd person singular present
NNP	Proper noun, singular	VBZ	Verb, 3rd person singular present
NNPS	Proper noun, plural	WDT	Wh-determiner
PDT	Predeterminer	WP	Wh-pronoun
POS	Possessive ending	WP$	Possessive wh-pronoun
PRP	Personal pronoun	WRB	Wh-adverb

Table 11.2. Patterns of tags for extracting two-word phrases

	First word	Second word	Third word (not extracted)
1	JJ	NN or NNS	anything
2	RB, RBR, or RBS	JJ	not NN nor NNS
3	JJ	JJ	not NN nor NNS
4	NN or NNS	JJ	not NN nor NNS
5	RB, RBR, or RBS	VB, VBD, VBN, or VBG	anything

11.2.2 Classification Based on Unsupervised Learning

It is not hard to imagine that opinion words and phrases are the dominating indicators for sentiment classification. Thus, using unsupervised learning based on such words and phrases would be quite natural. The method in [119] is such a technique. It performs classification based on some fixed syntactic phrases that are likely to be used to express opinions.

The algorithm makes use of a natural language processing technique called **part-of-speech (POS) tagging**. The part-of-speech of a word is a linguistic category that is defined by its syntactic or morphological behavior. Common POS categories in English grammar are: noun, verb, adjective, adverb, pronoun, preposition, conjunction, and interjection. Then, there are many categories which arise from different forms of these categories. For example, a verb can be a verb in its base form, in its past tense, etc. In this book, we use the standard **Penn Treebank POS Tags** as shown in Table 11.1. POS tagging is the task of labeling (or tagging) each word in a sentence with its appropriate part of speech. For details on part-of-speech tagging, please refer to the report by Santorini [104]. The Penn Treebank site is at http://www.cis.upenn.edu/~treebank/home.html.

The algorithm given in [119] consists of three steps:

Step 1: It extracts phrases containing adjectives or adverbs as adjectives and adverbs are good indicators of opinions. However, although an isolated adjective may indicate opinion, there may be insufficient context to determine its opinion orientation (called **semantic orientation** in [119]). For example, the adjective "unpredictable" may have a negative orientation in an automotive review, in a phrase such as "unpredictable steering," but it could have a positive orientation in a movie review, in a phrase such as "unpredictable plot." Therefore, the algorithm extracts two consecutive words, where one member of the pair is an adjective or adverb, and the other is a context word.

Two consecutive words are extracted if their POS tags conform to any of the patterns in Table 11.2. For example, the pattern in line 2 means that two consecutive words are extracted if the first word is an adverb and the second word is an adjective but the third word (which is not extracted) cannot be a noun. NNP and NNPS are avoided so that the names of entities in the review cannot influence the classification.

Example 6: In the sentence "This camera produces beautiful pictures", "beautiful pictures" will be extracted as it satisfies the first pattern. ■

Step 2: It estimates the semantic orientation of the extracted phrases using the **pointwise mutual information** (PMI) measure given in Equation (1):

$$PMI(term_1, term_2) = \log_2\left(\frac{\Pr(term_1 \wedge term_2)}{\Pr(term_1)\Pr(term_2)}\right). \tag{1}$$

Here, $\Pr(term_1 \wedge term_2)$ is the co-occurrence probability of $term_1$ and $term_2$, and $\Pr(term_1)\Pr(term_2)$ gives the probability that the two terms co-occur if they are statistically independent. The ratio between $\Pr(term_1 \wedge term_2)$ and $\Pr(term_1)\Pr(term_2)$ is thus a measure of the degree of statistical dependence between them. The log of this ratio is the amount of information that we acquire about the presence of one of the words when we observe the other.

The semantic/opinion orientation (*SO*) of a phrase is computed based on its association with the positive reference word "excellent" and its association with the negative reference word "poor":

$$SO(phrase) = PMI(phrase, \text{"excellent"}) - PMI(phrase, \text{"poor"}). \qquad (2)$$

The probabilities are calculated by issuing queries to a search engine and collecting the number of *hits*. For each search query, a search engine usually gives the number of relevant documents to the query, which is the number of hits. Thus, by searching the two terms together and separately, we can estimate the probabilities in Equation (1). Turney, the author of [119], used the AltaVista search engine because it has a NEAR operator, which constrains the search to documents that contain the words within ten words of one another in either order. Let *hits*(*query*) be the number of hits returned. Equation (2) can be rewritten as:

$$SO(phrase) = \log_2\left(\frac{hits(\text{phrase } NEAR \text{ "excellent"})hits(\text{"poor"})}{hits(\text{phrase } NEAR \text{ "poor"})hits(\text{"excellent"})}\right). \qquad (3)$$

To avoid division by zero, 0.01 is added to the hits.

Step 3: Given a review, the algorithm computes the average *SO* of all phrases in the review and classifies the review as recommended if the average *SO* is positive, not recommended otherwise.

Final classification accuracies on reviews from various domains range from 84% for automobile reviews to 66% for movie reviews.

To summarize this section, we can see that the main advantage of document level sentiment classification is that it provides a prevailing opinion on an entity, topic or event. The main shortcomings are:

- It does not give details on what people liked and/or disliked. In a typical evaluative document such as a review, the author usually writes specific aspects of an entity that he/she likes or dislikes. The ability to extract such details is very useful in practice.
- It is not easily applicable to non-reviews, e.g., forum and blog postings, because many such postings evaluate multiple entities and compare them. Also, some of them may not be intended to be evaluations of products but may still contain a few opinion sentences. In such cases, these opinion sentences need to be identified and analyzed.

11.3 Sentence Subjectivity and Sentiment Classification

Naturally the same document-level sentiment classification techniques can also be applied to individual sentences. The task of classifying a sentence as subjective or objective is often called **subjectivity classification** in the existing literature [36, 102, 103, 127, 130, 131, 136]. The resulting subjective sentences are also classified as expressing positive or negative opinions, which is called **sentence-level sentiment classification**.

Problem Definition: Given a sentence *s*, two sub-tasks are performed:

1. *Subjectivity classification*. Determine whether *s* is a subjective sentence or an objective sentence
2. *Sentence-level sentiment classification*. If *s* is subjective, determine whether it expresses a positive, negative, or neutral opinion

Notice that the quintuple (*e*, *a*, *oo*, *h*, *t*) is not used in defining the problem here because sentence-level classification is often an intermediate step. In most applications, one needs to know what entities or aspects of the entities are the targets of opinions. Knowing that some sentences have positive or negative opinions, but not about what, is of limited use. However, the two sub-tasks of the sentence-level classification are still important because (1) it filters out those sentences which contain no opinions, and (2) after we know what entities and aspects of the entities are talked about in a sentence, this step can help us determine whether the opinions about the entities and their aspects are positive or negative.

Most existing researches study both problems, although some of them focus only on one. Both problems are classification problems. Thus, traditional supervised learning methods are again applicable. For example, one of the early works reported in [124] performed subjectivity classification using the naïve Bayesian classifier. Subsequent researches also used other learning algorithms.

One of the bottlenecks in applying supervised learning is the manual effort involved in annotating a large number of training examples. To save the manual labeling effort, a bootstrapping approach to label training data automatically was reported in [103]. The algorithm works by first using two high precision classifiers (HP-Subj and HP-Obj) to automatically identify some subjective and objective sentences. The high-precision classifiers use lists of lexical items (single words or *n*-grams) that are good subjectivity clues. HP-Subj classifies a sentence as subjective if it contains two or more strong subjective clues. HP-Obj classifies a sentence as objective if there are no strong subjective clues. These classifiers will give very high precision but low recall. The extracted sentences are then added to the training data to learn patterns. The patterns (which form the subjectivity classifiers in the next iteration) are then used to automatically identify more subjective and objective sentences, which are then added to the training set, and the next iteration of the algorithm begins.

For pattern learning, a set of syntactic templates are provided to restrict the kinds of patterns to be learned. Some example syntactic templates and example patterns are shown below.

Syntactic template	Example pattern
<subj> passive-verb	<subj> was satisfied
<subj> active-verb	<subj> complained
active-verb <dobj>	endorsed <dobj>
noun aux <dobj>	fact is <dobj>
passive-verb prep <np>	was worried about <np>

Before discussing algorithms which also perform sentiment classification of subjective sentences, let us point out an assumption made in much of the research on the topic.

Assumption of sentence-level sentiment classification: The sentence expresses a single opinion from a single opinion holder.

This assumption is only appropriate for simple sentences with a single opinion, e.g., "The picture quality of this camera is amazing." However, for compound and complex sentences, a single sentence may express more than one opinion. For example, the sentence, "The picture quality of this camera is amazing and so is the battery life, but the viewfinder is too small for such a great camera," expresses both positive and negative opinions (it has mixed opinions). For "picture quality" and "battery life," the sentence is positive, but for "viewfinder," it is negative. It is also positive for the camera as a whole (i.e., the GENERAL aspect).

In [136], a study was reported that identifies subjective sentences and also determines their opinion orientations. For subjectivity, it applied supervised learning. For sentiment classification of each subjective sentence, it used a similar method to that in Sect. 11.2.2 but with many more seed words, and the score function was log-likelihood ratio. The same problem was also studied in [36] considering gradable adjectives and in [27] using semi-supervised learning. In [54, 55, 57], researchers also built models to identify some specific types of opinions.

As we mentioned earlier, sentence-level classification is not suitable for compound and complex sentences. It was pointed out in [130] that not only a single sentence may contain multiple opinions but also both subjective and factual clauses. It is useful to pinpoint such clauses. It is also important to identify the strength of opinions. A study of automatic sentiment classification was presented to classify clauses of every sentence by the *strength* of the opinions being expressed in individual clauses, down to four levels deep (*neutral*, *low*, *medium*, and *high*). The strength of *neutral* indicates the absence of opinion or subjectivity. Strength classification thus subsumes the task of classifying a sentence as subjective vs. objective. In [129], the problem was studied further using supervised learning by considering contextual sentiment influencers such as negation (e.g., *not* and *never*) and contrary (e.g., *but* and *however*). A list of influencers can be found in [98]. However, in many cases, identifying only clauses are insufficient because the opinions can be embedded in phrases, e.g., "Apple is doing very well in this terrible economy." In this sentence, the opinion on "Apple" is clearly positive but on "economy" it is negative.

Finally, as pointed out in Sect. 11.1.1, we should bear in mind that not all subjective sentences have opinions and those that do form only a subset of opinionated sentences. Many objective sentences can imply opinions too. Thus, to mine opinions from text, one needs to mine them from both subjective and objective sentences.

11.4 Opinion Lexicon Expansion

In the preceding sections, we mentioned that opinion words are employed in many sentiment classification tasks. We now discuss how such words are generated. In the research literature, **opinion words** are also known as **polar words**, **opinion-bearing words**, and **sentiment words**. Positive opinion words are used to express some desired states while negative opinion words are used to express some undesired states. Examples of positive opinion words are *beautiful*, *wonderful*, *good*, and *amazing*. Examples of negative opinion words are *bad*, *poor*, and *terrible*. Apart from individual words, there are also opinion phrases and idioms, e.g., *cost someone an arm and a leg*. Collectively, they are called the **opinion lexicon**. They are instrumental for opinion mining for obvious reasons.

Opinion words can, in fact, be divided into two types, the **base type** and the **comparative type**. All the examples above are of the base type. Opinion words of the comparative type are used to express comparative and superlative opinions. Examples of such words are *better*, *worse*, *best*, *worst*, etc., which are comparative and superlative forms of their base adjectives or adverbs, e.g., *good* and *bad*. Unlike opinion words of the base type, the words of the comparative type do not express a direct opinion on an entity but a comparative opinion on more than one entity, e.g., "Car-x is better than Car-y." This sentence tells us something quite interesting. It does not express an opinion that any of the two cars is good or bad. It just says that compared to Car-y, Car-x is better, and compared to Car-x, Car-y is worse. Thus, although we still can assign a comparative word as positive or negative based on whether it represents a desirable or undesirable state, we may not use it in the same way as an opinion word of the base type. We will discuss this issue further when we study comparative sentences. This section focuses on opinion words of the base type.

To compile or collect the opinion word list, three main approaches have been investigated: manual approach, dictionary-based approach, and corpus-based approach. The manual approach is very time consuming and thus not usually used alone, but it is used when combined with automated approaches as the final check because automated methods make mistakes. Below, we discuss the two automated approaches.

Dictionary-based approach: One of the simple techniques in this approach is based on bootstrapping using a small set of seed opinion words and an online dictionary, e.g., WordNet [81]. The strategy is to first collect a small set of opinion words manually with known orientations and then to grow this set by searching in the WordNet for their synonyms and antonyms. The newly found words are added to the seed list. The next iteration starts. The iterative process stops when no more new words are found. This approach is used in [37, 55]. After the process completes, manual inspection can be carried out to remove and/or correct errors. Researchers have also used additional information (e.g., glosses) in WordNet and additional techniques (e.g., machine learning) to generate better lists [1, 21, 22, 50]. Several opinion word lists have been produced [17, 23, 37, 108, 124].

The dictionary-based approach and the opinion words collected from it have a major shortcoming. The approach is unable to find opinion words with domain and context-specific orientations, which is quite common. For example, for a speaker phone, if it is quiet, it is usually negative. However, for a car, if it is quiet, it is positive. The corpus-based approach can help deal with this problem.

Corpus-based approach and sentiment consistency: The methods in the corpus-based approach rely on syntactic or co-occurrence patterns and also a seed list of opinion words to find other opinion words in a large corpus. One of the key ideas is the one proposed by Hazivassiloglou and McKeown [35]. The technique starts with a list of seed opinion adjectives, and uses them and a set of linguistic constraints or conventions on connectives to identify additional adjective opinion words and their orientations. One of the constraints is about the conjunction AND, which says that conjoined adjectives usually have the same orientation. For example, in the sentence, "This car is beautiful *and* spacious," if "beautiful" is known to be positive, it can be inferred that "spacious" is also positive. This is so because people usually express the same opinion on both sides of a conjunction. The following sentence is rather unnatural, "This car is beautiful and difficult to drive." If it is changed to "This car is beautiful but difficult to drive," it becomes acceptable. Rules or constraints are also designed for other connectives, OR, BUT, EITHER–OR, and NEITHER–NOR. This idea is called **sentiment consistency**. Of course, in practice, it is not always consistent. Learning is applied to a large corpus to determine if two conjoined adjectives are of the same or different orientations. Same- and different-orientation links between adjectives form a graph. Finally, clustering is performed on the graph to produce two sets of words: positive and negative. In [51], Kanayama and Nasukawa expanded this approach by introducing the idea of intra-sentential (within a sentence) and inter-sentential (between neighboring sentences) sentiment consistency (called *coherency* in [44]). The intra-sentential consistency is similar to that in [35]. Inter-sentential consistency applies the idea to neighboring sentences. That is, the same opinion orientation (positive or negative) is usually expressed in a few consecutive sentences. Opinion changes are indicated by adversative expressions such as *but* and *however*. Some criteria to determine whether to add a word to the positive or negative lexicon are also proposed. This study was based on Japanese text. In Sect. 11.5.4, a related but also quite different method will be described. Other related work includes [48, 49].

In [17], Ding et al. explored the idea of intra-sentential and inter-sentential sentiment consistency further. Instead of finding domain-dependent opinion words, they showed that the same word could indicate different orientations in different contexts even in the same domain. This fact was also clearly depicted by the basic rules of opinions in Sect. 11.5.2. For example, in the digital camera domain, the word "long" expresses opposite opinions in the two sentences: "The battery life is *long*" (positive) and "The time taken to focus is *long*" (negative). Thus, finding domain-dependent opinion words is insufficient. They then proposed to consider both possible opinion words and aspects together and use the pair (*aspect*, *opinion_word*) as the **opinion context**, e.g., the pair ("battery life", "long"). Their method thus determines opinion words and their orientations together with the aspects that they modify. The above rules about connectives are still applied. The work in [28] adopted the same context definition but used it for analyzing comparative sentences. In fact, the method in [112, 119] can also be considered as a method for finding context-specific opinions, but it does not use the sentiment consistency idea. Its opinion context is based on syntactic POS patterns rather than aspects and opinion words that modify them. All these context definitions, however, are still insufficient as the basic rules of opinions discussed in Sect. 11.5.2 show, i.e., many contexts can be more complex, e.g., consuming a large amount of resources. In [7], the problem of extracting opinion expressions with any number of words was studied. The Conditional Random Fields (CRF) method [62] was used as the sequence learning technique for extraction.

Using the corpus-based approach alone to identify all opinion words, however, is not as effective as the dictionary-based approach because it is hard to prepare a huge corpus to cover all English words. However, as we mentioned above, this approach has a major advantage that the dictionary-based approach does not have. It can help find domain- and context-specific opinion words and their orientations using a domain corpus.

Finally, we should realize that populating an opinion lexicon (domain dependent or not) is different from determining whether a word or phrase is actually expressing an opinion and what its orientation is in a particular sentence. Just because a word or phrase is listed in an opinion lexicon does not mean that it actually is expressing an opinion in a sentence. For example, in the sentence, "I am looking for a good health insurance," "good" does not express either a positive or negative opinion on any particular insurance. The same is true for opinion orientation. We should also remember that opinion words and phrases are not the only expressions that bear opinions. There are many others as we will see in Sect. 11.5.2.

11.5 Aspect-Based Opinion Mining

Although classifying opinionated texts at the document level or at the sentence level is useful in many cases, it does not provide the necessary detail needed for many other applications. A positive opinionated document about a particular entity does not mean that the author has positive opinions on all aspects of the entity. Likewise, a negative opinionated document does not mean that the author dislikes everything. In a typical opinionated document, the author writes both positive and negative aspects of the entity, although the general sentiment on the entity may be positive or negative. Document and sentence sentiment classification does not provide such information. To obtain these details, we need to go to the aspect level. That is, we need the full model of Sect. 11.1.1, i.e., aspect-based opinion mining (also called feature-based opinion mining). Instead of treating opinion mining simply as a classification of sentiments, aspect-based opinion mining introduces a suite of problems which require deeper natural language processing capabilities and also produce a richer set of results.

Recall that, at the aspect level, the mining objective is to discover every quintuple (e_i, a_{ij}, oo_{ijkl}, h_k, t_l) in a given document d. To achieve the objective, five tasks need to be performed. This section mainly focuses on the following two core tasks and they have also been studied more extensively by researchers (in Sect. 11.7, we will briefly discuss some other tasks):

1. **Aspect extraction**: Extract aspects that have been evaluated. For example, in the sentence, "The picture quality of this camera is amazing," the aspect is "picture quality" of the entity represented by "this camera."

Note that “this camera” does not indicate the GENERAL aspect because the evaluation is not about the camera as a whole, but about its picture quality. However, the sentence “I love this camera” evaluates the camera as a whole, i.e., the GENERAL aspect of the entity represented by “this camera.” Bear in mind whenever we talk about an aspect, we must know which entity it belongs to. In our discussion below, we often omit the entity just for simplicity of presentation.

2. **Aspect sentiment classification**: Determine whether the opinions on different aspects are positive, negative, or neutral. In the first example above, the opinion on the “picture quality” aspect is positive, and in the second example, the opinion on the GENERAL aspect is also positive.

11.5.1 Aspect Sentiment Classification

We study the second task first, determining the orientation of opinions expressed on each aspect in a sentence. Clearly, the sentence-level and clause-level sentiment classification methods discussed in Sect. 11.3 are useful here. That is, they can be applied to each sentence or clause which contains some aspects. The aspects in it will take the opinion orientation of the sentence or clause. However, these methods have difficulty dealing with mixed opinions in a sentence and opinions that need phrase level analysis, e.g., “Apple is doing very well in this terrible economy.” Clause-level analysis also needs techniques to identify clauses which itself is a challenging task, especially with informal text of blogs and forum discussions, which is full of grammatical errors. Here, we describe a **lexicon-based approach** to solving the problem [17, 37], which tries to avoid these problems and has been shown to perform quite well. The extension of this method to handling comparative sentences is discussed in Sect. 11.6. In the discussion below, we assume that entities and their aspects are known. Their extraction will be discussed in Sects. 11.5.3, 11.5.4, and 11.7.

The lexicon-based approach basically uses an **opinion lexicon**, i.e., a list of *opinion words* and *phrases*, to determine the orientations of opinions in a sentence [17, 37]. It also considers **opinion shifters** and **but-clauses**. The approach works as follows:

1. **Mark opinion words and phrases**: Given a sentence that contains one or more aspects, this step marks all opinion words and phrases in the sentence. Each positive word is assigned the opinion score of +1 and each negative word is assigned the opinion score of –1. For example, we have the sentence, “The picture quality of this camera is not great, but the battery life is long.” After this step, the sentence is turned into “The *picture quality* of this camera is not **great**[+1], but the *battery life* is

long" because "great" is a positive opinion word. The aspects are italicized. Although "long" indicates positive for "battery life," we assume that it is not known. In fact, "long" can be regarded as a context-dependent opinion word, which we have discussed in Sect. 11.4.

2. **Handle opinion shifters:** Opinion shifters (also called **valence shifters** [98]) are words and phrases that can shift or change opinion orientations. Negation words like *not*, *never*, *none*, *nobody*, *nowhere*, *neither*, and *cannot* are the most common type. In our example, this step turns the above sentence into "The *picture quality* of this camera is not **great**[-1], but the *battery life* is long" due to the negation word "not." Besides negation words, many other types of words and phrases can also alter opinion orientations. For example, some **modal auxiliary verbs** (e.g., *would*, *should*, *could*, *might*, *must*, and *ought*) are another type of opinion shifters, e.g., "The brake could be improved." So are some presuppositional items. This case is typical for adverbs like *barely* and *hardly* as shown by comparing "It works" with "It hardly works." "Works" indicates positive, but "hardly works" does not: it presupposes that better was expected. Words like *fail*, *omit*, *neglect* behave similarly, e.g., "This camera fails to impress me." Additionally, sarcasm changes orientation too, e.g., "What a great car, it failed to start the first day." Although it is easy to recognize such shifters manually, spotting them and handling them correctly in actual sentences by an automated system is by no means easy. Furthermore, not every appearance of an opinion shifter changes the opinion orientation, e.g., "not only … but also." Such cases need to be dealt with carefully.
3. **Handle but-clauses**: In English, *but* means *contrary*. A sentence containing *but* is handled by applying the following rule: the opinion orientation before *but* and after *but* are opposite to each other if the opinion on one side cannot be determined. After this step, the above sentence is turned into "The *picture quality* of this camera is not **great**[-1], but the *battery life* is *long*[+1]" due to "but" (note that [+1] is added at the end of the but-clause). Apart from *but*, phrases such as "with the exception of," "except that," and "except for" behave similarly to *but* and are handled in the same way. As in the case of negation, not every *but* means contrary, e.g., "not only … but also." Such non-but phrases containing "but" also need to be considered separately. Finally, we should note that contrary words and phrases do not always indicate an opinion change, e.g., "Car-x is great, but Car-y is better." Such cases need to be identified and dealt with separately.
4. **Aggregating opinions**: This step applies an opinion aggregation function to the resulting opinion scores to determine the final orientation of the opinion on each aspect in the sentence. Let the sentence be s, which

contains a set of aspects $\{a_1, \ldots, a_m\}$ and a set of opinion words or phrases $\{ow_1, \ldots, ow_n\}$ with their opinion scores obtained from steps 1, 2, and 3. The opinion orientation for each aspect a_i in s is determined by the following opinion aggregation function:

$$score(a_i, s) = \sum_{ow_j \in s} \frac{ow_j.oo}{dist(ow_j, a_i)}, \tag{5}$$

where ow_j is an opinion word/phrase in s, $dist(ow_j, a_i)$ is the distance between aspect a_i and opinion word ow_j in s. $ow_j.oo$ is the opinion score of ow_i. The multiplicative inverse is used to give lower weights to opinion words that are far away from aspect a_i. If the final score is positive, then the opinion on aspect a_i in s is positive. If the final score is negative, then the opinion on the aspect is negative. It is neutral otherwise.

This simple algorithm can perform quite well in many cases, but it is not sufficient in others. One main shortcoming is that opinion words and phrases do not cover all types of expressions that convey or imply opinions. There are in fact many other possible opinion bearing expressions. Most of them are also harder to deal with. Below, we list some of them, which we call the **basic rules of opinions**.

11.5.2 Basic Rules of Opinions

An opinion rule expresses a concept that implies a positive or negative opinion. In actual sentences, the concept can be expressed in many different ways in natural language. We present these rules using a formalism similar to the BNF form. The top level rules are as follows:

```
1.  POSITIVE  ::= P
2.              | PO
3.              | orientation shifter N
4               | orientation shifter NE
5.  NEGATIVE  ::= N
6.              | NE
7.              | orientation shifter P
8.              | orientation shifter PO
```

The non-terminals P and PO represent two types of **positive opinion expressions**. The non-terminal N and NE represent two types of **negative opinion expressions**. “opinion shifter N” and ”opinion shifter NE” represent the negation of N and NE, respectively, and “opinion shifter P” and “opinion shifter PO” represent the negation of P and PO, respectively. We can see that these are not expressed in the actual BNF form but a pseudo language stating some concepts. The reason is that we are unable to specify them precisely because for example, in an actual sentence, the opinion shifter may be in any form and can appear before or after N, NE, P, or PO. POSITIVE and NEGATIVE are the final orientations used to determine the opinions on the aspects in a sentence.

We now define N, NE, P, and PO, which contain no opinion shifters. These opinion expressions are grouped into six conceptual categories based on their characteristics.

1. *Opinion word or phrase*: This is the most commonly used category, in which opinion words or phrases alone can imply positive or negative opinions on aspects, e.g., "great" in "The picture quality is great." These words or phrases are reduced to P and N.

   9. P ::= a positive opinion word or phrase
   10. N ::= an negative opinion word or phrase

   Again, the details of the right-hand sides are not specified (which also apply to all the subsequent rules). It is assumed that a set of positive and negative opinion words/phrases exists for an application.
2. *Desirable or undesirable fact*: In this case, it is a factual statement, and the description uses no opinion words, but in the context of the entity, the description implies a positive or negative opinion. For example, the sentence "After my wife and I slept on it for two weeks, I noticed a mountain in the middle of the mattress" indicates a negative opinion about the mattress. However, the word "mountain" itself does not carry any opinion. Thus, we have the following two rules:

   11. P ::= desirable fact
   12. N ::= undesirable fact

3. *High, low, increased and decreased quantity of a positive or negative potential item*: For some aspects, a small value/quantity of them is negative, and a large value/quantity of them is positive, e.g., "The battery life is short" and "The battery life is long." We call such aspects **positive potential items** (**PPI**). Here "battery life" is a positive potential item. For some other aspects, a small value/quantity of them is positive, and a large value/quantity of them is negative, e.g., "This phone costs a lot" and "Sony reduced the price of the camera." We call such aspects **negative potential items** (**NPI**). "cost" and "price" are negative potential items. Both positive and negative potential items themselves express no opinions, i.e., "battery life" and "cost", but when they are modified by quantity adjectives or quantity change words or phrases, positive or negative opinions are implied. The following rules cover these cases:

13. PO ::= no, low, less or decreased quantity of NPI
14. | large, larger, or increased quantity of PPI
15. NE ::= no, low, less, or decreased quantity of PPI
16. | large, larger, or increased quantity of NPI
17. NPI ::= a negative potential item
18. PPI ::= a positive potential item

4. *Decreased and increased quantity of an opinionated item* (N and P): This set of rules is similar to the negation rules 3, 4, 7, and 8 above. Decreasing or increasing the quantity associated with an opinionated item (often nouns and noun phrases) can change the orientation of the opinion. For example, in the sentence "This drug reduced my pain significantly," "pain" is a negative opinion word, and the reduction of "pain" indicates a desirable effect of the drug. Hence, decreased pain implies a positive opinion on the drug. The concept of *decreasing* also extends to *removal* and *disappearance*, e.g., "My pain has disappeared after taking the drug." We then have the following rules:

   19. PO ::= less or decreased N
   20. | more or increased P
   21. NE ::= less or decreased P
   22. | more or increased N

   Rules 20 and 22 may not be needed as there is no change of opinion orientation, but they can change the opinion intensity. The key difference between this set of rules and the rules in the previous category is that no opinion words or phrases are involved in the previous category.
5. *Deviation from the norm or some desired value range*: In some application domains, the value of an aspect may have a desired range or norm. If it is above or below the normal range, it is negative, e.g., "This drug causes low (or high) blood pressure" and "This drug causes my blood pressure to reach 200." Notice that no opinion word appeared in these sentences. We have the following rules:

   23. PO ::= within the desired value range
   24. NE ::= above or below the desired value range

6. *Producing and consuming resources and wastes*: If an entity produces a lot of resources, it is positive. If it consumes a lot of resources, it is negative. For example, *water* is a resource. The sentence, "This washer uses a lot of water" gives a negative opinion about the washer. Likewise, if an entity produces a lot of wastes, it is negative. If it consumes a lot of wastes, it is positive. These give us the following rules:

25. PO ::= produce a large quantity of or more resource
26. | produce no, little or less waste
27. | consume no, little or less resource
28. | consume a large quantity of or more waste
29. NE ::= produce no, little or less resource
30. | produce some or more waste
31. | consume a large quantity of or more resource
32. | consume no, little or less waste

It should be stressed again that these are conceptual rules. They can be expressed in many ways using different words or phrases in actual sentences, and in different domains they may also manifest in different forms. These rules also show the difficulty of opinion mining because recognizing them in different domains are highly challenging.

We should also note that these rules may not be the only rules that govern expressions of positive and negative opinions. With further research, additional new rules may be discovered and the existing rules may be revised. It is also important to know that any manifestation of such rules in text does not always imply opinions. In other words, just because a rule is satisfied in a sentence does not mean that it actually expresses an opinion. For example, "I want a reliable car" does not express an opinion on any specific car, although the positive opinion word "reliable" appeared.

11.5.3 Aspect Extraction

Existing research on aspect extraction (more precisely, *aspect expression extraction*) is mainly carried out in online reviews. We thus focus on reviews here. There are two common review formats on the Web.

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. An example of such a review is given in Fig. 11.3.

Format 2 – **Free format:** The reviewer can write freely, i.e., no separation of pros and cons. An example of such a review is given in Fig. 11.4.

To extract aspects from Pros and Cons in reviews of Format 1 (not the detailed review, which is the same as that in Format 2), many information extraction techniques can be applied, e.g., Conditional Random Fields (CRF) [62], and Hidden Markov Models (HMM) [25], and mining sequential rules [72]. 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*,

My SLR is on the shelf
by camerafun4. Aug 09 ‘04
Pros: Great photos, easy to use, very small
Cons: Battery usage; included memory is stingy.
I had never used a digital camera prior to purchasing this Canon A70. I have always used a SLR … **Read the full review**

Fig. 11.3. An example of a review of format 1.

GREAT Camera., Jun 3, 2004
Reviewer: **jprice174** from Atlanta, Ga.
I did a lot of research last year before I bought this camera... It kinda hurt to leave behind my beloved nikon 35mm SLR, but I was going to Italy, and I needed something smaller, and digital.
The pictures coming out of this camera are amazing. The 'auto' feature takes great pictures most of the time. And with digital, you're not wasting film if the picture doesn't come out. …

Fig. 11.4. An example of a review of format 2.

but, etc. This observation helps the extraction algorithm to perform more accurately, see [72]. 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 complete sentences. The above algorithm can also be applied. However, experiments have shown that it is not as effective because complete sentences are more complex and contain a large amount of noise. Below, we describe some unsupervised methods for finding *explicit aspect expressions* that are nouns and noun phrases. The first method is due to [37]. The method requires a large number of reviews and consists of two steps:

1. Find frequent nouns and noun phrases: Nouns and noun phrases (or groups) are identified by a POS tagger. Their occurrence frequencies are counted, and only the frequent ones are kept. A frequency threshold can be 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 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.
2. Find infrequent aspects by exploiting the relationships between aspects and opinion words: The above step can miss many genuine aspect ex-

pressions 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.

This idea of using the modifying relationship of opinion words and aspects to extract aspects was later generalized to using dependency relations [146], which was further developed into the double-propagation method for simultaneously extracting both opinion words and aspects [101]. The double-propagation method will be described in Sect. 11.5.4.

The precision of step 1 of the above algorithm was improved in [99]. Their algorithm tries to remove those noun phrases that may not be product aspects/features. It evaluates each noun phrase by computing a pointwise mutual information (PMI) score between the phrase and some *meronymy discriminators* associated with the product class, e.g., a scanner class. The meronymy discriminators for the scanner class are, "of scanner," "scanner has," "scanner comes with," etc., which are used to find components or parts of scanners by searching on the Web. The PMI measure is a simplified version of that in Sect. 11.2.2.

$$PMI(a,d) = \frac{hits(a \wedge d)}{hits(a)hits(d)}, \tag{4}$$

where a is a candidate aspect identified in step 1 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 of the product because a and d do not co-occur frequently. The algorithm also distinguishes components/parts from attributes using WordNet's *is-a* hierarchy (which enumerates different kinds of properties) and morphological cues (e.g., "-iness," "-ity" suffixes).

Other related works on aspect extraction use existing knowledge, supervised learning, semi-supervised learning, topic modeling, and clustering. Several researchers also explored the idea of jointly modeling both aspects and opinions. Lu et al. [75] exploited an ontology (*Freebase*: http://www.freebase.com) to obtain aspects to a topic and used them to organize scattered opinions to generate structured summaries. Su et al. [111] proposed a clustering method with mutual reinforcement to identify implicit aspects. In [30], semi-supervised learning and domain knowledge are employed.

If reviews with manually annotated aspects and opinion expressions are available, standard supervised learning methods can be applied. Jin and Ho [41] applied a lexicalized hidden Markov model to learn patterns to extract aspects and opinion expressions. Jakob and Gurevych [39] used conditional random fields. Wu et al. [132] used dependency tree kernels.

Topic modeling methods have been attempted as an unsupervised and knowledge-lean approach. Titov and McDonald [116] showed that global topic models such as LDA (Latent Dirichlet allocation [5]) might not be suitable for detecting rateable aspects. They proposed multigrain topic models to discover local rateable aspects. Here, each discovered aspect is a unigram language model, i.e., a multinomial distribution over words. Such a representation is thus not as easy to interpret as aspects extracted by previous methods, but its advantage is that different words expressing the same or related aspects (more precisely aspect expressions) can usually be automatically grouped together under the same aspect. However, Titov and McDonald [116] did not separate aspects and opinion words in the discovery. Lin and He [69] proposed a joint topic-sentiment model also by extending LDA, where aspect words and opinion words were still not explicitly separated. To separate aspects and opinion words using topic models, Mei et al. [80] proposed to use a positive sentiment model and a negative sentiment model in additional to aspect models. Brody and Elhadad [8] proposed to first identify aspects using topic models and then identify aspect-specific opinion words by considering adjectives only. Zhao et al. [145] 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. Topic modeling-based approaches were also used by Liu et al. [74] and Lu et al. [76].

Another line of work is to associate aspects with opinion/sentiment ratings. It aims to predict ratings based on learned aspects or jointly model aspects and ratings. Titov and McDonald [115] proposed a statistical model that is able to discover aspects from text and to extract textual evidence from reviews supporting each aspect rating. Lu et al. [77] defined a problem of rated aspect summarization. They proposed to use the structured probabilistic latent semantic analysis method to learn aspects from a bag of phrases and a local/global method to predict aspect ratings. Wang et al. [121] proposed to infer both aspect ratings and aspect weights at the level of individual reviews based on learned latent aspects.

11.5.4 Simultaneous Opinion Lexicon Expansion and Aspect Extraction

In [100, 101], a method was proposed to extract both opinion words and aspects simultaneously by exploiting certain syntactic relations of opinion words and aspects. Although it was originally designed to work with product reviews, a reimplementation and extension of it has been applied on Twitter data, forum discussions, and blog postings. It has also been successfully used to analyze Chinese online discussions [139]. The method needs only an initial set of opinion word seeds as the input and no seed aspects are required. It is based on the observation 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 was found that opinion words have relations among themselves and so do targets. 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** [114] was adopted to describe these relations.

The algorithm uses only 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 nouns or noun phrases. Thus, the potential POS tags for opinion words are *JJ* (adjectives), *JJR* (comparative adjectives), and *JJS* (superlative adjectives), while those for aspects are *NN* (singular nouns) and *NNS* (plural nouns). The dependency relations describing relations between opinion words and aspects include *mod*, *pnmod*, *subj*, *s*, *obj*, *obj2*, and *desc*, while the relations for opinion words and aspects themselves contain only the conjunction relation *conj*. We use 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 $\langle \mathrm{POS}(w_i), \mathrm{R}, \mathrm{POS}(w_j) \rangle$, where $\mathrm{POS}(w_i)$ is the POS tag of word w_i and R is the relation. The values of $\mathrm{POS}(w_i)$ and R were listed above.

	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 good "screen".* *good→**mod**→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 best mp3 player.* *best→**mod**→player←**subj**←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 audio and "video"?* *video→**conj**→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 len.* *len→**obj**→has←**subj**←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 amazing and "easy" to use.* *easy→**conj**→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 sexy, "cool", accessory-available mp3 player, you can choose iPod.* *sexy→**mod**→player←**mod**←cool*

Table 11.3. Rules for aspect and opinion word extraction.
Column 1 is the rule ID, column 2 is the observed relation (line 1) and the constraints that it must satisfy (lines 2 – 4), 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.

The extraction process uses a rule-based approach with the relations defined above. 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 ⟨*JJ, mod, NNS*⟩. 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. The propagation performs four subtasks:

1. extracting aspects using opinion words
2. extracting aspects using the extracted aspects
3. extracting opinion words using the extracted aspects
4. extracting opinion words using both the given and the extracted opinion words

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 11.3. 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 between 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, we employ $R1_i$ 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. For example, we have the sentence "The phone has good screen" whose corresponding dependency tree is in Fig. 11.5. If we know that "good" is an opinion word, and it depends on "screen" through *mod* which is contained in $\{MR\}$ and "screen" is tagged as *NN*, $R1_1$ can be applied to extract "screen" as an aspect.

It should be noted that although this method only finds noun aspect words and adjective opinion words, it can be extended to aspects and opinion words of other parts-of-speech by adding more dependency relations. It also has a method to join words to form aspect phrases and a method to determine the opinion orientations of the extracted opinion words.

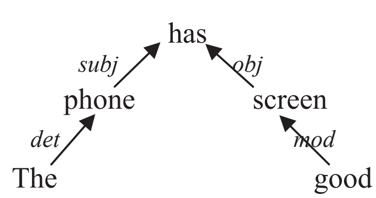


Fig. 11.5: The dependency tree for "The phone has good screen"

11.6 Mining Comparative Opinions

Directly or indirectly expressing positive or negative opinions about an entity and its aspects is only one form of evaluation. Comparing the entity with some other similar entities is another. **Comparisons** are related to but are also quite different from regular opinions. They not only have different semantic meanings but also different syntactic forms. For example, a typical regular opinion sentence is "The picture quality of this camera is great," and a typical comparative sentence is "The picture quality of Camera-x is better than that of Camera-y." This section first defines the problem and then presents some existing methods to solve it [18, 28, 42, 43].

11.6.1 Problem Definitions

In general, a comparative sentence expresses a relation based on similarities or differences of more than one entity. The comparison is usually conveyed using the *comparative* or *superlative* form of an adjective or adverb. A comparative sentence typically states that one entity has more or less of a certain attribute than another entity. A superlative sentence typically states that one entity has the most or least of a certain attribute among a set of similar entities. In general, a comparison can be between two or more entities, groups of entities, and one entity and the rest of the entities. It can also be between an entity and its previous versions.

Two types of comparatives: In English, comparatives are usually formed by adding the suffix -*er* and superlatives are formed by adding the suffix -*est* to their **base adjectives** and **adverbs**. For example, in "The battery life of Camera-x is longer than that of Camera-y," "longer" is the comparative form of the adjective "long." In "The battery life of this camera is the longest," "longest" is the superlative form of the adjective "long". We call this type of comparatives and superlatives **Type 1 comparatives and superlatives**. Note that for simplicity, we often use *comparative* to mean both *comparative* and *superlative* if superlative is not explicitly stated.

Adjectives and adverbs with two syllables or more and not ending in *y* do not form comparatives or superlatives by adding -*er* or -*est*. Instead, *more*, *most*, *less*, and *least* are used before such words, e.g., *more beautiful*. We call this type of comparatives and superlatives **Type 2 comparatives** and **Type 2 superlatives**. Both Type 1 and Type 2 are called **regular comparatives and superlatives**.

In English, there are also **irregular comparatives** and **superlatives**, i.e., *more*, *most*, *less*, *least*, *better*, *best*, *worse*, *worst*, *further/farther* and *furthest/farthest*, which do not follow the above rules. However, they behave similarly to Type 1 comparatives and are thus grouped under Type 1.

Apart from these standard comparatives and superlatives, many other words or phrases can also be used to express comparisons, e.g., *prefer* and *superior*. For example, the sentence, "Camera-x's quality is superior to Camera-y," says that "Camera-x" is better or preferred. In [42], Jindal and Liu identified a list of such words. Since these words behave similarly to Type 1 comparatives, they are also grouped under Type 1.

Types of comparative relations: Comparative relations or comparisons can be grouped into four main types. The first three types are called the **gradable comparisons** and the last one the **non-gradable comparisons**.

1. *Non-equal gradable comparisons*: Relations of the type *greater* or *less than* that express an ordering of some entities with regard to some of their shared aspects, e.g., "The Intel chip is faster than that of AMD". This type also includes user preferences, e.g., "I prefer Intel to AMD."
2. *Equative comparisons*: Relations of the type *equal to* that state two or more entities are equal with regard to some of their shared aspects, e.g., "The performance of Car-x is about the same as that of Car-y."
3. *Superlative comparisons*: Relations of the type *greater* or *less than all others* that rank one entity over *all* others, e.g., "The Intel chip is the fastest."
4. *Non-gradable comparisons*: Relations that compare aspects of two or more entities, but do not grade them. There are three main sub-types:
   - Entity A is similar to or different from entity B with regard to some of their shared aspects, e.g., "Coke tastes differently from Pepsi."
   - Entity A has aspect a_1, and entity B has aspect a_2 (a_1 and a_2 are usually substitutable), e.g., "Desktop PCs use external speakers but laptops use internal speakers."
   - Entity A has aspect a, but entity B does not have, e.g., "Phone-x has an earphone, but Phone-y does not have."

Comparative words used in non-equal gradable comparisons can be further categorized into two groups according to whether they express increased or decreased quantities, which are useful in opinion analysis.

- *Increasing comparatives*: Such a comparative expresses an increased quantity, e.g., *more* and *longer*.
- *Decreasing comparatives:* Such a comparative expresses a decreased quantity, e.g., *less* and *fewer*.

Objective of mining comparative opinions: Given a collection of opinionated documents D, discover in D all comparative opinion sextuples of the form:

$$(E_1, E_2, A, PE, h, t),$$

where E_1 and E_2 are the entity sets being compared based on their shared aspects A (entities in E_1 appear before entities in E_2 in the sentence), PE ($\in \{E_1, E_2\}$) is the preferred entity set of the opinion holder h, and t is the time when the comparative opinion is expressed.

Example 7: Consider the comparative sentence "*Canon's optics is better than those of Sony and Nikon*," written by John in 2010. The extracted comparative opinion is:

({Canon}, {Sony, Nikon}, {optics}, *preferred*:{Canon}, John, 2010)

The entity set E_1 is {Canon}, the entity set E_2 is {Sony, Nikon}, their shared aspect set A being compared is {optics}, the preferred entity set is {Canon}, the opinion holder h is John, and the time t when this comparative opinion was written is 2010. ■

To mine comparative opinions, the tasks of extracting entities, aspects, opinion holders, and times are the same as those for mining regular opinions. In [43], a method based on label sequential rules (LSR) is proposed to extract entities and aspects that are compared. A similar approach is described in [66] for extracting the compared entities. Clearly, the approaches discussed in previous sections are applicable as well, and so are many other information extraction methods. Below, we only focus on studying two comparative opinion mining specific problems, i.e., identifying comparative sentences and identifying the preferred entity set.

11.6.2 Identification of Comparative Sentences

Although most comparative sentences contain comparative adjectives and comparative adverbs, e.g., *better,* and *longer*, many sentences that contain such words are not comparative sentences, e.g., "I cannot agree with you more." Similarly, many sentences that do not contain such indicators are comparative sentences (usually non-gradable), e.g., "Cellphone-x has Bluetooth, but Cellphone-y does not have."

It was shown in [42] that almost every comparative sentence has a keyword or a key phrase indicating comparison. Using a set of 83 keywords and key phrases, 98% of the comparative sentences (recall = 98%) were identified with a precision of 32% using the authors' data set. The keywords and key phrases are:

1. Comparative adjectives (JJR) and comparative adverbs (RBR), e.g., *more*, *less*, *better*, and words ending with *-er*.
2. Superlative adjectives (JJS) and superlative adverbs (RBS), e.g., *most*, *least*, *best*, and words ending with *-est*.

3. Other indicative words and phrases such as *same*, *similar*, *differ*, *as well as*, *favor*, *beat*, *win, exceed*, *outperform, prefer*, *ahead, than*, *superior*, *inferior*, *number one, up against*, etc.

Since keywords and key phrases alone are able to achieve a high recall, the set of keywords and key phrases can be used to filter out those sentences that are unlikely to be comparative sentences. We can then improve the precision of the remaining set of sentences.

It was also observed in [42] that comparative sentences have strong patterns involving comparative keywords, which is not surprising. These patterns can be used as features in machine learning. To discover these patterns, class sequential rule (CSR) mining was used in [42]. Class sequential rule mining is a sequential pattern mining method (see Sect. 2.9.3). Each training example used for mining CSRs is a pair (s_i, y_i), where s_i is a sequence and y_i is a class, e.g., $y_i \in$ {*comparative*, *non-comparative*}. The sequence is generated from a sentence. Instead of using each full sentence, only words near a comparative keyword are used to generate each sequence. Each sequence is also labeled with a class indicating whether the sentence is a comparative sentence or not. Using the training data, CSRs can be generated. The detailed mining algorithm can be found in Sect. 2.9.3.

For classification model building, the left-hand side sequence patterns of the CSR rules with high conditional probabilities were used as features in [42]. Each sentence produces a vector. If the sentence matches a pattern, the feature value for the pattern of the vector is set to 1, and otherwise it is set to 0. Bayesian classification was employed for model building.

Classifying comparative sentences into four types: After comparative sentences are identified, we also want to classify them into the four types or classes, *non-equal gradable*, *equative*, *superlative*, and *non-gradable*. For this task, [42] showed that keywords and key phrases were already sufficient, i.e., the set of keywords and key phrases were used as features for machine learning. SVM was shown to give the best results.

11.6.3 Identification of Preferred Entities

Similar to opinion mining of normal sentences, opinion mining of comparative sentences also needs to determine whether a comparative sentence is opinionated or not. However, unlike normal sentences, it does not make much sense to apply sentiment classification to a comparative sentence as a whole because an opinionated comparative sentence does not express a direct positive or negative opinion. Instead, it compares multiple entities by ranking the entities based on their shared aspects to give a *comparative* *opinion*. In other words, it presents a preference order of the entities based on the comparison of some of their shared aspects. Since most comparative sentences compare two sets of entities, analysis of an opinionated comparative sentence means identifying the preferred entity set. However, for application purposes, one may assign positive opinions to the aspects of the entities in the preferred set, and negative opinions to the aspects of the entities in the not preferred set. Since little research has been done on classifying whether a comparative sentence is opinionated or not, below we only describe a method for identifying the preferred entity set [28].

The approach bears some resemblance to the lexicon-based approach to identifying opinion orientations on aspects. Thus, it needs opinion words used for comparisons. Similar to opinion words of the base type, these opinion words of comparative type can be divided into two categories.

1. *Comparative opinion words*: For Type 1 comparatives, this category includes words like *better*, *worse*, etc., which have explicit opinions. In sentences involving such words, it is usually easy to determine which entity set is the preferred one of the sentence author.

   In the case of Type 2 comparatives, formed by adding *more*, *less*, *most*, and *least* before adjectives/adverbs, the preferred entity set is determined by both words. The following rules are applicable:

Comparative Negative	::=	increasing comparative N
	|	decreasing comparative P
Comparative Positive	::=	increasing comparative P
	|	decreasing comparative N

   An **increasing comparative** expresses an increased value of a quantity, e.g., "*more*," and "*longer*," and a **decreasing comparative** expresses a decreased value of a quantity, e.g., "*less*," and "*fewer*." The first rule above says that the combination of an increasing comparative (e.g., *more*) and a negative opinion word (e.g., *awful*) implies a Type 2 **comparative negative** opinion (on the left). The other rules have similar meanings. P (respectively N) denotes a positive (negative) opinion word or phrase of the base type. In fact, the above four rules are already covered by the basic rules of opinions in Sect. 11.5.2.
2. *Context-dependent comparative opinion words*: In the case of Type 1 comparatives, such words include *higher*, *lower*, etc. For example, "Car-x has higher mileage per gallon than Car-y" carries a positive sentiment on "Car-x" and a negative sentiment on "Car-y" comparatively, i.e., "Car-x" is preferred. However, without domain knowledge it is hard to know whether "higher" is positive or negative. The combination of "higher" and "mileage" with the domain knowledge tells us that "higher mileage" is desirable. Again, these cases are already included in

the basic rules of opinions in Sect. 11.5.2. In this case, "mileage" is a *positive potential item*.

In the case of Type 2 comparatives, the situation is similar. However, in this case, the comparative word (*more*, *most*, *less* or *least*), the adjective/adverb, and the aspect are all important in determining the opinion or preference. If we know whether the comparative word is increasing or decreasing (which is easy since there are only four of them), then the opinion can be determined by applying the four rules in (1).

As discussed in Sect. 11.4, the pair (*aspect*, *opinion_word*) forms an opinion context. To determine whether a pair is positive or negative, the algorithm in [28] resorts to the external information, i.e., a large corpus of Pros and Cons from product reviews. It basically determines whether the *aspect* and *opinion_word* are more associated with each other in Pros or in Cons. If they are more associated in Pros, *opinion_word* is positive. Otherwise, it is negative. Using Pros and Cons is natural because they are short phrases and thus have little noise, and their opinion orientations are also known.

Due to the observation below, we can obtain comparative opinion words by simply converting opinion adjectives/adverbs of the base form to their comparative forms, which can be done automatically based on the English comparative formation rules described earlier and the WordNet.

Observation: If an adjective or adverb of the base form is positive (or negative), then its comparative or superlative form is also positive (or negative), e.g., *good*, *better*, and *best*.

After the conversion, these words are manually categorized into increasing and decreasing comparatives.

Once all the information is available, determining which entity set is preferred is relatively simple. Without negation, if the comparative is positive (or negative), then the entities before (or after) *than* is preferred. Otherwise, the entities after (or before) *than* are preferred. Additional details can be found in [28]. In [24], Fiszman et al. also studied the problem of identifying which entity has more of certain aspects in comparative sentences in biomedical texts, but it does not analyze opinions.

11.7 Some Other Problems

Besides the problems discussed in previous sections, there are still many other challenges in opinion mining. This section gives an introduction to some of them. As we will see, most of these problems are related to their general problems that have been studied before but the opinion text provides more clues for their solutions and also has additional requirements.

Entity, opinion holder, and time extraction: In some applications, it is useful to identify and extract opinion holders and the times when opinions are given. As we mentioned earlier, opinion holders are more useful for news articles or other types of formal documents in which the persons or organizations who expressed opinions are stated explicitly in the text. Such holders need to be identified by the system, and so do the dates and times when opinions are expressed. These extraction tasks are collectively called Named Entity Recognition (NER). They have been studied extensively in the literature. See a comprehensive survey of information extraction tasks and algorithms in [105].

In the case of social media on the Web, the opinion holders are often the authors of the discussion postings, bloggers, or reviewers, whose login ids are known although their true identities in the real world may be unknown. The date and time when an opinion is submitted are also known and displayed on the page, so their extraction is easy (see Chap. 9 on how to extract them).

Entity name extraction is also a NER problem, but there is a difference here. In a typical opinion mining application, the user wants to find opinions on some competing entities, e.g., competing products or brands. However, he/she often can only provide a few names because there are so many different brands and models. Furthermore, Web users also write names of the same product brands in many ways. For example, "Motorola" may be written as "Moto" or "Mot," and "Samsung" may be written as "Sammy." Product model names have even more variations. It is thus important for a system to automatically discover them from a relevant corpus (e.g., blogs and forum discussions). The key requirement of this discovery is that the discovered entities must be of the same class/type as entities provided by the user (e.g., phone brands and models).

In [67], this problem was modeled a **set expansion problem** [29, 95], which expands a set of given seed entities (e.g., product names). Formally, the problem is stated as follows: 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. Although this is a classification problem, in practice, the problem is often solved as a ranking problem, i.e., to rank the entities in D based on their likelihoods of belonging to C.

The classic methods for solving this problem in NLP are based on distributional similarity [63, 95]. The approach works by comparing the similarity of the surround words distributions of each candidate entity with the seed entities and then ranking the candidate entities based on the similarity values. However, it was shown in [67] that this approach is inaccurate, and PU learning based on S-EM (Chap. 5) performed considerably better based on the results from 10 corpora. To apply PU learning, the given seeds are used to automatically extract some positive examples, which are sentence segments that contain one of the seed product names. The rest of the sentences are treated as unlabeled examples. Additionally, it was also shown that S-EM outperforms the machine learning technique *Bayesian Sets* [29], which was designed specifically for set expansion.

Grouping aspect expressions indicating the same aspects: It is common that people use different words or phrases (which are called aspect expressions in Sect. 11.1) to describe the same aspect. For example, *photo* and *picture* refer to the same aspect in digital camera reviews. Identifying and grouping aspect expressions indicating the same aspect are essential for applications. Although WordNet [81] and other thesaurus dictionaries help to some extent, 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*.

Carenini et al. [11] proposed the first 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 (they are similar to those for information integration in Chap. 10). It requires a taxonomy of aspects to be given for a particular domain. The algorithm merges each discovered aspect expression to an aspect node in the taxonomy. Experiments based on digital camera and DVD reviews showed promising results.

In [138], Zhai et al. proposed a semi-supervised learning method to group aspect expressions into the user-specified aspect groups. Each group represents a specific aspect. To reflect the user needs, he/she first manually labels 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 (LU learning) based on labeled seeds and unlabeled examples. The method used the Expectation-Maximization (EM) algorithm, specifically, the naïve Bayesian EM formulation (see Sect. 5.1.1). When the algorithm ends, each unlabeled aspect expression is assigned to a suitable group.

The method in [138] uses two pieces of prior knowledge to provide a better initialization for EM. The two pieces of prior knowledge are (1) aspect expressions sharing some common words are likely to belong to the same group, e.g., "battery life" and "battery power," and (2) aspect expressions that are synonyms in a dictionary are likely to belong to the same group, e.g., "movie" and "picture." These two pieces of knowledge help EM produce better classification results.

Mapping implicit aspect expressions to aspects: 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. One way to map implicit aspect expressions to aspects is to manually compile a list of such mappings during training data annotation, which can then be used in the same domain in the future. However, we should 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."

Coreference resolution: This problem has been extensively studied in the NLP community. We use the following example blog to illustrate the problem: "I bought a Canon S500 camera yesterday. It looked beautiful. I took a few photos last night. They were amazing". "It" in the second sentence refers to "Canon S500 camera," which is an entity. "They" in the fourth sentence refers to "photos," which is an aspect of "Canon S500 camera". Recognizing these coreference relationships is called coreference resolution. Its usefulness in this case is clear. Without resolving them, we lose recall. That is, although we know that the second and fourth sentences express opinions, we do not know about what. Without knowing the opinion target, the opinion is of limited use.

In [16], the problem of **entity and aspect coreference resolution** was proposed. It determines which mentions of entities and/or aspects refer to the same entities. Here *entities* refer to both entities and aspects. Like most coreference resolution techniques, this paper took the supervised learning approach. The key interesting points were the design and testing of two opinion-related features to help classification. The first feature is based on opinion analysis of regular sentences (non-comparative sentences) and comparative sentences, and the idea of sentiment consistency. For example, we have the sentences, "The Sony camera is better than the Canon camera. It is cheap too." It is clear that "It" means "Sony" because in the first sentence, the opinion about "Sony" is positive (comparative positive), but it is negative (comparative negative) about "Canon," and the second sentence is positive. Thus, we can conclude that "It" refers to "Sony" because people usually express sentiments in a consistent way. It is unlikely that "It" refers to "Canon." As we can see, to obtain this feature, the system needs to have the ability to determine positive and negative opinions expressed in regular and comparative sentences.

The second feature considers what entities and aspects are modified by what opinion words. Consider these sentences, "The picture quality of the Canon camera is very good. It is not expensive either." The question is what "It" refers to, "Canon camera" or "picture quality." Clearly, we know that "It" refers to "Canon camera" because "picture quality" cannot be expensive. To obtain this feature, the system needs to identify what opinion words are usually associated with what entities or aspects, which means that the system needs to discover such relationships from the corpus. These two features can boost the coreference resolution accuracy.

Other NLP problems: The preceding problems are just some of the problems. In fact, there are many other NLP problems that need to be solved in order to produce accurate regular opinion quintuples (Sect. 11.1) and comparative opinion sextuples (Sect. 11.6.1). However, little research has been done to solve these problems. We use seven example sentences to illustrate some of them.

(1) Trying out Google chrome because Firefox keeps crashing.
(2) I am so happy because my new iPhone is nothing like my old ugly Nokia phone.
(3) After my wife and I slept on the mattress for only a week, I found a hill in the middle.
(4) Since I had a lot of pain on my back my doctor put me on the drug, and only two weeks after I have no more pain.
(5) Anyone knows a good Sony camera?
(6) If I can find a good Sony camera, I will buy it.
(7) What a great car, it stopped working in the second day.

For sentence (1), the opinion about Firefox is clearly negative, but for Google chrome, there is no opinion. We need to segment the sentence into clauses to decide that "crashing" only applies to Firefox. "Trying out" also indicates that there is no opinion yet. For sentence (2), it is easy to know that the opinion on the Nokia phone is negative, but it is not so easy to know whether the opinion on iPhone is positive or negative unless we realize that this is a comparative sentence. For sentence (3), the difficulty is that there is no adjective or adverb opinion word. In fact, this is an objective sentence rather than a subjective sentence. However, "hill", which is a noun, does imply a negative opinion for the mattress although "hill" itself bears no opinion. The issue is how to detect such cases. Some initial work has been done in [140] but the accuracy is still low. Sentence (4) shows an indirect opinion. This sentence has two conflict opinion expressions "a lot of pain" and "no more pain." The question is whether the drug is effective or not. Sentence (5) is a question and expresses no opinion, although the opinion word "good" is there. However, in some cases, question sentences can express opinions, e.g., "Any idea how to repair this lousy Sony camera?" Sentence (6) is a conditional sentence, which again has no opinion. However, conditional sentences can express opinions in many cases too, e.g., "If you are looking for a good camera, get this Sony" [85]. The first part of sentence (7) is sarcastic. Sarcasm is difficult to deal with. Some initial work has been done in [118].

In summary, to handle all these sentences and their associated problems, we need deeper sentence analysis and in many cases even at the semantic level. So far, not much research has been done.

11.8 Opinion Search and Retrieval

As Web search has proven to be valuable, it is not hard to imagine that opinion search will also be of great use. Two typical kinds of opinion search queries are as follows:

1. Find public opinions on a particular entity or an aspect of the entity, e.g., find customer opinions on a digital camera or the picture quality of the camera, and find public opinions on a political topic.
2. Find opinions of a person or organization (i.e., opinion holder) on a particular entity or an aspect of the entity, e.g., find Barack Obama's opinion on abortion. This type of search is particularly relevant to news articles, where individuals or organizations who express opinions are explicitly stated.

For the first type of queries, the user may simply give the name of the entity or the name of the aspect together with the name of the entity. For the second type of queries, the user may give the name of the opinion holder and the name of the entity.

Similar to traditional Web search, opinion search also has two major tasks: 1) retrieving relevant documents/sentences to the user query and 2) ranking the retrieved documents/sentences. However, there are also major differences. On retrieval, opinion search needs to perform two sub-tasks:

1. Find documents or sentences that are relevant to the query topic. This is the only task performed in the traditional Web search or retrieval.
2. Determine whether the documents or sentences express opinions and whether the opinions are positive or negative. This is the task of sentiment analysis. Traditional search does not perform this sub-task.

As for ranking, traditional Web search engines rank Web pages based on authority and relevance scores (Chap. 6). The basic premise is that the top ranked pages (ideally the first page) contain sufficient information to satisfy the user's information need. This paradigm is adequate for factual information search because *one fact equals to any number of the same fact*. That is, if the first page contains the required information, there is no need to see the rest of the relevant pages. For opinion search, this paradigm is fine for the second type of queries because the opinion holder usually has only one opinion on a particular entity or topic, and the opinion is contained in a single document or page. However, for the first type of opinion queries, this paradigm needs to be modified because ranking in opinion search has two objectives. First, it needs to rank those opinionated documents or sentences with high utilities or information contents at the top (see Sect. 11.10). Second, it needs to reflect the natural distribution of positive and negative opinions. This second objective is important because in most applications the actual proportions of positive and negative opinions are the most important pieces of information. Only reading the top ranked results as in the traditional search is problematic because the top result only represents the opinion of a single opinion holder. Thus, ranking in opinion search needs to capture the natural distribution of positive and negative sentiments of the whole population. One simple solution is to produce two rankings, one for positive opinions and one for negative opinions, and also display the numbers of positive and negative opinions.

Providing an aspect-based summary for each opinion search will be even better. However, it is an extremely challenging problem because aspect extraction, aspect grouping, and associating entities to its aspects are all very difficult problems. Without effective solution for them, such summary will not be possible.

To give a favor of opinion search, we present an example system [143], which is the winner of the blog track in the 2007 TREC evaluation (http://trec.nist.gov/). The task is exactly opinion search (or retrieval). This system has two components. The first component is for retrieving relevant documents for each query. The second component is for classifying the retrieved documents as opinionated or not-opinionated (subjectivity classification). The opinionated documents are further classified into positive, negative, or mixed (containing both positive and negative opinions).

Retrieval component: This component performs the traditional information retrieval (IR) task. This component considers both keywords and concepts. Concepts are named entities (e.g., names of people or organizations) or various types of phrases from dictionaries and other sources (e.g., Wikipedia entries). The strategy for processing a user query is as follows [142, 143]: It first recognizes and disambiguates the concepts within the user query. It then broadens the search query with its synonyms. After that, it recognizes concepts in the retrieved documents and also performs pseudo-feedback to automatically extract relevant words from the top-ranked documents to expand the query. Finally, it computes a similarity (or relevance score) of each document with the expanded query using both concepts and keywords.

Opinion classification component: This component performs two tasks: (1) classifying each document into one of the two categories, opinionated and not-opinionated, and (2) classifying each opinionated document as expressing a positive, negative, or mixed opinion. For both tasks, the system uses supervised learning. For the first task, it obtains a large amount of opinionated (subjective) training data from review sites such as rateitall.com and epinion.com. The data are also collected from different domains involving consumer goods and services as well as government policies and political viewpoints. The not-opinionated training data are obtained from sites that give objective information such as Wikipedia. From these training data, a SVM classifier is constructed.

This classifier is then applied to each retrieved document as follows: The document is first partitioned into sentences. The SVM classifier then classifies each sentence as opinionated or not-opinionated. If a sentence is classified to be opinionated, its strength as determined by SVM is also noted. A document is regarded opinionated if there is at least one sentence that is classified as opinionated. To ensure that the opinion of the sentence is directed to the query topic, the system requires that enough query concepts/words are found in its vicinity. The totality of the opinionated sentences and their strengths in a document together with the document's similarity with the query is used to rank the document.

To determine whether an opinionated document expresses a positive, negative or mixed opinion, the second classifier is constructed. The training data are reviews from review sites containing review ratings (e.g., rateitall.com). A low rating indicates a negative opinion while a high rating indicates a positive opinion. Using positive and negative reviews as training data, a sentiment classifier is built to classify each document as expressing positive, negative, or mixed opinion.

There are many other approaches for opinion retrieval. The readers are encouraged to read the papers at the TREC site (http://trec.nist.gov/pubs/trec16/t16_proceedings.html) and the overview paper of 2007 TREC blog track [78].

11.9 Opinion Spam Detection

In Sect. 6.10, we discussed Web spam, which refers to the use of "illegitimate means" to boost the search rank position of some target Web pages. The reason for spamming is because of the economic and/or publicity value of the rank position of a page returned by a search engine. In the context of opinions, the problem is similar but also quite different.

It has become a common practice for people to find and to read opinions on the Web for many purposes. For example, if one wants to buy a product, one typically goes to a merchant or review site (e.g., amazon.com) to read some reviews of existing users of the product. If one sees many positive reviews of the product, one is very likely to buy the product. However, if one sees many negative reviews, he/she will most likely choose another product. Positive opinions can result in significant financial gains and/or fames for organizations and individuals. This, unfortunately, gives good incentives for **opinion spam**, which refers to human activities (e.g., write spam reviews) that try to deliberately mislead readers or automated opinion mining systems by giving undeserving positive opinions to some target entities in order to promote the entities and/or by giving unjust or false negative opinions to some other entities in order to damage their reputation. Such opinions are also called **fake opinions**, **bogus opinions**, or **fake reviews**. We can predict that as opinions on the Web are increasingly used in practice by consumers and organizations, the problem of detecting spam opinions will become more and more critical.

Opinion spam is very different from Web spam because the two main types of Web spam, i.e., *content spam* and *link spam*, seldom occur in opinion documents such as product reviews. Recall that link spam is spam on hyperlinks, which almost does not exist in reviews as there is usually no links among reviews. Content spam tries to add irrelevant or remotely relevant words in target Web pages in order to fool search engines, which again hardly occurs in reviews. This section uses consumer reviews of products as an example to study opinion spam.

11.9.1 Types of Spam and Spammers

There are generally three types of spam reviews as identified in [44, 45]:

Type 1 (fake review): These are reviews that deliberately mislead readers or opinion mining systems by giving undeserving positive opinions to some target entities in order to promote the entities and/or by giving unjust or malicious negative opinions to some other entities in order to damage their reputation.

Table 11.4. Spam reviews vs. product quality

	Positive spam review	Negative spam review
Good quality product	1	***2***
Bad quality product	***3***	4
Average quality product	***5***	***6***

Type 2 (review on brand only): These reviews do not comment on the specific products that they are supposed to review, but only comment on the brands, the manufacturers, or the sellers of the products. Although they may be useful, they are considered as spam because they are not targeted at the specific products and are often biased. For example, in a review for a HP printer, the reviewer only wrote "I hate HP. I never buy any of their products".

Type 3 (non-review): These are not reviews or opinionated although they appear as reviews. There are two main sub-types: (1) advertisements and (2) other irrelevant texts containing no opinions (e.g., questions, answers, and random texts).

Harmful Fake Reviews: According to [44], types 2 and 3 spam reviews are rare and also easy to detect. We thus focus on type 1 fake reviews. In order to detect such reviews, let us first discuss what kinds of fake reviews are harmful. Table 11.4 gives a simple view of type 1 spam. The goal of spam reviews in regions 1, 3 and 5 is to promote the product. Although opinions expressed in region 1 may be true, the reviewers do not announce their conflict of interests. Note that good, bad, and average products can be defined based on the average rating given to the product (this definition can be dangerous if there are spammer groups that work together, see below). The goal of spam reviews in regions 2, 4, and 6 is to damage the reputation of some entities. Although opinions in reviews of region 4 may be true, the reviewers do not announce their conflict of interests and have bad intensions. Clearly, spam reviews in regions 1 and 4 are not so damaging, while spam reviews in regions 2, 3, 5, and 6 are very harmful. Thus, spam detection techniques should focus on identifying reviews in these regions.

Individual Spammers and Group Spammers: A spammer may act individually (e.g., the author of a book) or as a member of a group (e.g., a group of employees of a company).

Individual spammers: In this case, a spammer, who does not work with anyone else, writes spam reviews. The spammer may register at a review site as a single user, or as many fake users using different user-ids. He/she can also register at multiple review sites and write spam reviews.

Group spammers: A group of spammers works together to promote a target entity and/or to damage the reputation of another. They may also register at multiple sites and spam on these sites. Group spam can be very damaging because they may take control of the sentiment on a product and completely mislead potential customers.

11.9.2 Hiding Techniques

In order to avoid being detected, spammers may take a variety of precautions. We study individual and group spammers separately. The lists are by no means exhaustive and should be considered as just examples.

An Individual Spammer

1. The spammer builds up reputation by reviewing other products in the same or different categories/brands that he/she does not care about and give them agreeable ratings and reasonable reviews. Then, he/she becomes a trustworthy reviewer. However, he/she may write spam reviews on the products that he/she really cares about. This hiding method is useful because some sites rank reviewers based on their reviews that are found helpful by readers, e.g., amazon.com. Some sites also have trust systems that allow readers to assign trust scores to reviewers.
2. The spammer registers multiple times at a site using different user-ids and write multiple spam reviews under these user-ids so that their reviews or ratings will not appear as outliers. The spammer may even use different machines to avoid being detected by server log-based detection methods that can compare IP addresses of reviewers.
3. Spammers write either only positive reviews on his/her own products or only negative reviews on the products of his/her competitors, but not both. This is to hide from spam detection methods that compare one's reviews on competing products from different brands.
4. The spammer gives a reasonably high rating but write a critical (negative) review. This may fool detection methods that find outliers based on ratings alone. Yet, automated review mining systems will pick up all the negative sentiments in the actual review content.

A Group of Spammers

1. Every member of the group reviews the same product to lower the rating deviation.
2. Every member of the group writes a review roughly at the time when the product is launched in order to take control of sentiment on the

product. It is generally not a good idea to write many spam reviews at the same time after many reviews have been written by others because a spike will appear, which can be easily detected.
3. Members of the group write reviews at random intervals to hide spikes.
4. If the group is sufficiently large, it may be divided into sub-groups so that each sub-group can spam at different Web sites (instead of only spam at the same site) to avoid being detected by methods that compare average ratings and content similarities of reviews from different sites.

11.9.3 Spam Detection Based on Supervised Learning

In general, spam detection can be formulated as a classification problem with two classes, *spam* and *non-spam*. Due to the specific nature of different types of spam, they need to be dealt with differently. For spam reviews of type 2 and type 3, they can be detected based on traditional classification learning using manually labeled spam and non-spam reviews because these two types of spam reviews are easily recognizable manually. The main task is to find a set of effective features for model building. In [44, 45], three sets of features were identified for learning:

Review centric features: These are features about the content of reviews. Example features include actual words in a review, the number of times that brand names are mentioned, the percentage of opinion words, the review length, and the number of helpful feedbacks.

Reviewer centric features: These are features about each reviewer. Example features include the average rating given by the reviewer, the standard deviation in rating, the ratio of the number of reviews that the reviewer wrote which were the first reviews of the products to the total number of reviews that he/she wrote, and the ratio of the number of cases in which he/she was the only reviewer.

Product centric features: These are features about each product. Example features include the price of the product, the sales rank of the product (amazon.com assigns sales rank to 'now selling products' according to their sales volumes), the average review rating of the product, and the standard deviation in ratings of the reviews for the product.

Logistic regression was used in learning. Experimental results based on a large number of amazon.com reviews showed that type 2 and type 3 spam reviews are fairly easy to detect.

However, this cannot be said about type 1 spam, i.e., fake opinions or reviews. In fact, it is very difficult to detect such reviews because manual labeling training data is very hard, if not impossible. The problem is that identifying spam reviews by simply reading the reviews is extremely difficult because a spammer can carefully craft a spam review that is just like any innocent review.

Since manually labeling training data is hard, other ways have to be explored in order to find training examples for detecting possible fake reviews. In [44], it exploited duplicate reviews. In their study of 5.8 million reviews, 2.14 million reviewers and 6.7 million products from amazon.com, they found a large number of duplicate and near-duplicate reviews, which indicates that review spam is widespread. These duplicates (which include near-duplicates) can be divided into four groups:

1. Duplicates from the same userid on the same product
2. Duplicates from different userids on the same product
3. Duplicates from the same userid on different products
4. Duplicates from different userids on different products

The first type of duplicates can be the results of reviewers mistakenly clicking the submit button multiple times (which of course can be detected based on the submission dates and times), or the same reviewers coming back to write updated reviews after using the product for some time. However, the last three kinds of duplicates are almost certainly fake reviews. Further sanity check was performed on these duplicate reviews because amazon.com cross-posts reviews to different formats of the same product, e.g., hardcover and paperback of the same book. Such duplicates were removed. These three types of duplicates and near duplicates were treated as type 1 spam reviews, and the rest of the reviews were treated as non-spam reviews. Logistic regression was used to build a classification model. The experiments showed some tentative but interesting results.

- Negative outlier reviews (ratings with significant negative deviations from the average rating) tend to be heavily spammed. This is quite intuitive. Positive outlier reviews are not badly spammed.
- Those reviews that are the only reviews of some products are likely to be spammed. This can be explained by the tendency of promoting an unpopular product by writing a spam review.
- Top-ranked reviewers are more likely to be spammers. Amazon.com gives a rank to each member/reviewer based on the frequency that he/she gets helpful feedback on his/her reviews. Additional analysis showed that top-ranked reviewers generally wrote a large number of reviews. People who wrote a large number of reviews are natural suspects. Some top reviewers wrote thousands or even tens of thousands of reviews, which is unlikely for an ordinary consumer.
- Spam reviews can get good helpful feedbacks and non-spam reviews can get bad feedbacks. This is important as it shows that if usefulness or

quality of a review is defined based on the helpful feedbacks that the review gets, people can be readily fooled by spam reviews. Note that the number of helpful feedbacks can be spammed too.

- Products of lower sale ranks are more likely to be spammed. This is good news because spam activities seem to be limited to low selling products, which is actually quite intuitive as it is difficult to damage the reputation of a popular product by writing a spam review.

Finally, it should be noted again that these results are only tentative because (1) it is not confirmed that the three types of duplicates are absolutely spam reviews, and (2) many spam reviews are not duplicated and they are not considered as spam in model building but as non-spam due to the difficulty of manual labeling. For additional analysis and more spam detection strategies, please refer to [44, 45].

11.9.4 Spam Detection Based on Abnormal Behaviors

Due to the difficulty of manually labeling training data, treating opinion spam detection as a supervised learning problem is problematic. This section describes two techniques that try to identify atypical behaviors of reviewers for detecting spammers. For example, if a reviewer wrote all negative reviews for a brand but other reviewers were all positive about the brand, then this reviewer is naturally a spam suspect.

The first technique, which is due to [68], identifies several unusual reviewer behavior models based on different review patterns that suggest spamming. Each model assigns a numeric spamming behavior score to a reviewer by measuring the extent to which the reviewer practices spamming behavior of the type. All the scores are then combined to produce the final spam score. The spamming behavior models are:

(a) **Targeting products**: To game an online review system, it is hypothesized that a spammer will direct most of his efforts on promoting or victimizing a few products which are collectively called the *targeted products*. He is expected to monitor the targeted products closely and mitigate the ratings by writing fake reviews when time is appropriate.

(b) **Targeting groups**: This spam behavior model defines the pattern of spammers manipulating ratings of a set of products sharing some attribute(s) within a short span of time. For example, a spammer may target several products of a brand within a few hours. This pattern of ratings saves the spammers' time as they do not need to log on to the review system many times. To achieve maximum impact, the ratings given to these target groups of products are either very high or low.

(c) **General rating deviation**: A reasonable reviewer is expected to give ratings similar to other raters of the same product. As spammers attempt to promote or demote products, their ratings could be quite different from other reviewers.
(d) **Early rating deviation**: Early deviation captures the behavior of a spammer contributing a spam review soon after product is launched. Such spam reviews are likely to attract attention from other reviewers, allowing spammers to manipulate the views of subsequent reviewers.

The second technique, which is due to [46], identifies unusual reviewer behavior patterns through unexpected rule discovery. For example, if a reviewer wrote all negative reviews on products of a brand but other reviewers are generally positive about the brand, this reviewer is a spam suspect. To find unusual behaviors, the conventional approach is to write an application-specific heuristic program to find such behaviors as the first technique above. However, this is undesirable. It is much better to propose a general framework for solving this class of problems so that the resulting system can also be applied to other domains. Such a general approach is proposed in [46], which shows that the problem can be formulated as finding unexpected class association rules/patterns from data (Sect. 2.5).

Recall that the data for mining class association rules (CAR) consists of a set of data records, which are described by a set of normal attributes $A = \{A_1, \cdots, A_n\}$, and a class attribute $C = \{c_1, \cdots, c_m\}$ of m discrete values, called *classes* (Sect. 2.5). A CAR rule is of the form: $X \rightarrow c_i$, where X is a set of conditions from the attributes in A and c_i is a class in C. Such a rule gives the conditional probability of $\Pr(c_i \mid X)$ (called the *confidence*) and the joint probability $\Pr(X, c_i)$ (called the *support*).

For the spam detection application, the data for mining is produced as follows: Each review forms a data record with a set of attributes, e.g., *reviewer-id*, *brand-id*, *product-id*, and a class. The class represents the sentiment of the reviewer on the product, *positive*, *negative*, or *neutral* based on the review rating. In most review sites (e.g., amazon.com), each review has a rating between 1 (lowest) and 5 (highest) assigned by its reviewer. We can assign the rating of 4 or 5 as positive, 3 as neutral, and 1 or 2 as negative. A rule could be that a reviewer gives all positive ratings to a particular brand of products. The method in [46] finds four types of unexpected rules based on four unexpectedness definitions. The unexpected rules represent atypical behaviors of reviewers. Below, an example behavior is given for each type of unexpectedness definition. Their detailed definitions, which can be found in [46], are quite involved.

- **Confidence unexpectedness**: Using this measure, we can find reviewers who give all high ratings to products of a brand, but most other reviewers are generally negative about the brand.

- **Support unexpectedness**: Using this measure, we can find reviewers who write multiple reviews for a single product, while other reviewers only write one review.
- **Attribute distribution unexpectedness**: Using this measure, we can find that most positive reviews for a brand of products are from only one reviewer although there are a large number of reviewers who have reviewed the products of the brand.
- **Attribute unexpectedness**: Using this measure, we can find reviewers who write only positive reviews to one brand and only negative reviews to another brand.

The advantage of this approach is that all the measures are defined based on CARs rules, and are not specific to any application domain, and thus can be used in other domains to find unexpected rules. The weakness is that some behaviors cannot be detected, e.g., time-related behaviors, because class association rules do not consider time.

Experimental results of both papers [46, 68] using amazon.com reviews showed that many spammers can be detected based on their behaviors.

11.9.5 Group Spam Detection

A group spam detection algorithm was reported in [84]. It finds groups of spammers who work together to promote or demote some products. The method works in two steps:

1. **Frequent pattern mining**: First, it pre-processes the review data to produce a set of transactions. Each transaction represents a unique product and consists of all the reviewers (their ids) who have reviewed that product. Using all the transactions, it performs frequent pattern mining. The patterns give us a set of candidate groups who might have spammed together. The reason for using frequent pattern mining is as follows: If a group of reviewers who only worked together once to promote or to demote a single product, it can be hard to detect based on their collective or group behavior. However, these fake reviewers (especially those who get paid to write) cannot be just writing one review for a single product because they would not make enough money that way. Instead, they work on many products, i.e., write many reviews about many products, which unfortunately also give them away. Frequent pattern mining can be used to find them working together on multiple products.
2. **Rank groups based on a set of group spam indicators**: The groups discovered in step 1 may not all be spammer groups. Many of the re-

viewers are grouped together in pattern mining simply due to chance. Then, this step first uses a set of indicators to catch different types of unusual group behaviors. These indicators include writing reviews together in a short time window, writing reviews right after the product launch, group content similarity, group rating deviation, etc. (see [84] for details). It then ranks the discovered groups from step 1 based on their indicator values using SVM rank (also called Ranking SVM) [47].

11.10 Utility of Reviews

A related problem that has also been studied in the past few years is the determination of the usefulness, helpfulness, or utility of each review [31, 57, 73, 144]. This is a meaningful task as it is desirable to rank reviews based on utilities or qualities when showing reviews to the user, with the most useful reviews first. In fact, many review aggregation sites have been practicing this for years. They obtain the helpfulness or utility score of each review by asking readers to provide helpfulness feedbacks to each review. For example, in amazon.com, the reader can indicate whether he/she finds a review helpful by responding to the question "*Was the review helpful to you?*" just below each review. The feedback results from all those responded are then aggregated and displayed right before each review, e.g., "*15 of 16 people found the following review helpful*." Although most review sites already provide the service, automatically determining the quality of a review is still useful because many reviews have few or no feedbacks. This is especially true for new reviews.

Determining the utility of reviews is usually formulated as a regression problem. The learned model assigns a utility value to each review, which can be used in review ranking. In this area of research, the ground truth data used for both training and testing are usually the user-helpfulness feedback given to each review, which as we discussed above is provided for each review at many review sites. So unlike fake review detection, the training and testing data here is not an issue.

Researchers have used many types of features for model building. Example features include review length, review rating (the number of stars), counts of some specific POS tags, opinion words, tf-idf weighting scores, wh-words, product attribute mentions, comparison with product specifications, comparison with editorial reviews, and many more. Subjectivity classification was also applied in [31]. In [73], Liu et al. formulated the problem slightly differently. They made it a binary classification problem. Instead of using the original helpfulness feedback as the target or dependent variable, they performed manual annotation based on whether the review evaluates many product aspects or not.

Finally, we should note that review utility regression/classification and review spam detections are different concepts. Not-helpful or low quality reviews are not necessarily fake reviews or spam, and helpful reviews may not be non-spam. A user often determines whether a review is helpful or not based on whether the review expresses opinions on many aspects of the product. A spammer can satisfy this requirement by carefully crafting a review that is just like a normal helpful review. Using the number of helpful feedbacks to define review quality is also problematic because user feedbacks can be spammed too. Feedback spam is a sub-problem of click fraud in search advertising, where a person or robot clicks on some online advertisements to give the impression of real customer clicks. Here, a robot or a human spammer can also click on helpfulness feedback button to increase the helpfulness of a review. Another important point is that a low quality review is still a valid review and should not be discarded, but a spam review is untruthful and/or malicious and should be removed once detected.

Bibliographic Notes

An attempt to define the opinion mining and sentiment analysis problem was made in the first edition of this book. Improvements were made in my book chapter "Sentiment Analysis and Subjectivity" [70] for the second edition of *Handbook of Natural Language Processing* [38]. The purpose was to provide a common framework for different research directions and to abstract a structure from the intimidating unstructured text. Needless to say, the definitions were influenced by many early researches. The main idea was from the aspect-based opinion mining model proposed in [37], which was then called feature-based opinion mining. The improvements in [70] and in Sect. 11.1 of this chapter were also shaped by my 1.5 years of involvement in a startup company on opinion mining and hands-on experiences in serving industry clients and understanding their diverse needs.

Much of the early research on opinion mining focused on sentiment classification at the document and sentence levels. Representative works on classification at the document level in the early years include those by Turney [119] and Pang et al. [94]. They have been discussed in this chapter. Representative works on classification at the sentence level include those by Hatzivassiloglou and Wiebe [36] and Riloff and Wiebe [103] among others, which determines whether a sentence is subjective or objective. Sentence level sentiment or opinion classification (positive, negative and neutral) was studied by Kim and Hovy [55], Wiebe and Riloff [126], among others. Some of these methods have been discussed in Sect. 11.3.3.

Other related works at both the document and sentence levels include those by Dave et al [15], Tong [117], Das and Chen [14], Morinaga et al. [83], Beineke et al. [3], Nasukawa and Yi [86], Nigam and Hurst [88], Gamon [26], Gamon et al. [27], Pang and Lee [92, 93], Ng et al. [87], McDonald et al. [79], Wilson et al. [129-131], Yessenalina et al. [134], Li et al. [65], and many others. A survey on this literature can be found in [91].

The model of aspect-based opinion mining and summarization was introduced by Hu and Liu [37] and Liu et al. [72] (originally called feature-based opinion mining and summarization). Some initial methods for performing the task were also proposed. Popescu and Etzioni [99], Carenini et al [11], Ku et al. [60, 61], Ding et al. [17], Jin and Ho [41], and Lu et al. [75] explored the problem further. These works typically find aspects first and then determine their associated opinions, e.g., [17, 37, 99]. Recently, researchers also proposed many statistical models to find aspects and their associated opinions at the same time, e.g., those by Titov and McDonald [115, 116], Brody and Elhadad [8], Wang et al. [121], Lin and He [69], Mei et al. [80], Zhao et al. [145], Lu et al. [77], etc. We have briefly discussed these ideas in Sect. 11.5.3. The work of Qiu et al. [100, 101] deals with the same problem but by exploiting some syntactic relations between opinion words and their target aspects to extract both. Other related works include [9, 12, 16, 18, 20, 33, 34, 39, 40, 48, 49, 53, 56, 58, 59, 64, 87, 89, 97, 107, 110, 113, 122, 125, 131, 135, 137, 138, 141, 146]

Most document level, sentence level, and aspect level techniques need a list of opinion words or phrases, which is called the opinion lexicon. There are two types of approaches to compiling and expanding an opinion lexicon: (1) corpus-based approaches and (2) dictionary-based approaches. Corpus-based approaches find co-occurrence patterns of words to determine their opinion orientations, which have been studied by Turney [119], Riloff and Wiebe [103], Hatzivassiloglou and McKeown [35], Yu and Hatzivassiloglou [136], Grefenstette et al. [32], Kanayama and Nasukawa [51], Ding et al. [17], Murthy and Liu [28], and Kaji and Kitsuregawa [48, 49]. Qiu et al. [100, 101] proposed a double-propagation method, which discovers aspects and expands a given opinion lexicon simultaneously. Dictionary-based approaches use synonyms, antonyms, hierarchies, and gloss in WordNet to determine word opinions, e.g., Hu and Liu [37], Kamps et al. [50], Valitutti et al. [120], Kim and Hovy [55], Esuli and Sebastiani [21, 22], Andreevskaia and Bergler [1], and Dragut et al. [19].

On mining comparative sentences, Jindal and Liu [42, 43] defined the problem and proposed some initial techniques, which were improved by Ganapathibhotla and Liu [28] and Ding et al. [18]. Li et al. [66] further studied the extraction of compared entities. Research in linguistics on syntax and semantics of comparatives can be found in [52, 82].

The problem of opinion spam was introduced by Jindal and Liu in [44, 45]. They also proposed a supervised approach to detect fake reviews. Detecting spammers (or reviewers who write fake reviews) by studying their atypical behaviors were investigated by Lim et al. [68] and Jindal et al. [46]. Group spam detection was studied by Mukherjee et al. in [84].

Bibliography

1. Andreevskaia, A. and S. Bergler. Mining WordNet for fuzzy sentiment: Sentiment tag extraction from WordNet glosses. In *Proceedings of Conference of the European Chapter of the Association for Computational Linguistics (EACL-06)*, 2006.
2. Aue, A. and M. Gamon. Customizing sentiment classifiers to new domains: a case study. In *Proceedings of Recent Advances in Natural Language Processing (RANLP-2005)*, 2005.
3. Beineke, P., T. Hastie, C. Manning, and S. Vaithyanathan. An exploration of sentiment summarization. In *Proceedings of AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications*, 2003.
4. 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.
5. Blei, D., A. Ng, and M. Jordan. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 2003, 3: p. 993-1022.
6. Blitzer, J., M. Dredze, and F. Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL-2007)*, 2007.
7. Breck, E., Y. Choi, and C. Cardie. Identifying expressions of opinion in context. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-2007)*, 2007.
8. 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*, 2010.
9. Carenini, G., R. Ng, and A. Pauls. Interactive multimedia summaries of evaluative text. In *Proceedings of 10th Intl. Conf. on Intelligent User Interfaces (IUI-2006)*, 2006.
10. Carenini, G., R. Ng, and A. Pauls. Multi-document summarization of evaluative text. In *Proceedings of the European Chapter of the Association for Computational Linguistics (EACL-2006)*, 2006.
11. Carenini, G., R. Ng, and E. Zwart. Extracting knowledge from evaluative text. In *Proceedings of Third Intl. Conf. on Knowledge Capture (K-CAP-05)*, 2005.

12. Choi, Y. and C. Cardie. Hierarchical sequential learning for extracting opinions and their attributes. In *Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL-2010)*, 2010.
13. 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.
14. Das, S. and M. Chen. Yahoo! for Amazon: Extracting market sentiment from stock message boards. In *Proceedings of APFA-2001*, 2001.
15. Dave, K., S. Lawrence, and D. Pennock. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of International Conference on World Wide Web (WWW-2003)*, 2003.
16. Ding, X. and B. Liu. Resolving Object and Attribute Coreference in Opinion Mining. In *Proceedings of International Conference on Computational Linguistics (COLING-2010)*, 2010.
17. Ding, X., B. Liu, and P. Yu. A holistic lexicon-based approach to opinion mining. In *Proceedings of the Conference on Web Search and Web Data Mining (WSDM-2008)*, 2008.
18. Ding, X., B. Liu, and L. Zhang. Entity discovery and assignment for opinion mining applications. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2009)*, 2009.
19. Dragut, E., C. Yu, P. Sistla, and W. Meng. Construction of a sentimental word dictionary. In *Proceedings of ACM International Conference on Information and Knowledge Management (CIKM-2010)*, 2010.
20. Eguchi, K. and V. Lavrenko. Sentiment retrieval using generative models. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-2006)*, 2006.
21. Esuli, A. and F. Sebastiani. Determining term subjectivity and term orientation for opinion mining. In *Proceedings of Conf. of the European Chapter of the Association for Computational Linguistics (EACL-2006)*, 2006.
22. Esuli, A. and F. Sebastiani. Determining the semantic orientation of terms through gloss classification. In *Proceedings of ACM International Conference on Information and Knowledge Management (CIKM-2005)*, 2005.
23. Esuli, A. and F. Sebastiani. SentiWordNet: A publicly available lexical resource for opinion mining. In *Proceedings of Language Resources and Evaluation (LREC-2006)*, 2006.
24. Fiszman, M., D. Demner-Fushman, F. Lang, P. Goetz, and T. Rindflesch. Interpreting comparative constructions in biomedical text. In *Proceedings of BioNLP*, 2007.
25. Freitag, D. and A. McCallum. Information extraction with HMM structures learned by stochastic optimization. In *Proceedings of National Conf. on Artificial Intelligence (AAAI-2000)*, 2000.

26. Gamon, M. Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis. In *Proceedings of Intl. Conf. on Computational Linguistics*, 2004.
27. Gamon, M., A. Aue, S. Corston-Oliver, and E. Ringger. Pulse: Mining customer opinions from free text. *Advances in Intelligent Data Analysis VI*, 2005: p. 121-132.
28. Ganapathibhotla, M. and B. Liu. Mining opinions in comparative sentences. In *Proceedings of International Conference on Computational Linguistics (Coling-2008)*, 2008.
29. Ghahramani, Z. and K. Heller. Bayesian sets. *Advances in Neural Information Processing Systems*, 2006, 18: p. 435.
30. Ghani, R., K. Probst, Y. Liu, M. Krema, and A. Fano. Text mining for product attribute extraction. *ACM SIGKDD Explorations Newsletter*, 2006, 8(1): p. 41-48.
31. Ghose, A. and P. Ipeirotis. Designing novel review ranking systems: predicting the usefulness and impact of reviews. In *Proceedings of the International Conference on Electronic Commerce* 2007.
32. Grefenstette, G., Y. Qu, D. Evans, and J. Shanahan. Validating the coverage of lexical resources for affect analysis and automatically classifying new words along semantic axes. In *Proceedings of AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications*, 2004.
33. 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.
34. Hassan, A. and D. Radev. Identifying text polarity using random walks. In *Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL-2010)*, 2010.
35. Hatzivassiloglou, V. and K. McKeown. Predicting the semantic orientation of adjectives. In *Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL-1997)*, 1997.
36. Hatzivassiloglou, V. and J. Wiebe. Effects of adjective orientation and gradability on sentence subjectivity. In *Proceedings of Interntional Conference on Computational Linguistics (COLING-2000)*, 2000.
37. 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)*, 2004.
38. Indurkhya, N. and F. Damerau. Handbook of Natural Language Processing. *Chapman & Hall/Crc Machine Learning & Pattern Recognition*, 2010.
39. 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.
40. Jijkoun, V., M.d. Rijke, and W. Weerkamp. Generating Focused Topic-Specific Sentiment Lexicons. In *Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL-2010)*, 2010.

41. 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)*, 2009.
42. Jindal, N. and B. Liu. Identifying comparative sentences in text documents. In *Proceedings of ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR-2006)*, 2006.
43. Jindal, N. and B. Liu. Mining comparative sentences and relations. In *Proceedings of National Conf. on Artificial Intelligence (AAAI-2006)*, 2006.
44. Jindal, N. and B. Liu. Opinion spam and analysis. In *Proceedings of the Conference on Web Search and Web Data Mining (WSDM-2008)*, 2008.
45. Jindal, N. and B. Liu. Review spam detection. In *Proceedings of WWW (Poster paper)*, 2007.
46. Jindal, N., B. Liu, and E. Lim. Finding Unusual Review Patterns Using Unexpected Rules. In *Proceedings of ACM International Conference on Information and Knowledge Management (CIKM-2010)*, 2010.
47. Joachims, T. Optimizing search engines using clickthrough data. In *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD-2002)*, 2002.
48. Kaji, N. and M. Kitsuregawa. Automatic construction of polarity-tagged corpus from HTML documents. In *Proceedings of COLING/ACL 2006 Main Conference Poster Sessions (COLING-ACL-2006)*, 2006.
49. Kaji, N. and M. Kitsuregawa. Building lexicon for sentiment analysis from massive collection of HTML documents. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-2007)*, 2007.
50. Kamps, J., M. Marx, R. Mokken, and M. De Rijke. Using WordNet to measure semantic orientation of adjectives. In *Proc. of LREC-2004*, 2004.
51. Kanayama, H. and T. Nasukawa. Fully automatic lexicon expansion for domain-oriented sentiment analysis. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-2006)*, 2006.
52. Kennedy, C. Comparatives, Semantics of. In *Encyclopedia of Language and Linguistics, Second Edition*. 2005, Elsevier.
53. Kim, S. and E. Hovy. Automatic identification of pro and con reasons in online reviews. In *Proceedings of COLING/ACL 2006 Main Conference Poster Sessions (ACL-2006)*, 2006.
54. Kim, S. and E. Hovy. Crystal: Analyzing predictive opinions on the web. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP/CoNLL-2007)*, 2007.
55. Kim, S. and E. Hovy. Determining the sentiment of opinions. In *Proceedings of Interntional Conference on Computational Linguistics (COLING-2004)*, 2004.
56. Kim, S. and E. Hovy. Identifying and analyzing judgment opinions. In *Proceedings of Human Language Technology Conference of the North American Chapter of the ACL*, 2006.

57. Kim, S., P. Pantel, T. Chklovski, and M. Pennacchiotti. Automatically assessing review helpfulness. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2006)*, 2006: Association for Computational Linguistics.
58. Kobayashi, N., R. Iida, K. Inui, and Y. Matsumoto. Opinion mining on the Web by extracting subject-attribute-value relations. In *Proceedings of AAAI-CAAW'06*, 2006.
59. 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*, 2007.
60. Ku, L., H. Ho, and H. Chen. Novel relationship discovery using opinions mined from the web. In *Proceedings of National Conf. on Artificial Intelligence (AAAI-2006)*, 2006.
61. Ku, L., Y. Liang, and H. Chen. Opinion extraction, summarization and tracking in news and blog corpora. In *Proceedings of AAAI-CAAW'06*, 2006.
62. Lafferty, J., A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of International Conference on Machine Learning (ICML-2001)*, 2001.
63. Lee, L. Measures of distributional similarity. In *Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL-1999)*, 1999.
64. Li, B., L. Zhou, S. Feng, and K.-F. Wong. A Unified Graph Model for Sentence-Based Opinion Retrieval. In *Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL-2010)*, 2010.
65. Li, S., C.-R. Huang, G. Zhou, and S.Y.M. Lee. Employing Personal/Impersonal Views in Supervised and Semi-Supervised Sentiment Classification. In *Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL-2010)*, 2010.
66. Li, S., C. Lin, Y. Song, and Z. Li. Comparable entity mining from comparative questions. In *Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL-2010)*, 2010.
67. 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)*, 2010.
68. Lim, E., V. Nguyen, N. Jindal, B. Liu, and H. Lauw. Detecting Product Review Spammers using Rating Behaviors. In *Proceedings of ACM International Conference on Information and Knowledge Management (CIKM-2010)*, 2010.
69. 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.
70. Liu, B. Sentiment analysis and subjectivity. In *Handbook of Natural Language Processing, Second Edition*, N. Indurkhya and F.J. Damerau, Editors. 2010.

71. Liu, B. *Web data mining: Exploring Hyperlinks, Contents, and Usage Data, The First Edition*. 2006: Springer.
72. 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.
73. Liu, J., Y. Cao, C. Lin, Y. Huang, and M. Zhou. Low-quality product review detection in opinion summarization. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL-2007)*, 2007.
74. Liu, Y., X. Huang, A. An, and X. Yu. ARSA: a sentiment-aware model for predicting sales performance using blogs. In *Proceedings of ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR-2007)*, 2007.
75. Lu, Y., H. Duan, H. Wang, and C. Zhai. Exploiting Structured Ontology to Organize Scattered Online Opinions. In *Proceedings of Interntional Conference on Computational Linguistics (COLING-2010)*, 2010.
76. Lu, Y. and C. Zhai. Opinion integration through semi-supervised topic modeling. In *Proceedings of International Conference on World Wide Web (WWW-2008)*, 2008.
77. 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.
78. Macdonald, C., I. Ounis, and I. Soboroff. Overview of the TREC 2007 blog track, 2007.
79. McDonald, R., K. Hannan, T. Neylon, M. Wells, and J. Reynar. Structured models for fine-to-coarse sentiment analysis. In *Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL-2007)*, 2007.
80. 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.
81. Miller, G., R. Beckwith, C. Fellbaum, D. Gross, and K. Miller. *WordNet: An on-line lexical database*. 1990: Oxford Univ. Press.
82. Moltmann, F. *Coordination and Comparatives, PhD thesis*. 1987, MIT: Cambridge.
83. Morinaga, S., K. Yamanishi, K. Tateishi, and T. Fukushima. Mining product reputations on the web. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2002)*, 2002.
84. Mukherjee, A., B. Liu, J. Wang, N. Glance, and N. Jindal. Detecting Group Review Spam. In *Proceedings of International Conference on World Wide Web (WWW-2011, poster paper)*, 2011.
85. Narayanan, R., B. Liu, and A. Choudhary. Sentiment analysis of conditional sentences. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-2009)*, 2009.
86. Nasukawa, T. and J. Yi. Sentiment analysis: Capturing favorability using natural language processing. In *Proceedings of the K-CAP-03, 2nd Intl. Conf. on Knowledge Capture*, 2003.

87. Ng, V., S. Dasgupta, and S. Arifin. Examining the role of linguistic knowledge sources in the automatic identification and classification of reviews. In *Proceedings of COLING/ACL 2006 Main Conference Poster Sessions (COLING/ACL-2006)*, 2006.
88. Nigam, K. and M. Hurst. Towards a robust metric of opinion. In *Proceedings of AAAI Spring Symp. on Exploring Attitude and Affect in Text*, 2004.
89. Nishikawa, H., T. Hasegawa, Y. Matsuo, and G. Kikui. Optimizing informativeness and readability for sentiment summarization. In *Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL-2010)*, 2010.
90. Pan, S., X. Ni, J. Sun, Q. Yang, and Z. Chen. Cross-domain sentiment classification via spectral feature alignment. In *Proceedings of International Conference on World Wide Web (WWW-2010)*, 2010.
91. Pang, B. and L. Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2008, 2(1-2): p. 1-135.
92. Pang, B. and L. Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of Meeting of the Association for Computational Linguistics (ACL-2005)*, 2005.
93. Pang, B. and L. Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of Meeting of the Association for Computational Linguistics (ACL-2004)*, 2004.
94. 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.
95. Pantel, P., E. Crestan, A. Borkovsky, A. Popescu, and V. Vyas. Web-scale distributional similarity and entity set expansion. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-2009)*, 2009.
96. Parrott, W. *Emotions in social psychology: Essential readings*. 2001: Psychology Pr.
97. Paul, M., C. Zhai, and R. Girju. Summarizing Contrastive Viewpoints in Opinionated Text. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-2010)*, 2010.
98. Polanyi, L. and A. Zaenen. Contextual valence shifters. In *Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text*, 2004.
99. 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.
100. Qiu, G., B. Liu, J. Bu, and C. Chen. Expanding domain sentiment lexicon through double propagation. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI-2009)*, 2009.
101. Qiu, G., B. Liu, J. Bu, and C. Chen. Opinion Word Expansion and Target Extraction through Double Propagation. *To appear in Computational Linguistics*, 2010.

102. Riloff, E., S. Patwardhan, and J. Wiebe. Feature subsumption for opinion analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2006)*, 2006.
103. Riloff, E. and J. Wiebe. Learning extraction patterns for subjective expressions. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-2003)*, 2003.
104. Santorini, B. *Part-of-speech tagging guidelines for the Penn Treebank Project*. 1990: University of Pennsylvania, School of Engineering and Applied Science, Dept. of Computer and Information Science.
105. Sarawagi, S. Information extraction*. Foundations and Trends in Databases*, 2008, 1(3): p. 261-377.
106. Seki, Y., K. Eguchi, N. Kando, and M. Aono. Opinion-focused summarization and its analysis at DUC 2006. In *Proceedings of the Document Understanding Conference (DUC)*, 2006.
107. Shanahan, J., Y. Qu, and J. Wiebe. *Computing attitude and affect in text: theory and applications*. 2006: Springer-Verlag.
108. Stone, P. The general inquirer: A computer approach to content analysis*. Journal of Regional Science*, 1968, 8(1).
109. Stoyanov, V. and C. Cardie. Partially supervised coreference resolution for opinion summarization through structured rule learning. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-2006)*, 2006.
110. Stoyanov, V. and C. Cardie. Toward opinion summarization: Linking the sources. In *Proceedings of Workshop on Sentiment and Subjectivity in Text*, 2006.
111. 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.
112. Takamura, H., T. Inui, and M. Okumura. Extracting semantic orientations of phrases from dictionary. In *Proceedings of the Joint Human Language Technology/North American Chapter of the ACL Conference (HLT-NAACL-2007)*, 2007.
113. Tata, S. and B. Di Eugenio. Generating fine-grained reviews of songs from album reviews. In *Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL-2010)*, 2010.
114. Tesniere, L. *Élements de syntaxe structurale: Préf. de Jean Fourquet*. 1959: C. Klincksieck.
115. 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)*, 2008.
116. Titov, I. and R. McDonald. Modeling online reviews with multi-grain topic models. In *Proceedings of International Conference on World Wide Web (WWW-2008)*, 2008.
117. Tong, R. An operational system for detecting and tracking opinions in on-line discussion. In *Proceedings of SIGIR Workshop on Operational Text Classification*, 2001.

118. Tsur, O., D. Davidov, and A. Rappoport. In *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media (ICWSM-2010)*. 2010.
119. Turney, P. 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.
120. Valitutti, A., C. Strapparava, and O. Stock. Developing affective lexical resources. *PsychNology Journal*, 2004, 2(1): p. 61-83.
121. 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.
122. Wei, W. and J. Gulla. Sentiment learning on product reviews via sentiment ontology tree. In *Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL-2010)*, 2010.
123. Wiebe, J. Learning subjective adjectives from corpora. In *Proceedings of National Conf. on Artificial Intelligence (AAAI-2000)*, 2000.
124. Wiebe, J., R. Bruce, and T. O'Hara. Development and use of a gold-standard data set for subjectivity classifications. In *Proceedings of the Association for Computational Linguistics (ACL-1999)*, 1999.
125. Wiebe, J. and R. Mihalcea. Word sense and subjectivity. In *Proceedings of Intl. Conf. on Computational Linguistics and 44th Annual Meeting of the ACL (COLING/ACL-2006)*, 2006.
126. Wiebe, J. and E. Riloff. Creating subjective and objective sentence classifiers from unannotated texts. *Computational Linguistics and Intelligent Text Processing*, 2005: p. 486-497.
127. Wiebe, J., T. Wilson, R. Bruce, M. Bell, and M. Martin. Learning subjective language. *Computational Linguistics*, 2004, 30(3): p. 277-308.
128. Wiebe, J., T. Wilson, and C. Cardie. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 2005, 39(2): p. 165-210.
129. 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.
130. Wilson, T., J. Wiebe, and R. Hwa. Just how mad are you? Finding strong and weak opinion clauses. In *Proceedings of National Conference on Artificial Intelligence (AAAI-2004)*, 2004.
131. Wilson, T., J. Wiebe, and R. Hwa. Recognizing strong and weak opinion clauses. *Computational Intelligence*, 2006, 22(2): p. 73-99.
132. 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.
133. Yang, H., L. Si, and J. Callan. Knowledge transfer and opinion detection in the TREC2006 blog track. In *Proceedings of TREC*, 2006.

134. Yessenalina, A., Y. Yue, and C. Cardie. Multi-level Structured Models for Document-level Sentiment Classification. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-2010)*, 2010.
135. 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 IEEE International Conference on Data Mining (ICDM-2003)*, 2003.
136. Yu, H. and V. Hatzivassiloglou. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-2003)*, 2003.
137. 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.
138. 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.
139. Zhai, Z., B. Liu, L. Zhang, H. Xu, and P. Jia. *Identifying evaluative opinions in online discussions*. In *Proceedings of AAAI*. 2011.
140. Zhang, L. and B. Liu. *Identifying noun product features that imply opinions*. In *Proceedings of ACL (short paper),*. 2011.
141. 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.
142. Zhang, W., L. Jia, C. Yu, and W. Meng. Improve the effectiveness of the opinion retrieval and opinion polarity classification. In *Proceedings of ACM International Conference on Information and Knowledge Management (CIKM-2008)*, 2008.
143. Zhang, W. and C. Yu. *UIC at TREC 2007 Blog Report*. 2007.
144. Zhang, Z. and B. Varadarajan. Utility scoring of product reviews. In *Proceedings of ACM International Conference on Information and Knowledge Management (CIKM-2006)*, 2006.
145. 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.
146. 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.