# Sentiment Classification

Andrea Esuli

# Supervised/unsupervised

**Supervised learning** methods are the most commonly used one, yet also some **unsupervised** methods have been successfully.

Unsupervised methods rely on the shared and recurrent characteristics of the sentiment dimension across topics to perform classification by means of hand-made heuristics and simple language models.

Supervised methods rely on a **training set** of labeled examples that describe the correct classification label to be assigned to a number of documents.
A learning algorithm then exploits the examples to model a general classification function.

# Unsupervised methods

# Unsupervised Sentiment Classification

Unsupervised methods do not require labeled examples.

Knowledge about the task is usually added by using lexical resources and hard-coded heuristics, e.g.:

- Lexicons + patterns: VADER
- Patterns + Simple language model: SO-PMI

Neural language models have been found that they learn to recognize sentiment with no explicit knowledge about the task.

# VADER

VADER (Valence Aware Dictionary for sEntiment Reasoning) uses a curated lexicon derived from well known sentiment lexicons that assigns a positivity/negativity score to 7k+ words/emoticons.

It also uses a number of hand-written pattern matching rules (e.g., negation, intensifiers) to modify the contribution of the original word scores to the overall sentiment of text.

[Hutto and Gilbert. VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. ICWSM 2014.]()

VADER is integrated into [NLTK]()

```
NEGATE = {"aint", "arent", "cannot", "cant", "couldn
 "ain't", "aren't", "can't", "couldn't", "daren't",
 "dont", "hadnt", "hasnt", "havent", "isnt", "mightn
 "don't", "hadn't", "hasn't", "haven't", "isn't", "m
 "neednt", "needn't", "never", "none", "nope", "nor"
 "oughtnt", "shant", "shouldnt", "uhuh", "wasnt", "w
 "oughtn't", "shan't", "shouldn't", "uh-uh", "wasn't
 "without", "wont", "wouldnt", "won't", "wouldn't",

# booster/dampener 'intensifiers' or 'degree adverbs
# http://en.wiktionary.org/wiki/Category:English_deg

BOOSTER_DICT = \
{"absolutely": B_INCR, "amazingly": B_INCR, "awfully
 "decidedly": B_INCR, "deeply": B_INCR, "effing": B_
 "entirely": B_INCR, "especially": B_INCR, "exceptio
 "fabulously": B_INCR, "flipping": B_INCR, "flippin"
 "fricking": B_INCR, "frickin": B_INCR, "frigging":
 "greatly": B_INCR, "hella": B_INCR, "highly": B_INC
 "intensely": B_INCR, "majorly": B_INCR, "more": B_I
 "purely": B_INCR, "quite": B_INCR, "really": B_INCR
 "so": B_INCR, "substantially": B_INCR,
 "thoroughly": B_INCR, "totally": B_INCR, "tremendou
 "uber": B_INCR, "unbelievably": B_INCR, "unusually"
 "very": B_INCR,
 "almost": B_DECR, "barely": B_DECR, "hardly": B_DEC
 "kind of": B_DECR, "kinda": B_DECR, "kindof": B_DEC
 "less": B_DECR, "little": B_DECR, "marginally": B_D
 "scarcely": B_DECR, "slightly": B_DECR, "somewhat":
 "sort of": B_DECR, "sorta": B_DECR, "sortof": B_DEC

# check for special case idioms using a sentiment-la
SPECIAL_CASE_IDIOMS = {"the shit": 3, "the bomb": 3,
                       "cut the mustard": 2, "kiss o
```

# VADER

```
from nltk.sentiment.vader import SentimentIntensityAnalyzer
vader = SentimentIntensityAnalyzer()

vader.polarity_scores('the best experience I had')
Out: {'neg': 0.0, 'neu': 0.417, 'pos': 0.583, 'compound': 0.6369}

vader.polarity_scores('not the best experience I had')
Out: {'neg': 0.457, 'neu': 0.543, 'pos': 0.0, 'compound': -0.5216}
```

VADER can be used to bootstrap a training set for *supervised learning.*

In this case we can talk of a *weakly-supervised* or *semi-supervised* approach, since training data are not all validated by a human, and can contain errors.

```
NEGATE = {"aint", "arent", "cannot", "cant", "couldn
 "ain't", "aren't", "can't", "couldn't", "daren't",
 "dont", "hadnt", "hasnt", "havent", "isnt", "mightn
 "don't", "hadn't", "hasn't", "haven't", "isn't", "m
 "neednt", "needn't", "never", "none", "nope", "nor"
 "oughtnt", "shant", "shouldnt", "uhuh", "wasnt", "w
 "oughtn't", "shan't", "shouldn't", "uh-uh", "wasn't
 "without", "wont", "wouldnt", "won't", "wouldn't",

# booster/dampener 'intensifiers' or 'degree adverbs
# http://en.wiktionary.org/wiki/Category:English_deg

BOOSTER_DICT = \
{"absolutely": B_INCR, "amazingly": B_INCR, "awfully
 "decidedly": B_INCR, "deeply": B_INCR, "effing": B_
 "entirely": B_INCR, "especially": B_INCR, "exceptio
 "fabulously": B_INCR, "flipping": B_INCR, "flippin"
 "fricking": B_INCR, "frickin": B_INCR, "frigging":
 "greatly": B_INCR, "hella": B_INCR, "highly": B_INC
 "intensely": B_INCR, "majorly": B_INCR, "more": B_I
 "purely": B_INCR, "quite": B_INCR, "really": B_INCR
 "so": B_INCR, "substantially": B_INCR,
 "thoroughly": B_INCR, "totally": B_INCR, "tremendou
 "uber": B_INCR, "unbelievably": B_INCR, "unusually"
 "very": B_INCR,
 "almost": B_DECR, "barely": B_DECR, "hardly": B_DEC
 "kind of": B_DECR, "kinda": B_DECR, "kindof": B_DEC
 "less": B_DECR, "little": B_DECR, "marginally": B_D
 "scarcely": B_DECR, "slightly": B_DECR, "somewhat":
 "sort of": B_DECR, "sorta": B_DECR, "sortof": B_DEC

# check for special case idioms using a sentiment-la
SPECIAL_CASE_IDIOMS = {"the shit": 3, "the bomb": 3,
                       "cut the mustard": 2, "kiss o
```

# Thumbs Up or Thumbs Down?

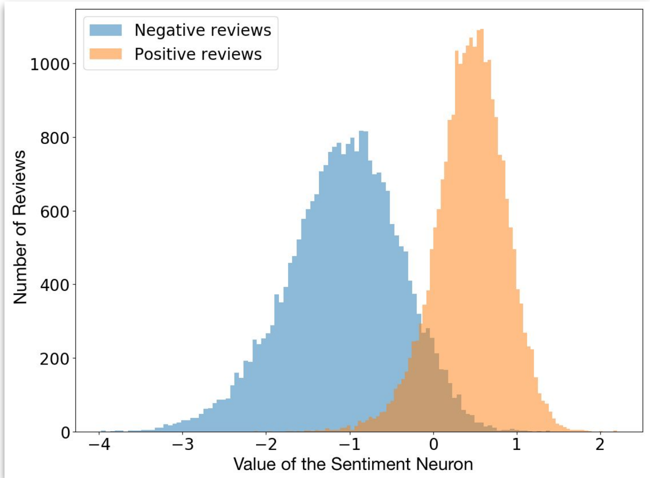
Pointwise Mutual Information has been applied to determine the overall sentiment of text.

- Short phrases extracted from text using POS patterns, e.g.:
  `JJ+NN, RB+JJ, JJ+JJ, NN+JJ, RB+VB`
- SO-PMI score of each phrase is computed using a search engine and proximity queries, e.g.: `"very solid" NEAR good`
- SO-PMI scores for phrases are averaged to produce the document score.

Turney. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. ACL 2002

# Sentiment Classification from a single neuron

A char-level LSTM with 4096 units has been trained on **82 millions** of reviews from Amazon.




After training one of the units had a very high correlation with sentiment, resulting in state-of-the-art accuracy when used as a classifier.

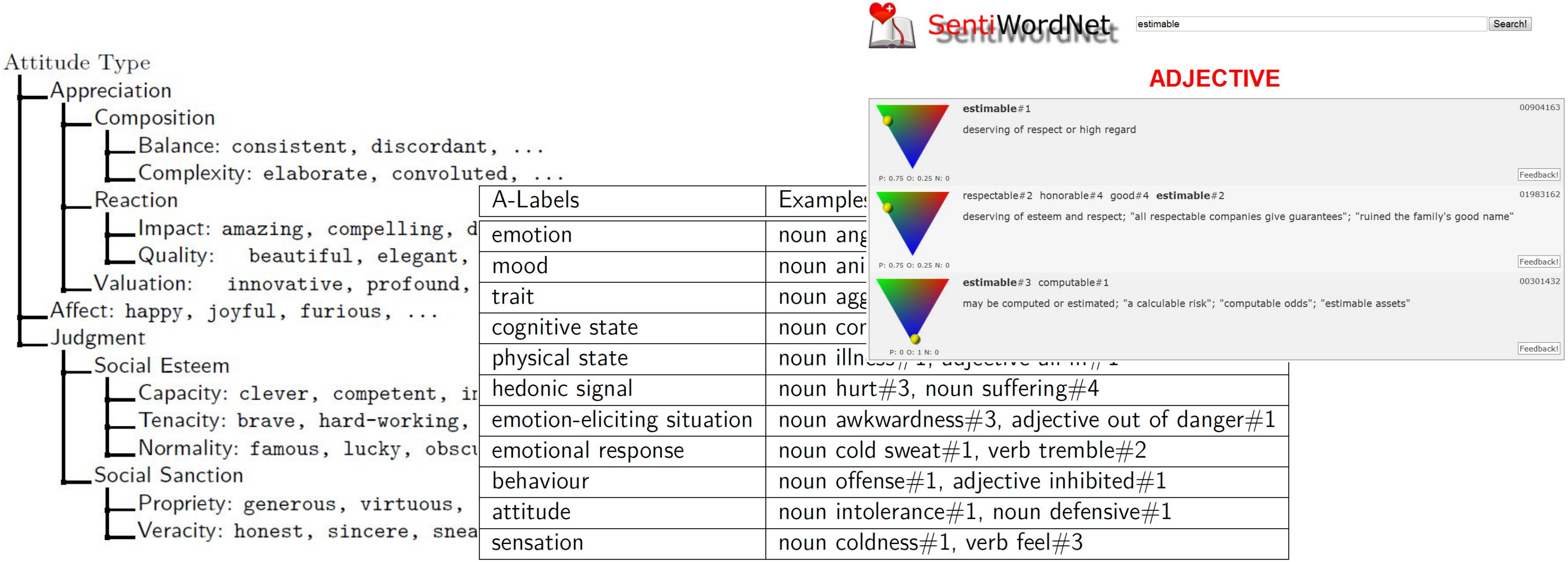
By fixing the sentiment unit to a given value, the generation process has been forced to produce reviews with a given sentiment polarity.

[Blog post](#) - [Radford et al. Learning to Generate Reviews and Discovering Sentiment. Arxiv 1704.01444](#)

# Supervised methods

# Supervised methods

Supervised methods use a traditional ML pipeline, typically exploiting the use of lexical resources to improve the number and quality of sentiment-related features extracted from text.



# Sentiment features

Sentiment lexicon can be exploited to add sentiment information in text representation.

In this way a general knowledge about language connects words that are observed in the training set with words that occur only in the test set (which would have been considered out-of-vocabulary words).

good → SWN_Pos

gentle → SWN_Pos

bad → SWN_Neg

hostile → SWN_Neg

# Distant supervision

Producing training data for supervised learning may have a relevant cost.

Distant supervision exploits "cheap" methods that "weakly" label examples to bootstrap a training set, e.g.:

- labeling tweets with 😄 as positive and those with 😒 as negative.
- using VADER to perform a first labeling (skipping low confidence labels).

The rationale behind distant supervision is that:

- noisy information in training data will cancel out in the learning phase.
- discriminant features that have a decent correlation with the weak labeling emerge among the other.

# Distant supervision likes sentiment

Distant supervision fits better with sentiment analysis than with topic-related analysis because in the former it is easier to define negative examples.

A negative sentiment is a concept on its own, opposite to a positive one.

The "negation" of a topic is just the absence of the topic. It is harder to define a heuristic to label negative docs.

- How to automatically mark a negative example for a "soccer" classifier?
- Just use random sampling [when nothing else works](when nothing else works).