Automated opinion mining and summarization systems are thus needed, as subjective biases and mental limitations can be overcome with an objective sentiment analysis system.

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 [36, 37]. A comparative opinion is usually expressed using the comparative or superlative form of an adjective or adverb, although not always.

Objective of opinion mining: Given a collection of opinionated documents D, discover all opinion quintuples (ei, aij , ooijkl, hk, tl) 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 ei. A Survey of Opinion Mining and Sentiment Analysis 419 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 ei indicates a unique aspect aij . Task 3 (opinion holder and time extraction): Extract these pieces of information from the text or unstructured 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 (ei, aij , ooijkl, hk, tl) expressed in D based on the results of the above tasks. We use an example blog to illustrate these tasks (a sentence id is associated with each sentence):

Definition 13.5 (Sentence Subjectivity) An objective sentence presents some factual information about the world, while a subjective sentence expresses some personal feelings, views or beliefs. There is some confusion among researchers to equate subjectivity with opinion. 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 [105], which we will discuss in Sect. 3.

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 [15, 18, 24, 37]. 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 as 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 as Type 2 comparatives and 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 [36], 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”. A Survey of Opinion Mining and Sentiment Analysis 443

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 a1, and entity B has aspect a2 (a1 and a2 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 (E1, E2, A, P E, h, t), where E1 and E2 are the entity sets being compared based on their shared aspects A (entities in E1 appear before entities in E2 in the sentence), P E(∈ {E1, E2}) is the preferred entity set of the opinion holder h, and t is the time when the comparative opinion is expressed.

Example 13.11 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 E1 is {Canon}, the entity set E2 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 [37], a method based on label sequential rules (LSR) is proposed to extract entities and aspects that are compared. A similar approach 444 MINING TEXT DATA is described in [54] for extracting the compared entities. Clearly, the approaches discussed in previous sections are applicable as well, and so are many other information extraction methods. See [37, 24, 18] for some existing methods for performing sentiment analysis of comparative sentences, i.e., identifying comparative sentences and identifying the preferred entity set.

Entity recognition:

Grouping aspect expressions indicating the same aspects: It is common that people use different words or phrases (which are called aspect expressions in Sect. 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 [69] 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 [12] 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. It requires a taxonomy of aspects to be given for a particular domain. The algorithm merges each discovered aspect 446 MINING TEXT DATA expression to an aspect node in the taxonomy. Experiments based on digital camera and DVD reviews showed promising results. In [114], 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 based on labeled seeds and unlabeled examples. The method used the Expectation-Maximization (EM) algorithm. Two pieces of prior knowledge were used to provide a better initialization for EM, i.e., (1) aspect expressions sharing some common words are likely to belong to the same group, and (2) aspect expressions that are synonyms in a dictionary are likely to belong to the same group.

Article 2:

3. Relation Extraction

Another important task in information extraction is relation extraction. Relation extraction is the task of detecting and characterizing the semantic relations between entities in text. For example, from the following sentence fragment, Facebook co-founder Mark Zuckerberg we can extract the following relation, FounderOf(Mark Zuckerberg, Facebook). Much of the work on relation extraction is based on the task definition from the Automatic Content Extraction (ACE) program [1]. ACE focuses on binary relations, i.e. relations between two entities. The two entities involved are also referred to as arguments. A set of major relation types and their subtypes are defined by ACE. Examples of ACE major relation types include physical (e.g. an entity is physically near another entity), personal/social (e.g. a person is a family member of another person), and employment/affiliation (e.g. a person is employed by an organization). ACE makes a distinction between relation extraction and relation mention extraction. The former refers to identifying the semantic relation between a pair of entities based on all the evidence we can gather from the corpus, whereas the latter refers to identifying individual mentions of entity relations. Because corpus-level relation extraction to a large extent still relies on accurate mention-level relation extraction, in the rest of this chapter we do not make any distinction between these two problems unless necessary. Various techniques have been proposed for relation extraction. The most common and straightforward approach is to treat the task as a classification problem: Given a pair of entities co-occurring in the same sentence, can we classify the relation between the two entities into one of the predefined relation types? Although it is also possible for relation mentions to cross sentence boundaries, such cases are less frequent and hard to detect. Existing work therefore mostly focuses on relation extraction within sentence boundaries. There have been a number of studies following the classification approach [38, 71, 37, 18, 19]. Feature engineering is the most critical step of this approach. An extension of the feature-based classification approach is to define kernels rather than features and to apply kernel machines such as support vector machines to perform classification. Ker- Information Extraction from Text 23 nels defined over word sequences [14], dependency trees [26], dependency paths [13] and parse trees [67, 68] have been proposed. Both feature-based and kernel-based classification methods require a large amount of training data. Another major line of work on relation extraction is weakly supervised relation extraction from large corpora that does not rely on the availability of manually labeled training data. One approach is the bootstrapping idea to start with a small set of seed examples and iteratively find new relation instances as well as new extraction patterns. Representative work includes the Snowball system [3]. Another approach is distant supervision that makes use of known relation instances from existing knowledge bases such as Freebase [50].

3.1 Feature-based Classification

A typical approach to relation extraction is to treat the task as a classification problem [38, 71, 37, 18, 19]. Specifically, any pair of entities co-occurring in the same sentence is considered a candidate relation instance. The goal is to assign a class label to this instance where the class label is either one of the predefined relation types or nil for unrelated entity pairs. Alternatively, a two-stage classification can be performed where at the first stage whether two entities are related is determined and at the second stage the relation type for each related entity pair is determined. Classification approach assumes that a training corpus exists in which all relation mentions for each predefined relation type have been manually annotated. These relation mentions are used as positive training examples. Entity pairs co-occurring in the same sentence but not labeled are used as negative training examples. Each candidate relation instance is represented by a set of features that are carefully chosen. Standard learning algorithms such as support vector machines and logistic regression can then be used to train relation classifiers. Feature engineering is a critical step for this classification approach. Researchers have examined a wide range of lexical, syntactic and semantic features.

Entity features: Oftentimes the two argument entities, including the entity words themselves and the entity types, are correlated with certain relation types. In the ACE data sets, for example, entity words such as father, mother, brother and sister and the person entity type are all strong indicators of the family relation subtype.

Lexical contextual features: Intuitively the contexts surrounding the two argument entities are important. The simplest way to incorporate evidence from contexts is to use lexical features. For example, if the word founded occurs between the two arguments, they are more likely to have the FounderOf relation.

Syntactic contextual features: Syntactic relations between the two arguments or between an argument and another word can often be useful. For example, if the first argument is the subject of the verb founded and the second argument is the object of the verb founded, then one can almost immediately tell that the FounderOf relation exists between the two arguments. Syntactic features can be derived from parse trees of the sentence containing the relation instance.

Background knowledge: Chan and Roth studied the use of background knowledge for relation extraction [18]. An example is to make use of Wikipedia. If two arguments co-occur in the same Wikipedia article, the content of the article can be used to check whether the two entities are related. Another example is word clusters. For example, if we can group all names of companies such as IBM and Apple into the same word cluster, we achieve a level of abstraction higher than words and lower than the general entity type organization. This level of abstraction may help extraction of certain relation types such as Acquire between two companies.