Speech and Language Processing. Daniel Jurafsky & James H. Martin. Copyright © 2026. All rights reserved. Draft of January 6, 2026.

CHAPTER

22 Lexicons for Sentiment, Affect, and Connotation

Some day we'll be able to measure the power of words
Maya Angelou

affective

In this chapter we turn to tools for interpreting **affective** meaning, extending our study of sentiment analysis in Appendix K. We use the word 'affective', following the tradition in **affective computing** (Picard, 1995) to mean emotion, sentiment, personality, mood, and attitudes. Affective meaning is closely related to **subjectivity**, the study of a speaker or writer's evaluations, opinions, emotions, and speculations (Wiebe et al., 1999).

subjectivity

How should affective meaning be defined? One influential typology of affective states comes from Scherer (2000), who defines each class of affective states by factors like its cognitive realization and time course (Fig. 22.1).

Emotion: Relatively brief episode of 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 stance: Affective stance taken toward another person in a specific interaction, coloring the interpersonal exchange in that situation.
(*distant, cold, warm, supportive, contemptuous, friendly*)

Attitude: Relatively enduring, affectively colored beliefs, preferences, and predispositions towards objects or persons.
(*liking, loving, hating, valuing, desiring*)

Personality traits: Emotionally laden, stable personality dispositions and behavior tendencies, typical for a person.
(*nervous, anxious, reckless, morose, hostile, jealous*)

Figure 22.1 The Scherer typology of affective states (Scherer, 2000).

We can design extractors for each of these kinds of affective states. Appendix K already introduced *sentiment analysis*, the task of extracting the positive or negative orientation that a writer expresses in a text. This corresponds in Scherer's typology to the extraction of **attitudes**: figuring out what people like or dislike, from affect-rich texts like consumer reviews of books or movies, newspaper editorials, or public sentiment in blogs or tweets.

Detecting **emotion** and **moods** is useful for detecting whether a student is confused, engaged, or certain when interacting with a tutorial system, whether a caller to a help line is frustrated, whether someone's blog posts or tweets indicated depression. Detecting emotions like fear in novels, for example, could help us trace what groups or situations are feared and how that changes over time.

Detecting different **interpersonal stances** can be useful when extracting information from human-human conversations. The goal here is to detect stances like friendliness or awkwardness in interviews or friendly conversations, for example for summarizing meetings or finding parts of a conversation where people are especially excited or engaged, conversational **hot spots** that can help in meeting summarization. Detecting the **personality** of a user—such as whether the user is an **extrovert** or the extent to which they are **open to experience**— can help improve conversational agents, which seem to work better if they match users' personality expectations (Mairesse and Walker, 2008). And affect is important for generation as well as recognition; synthesizing affect is important for conversational agents in various domains, including literacy tutors such as children's storybooks, or computer games.

In Appendix K we introduced the use of naive Bayes classification to classify a document's sentiment. Various classifiers have been successfully applied to many of these tasks, using all the words in the training set as input to a classifier which then determines the affect status of the text.

In this chapter we focus on an alternative model, in which instead of using every word as a feature, we focus only on certain words, ones that carry particularly strong cues to affect or sentiment. We call these lists of words **affective lexicons** or **sentiment lexicons**. These lexicons presuppose a fact about semantics: that words have *affective meanings* or **connotations**. The word *connotation* has different meanings in different fields, but here we use it to mean the aspects of a word's meaning that are related to a writer or reader's emotions, sentiment, opinions, or evaluations. In addition to their ability to help determine the affective status of a text, connotation lexicons can be useful features for other kinds of affective tasks, and for computational social science analysis.

connotations

In the next sections we introduce basic theories of emotion, show how sentiment lexicons are a special case of emotion lexicons, and mention some useful lexicons. We then survey three ways for building lexicons: human labeling, semi-supervised, and supervised. Finally, we talk about how to detect affect toward a particular entity, and introduce connotation frames.

22.1 Defining Emotion

emotion

One of the most important affective classes is **emotion**, which Scherer (2000) defines as a "relatively brief episode of response to the evaluation of an external or internal event as being of major significance".

Detecting emotion has the potential to improve a number of language processing tasks. Emotion recognition could help dialogue systems like tutoring systems detect that a student was unhappy, bored, hesitant, confident, and so on. Automatically detecting emotions in reviews or customer responses (anger, dissatisfaction, trust) could help businesses recognize specific problem areas or ones that are going well. Emotion can play a role in medical NLP tasks like helping diagnose depression or suicidal intent. Detecting emotions expressed toward characters in novels might play a role in understanding how different social groups were viewed by society at different times.

Computational models of emotion in NLP have mainly been based on two families of theories of emotion (out of the many studied in the field of affective science). In one of these families, emotions are viewed as fixed atomic units, limited in number, and from which others are generated, often called **basic emotions** (Tomkins

basic emotions

1962, Plutchik 1962), a model dating back to Darwin. Perhaps the most well-known of this family of theories are the 6 emotions proposed by Ekman (e.g., Ekman 1999) to be universally present in all cultures: *surprise, happiness, anger, fear, disgust, sadness*. Another atomic theory is the Plutchik (1980) wheel of emotion, consisting of 8 basic emotions in four opposing pairs: *joy–sadness*, *anger–fear*, *trust–disgust*, and *anticipation–surprise*, together with the emotions derived from them, shown in Fig. 22.2.

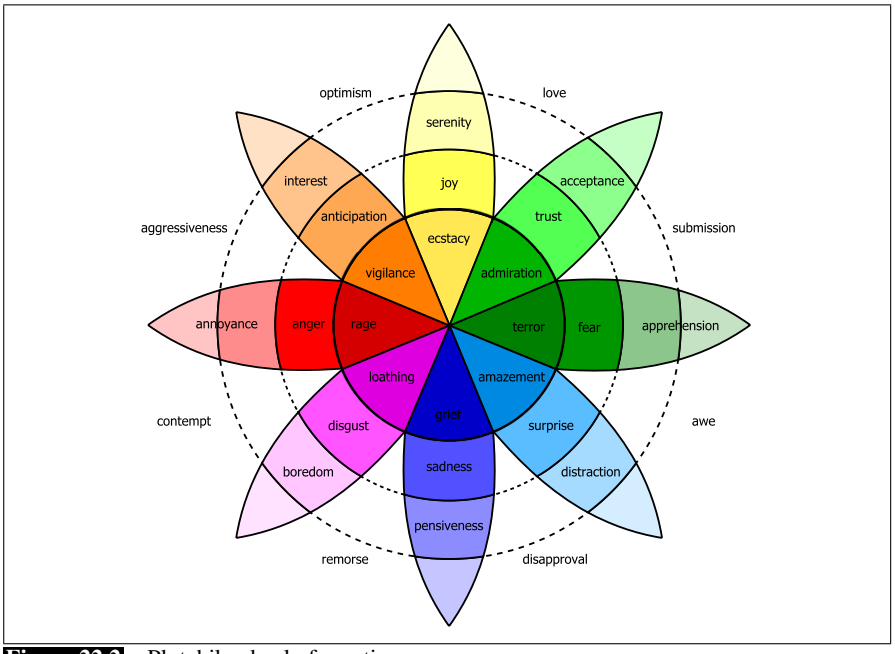

Figure 22.2 Plutchik wheel of emotion.

The second class of emotion theories widely used in NLP views emotion as a space in 2 or 3 dimensions (Russell, 1980). Most models include the two dimensions **valence** and **arousal**, and many add a third, **dominance**. These can be defined as:

valence: the pleasantness of the stimulus

arousal: the level of alertness, activeness, or energy provoked by the stimulus

dominance: the degree of control or dominance exerted by the stimulus or the emotion

Sentiment can be viewed as a special case of this second view of emotions as points in space. In particular, the **valence** dimension, measuring how pleasant or unpleasant a word is, is often used directly as a measure of sentiment.

In these lexicon-based models of affect, the affective meaning of a word is generally fixed, irrespective of the linguistic context in which a word is used, or the dialect or culture of the speaker. By contrast, other models in affective science represent emotions as much richer processes involving cognition (Barrett et al., 2007). In **appraisal theory**, for example, emotions are complex processes, in which a person considers how an event is congruent with their goals, taking into account variables like the agency, certainty, urgency, novelty and control associated with the event (Moors et al., 2013). Computational models in NLP taking into account these richer theories of emotion will likely play an important role in future work.

22.2 Available Sentiment and Affect Lexicons

A wide variety of affect lexicons have been created and released. The most basic lexicons label words along one dimension of semantic variability, generally called "sentiment" or "valence".

General Inquirer

In the simplest lexicons this dimension is represented in a binary fashion, with a wordlist for positive words and a wordlist for negative words. The oldest is the **General Inquirer** (Stone et al., 1966), which drew on content analysis and on early work in the cognitive psychology of word meaning (Osgood et al., 1957). The General Inquirer has a lexicon of 1915 positive words and a lexicon of 2291 negative words (as well as other lexicons discussed below). The MPQA Subjectivity lexicon (Wilson et al., 2005) has 2718 positive and 4912 negative words drawn from prior lexicons plus a bootstrapped list of subjective words and phrases (Riloff and Wiebe, 2003). Each entry in the lexicon is hand-labeled for sentiment and also labeled for reliability (strongly subjective or weakly subjective). The polarity lexicon of Hu and Liu (2004) gives 2006 positive and 4783 negative words, drawn from product reviews, labeled using a bootstrapping method from WordNet.

Positive	admire, amazing, assure, celebration, charm, eager, enthusiastic, excellent, fancy, fantastic, frolic, graceful, happy, joy, luck, majesty, mercy, nice, patience, perfect, proud, rejoice, relief, respect, satisfactorily, sensational, super, terrific, thank, vivid, wise, wonderful, zest
Negative	abominable, anger, anxious, bad, catastrophe, cheap, complaint, condescending, deceit, defective, disappointment, embarrass, fake, fear, filthy, fool, guilt, hate, idiot, inflict, lazy, miserable, mourn, nervous, objection, pest, plot, reject, scream, silly, terrible, unfriendly, vile, wicked

Figure 22.3 Some words with consistent sentiment across the General Inquirer (Stone et al., 1966), the MPQA Subjectivity lexicon (Wilson et al., 2005), and the polarity lexicon of Hu and Liu (2004).

Slightly more general than these sentiment lexicons are lexicons that assign each word a value on all three affective dimensions. The NRC Valence, Arousal, and Dominance (VAD) lexicon (Mohammad, 2018a) assigns valence, arousal, and dominance scores to 20,000 words. Some examples are shown in Fig. 22.4.

Valence		Arousal		Dominance	
vacation	.840	enraged	.962	powerful	.991
delightful	.918	party	.840	authority	.935
whistle	.653	organized	.337	saxophone	.482
consolation	.408	effortless	.120	discouraged	.0090
torture	.115	napping	.046	weak	.045

Figure 22.4 Values of sample words on the emotional dimensions of Mohammad (2018a).

EmoLex

The NRC Word-Emotion Association Lexicon, also called **EmoLex** (Mohammad and Turney, 2013), uses the Plutchik (1980) 8 basic emotions defined above. The lexicon includes around 14,000 words including words from prior lexicons as well as frequent nouns, verbs, adverbs and adjectives. Values from the lexicon for some sample words:

Word	anger	anticipation	disgust	fear	joy	sadness	surprise	trust	positive	negative
reward	0	1	0	0	1	0	1	1	1	0
worry	0	1	0	1	0	1	0	0	0	1
tenderness	0	0	0	0	1	0	0	0	1	0
sweetheart	0	1	0	0	1	1	0	1	1	0
suddenly	0	0	0	0	0	0	1	0	0	0
thirst	0	1	0	0	0	1	1	0	0	0
garbage	0	0	1	0	0	0	0	0	0	1

For a smaller set of 5,814 words, the NRC Emotion/Affect Intensity Lexicon (Mohammad, 2018b) contains real-valued scores of association for anger, fear, joy, and sadness; Fig. 22.5 shows examples.

Anger		Fear		Joy		Sadness	
outraged	0.964	horror	0.923	superb	0.864	sad	0.844
violence	0.742	anguish	0.703	cheered	0.773	guilt	0.750
coup	0.578	pestilence	0.625	rainbow	0.531	unkind	0.547
oust	0.484	stressed	0.531	gesture	0.387	difficulties	0.421
suspicious	0.484	failing	0.531	warms	0.391	beggar	0.422
nurture	0.059	confident	0.094	hardship	.031	sing	0.017

Figure 22.5 Sample emotional intensities for words for anger, fear, joy, and sadness from Mohammad (2018b).

LIWC

LIWC, **Linguistic Inquiry and Word Count**, is a widely used set of 73 lexicons containing over 2300 words (Pennebaker et al., 2007), designed to capture aspects of lexical meaning relevant for social psychological tasks. In addition to sentiment-related lexicons like ones for negative emotion (*bad, weird, hate, problem, tough*) and positive emotion (*love, nice, sweet*), LIWC includes lexicons for categories like anger, sadness, cognitive mechanisms, perception, tentative, and inhibition, shown in Fig. 22.6.

There are various other hand-built affective lexicons. The General Inquirer includes additional lexicons for dimensions like strong vs. weak, active vs. passive, overstated vs. understated, as well as lexicons for categories like pleasure, pain, virtue, vice, motivation, and cognitive orientation.

concrete
abstract

Another useful feature for various tasks is the distinction between **concrete** words like *banana* or *bathrobe* and **abstract** words like *belief* and *although*. The lexicon in Brysbaert et al. (2014) used crowdsourcing to assign a rating from 1 to 5 of the concreteness of 40,000 words, thus assigning *banana*, *bathrobe*, and *bagel* 5, *belief* 1.19, *although* 1.07, and in between words like *brisk* a 2.5.

22.3 Creating Affect Lexicons by Human Labeling

crowdsourcing

The earliest method used to build affect lexicons, and still in common use, is to have humans label each word. This is now most commonly done via **crowdsourcing**: breaking the task into small pieces and distributing them to a large number of anno-

Positive Emotion	Negative Emotion	Insight	Inhibition	Family	Negate
appreciat*	anger*	aware*	avoid*	brother*	aren't
comfort*	bore*	believe	careful*	cousin*	cannot
great	cry	decid*	hesitat*	daughter*	didn't
happy	despair*	feel	limit*	family	neither
interest	fail*	figur*	oppos*	father*	never
joy*	fear	know	prevent*	grandf*	no
perfect*	griev*	knew	reluctan*	grandm*	nobod*
please*	hate*	means	safe*	husband	none
safe*	panic*	notice*	stop	mom	nor
terrific	suffers	recogni*	stubborn*	mother	nothing
value	terrify	sense	wait	niece*	nowhere
wow*	violent*	think	wary	wife	without

Figure 22.6 Samples from 5 of the 73 lexical categories in LIWC (Pennebaker et al., 2007). The * means the previous letters are a word prefix and all words with that prefix are included in the category.

tators. Let's take a look at some of the methodological choices for two crowdsourced emotion lexicons.

The NRC Emotion Lexicon (EmoLex) (Mohammad and Turney, 2013), labeled emotions in two steps. To ensure that the annotators were judging the correct sense of the word, they first answered a multiple-choice synonym question that primed the correct sense of the word (without requiring the annotator to read a potentially confusing sense definition). These were created automatically using the headwords associated with the thesaurus category of the sense in question in the Macquarie dictionary and the headwords of 3 random distractor categories. An example:

```
Which word is closest in meaning (most related) to startle?
```

- `automobile`
- `shake`
- `honesty`
- `entertain`

For each word (e.g. *startle*), the annotator was then asked to rate how associated that word is with each of the 8 emotions (*joy*, *fear*, *anger*, etc.). The associations were rated on a scale of *not*, *weakly*, *moderately*, and *strongly* associated. Outlier ratings were removed, and then each term was assigned the class chosen by the majority of the annotators, with ties broken by choosing the stronger intensity, and then the 4 levels were mapped into a binary label for each word (no and weak mapped to 0, moderate and strong mapped to 1).

best-worst scaling

The NRC VAD Lexicon (Mohammad, 2018a) was built by selecting words and emoticons from prior lexicons and annotating them with crowd-sourcing using **best-worst scaling** (Louviere et al. 2015, Kiritchenko and Mohammad 2017). In best-worst scaling, annotators are given N items (usually 4) and are asked which item is the **best** (highest) and which is the **worst** (lowest) in terms of some property. The set of words used to describe the ends of the scales are taken from prior literature. For valence, for example, the raters were asked:

> Q1. Which of the four words below is associated with the MOST happiness / pleasure / positiveness / satisfaction / contentedness / hopefulness OR LEAST unhappiness / annoyance / negativeness / dissatisfaction /

melancholy / despair? (Four words listed as options.)
Q2. Which of the four words below is associated with the LEAST happiness / pleasure / positiveness / satisfaction / contentedness / hopefulness OR MOST unhappiness / annoyance / negativeness / dissatisfaction / melancholy / despair? (Four words listed as options.)

The score for each word in the lexicon is the proportion of times the item was chosen as the best (highest V/A/D) minus the proportion of times the item was chosen as the worst (lowest V/A/D). The agreement between annotations are evaluated by **split-half reliability**: split the corpus in half and compute the correlations between the annotations in the two halves.

split-half reliability

22.4 Semi-supervised Induction of Affect Lexicons

Another common way to learn sentiment lexicons is to start from a set of seed words that define two poles of a semantic axis (words like *good* or *bad*), and then find ways to label each word w by its similarity to the two seed sets. Here we summarize two families of seed-based semi-supervised lexicon induction algorithms, axis-based and graph-based.

22.4.1 Semantic Axis Methods

One of the most well-known lexicon induction methods, the Turney and Littman (2003) algorithm, is given seed words like *good* or *bad*, and then for each word w to be labeled, measures both how similar it is to *good* and how different it is from *bad*. Here we describe a slight extension of the algorithm due to An et al. (2018), which is based on computing a **semantic axis**.

In the first step, we choose seed words by hand. There are two methods for dealing with the fact that the affect of a word is different in different contexts: (1) start with a single large seed lexicon and rely on the induction algorithm to finetune it to the domain, or (2) choose different seed words for different genres. Hellrich et al. (2019) suggests that for modeling affect across different historical time periods, starting with a large modern affect dictionary is better than small seedsets tuned to be stable across time. As an example of the second approach, Hamilton et al. (2016) define one set of seed words for general sentiment analysis, a different set for Twitter, and yet another set for sentiment in financial text:

Domain	Positive seeds	Negative seeds
General	good, lovely, excellent, fortunate, pleasant, delightful, perfect, loved, love, happy	bad, horrible, poor, unfortunate, unpleasant, disgusting, evil, hated, hate, unhappy
Twitter	love, loved, loves, awesome, nice, amazing, best, fantastic, correct, happy	hate, hated, hates, terrible, nasty, awful, worst, horrible, wrong, sad
Finance	successful, excellent, profit, beneficial, improving, improved, success, gains, positive	negligent, loss, volatile, wrong, losses, damages, bad, litigation, failure, down, negative

In the second step, we compute embeddings for each of the pole words. These embeddings can be off-the-shelf word2vec embeddings, or can be computed directly

on a specific corpus (for example using a financial corpus if a finance lexicon is the goal), or we can finetune off-the-shelf embeddings to a corpus. Fine-tuning is especially important if we have a very specific genre of text but don't have enough data to train good embeddings. In finetuning, we begin with off-the-shelf embeddings like word2vec, and continue training them on the small target corpus.

Once we have embeddings for each pole word, we create an embedding that represents each pole by taking the centroid of the embeddings of each of the seed words; recall that the centroid is the multidimensional version of the mean. Given a set of embeddings for the positive seed words $S^+ = \{E(w_1^+), E(w_2^+), ..., E(w_n^+)\}$, and embeddings for the negative seed words $S^- = \{E(w_1^-), E(w_2^-), ..., E(w_m^-)\}$, the pole centroids are:

$$\mathbf{V}^+ = \frac{1}{n}\sum_{1}^{n} E(w_i^+)$$
$$\mathbf{V}^- = \frac{1}{m}\sum_{1}^{m} E(w_i^-) \tag{22.1}$$

The semantic axis defined by the poles is computed just by subtracting the two vectors:

$$\mathbf{V}_{axis} = \mathbf{V}^+ - \mathbf{V}^- \tag{22.2}$$

$\mathbf{V}_{axis}$, the semantic axis, is a vector in the direction of positive sentiment. Finally, we compute (via cosine similarity) the angle between the vector in the direction of positive sentiment and the direction of w's embedding. A higher cosine means that w is more aligned with S^+ than S^-.

$$\begin{aligned} \text{score}(w) &= \cos\left(E(w), \mathbf{V}_{\text{axis}}\right) \\ &= \frac{E(w)\cdot \mathbf{V}_{\text{axis}}}{\|E(w)\|\|\mathbf{V}_{\text{axis}}\|} \end{aligned} \tag{22.3}$$

If a dictionary of words with sentiment scores is sufficient, we're done! Or if we need to group words into a positive and a negative lexicon, we can use a threshold or other method to give us discrete lexicons.

22.4.2 Label Propagation

An alternative family of methods defines lexicons by propagating sentiment labels on graphs, an idea suggested in early work by Hatzivassiloglou and McKeown (1997). We'll describe the simple SentProp (Sentiment Propagation) algorithm of Hamilton et al. (2016), which has four steps:

1. **Define a graph**: Given word embeddings, build a weighted lexical graph by connecting each word with its k nearest neighbors (according to cosine similarity). The weights of the edge between words w_i and w_j are set as:

$$\mathbf{E}_{i,j} = \arccos\left(-\frac{\mathbf{w_i}^\top \mathbf{w_j}}{\|\mathbf{w_i}\|\|\mathbf{w_j}\|}\right). \tag{22.4}$$

2. **Define a seed set:** Choose positive and negative seed words.

3. **Propagate polarities from the seed set:** Now we perform a random walk on this graph, starting at the seed set. In a random walk, we start at a node and then choose a node to move to with probability proportional to the edge probability. A word's polarity score for a seed set is proportional to the probability of a random walk from the seed set landing on that word (Fig. 22.7).
4. **Create word scores**: We walk from both positive and negative seed sets, resulting in positive ($\text{rawscore}^+(w_i)$) and negative ($\text{rawscore}^-(w_i)$) raw label scores. We then combine these values into a positive-polarity score as:

$$\text{score}^+(w_i) = \frac{\text{rawscore}^+(w_i)}{\text{rawscore}^+(w_i) + \text{rawscore}^-(w_i)} \tag{22.5}$$

   It's often helpful to standardize the scores to have zero mean and unit variance within a corpus.
5. **Assign confidence to each score:** Because sentiment scores are influenced by the seed set, we'd like to know how much the score of a word would change if a different seed set is used. We can use bootstrap sampling to get confidence regions, by computing the propagation B times over random subsets of the positive and negative seed sets (for example using $B = 50$ and choosing 7 of the 10 seed words each time). The standard deviation of the bootstrap sampled polarity scores gives a confidence measure.

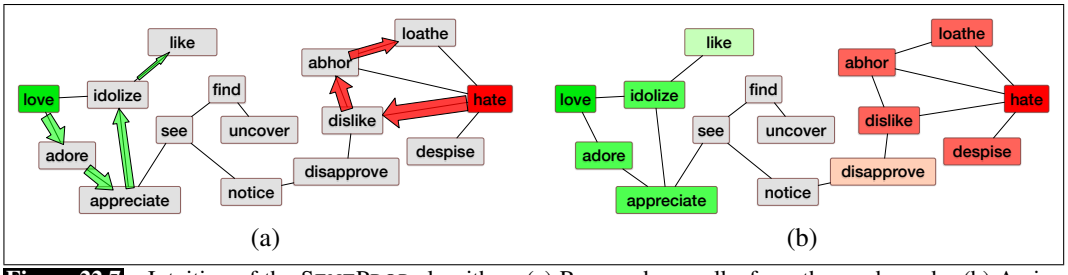

Figure 22.7 Intuition of the SENTPROP algorithm. (a) Run random walks from the seed words. (b) Assign polarity scores (shown here as colors green or red) based on the frequency of random walk visits.

22.4.3 Other Methods

The core of semisupervised algorithms is the metric for measuring similarity with the seed words. The Turney and Littman (2003) and Hamilton et al. (2016) approaches above used embedding cosine as the distance metric: words were labeled as positive basically if their embeddings had high cosines with positive seeds and low cosines with negative seeds. Other methods have chosen other kinds of distance metrics besides embedding cosine.

For example the Hatzivassiloglou and McKeown (1997) algorithm uses syntactic cues; two adjectives are considered similar if they were frequently conjoined by *and* and rarely conjoined by *but*. This is based on the intuition that adjectives conjoined by the words *and* tend to have the same polarity; positive adjectives are generally coordinated with positive, negative with negative:

fair and legitimate, corrupt and brutal

but less often positive adjectives coordinated with negative:

*fair and brutal, *corrupt and legitimate

By contrast, adjectives conjoined by *but* are likely to be of opposite polarity:

fair but brutal

Another cue to opposite polarity comes from morphological negation (*un-*, *im-*, *-less*). Adjectives with the same root but differing in a morphological negative (*adequate/inadequate*, *thoughtful/thoughtless*) tend to be of opposite polarity.

Yet another method for finding words that have a similar polarity to seed words is to make use of a thesaurus like WordNet (Kim and Hovy 2004, Hu and Liu 2004). A word's synonyms presumably share its polarity while a word's antonyms probably have the opposite polarity. After a seed lexicon is built, each lexicon is updated as follows, possibly iterated.

Lex^+: Add synonyms of positive words (*well*) and antonyms (like *fine*) of negative words

Lex^-: Add synonyms of negative words (*awful*) and antonyms (like *evil*) of positive words

SentiWordNet

An extension of this algorithm assigns polarity to WordNet senses, called **SentiWordNet** (Baccianella et al., 2010). Fig. 22.8 shows some examples.

Synset	Pos	Neg	Obj
good#6 'agreeable or pleasing'	1	0	0
respectable#2 honorable#4 good#4 estimable#2 'deserving of esteem'	0.75	0	0.25
estimable#3 computable#1 'may be computed or estimated'	0	0	1
sting#1 burn#4 bite#2 'cause a sharp or stinging pain'	0	0.875	.125
acute#6 'of critical importance and consequence'	0.625	0.125	.250
acute#4 'of an angle; less than 90 degrees'	0	0	1
acute#1 'having or experiencing a rapid onset and short but severe course'	0	0.5	0.5

Figure 22.8 Examples from SentiWordNet 3.0 (Baccianella et al., 2010). Note the differences between senses of homonymous words: *estimable#3* is purely objective, while *estimable#2* is positive; *acute* can be positive (*acute#6*), negative (*acute#1*), or neutral (*acute #4*).

In this algorithm, polarity is assigned to entire synsets rather than words. A positive lexicon is built from all the synsets associated with 7 positive words, and a negative lexicon from synsets associated with 7 negative words. A classifier is then trained from this data to take a WordNet gloss and decide if the sense being defined is positive, negative or neutral. A further step (involving a random-walk algorithm) assigns a score to each WordNet synset for its degree of positivity, negativity, and neutrality.

In summary, semisupervised algorithms use a human-defined set of seed words for the two poles of a dimension, and use similarity metrics like embedding cosine, coordination, morphology, or thesaurus structure to score words by how similar they are to the positive seeds and how dissimilar to the negative seeds.

22.5 Supervised Learning of Word Sentiment

Semi-supervised methods require only minimal human supervision (in the form of seed sets). But sometimes a supervision signal exists in the world and can be made use of. One such signal is the scores associated with *online reviews*.

The web contains an enormous number of online reviews for restaurants, movies, books, or other products, each of which have the text of the review along with an

associated review score: a value that may range from 1 star to 5 stars, or scoring 1 to 10. Fig. 22.9 shows samples extracted from restaurant, book, and movie reviews.

	Movie review excerpts (IMDb)
10	A great movie. This film is just a wonderful experience. It's surreal, zany, witty and slapstick all at the same time. And terrific performances too.
1	This was probably the worst movie I have ever seen. The story went nowhere even though they could have done some interesting stuff with it.
	Restaurant review excerpts (Yelp)
5	The service was impeccable. The food was cooked and seasoned perfectly... The watermelon was perfectly square ... The grilled octopus was ... mouthwatering...
2	...it took a while to get our waters, we got our entree before our starter, and we never received silverware or napkins until we requested them...
	Book review excerpts (GoodReads)
1	I am going to try and stop being deceived by eye-catching titles. I so wanted to like this book and was so disappointed by it.
5	This book is hilarious. I would recommend it to anyone looking for a satirical read with a romantic twist and a narrator that keeps butting in
	Product review excerpts (Amazon)
5	The lid on this blender though is probably what I like the best about it... enables you to pour into something without even taking the lid off! ... the perfect pitcher! ... works fantastic.
1	I hate this blender... It is nearly impossible to get frozen fruit and ice to turn into a smoothie... You have to add a TON of liquid. I also wish it had a spout ...

Figure 22.9 Excerpts from some reviews from various review websites, all on a scale of 1 to 5 stars except IMDb, which is on a scale of 1 to 10 stars.

We can use this review score as supervision: positive words are more likely to appear in 5-star reviews; negative words in 1-star reviews. And instead of just a binary polarity, this kind of supervision allows us to assign a word a more complex representation of its polarity: its distribution over stars (or other scores).

Thus in a ten-star system we could represent the sentiment of each word as a 10-tuple, each number a score representing the word's association with that polarity level. This association can be a raw count, or a likelihood $P(w|c)$, or some other function of the count, for each class c from 1 to 10.

For example, we could compute the IMDb likelihood of a word like *disappoint(ed/ing)* occurring in a 1 star review by dividing the number of times *disappoint(ed/ing)* occurs in 1-star reviews in the IMDb dataset (8,557) by the total number of words occurring in 1-star reviews (25,395,214), so the IMDb estimate of $P(disappointing|1)$ is .0003.

A slight modification of this weighting, the normalized likelihood, can be used as an illuminating visualization (Potts, 2011)[1]

$$\begin{aligned} P(w|c) &= \frac{count(w,c)}{\sum_{w \in C} count(w,c)} \\ PottsScore(w) &= \frac{P(w|c)}{\sum_c P(w|c)} \end{aligned} \tag{22.6}$$

Dividing the IMDb estimate $P(disappointing|1)$ of .0003 by the sum of the likelihood $P(w|c)$ over all categories gives a Potts score of 0.10. The word *disappointing*

[1] Each element of the Potts score of a word w and category c can be shown to be a variant of the pointwise mutual information $\text{pmi}(w,c)$ without the log term; see Exercise 22.1.

thus is associated with the vector [.10, .12, .14, .14, .13, .11, .08, .06, .06, .05]. The

Potts diagram

Potts diagram (Potts, 2011) is a visualization of these word scores, representing the prior sentiment of a word as a distribution over the rating categories.

Fig. 22.10 shows the Potts diagrams for 3 positive and 3 negative scalar adjectives. Note that the curve for strongly positive scalars have the shape of the letter J, while strongly negative scalars look like a reverse J. By contrast, weakly positive and negative scalars have a hump-shape, with the maximum either below the mean (weakly negative words like *disappointing*) or above the mean (weakly positive words like *good*). These shapes offer an illuminating typology of affective meaning.

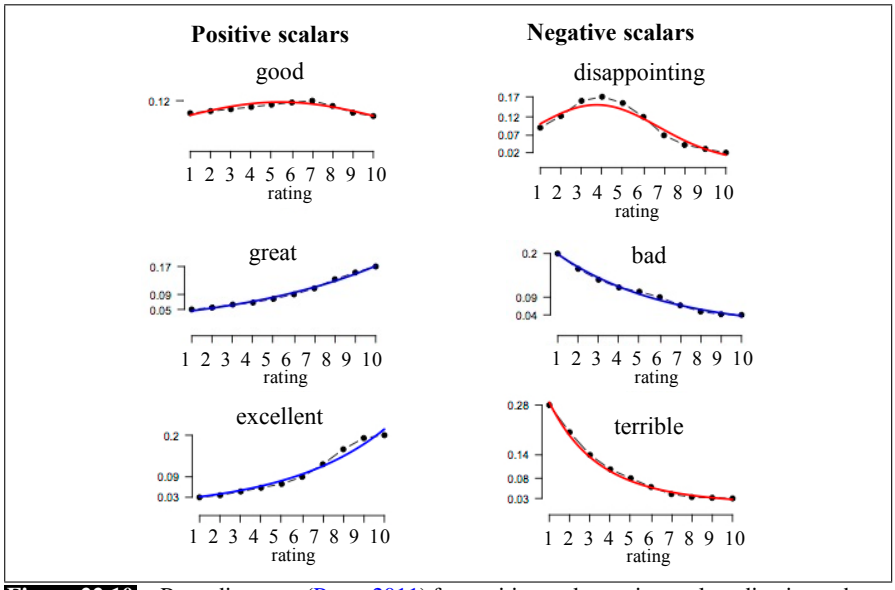

Figure 22.10 Potts diagrams (Potts, 2011) for positive and negative scalar adjectives, showing the J-shape and reverse J-shape for strongly positive and negative adjectives, and the hump-shape for more weakly polarized adjectives.

Fig. 22.11 shows the Potts diagrams for emphasizing and attenuating adverbs. Note that emphatics tend to have a J-shape (most likely to occur in the most positive reviews) or a U-shape (most likely to occur in the strongly positive and negative). Attenuators all have the hump-shape, emphasizing the middle of the scale and downplaying both extremes. The diagrams can be used both as a typology of lexical sentiment, and also play a role in modeling sentiment compositionality.

In addition to functions like posterior $P(c|w)$, likelihood $P(w|c)$, or normalized likelihood (Eq. 22.6) many other functions of the count of a word occurring with a sentiment label have been used. We'll introduce some of these on page 16, including ideas like normalizing the counts per writer in Eq. 22.14.

22.5.1 Log Odds Ratio Informative Dirichlet Prior

One thing we often want to do with word polarity is to distinguish between words that are more likely to be used in one category of texts than in another. We may, for example, want to know the words most associated with 1 star reviews versus those associated with 5 star reviews. These differences may not be just related to sentiment. We might want to find words used more often by Democratic than Republican

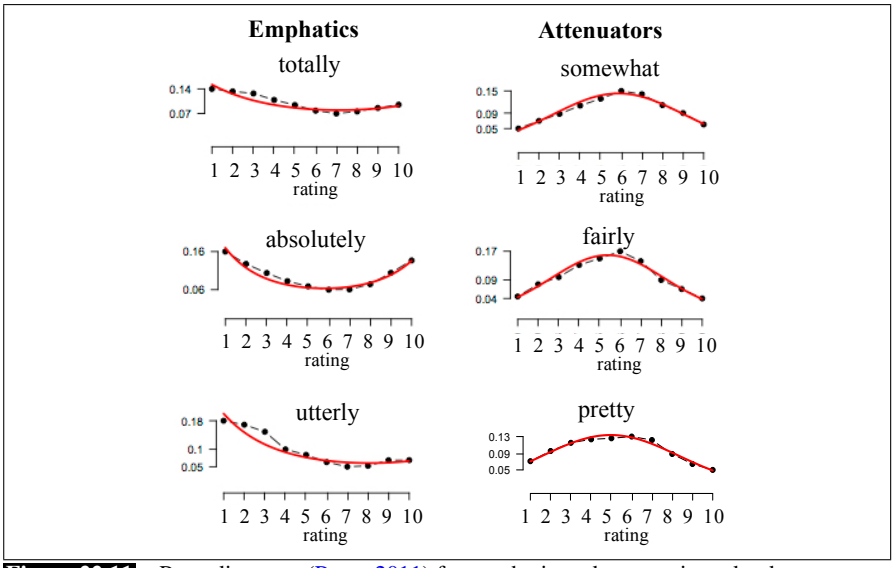

Figure 22.11 Potts diagrams (Potts, 2011) for emphatic and attenuating adverbs.

members of Congress, or words used more often in menus of expensive restaurants than cheap restaurants.

Given two classes of documents, to find words more associated with one category than another, we could measure the difference in frequencies (is a word w more frequent in class A or class B?). Or instead of the difference in frequencies we could compute the ratio of frequencies, or compute the log odds ratio (the log of the ratio between the odds of the two words). We could then sort words by whichever association measure we pick, ranging from words overrepresented in category A to words overrepresented in category B.

The problem with simple log-likelihood or log odds methods is that they overemphasize differences in very rare words, and often also in very frequent words. Very rare words will seem to occur very differently in the two corpora since with tiny counts there may be statistical fluctuations, or even zero occurrences in one corpus compared to non-zero occurrences in the other. Very frequent words will also seem different since all counts are large.

In this section we walk through the details of one solution to this problem: the "log odds ratio informative Dirichlet prior" method of Monroe et al. (2008) that is a particularly useful method for finding words that are statistically overrepresented in one particular category of texts compared to another. It's based on the idea of using another large corpus to get a prior estimate of what we expect the frequency of each word to be.

log likelihood ratio

Let's start with the goal: assume we want to know whether the word *horrible* occurs more in corpus i or corpus j. We could compute the **log likelihood ratio**, using $f^i(w)$ to mean the frequency of word w in corpus i, and n^i to mean the total number of words in corpus i:

$$\begin{aligned}\text{llr}(horrible) &= \log \frac{P^i(horrible)}{P^j(horrible)} \\ &= \log P^i(horrible) - \log P^j(horrible) \\ &= \log \frac{\text{f}^i(horrible)}{n^i} - \log \frac{\text{f}^j(horrible)}{n^j} \end{aligned} \tag{22.7}$$

log odds ratio

Instead, let's compute the **log odds ratio**: does *horrible* have higher odds in i or in j:

$$\begin{aligned}
\text{lor}(horrible) &= \log\left(\frac{P^i(horrible)}{1-P^i(horrible)}\right) - \log\left(\frac{P^j(horrible)}{1-P^j(horrible)}\right) \\
&= \log\left(\frac{\frac{\text{f}^i(horrible)}{n^i}}{1-\frac{\text{f}^i(horrible)}{n^i}}\right) - \log\left(\frac{\frac{\text{f}^j(horrible)}{n^j}}{1-\frac{\text{f}^j(horrible)}{n^j}}\right) \\
&= \log\left(\frac{\text{f}^i(horrible)}{n^i-\text{f}^i(horrible)}\right) - \log\left(\frac{\text{f}^j(horrible)}{n^j-\text{f}^j(horrible)}\right) \qquad (22.8)
\end{aligned}$$

The Dirichlet intuition is to use a large background corpus to get a prior estimate of what we expect the frequency of each word w to be. We'll do this very simply by adding the counts from that corpus to the numerator and denominator, so that we're essentially shrinking the counts toward that prior. It's like asking how large are the differences between i and j given what we would expect given their frequencies in a well-estimated large background corpus.

The method estimates the difference between the frequency of word w in two corpora i and j via the prior-modified log odds ratio for w, $\delta_w^{(i-j)}$, which is estimated as:

$$\delta_w^{(i-j)} = \log\left(\frac{f_w^i+\alpha_w}{n^i+\alpha_0-(f_w^i+\alpha_w)}\right) - \log\left(\frac{f_w^j+\alpha_w}{n^j+\alpha_0-(f_w^j+\alpha_w)}\right) \qquad (22.9)$$

(where n^i is the size of corpus i, n^j is the size of corpus j, f_w^i is the count of word w in corpus i, f_w^j is the count of word w in corpus j, α_0 is the scaled size of the background corpus, and α_w is the scaled count of word w in the background corpus.)

In addition, Monroe et al. (2008) make use of an estimate for the variance of the log–odds–ratio:

$$\sigma^2\left(\hat{\delta}_w^{(i-j)}\right) \approx \frac{1}{f_w^i+\alpha_w} + \frac{1}{f_w^j+\alpha_w} \qquad (22.10)$$

The final statistic for a word is then the z–score of its log–odds–ratio:

$$\frac{\hat{\delta}_w^{(i-j)}}{\sqrt{\sigma^2\left(\hat{\delta}_w^{(i-j)}\right)}} \qquad (22.11)$$

The Monroe et al. (2008) method thus modifies the commonly used log odds ratio in two ways: it uses the z-scores of the log odds ratio, which controls for the amount of variance in a word's frequency, and it uses counts from a background corpus to provide a prior count for words.

Fig. 22.12 shows the method applied to a dataset of restaurant reviews from Yelp, comparing the words used in 1-star reviews to the words used in 5-star reviews (Jurafsky et al., 2014). The largest difference is in obvious sentiment words, with the 1-star reviews using negative sentiment words like *worse, bad, awful* and the 5-star reviews using positive sentiment words like *great, best, amazing*. But there are other illuminating differences. 1-star reviews use logical negation (*no, not*), while 5-star reviews use emphatics and emphasize universality (*very, highly, every, always*). 1-star reviews use first person plurals (*we, us, our*) while 5 star reviews use the second

person. 1-star reviews talk about people (*manager, waiter, customer*) while 5-star reviews talk about dessert and properties of expensive restaurants like courses and atmosphere. See Jurafsky et al. (2014) for more details.

Class	Words in 1-star reviews	Class	Words in 5-star reviews
Negative	*worst, rude, terrible, horrible, bad, awful, disgusting, bland, tasteless, gross, mediocre, overpriced, worse, poor*	**Positive**	*great, best, love(d), delicious, amazing, favorite, perfect, excellent, awesome, friendly, fantastic, fresh, wonderful, incredible, sweet, yum(my)*
Negation	*no, not*	**Emphatics/ universals**	*very, highly, perfectly, definitely, absolutely, everything, every, always*
1Pl pro	*we, us, our*	**2 pro**	*you*
3 pro	*she, he, her, him*	**Articles**	*a, the*
Past verb	*was, were, asked, told, said, did, charged, waited, left, took*	**Advice**	*try, recommend*
Sequencers	*after, then*	**Conjunct**	*also, as, well, with, and*
Nouns	*manager, waitress, waiter, customer, customers, attitude, waste, poisoning, money, bill, minutes*	**Nouns**	*atmosphere, dessert, chocolate, wine, course, menu*
Irrealis modals	*would, should*	**Auxiliaries**	*is/'s, can, 've, are*
Comp	*to, that*	**Prep, other**	*in, of, die, city, mouth*

Figure 22.12 The top 50 words associated with one–star and five-star restaurant reviews in a Yelp dataset of 900,000 reviews, using the Monroe et al. (2008) method (Jurafsky et al., 2014).

22.6 Using Lexicons for Sentiment Recognition

In Appendix K we introduced the naive Bayes algorithm for sentiment analysis. The lexicons we have focused on throughout the chapter so far can be used in a number of ways to improve sentiment detection.

In the simplest case, lexicons can be used when we don't have sufficient training data to build a supervised sentiment analyzer; it can often be expensive to have a human assign sentiment to each document to train the supervised classifier.

In such situations, lexicons can be used in a rule-based algorithm for classification. The simplest version is just to use the ratio of positive to negative words: if a document has more positive than negative words (using the lexicon to decide the polarity of each word in the document), it is classified as positive. Often a threshold λ is used, in which a document is classified as positive only if the ratio is greater than λ. If the sentiment lexicon includes positive and negative weights for each word, θ_w^+ and θ_w^-, these can be used as well. Here's a simple such sentiment algorithm:

$$
\begin{aligned}
f^+ &= \sum_{w \text{ s.t. } w \in positivelexicon} \theta_w^+ count(w) \\
f^- &= \sum_{w \text{ s.t. } w \in negativelexicon} \theta_w^- count(w) \\
sentiment &= \begin{cases} + & \text{if } \frac{f^+}{f^-} > \lambda \\ - & \text{if } \frac{f^-}{f^+} > \lambda \\ 0 & \text{otherwise.} \end{cases}
\end{aligned} \tag{22.12}
$$

If supervised training data is available, these counts computed from sentiment lexicons, sometimes weighted or normalized in various ways, can also be used as features in a classifier along with other lexical or non-lexical features. We return to such algorithms in Section 22.7.

22.7 Using Lexicons for Affect Recognition

Detection of emotion (and the other kinds of affective meaning described by Scherer (2000)) can be done by generalizing the algorithms described above for detecting sentiment.

The most common algorithms involve supervised classification: a training set is labeled for the affective meaning to be detected, and a classifier is built using features extracted from the training set. As with sentiment analysis, if the training set is large enough, and the test set is sufficiently similar to the training set, simply using all the words or all the bigrams as features in a powerful classifier like logistic regression or SVM is an excellent algorithm whose performance is hard to beat. Thus we can treat affective meaning classification of a text sample as simple document classification.

Some modifications are nonetheless often necessary for very large datasets. For example, the Schwartz et al. (2013) study of personality, gender, and age using 700 million words of Facebook posts used only a subset of the n-grams of lengths 1-3. Only words and phrases used by at least 1% of the subjects were included as features, and 2-grams and 3-grams were only kept if they had sufficiently high PMI (PMI greater than $2 * \mathit{length}$, where *length* is the number of words):

$$\text{pmi}(\mathit{phrase}) = \log \frac{p(\mathit{phrase})}{\prod_{w \in \mathit{phrase}} p(w)} \tag{22.13}$$

Various weights can be used for the features, including the raw count in the training set, or some normalized probability or log probability. Schwartz et al. (2013), for example, turn feature counts into phrase likelihoods by normalizing them by each subject's total word use.

$$p(\mathit{phrase}|\mathit{subject}) = \frac{\text{freq}(\mathit{phrase}, \mathit{subject})}{\sum_{\mathit{phrase}' \in \text{vocab}(\mathit{subject})} \text{freq}(\mathit{phrase}', \mathit{subject})} \tag{22.14}$$

If the training data is sparser, or not as similar to the test set, any of the lexicons we've discussed can play a helpful role, either alone or in combination with all the words and n-grams.

Many possible values can be used for lexicon features. The simplest is just an indicator function, in which the value of a feature f_L takes the value 1 if a particular text has any word from the relevant lexicon L. Using the notation of Appendix K, in which a feature value is defined for a particular output class c and document x.

$$f_L(c,x) = \begin{cases} 1 & \text{if } \exists w : w \in L \ \& \ w \in x \ \& \ \mathit{class} = c \\ 0 & \text{otherwise} \end{cases}$$

Alternatively the value of a feature f_L for a particular lexicon L can be the total

number of word *tokens* in the document that occur in L:

$$f_L = \sum_{w \in L} count(w)$$

For lexica in which each word is associated with a score or weight, the count can be multiplied by a weight θ_w^L:

$$f_L = \sum_{w \in L} \theta_w^L count(w)$$

Counts can alternatively be logged or normalized per writer as in Eq. 22.14.

However they are defined, these lexicon features are then used in a supervised classifier to predict the desired affective category for the text or document. Once a classifier is trained, we can examine which lexicon features are associated with which classes. For a classifier like logistic regression the feature weight gives an indication of how associated the feature is with the class.

22.8 Lexicon-based methods for Entity-Centric Affect

What if we want to get an affect score not for an entire document, but for a particular entity in the text? The entity-centric method of Field and Tsvetkov (2019) combines affect lexicons with contextual embeddings to assign an affect score to an entity in text. In the context of affect about people, they relabel the Valence/Arousal/Dominance dimension as Sentiment/Agency/Power. The algorithm first trains classifiers to map embeddings to scores:

1. For each word w in the training corpus:
   (a) Use off-the-shelf pretrained encoders (like BERT) to extract a contextual embedding $\mathbf{e}$ for each instance of the word. No additional finetuning is done.
   (b) Average over the $\mathbf{e}$ embeddings of each instance of w to obtain a single embedding vector for one training point w.
   (c) Use the NRC VAD Lexicon to get S, A, and P scores for w.
2. Train (three) regression models on all words w to predict V, A, D scores from a word's average embedding.

Now given an entity mention m in a text, we assign affect scores as follows:

1. Use the same pretrained LM to get contextual embeddings for m in context.
2. Feed this embedding through the 3 regression models to get S, A, P scores for the entity.

This results in a (S,A,P) tuple for a given entity mention; To get scores for the representation of an entity in a complete document, we can run coreference resolution and average the (S,A,P) scores for all the mentions. Fig. 22.13 shows the scores from their algorithm for characters from the movie *The Dark Knight* when run on Wikipedia plot summary texts with gold coreference.

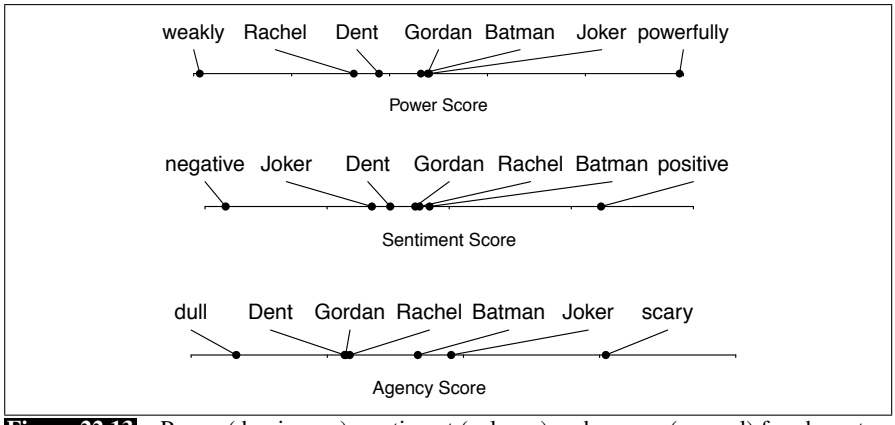

Figure 22.13 Power (dominance), sentiment (valence) and agency (arousal) for characters in the movie *The Dark Knight* computed from embeddings trained on the NRC VAD Lexicon. Note the protagonist (Batman) and the antagonist (the Joker) have high power and agency scores but differ in sentiment, while the love interest Rachel has low power and agency but high sentiment.

22.9 Connotation Frames

connotation frame

The lexicons we've described so far define a word as a point in affective space. A **connotation frame**, by contrast, is a lexicon that incorporates a richer kind of grammatical structure, by combining affective lexicons with the frame semantic lexicons of Chapter 21. The basic insight of connotation frame lexicons is that a predicate like a verb expresses connotations about the verb's arguments (Rashkin et al. 2016, Rashkin et al. 2017).

Consider sentences like:

(22.15) Country A violated the sovereignty of Country B

(22.16) the teenager ... survived the Boston Marathon bombing"

By using the verb *violate* in (22.15), the author is expressing their sympathies with Country B, portraying Country B as a victim, and expressing antagonism toward the agent Country A. By contrast, in using the verb *survive*, the author of (22.16) is expressing that the bombing is a negative experience, and the subject of the sentence, the teenager, is a sympathetic character. These aspects of connotation are inherent in the meaning of the verbs *violate* and *survive*, as shown in Fig. 22.14.

The connotation frame lexicons of Rashkin et al. (2016) and Rashkin et al. (2017) also express other connotative aspects of the predicate toward each argument, including the *effect* (something bad happened to x) *value*: (x is valuable), and *mental state*: (x is distressed by the event). Connotation frames can also mark the *power differential* between the arguments (using the verb *implore* means that the theme argument has greater power than the agent), and the *agency* of each argument (*waited* is low agency). Fig. 22.15 shows a visualization from Sap et al. (2017).

Connotation frames can be built by hand (Sap et al., 2017), or they can be learned by supervised learning (Rashkin et al., 2016), for example using hand-labeled training data to supervise classifiers for each of the individual relations, e.g., whether S(writer $\rightarrow$ Role1) is + or -, and then improving accuracy via global constraints across all relations.

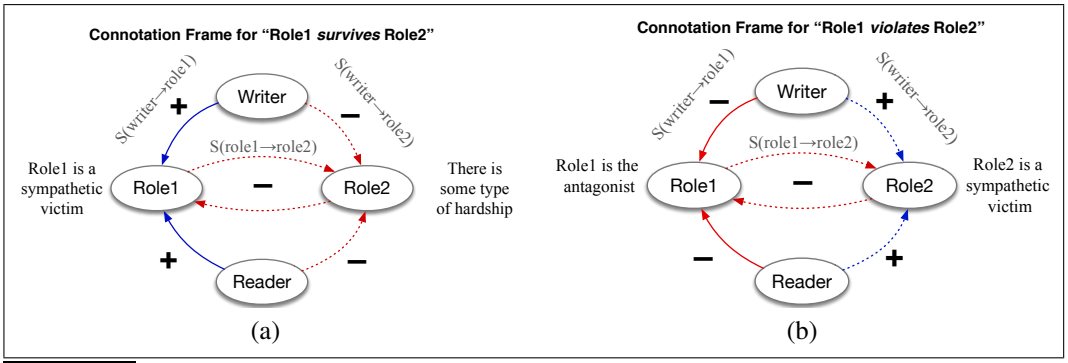

Figure 22.14 Connotation frames for *survive* and *violate*. (a) For *survive*, the writer and reader have positive sentiment toward Role1, the subject, and negative sentiment toward Role2, the direct object. (b) For *violate*, the writer and reader have positive sentiment instead toward Role2, the direct object.

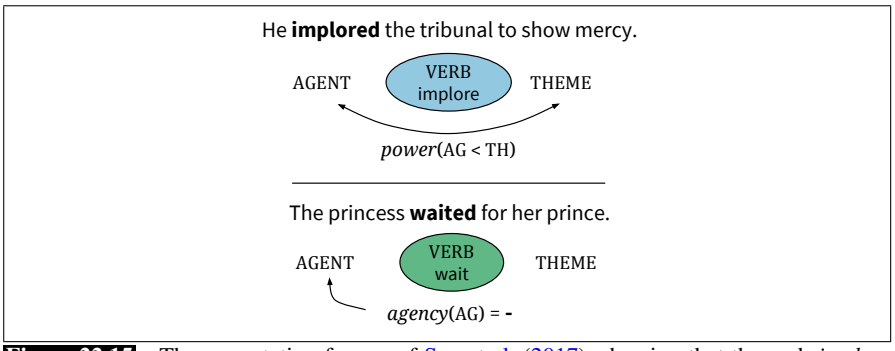

Figure 22.15 The connotation frames of Sap et al. (2017), showing that the verb *implore* implies the agent has lower power than the theme (in contrast, say, with a verb like *demanded*), and showing the low level of agency of the subject of *waited*. Figure from Sap et al. (2017).

22.10 Summary

- Many kinds of affective states can be distinguished, including *emotions*, *moods*, *attitudes* (which include *sentiment*), *interpersonal stance*, and *personality*.
- **Emotion** can be represented by fixed atomic units often called **basic emotions**, or as points in space defined by dimensions like **valence** and **arousal**.
- Words have **connotational** aspects related to these affective states, and this connotational aspect of word meaning can be represented in lexicons.
- Affective lexicons can be built by hand, using **crowd sourcing** to label the affective content of each word.
- Lexicons can be built with **semi-supervised**, bootstrapping from seed words using similarity metrics like embedding cosine.
- Lexicons can be learned in a **fully supervised** manner, when a convenient training signal can be found in the world, such as ratings assigned by users on a review site.
- Words can be assigned weights in a lexicon by using various functions of word counts in training texts, and ratio metrics like **log odds ratio informative Dirichlet prior**.
- Affect can be detected, just like sentiment, by using standard supervised **text classification** techniques, using all the words or bigrams in a text as features.

Additional features can be drawn from counts of words in lexicons.

- Lexicons can also be used to detect affect in a **rule-based classifier** by picking the simple majority sentiment based on counts of words in each lexicon.
- **Connotation frames** express richer relations of affective meaning that a predicate encodes about its arguments.

Historical Notes

The idea of formally representing the subjective meaning of words began with Osgood et al. (1957), the same pioneering study that first proposed the vector space model of meaning described in Chapter 5. Osgood et al. (1957) had participants rate words on various scales, and ran factor analysis on the ratings. The most significant factor they uncovered was the evaluative dimension, which distinguished between pairs like *good/bad*, *valuable/worthless*, *pleasant/unpleasant*. This work influenced the development of early dictionaries of sentiment and affective meaning in the field of **content analysis** (Stone et al., 1966).

subjectivity

Wiebe (1994) began an influential line of work on detecting **subjectivity** in text, beginning with the task of identifying subjective sentences and the subjective characters who are described in the text as holding private states, beliefs or attitudes. Learned sentiment lexicons such as the polarity lexicons of Hatzivassiloglou and McKeown (1997) were shown to be a useful feature in subjectivity detection (Hatzivassiloglou and Wiebe 2000, Wiebe 2000).

The term **sentiment** seems to have been introduced in 2001 by Das and Chen (2001), to describe the task of measuring market sentiment by looking at the words in stock trading message boards. In the same paper Das and Chen (2001) also proposed the use of a sentiment lexicon. The list of words in the lexicon was created by hand, but each word was assigned weights according to how much it discriminated a particular class (say buy versus sell) by maximizing across-class variation and minimizing within-class variation. The term *sentiment*, and the use of lexicons, caught on quite quickly (e.g., inter alia, Turney 2002). Pang et al. (2002) first showed the power of using all the words without a sentiment lexicon; see also Wang and Manning (2012).

Most of the semi-supervised methods we describe for extending sentiment dictionaries drew on the early idea that synonyms and antonyms tend to co-occur in the same sentence (Miller and Charles 1991, Justeson and Katz 1991, Riloff and Shepherd 1997). Other semi-supervised methods for learning cues to affective meaning rely on information extraction techniques, like the AutoSlog pattern extractors (Riloff and Wiebe, 2003). Graph based algorithms for sentiment were first suggested by Hatzivassiloglou and McKeown (1997), and graph propagation became a standard method (Zhu and Ghahramani 2002, Zhu et al. 2003, Zhou et al. 2004, Velikovich et al. 2010). Crowdsourcing can also be used to improve precision by filtering the result of semi-supervised lexicon learning (Riloff and Shepherd 1997, Fast et al. 2016).

Much recent work focuses on ways to learn embeddings that directly encode sentiment or other properties, such as the DENSIFIER algorithm of Rothe et al. (2016) that learns to transform the embedding space to focus on sentiment (or other) information.

Exercises

22.1 Show that the relationship between a word w and a category c in the Potts Score in Eq. 22.6 is a variant of the pointwise mutual information $\text{pmi}(w,c)$ without the log term.

An, J., H. Kwak, and Y.-Y. Ahn. 2018. SemAxis: A lightweight framework to characterize domain-specific word semantics beyond sentiment. *ACL*.

Baccianella, S., A. Esuli, and F. Sebastiani. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. *LREC*.

Barrett, L. F., B. Mesquita, K. N. Ochsner, and J. J. Gross. 2007. The experience of emotion. *Annual Review of Psychology*, 58:373–403.

Brysbaert, M., A. B. Warriner, and V. Kuperman. 2014. Concreteness ratings for 40 thousand generally known English word lemmas. *Behavior Research Methods*, 46(3):904–911.

Das, S. R. and M. Y. Chen. 2001. Yahoo! for Amazon: Sentiment parsing from small talk on the web. EFA 2001 Barcelona Meetings. http://ssrn.com/abstract=276189.

Ekman, P. 1999. Basic emotions. In T. Dalgleish and M. J. Power, eds, *Handbook of Cognition and Emotion*, 45–60. Wiley.

Fast, E., B. Chen, and M. S. Bernstein. 2016. Empath: Understanding Topic Signals in Large-Scale Text. *CHI*.

Field, A. and Y. Tsvetkov. 2019. Entity-centric contextual affective analysis. *ACL*.

Hamilton, W. L., K. Clark, J. Leskovec, and D. Jurafsky. 2016. Inducing domain-specific sentiment lexicons from unlabeled corpora. *EMNLP*.

Hatzivassiloglou, V. and K. McKeown. 1997. Predicting the semantic orientation of adjectives. *ACL*.

Hatzivassiloglou, V. and J. Wiebe. 2000. Effects of adjective orientation and gradability on sentence subjectivity. *COLING*.

Hellrich, J., S. Buechel, and U. Hahn. 2019. Modeling word emotion in historical language: Quantity beats supposed stability in seed word selection. *3rd Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*.

Hu, M. and B. Liu. 2004. Mining and summarizing customer reviews. *SIGKDD-04*.

Jurafsky, D., V. Chahuneau, B. R. Routledge, and N. A. Smith. 2014. Narrative framing of consumer sentiment in online restaurant reviews. *First Monday*, 19(4).

Justeson, J. S. and S. M. Katz. 1991. Co-occurrences of antonymous adjectives and their contexts. *Computational linguistics*, 17(1):1–19.

Kim, S. M. and E. H. Hovy. 2004. Determining the sentiment of opinions. *COLING*.

Kiritchenko, S. and S. M. Mohammad. 2017. Best-worst scaling more reliable than rating scales: A case study on sentiment intensity annotation. *ACL*.

Louviere, J. J., T. N. Flynn, and A. A. J. Marley. 2015. *Best-worst scaling: Theory, methods and applications*. Cambridge University Press.

Mairesse, F. and M. A. Walker. 2008. Trainable generation of big-five personality styles through data-driven parameter estimation. *ACL*.

Miller, G. A. and W. G. Charles. 1991. Contextual correlates of semantics similarity. *Language and Cognitive Processes*, 6(1):1–28.

Mohammad, S. M. 2018a. Obtaining reliable human ratings of valence, arousal, and dominance for 20,000 English words. *ACL*.

Mohammad, S. M. 2018b. Word affect intensities. *LREC*.

Mohammad, S. M. and P. D. Turney. 2013. Crowdsourcing a word-emotion association lexicon. *Computational Intelligence*, 29(3):436–465.

Monroe, B. L., M. P. Colaresi, and K. M. Quinn. 2008. Fightin'words: Lexical feature selection and evaluation for identifying the content of political conflict. *Political Analysis*, 16(4):372–403.

Moors, A., P. C. Ellsworth, K. R. Scherer, and N. H. Frijda. 2013. Appraisal theories of emotion: State of the art and future development. *Emotion Review*, 5(2):119–124.

Osgood, C. E., G. J. Suci, and P. H. Tannenbaum. 1957. *The Measurement of Meaning*. University of Illinois Press.

Pang, B., L. Lee, and S. Vaithyanathan. 2002. Thumbs up? Sentiment classification using machine learning techniques. *EMNLP*.

Pennebaker, J. W., R. J. Booth, and M. E. Francis. 2007. *Linguistic Inquiry and Word Count: LIWC 2007*. Austin, TX.

Picard, R. W. 1995. Affective computing. Technical Report 321, MIT Media Lab Perceptual Computing Technical Report. Revised November 26, 1995.

Plutchik, R. 1962. *The emotions: Facts, theories, and a new model*. Random House.

Plutchik, R. 1980. A general psychoevolutionary theory of emotion. In R. Plutchik and H. Kellerman, eds, *Emotion: Theory, Research, and Experience, Volume 1*, 3–33. Academic Press.

Potts, C. 2011. On the negativity of negation. In N. Li and D. Lutz, eds, *Proceedings of Semantics and Linguistic Theory 20*, 636–659. CLC Publications, Ithaca, NY.

Rashkin, H., E. Bell, Y. Choi, and S. Volkova. 2017. Multilingual connotation frames: A case study on social media for targeted sentiment analysis and forecast. *ACL*.

Rashkin, H., S. Singh, and Y. Choi. 2016. Connotation frames: A data-driven investigation. *ACL*.

Riloff, E. and J. Shepherd. 1997. A corpus-based approach for building semantic lexicons. *EMNLP*.

Riloff, E. and J. Wiebe. 2003. Learning extraction patterns for subjective expressions. *EMNLP*.

Rothe, S., S. Ebert, and H. Schütze. 2016. Ultradense Word Embeddings by Orthogonal Transformation. *NAACL HLT*.

Russell, J. A. 1980. A circumplex model of affect. *Journal of personality and social psychology*, 39(6):1161–1178.

Sap, M., M. C. Prasettio, A. Holtzman, H. Rashkin, and Y. Choi. 2017. Connotation frames of power and agency in modern films. *EMNLP*.

Scherer, K. R. 2000. Psychological models of emotion. In J. C. Borod, ed., *The neuropsychology of emotion*, 137–162. Oxford.

Schwartz, H. A., J. C. Eichstaedt, M. L. Kern, L. Dziurzynski, S. M. Ramones, M. Agrawal, A. Shah, M. Kosinski, D. Stillwell, M. E. P. Seligman, and L. H. Ungar. 2013. Personality, gender, and age in the language of social media: The open-vocabulary approach. *PloS one*, 8(9):e73791.

Stone, P., D. Dunphry, M. Smith, and D. Ogilvie. 1966. *The General Inquirer: A Computer Approach to Content Analysis*. MIT Press.

Tomkins, S. S. 1962. *Affect, imagery, consciousness: Vol. I. The positive affects*. Springer.

Turney, P. D. 2002. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. *ACL*.

Turney, P. D. and M. Littman. 2003. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems (TOIS)*, 21:315–346.

Velikovich, L., S. Blair-Goldensohn, K. Hannan, and R. McDonald. 2010. The viability of web-derived polarity lexicons. *NAACL HLT*.

Wang, S. and C. D. Manning. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. *ACL*.

Wiebe, J. 1994. Tracking point of view in narrative. *Computational Linguistics*, 20(2):233–287.

Wiebe, J. 2000. Learning subjective adjectives from corpora. *AAAI*.

Wiebe, J., R. F. Bruce, and T. P. O'Hara. 1999. Development and use of a gold-standard data set for subjectivity classifications. *ACL*.

Wilson, T., J. Wiebe, and P. Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. *EMNLP*.

Zhou, D., O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. 2004. Learning with local and global consistency. *NeurIPS*.

Zhu, X. and Z. Ghahramani. 2002. Learning from labeled and unlabeled data with label propagation. Technical Report CMU-CALD-02, CMU.

Zhu, X., Z. Ghahramani, and J. Lafferty. 2003. Semi-supervised learning using gaussian fields and harmonic functions. *ICML*.