Sentiment Analysis

Promothesh Chatterjee*

Public *sentiment* is everything. With public *sentiment*, nothing can fail. Without it, nothing can succeed.

— Abraham Lincoln

Sentiment Analysis

The aim of Marketing is often defined as the ability of companies to meet customer needs profitably over a long period of time. This is only possible by satisfying the customers and differentiating the brand offering. Consequently, it is imperative to understand the ever changing consumer preferences to come up with appropriate branding and positioning strategies.

Given the burgeoning worldwide access to the internet, huge amount of text data is being generated which provides a promising way to discover consumer opinion about products and services. For instance, by writing blogs, sharing posts in social media, or reviewing products/services, consumers are constantly generating content. Mining this user generated content (UGC) can be an important way for companies to make informed decision on brand image and brand positioning.

Even in the field of finance, researchers have explored how the degree of positivity or negativity in texts (blogs, tweets, opinions, media articles, corporate disclosures) affects stocks prices. Human resource issues can also be addressed using sentiment analysis. Do the employees feel a sense of belongingness, or is there interpersonal conflict? Sentiment analysis can help us answer these questions and their impact on work, satisfaction, and engagement? Text analytic techniques on employee emails, social media comments, digital memos, etc. can be used to extract these insights.

Sentiment analysis finds uses in many areas. For instance, is the movie review positive or negative? How is the general sentiment of public toward economy? What do people think of a political candidate? So what exactly is sentiment analysis?

What is Sentiment Analysis?—

Sentiment analysis goes by many names- opinion extraction, opinion mining, sentiment mining, subjectivity analysis etc and refers to the usage of computational methods to study opinions, attitudes, and emotions found in text.

Using a more specific definition instead of general label of attitude/opinion/emotion etc., we can say sentiment analysis is the detection of attitudes "enduring, affectively (refers to emotions, moods etc., see below at Scherer's typology of affective states) colored beliefs, dispositions towards objects or persons". Typically, there is source or holder of the attitude, there is a target or aspect of the attitude, there is the type of attitude (love, hate etc.) or simply polarity of the attitude (positive, negative, neutral) along with it's strength. Furthermore, sentiment analysis could be at document level, sentence level or feature level.

Simplest task in sentiment analysis could be detecting whether the attitude of the text is positive or negative. A more complicated task could include rating the attitude of the text on a predefined scale (say, 1 to 5). More advanced sentiment tasks could include detecting the target, source and different types of attitudes.

Sentiment analysis operates at three levels: sentence level, document level, and aspect level. At the sentence level, documents or paragraphs are divided into sentences, and each sentence is analyzed to determine its

sentiment. At the document level, the overall sentiment of the entire document or text is assessed. This method focuses on extracting the general sentiment by ignoring repetitive details and noise. The main challenge in this type of analysis is to correctly interpret how different words and phrases throughout the document contribute to its overall sentiment, requiring a deep understanding of the complex relationships between sentiments and words. At the aspect level, the sentiment concerning specific features or aspects of a product is evaluated. For instance, in the statement "the processor speed is high, but this product is overpriced," the sentiments about specific aspects like speed and cost are analyzed. Here, speed is an explicit aspect as it is directly mentioned, whereas cost is an implicit aspect, hinted at but not directly stated. Aspect-level sentiment analysis is generally more challenging due to the difficulty in identifying these subtle features.

What makes reviews hard to classify? It is difficult for sentiment analysis systems to assess subtle meanings. For example, check out this sarcastic review for a perfume. "If you are reading this because it is your darling fragrance, please wear it at home exclusively, and tape the windows shut."

For a better understanding of sentiment analysis, it is important to understand some theories of affect as they can be very useful for:

- Formulating annotation schemes
- Understanding natural and derived labels and clusters
- Reducing the dimensionality of existing labels
- Finding useful sentiment contrasts and alignments

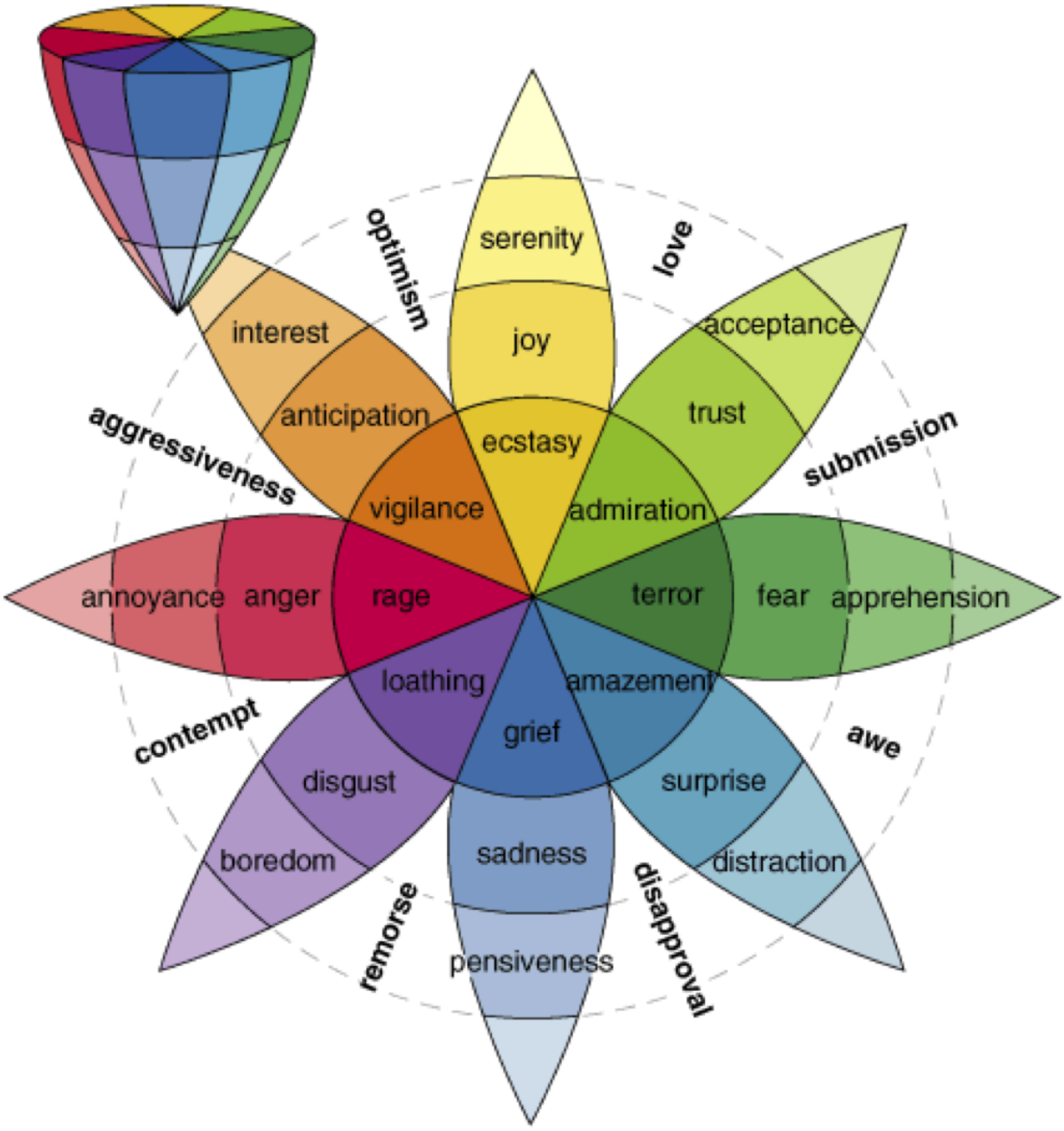
Popularly used theories of affect used in sentiment analysis include:

1) Scherer's Typology of Affective States

Type of affective state: brief definition (*examples*)	Intensity	Duration	Synchronization	Event focus	Appraisal elicitation	Rapidity of change	Behavioral impact
Emotion: relatively brief episode of synchronized response of all or most organismic subsystems in response to the evaluation of an external or internal event as being of major significance (*angry*, *sad*, *joyful*, *fearful*, *ashamed*, *proud*, *elated*, *desperate*)	+ +–+ ++	+	+ + +	+ + +	+ + +	+ + +	+ + +
Mood: diffuse affect state, most pronounced as change in subjective feeling, of low intensity but relatively long duration, often without apparent cause (*cheerful*, *gloomy*, *irritable*, *listless*, *depressed*, *buoyant*)	+–+ +	++	+	+	+	++	+
Interpersonal stances: affective stance taken toward another person in a specific interaction, colouring the interpersonal exchange in that situation (*distant*, *cold*, *warm*, *supportive*, *contemptuous*)	+–+ +	+–+ +	+	++	+	+ + +	++
Attitudes: relatively enduring, affectively coloured beliefs, preferences, and predispositions towards objects or persons (*liking*, *loving*, *hating*, *valueing*, *desiring*)	0–+ +	+ +–+ ++	0	0	+	0–+	+
Personality traits: emotionally laden, stable personality dispositions and behavior tendencies, typical for a person (*nervous*, *anxious*, *reckless*, *morose*, *hostile*, *envious*, *jealous*)	0–+	+ + +	0	0	0	0	+

0: low, +: medium, ++: high, + + +: very high, –: indicates a range.

2) Plutchik's (2002) Wheel of Emotions: is used to find out how emotions relate to one another. The color scheme helps in visualization of the emotions.



3) Ekman's (1985) theory of emotions connects facial expressions to six basic emotions: surprise, happiness, anger, fear, disgust, sadness

When building sentiment analysis related products, one must keep in mind the following things:

a) Sentiment is often multidimensional.
b) Writer and reader approach the sentiment information from different perspectives.
c) Because emotions and social interactions are complex, to understand the full picture we need to know who was talking, to whom, and why, before trying to understand their attitudes and emotions.
d) Human interaction is very context dependent, the same thing might mean different things in different contexts. For instance, what could be perceived as sincere in one context and could be wryly ironic in another.

Issues relating to Language Structure—

All we have done in terms of preprocessing is to tokenize, stem, remove punctuation, lower case the words etc. In sentiment analysis, special attention has to be given to preprocessing such as dealing with Twitter peculiarities such as names, hash tags, retweets, emoticons, capitalization for emphasis etc. Dealing with sentiment analysis, some more specific issues crop up.

1) Dealing with Negatives The basic idea is that semantic words behave differently under negation. For words which show relatively moderate polarity such as 'good' and 'bad', their negation 'not good' and 'not bad' behave like opposites. However, if you consider superlative words such as 'superb' and 'terrible', their negation ('not superb' and 'not terrible') is not really their opposite.

One approach to dealing with negatives could be to append a negative between every negation and clause level punctuation (Das and Chen 2001; Pang, Lee, and Vaithyanathan 2002). For instance, consider the

following sentence:

I don't think I will enjoy that movie: it is probably too sentimental.

The proposed algorithm will tokenize the sentence like this:

"i", "don't", "think_NEG", "i_NEG", "will_NEG", "enjoy_NEG", "that_NEG", "movie_NEG".

The algorithm has tokenized the word "enjoy" into two tokens, one explicitly negative and other outside the scope of negation making it easier for the sentiment analysis system to learn how the two behave. Negation marking is important for shorter documents, however longer documents generally have many sentiment cues and do not depend on a single one.

2) Scope marking This refers to marking quotation which report things like say, claim, etc.

For instance, consider the following sentence.

This "amazing" car was a piece of junk.

For attitude verbs, the strategy is the same as the one for negation: _REPORT marking between the relevant predicates and clause-level punctuation.

3) Part of Speech Tagging In order to assess the product features, research often uses a part-of-speech tagger to annotate each word with its partofspeech (POS)(whether a word is a noun, an adjective, a verb etc.). Typically nouns and noun phrases are popular possibilities for product features.

Look at the different pos tags that can be generated (this uses Penn Treebank, a popular POS tagset):

https://www.sketchengine.eu/penn-treebank-tagset/

POS tagging results in word–tag pairs which can be used as features or components of features for further analysis. We will explore this further in the section on using R for sentiment analysis.

4) Parsing Parsing means breaking a sentence into its constituent parts. Let's first understand the notion of noun and verb phrases with the help of some examples:

A rugged handset (Noun phrase) The red rose(Noun phrase) X is playing (Verb phrase) Y cannot run (Verb phrase)

Thus, noun/verb phrase is a group of words that act as a noun/verb in a sentence. POS tagging can tag individual word with its part-of-speech but may not help in understanding the sentence. How words combine together in a sentence give us a better understanding. For instance, a POS Tagger will tell us: 'The': Determiner, 'red': Adjective, 'rose': Noun

However, it does not specify that the color 'red' specifies the 'rose'. Thus to understand relationship between words, we need some sort of grouping. Hence parsing is important.

Parsing, in the context of natural language processing (NLP), is the process of analyzing a string of symbols, either in natural language or in computer languages, according to the rules of a formal grammar. The goal of parsing is to determine the structure of the input sentence or code according to a given set of grammar rules, and produce a parse tree that represents this structure.

There are two main types of parsing in NLP:

1. **Syntactic Parsing:** This involves determining the structure of a sentence. This can be further classified into constituency parsing and dependency parsing. Constituency parsing involves breaking a text down into sub-phrases or "constituents", whereas dependency parsing focuses on the grammatical relationships between words.
2. **Semantic Parsing:** This involves understanding the meaning of a sentence. It maps sentences to logical forms that represent their meaning.

In the context of sentiment analysis, parsing can be used to improve the accuracy of the analysis. Sentiment analysis is the process of determining the emotional tone behind words to understand the attitudes, opinions, and emotions of a speaker or writer. Here's how parsing can be used:

- **Identifying context and relationships:** Parsing can help identify the relationships between words in a sentence, which can provide context and help determine the overall sentiment. For example, in the sentence "I don't like this product," a parser can identify that "don't" is negating "like," leading to a negative sentiment.
- **Handling complex sentences:** Parsing can also help handle more complex sentences. For example, in the sentence "I like the product but the customer service was terrible," a parser can identify the contrast introduced by "but" and understand that the sentiment towards the "product" is positive, but the sentiment towards the "customer service" is negative.
- **Improving feature extraction:** In machine learning models used for sentiment analysis, parsing can help in feature extraction, which is the process of selecting the relevant pieces of data for analysis. Parsed data can make the feature extraction process more accurate, improving the overall performance of the sentiment analysis model.

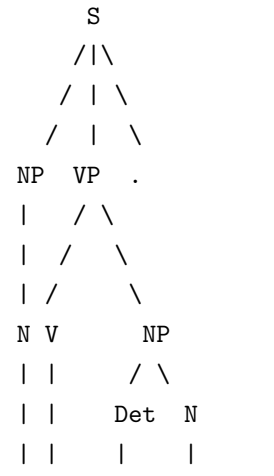
So, while parsing can be a complex and computationally intensive process, it can significantly enhance the performance and accuracy of sentiment analysis systems, especially when dealing with complex sentences and nuanced sentiments.

1) Syntactic parsing, also known as constituency parsing, involves breaking down a text into its grammatical components (constituents) and showing their syntactic relations to each other. This is typically represented as a tree structure, known as a parse tree.

The tree structure of syntactic parsing works as follows:

- The root of the tree is typically the Sentence (S) itself.
- Each internal node in the tree represents a constituent (a word or a group of words that function as a single unit). This could be a noun phrase (NP), verb phrase (VP), adjective phrase (AP), etc.
- Each leaf node (the nodes at the very bottom of the tree) represents a word in the sentence.
- The branches or edges of the tree show the relationships between these constituents.

For example, let's consider the sentence "Tom buys an iPhone". The parse tree of this sentence might look like this:



- `S` is the sentence.
- `NP` stands for Noun Phrase, `VP` stands for Verb Phrase, `Det` stands for Determiner (an article like "a" or "an"), `N` stands for Noun, and `V` stands for Verb.
- `.` represents the end of the sentence.
- In this case, "Tom" is a noun phrase, "buys an iPhone" is a verb phrase, and "an iPhone" is a noun phrase within the verb phrase.
- The tree structure shows that "Tom" is the subject of the sentence, "buys" is the verb, and "an iPhone" is the object.

This tree representation helps to understand the structure of the sentence and the relationships between its components. It's a crucial part of many natural language processing tasks, including machine translation, sentiment analysis, named entity recognition, and more.

2. Dependency parsing is a technique used in natural language processing that involves analyzing the grammatical structure of a sentence based on the dependencies between its words.

A "dependency" in this context refers to the grammatical relationship between words in a sentence. For instance, in the sentence "The cat sat on the mat," the word "sat" is dependent on "cat" (because "cat" is the one doing the sitting), and "on the mat" is dependent on "sat" (because "sat" is the action that is being further described).

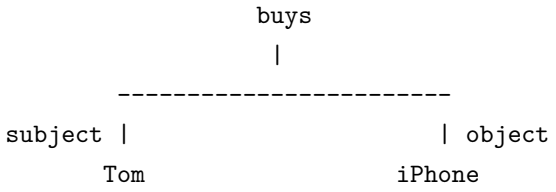
A dependency parser analyzes sentences in this way, producing a tree-like structure (a dependency tree or graph) that shows how the words in a sentence relate to one another. This tree can then be used for various applications, such as information extraction, machine translation, or sentiment analysis.

In a dependency tree, arrows are used to indicate dependencies, pointing from the "head" (the word that others depend on) to the "dependent" (the word that is dependent on the head). For instance, in the sentence "The cat sat on the mat," the word "cat" is the head of "The", and "sat" is the head of "cat".

Dependency parsing is a crucial part of understanding natural language because it helps us understand the relationships between words in a sentence, which in turn helps us understand the meaning of the sentence as a whole.

Dependency Parsing groups the words in a sentence based on the relationship between the words. For instance, the sentence 'Tom buys iPhone' has the following dependency parsing:

"buys" is the root of the sentence, as it's the main action. "Tom" is the subject (nsubj) of "buys". "iPhone" is the object (obj) of "buys".



Dependency parsing approach abstracts away unnecessary information, keeping only the necessary information. Dependency parsing further provides an approximation to the semantic relationship between predicates and their arguments which comes useful applications regarding question answering and information extraction etc.

Lexicon method to Sentiment Analysis

Using a lexicon or dictionary involves calculating the sentiment of the text from the semantic meaning of word or phrases. Typically, each word in the dictionary has a tag of a positive or negative sentiment value assigned to it. The way we approach sentiment analysis in lexicon-based approach by treating text as a bag of words and then assessing sentiment values from the dictionary. Finally, by summing or averaging, the overall sentiment for the text is computed.

Using a Classifier for Sentiment Analysis: Binarized Multinomial Naïve Bayes

A sentiment analysis task can be modeled as a classification problem too, where a certain classifier is given some text and returns a category, e.g. positive, negative, or neutral. Of course, this approach needs a training dataset for the classifier to work. We will briefly discuss a version of Naïve Bayes, called Binarized Multinomial Naïve Bayes when we look at the R implementation (of course, we discuss these methods and the underlying math in much greater details in our section on classification methods).

The basic idea here is that word occurrence matters more than word frequency. For instance, if the word 'terrible' occurs 5 times or once, it does not change the meaning. In Binarized Multinomial Naïve Bayes, the algorithm clips the unique word counts in each document at 1. The basic procedure is to remove all duplicate words from document and then perform Naive Bayes. Binary seems to work better than regular Naïve Bayes (Metsis et al. 2006). Note, this is not the same as Multivariate Bernoulli Naïve Bayes. One other option is to use log(freq(words)) instead of clipping each word frequency at 1.

Resources

1) Bing Liu Tutorial: https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html#lexicon

2) https://www.cs.uic.edu/~liub/FBS/NLP-handbook-sentiment-analysis.pdf

3) http://sentiment.christopherpotts.net/index.html

4) Dependency Parsing: https://web.stanford.edu/~jurafsky/slp3/15.pdf

References Scherer, Klaus R. 1984. Emotion as a Multicomponent Process: A model and some cross-cultural data. In P. Shaver, ed., Review of Personality and Social Psych 5: 37-63.

Ekman, Paul. 1985. Telling Lies: Clues to Deceit in the Marketplace, Politics, and Marriage. New York: Norton.

Sanjiv Das and Mike Chen. 2001. Yahoo! for Amazon: extracting market sentiment from stock message boards. In Proceedings of the 8th Asia Pacific Finance Association Annual Conference.

Pang, Bo; Lillian Lee; and Shivakumar Vaithyanathan. 2002. Thumbs up? sentiment classification using machine learning techniques. In Proceedings of the Conference on Empirical Methods in Natural Language Processing. ACL.

Plutchik, Robert. 2002. Emotions and Life. Washington, D.C.: American Psychological Association.

V. Metsis, I. Androutsopoulos, G. Paliouras. 2006. Spam Filtering with Naive Bayes – Which Naive Bayes? CEAS 2006 -Third Conference on Email and AntiSpam.