The Emotional Spark - Concepts, Challenges, and Best Practices in Sentiment Analysis

Dr. Stefan Gindl, Freelancer
Data Scientist | Software Developer | Assistant Professor
stefan.gindl@gmail.com

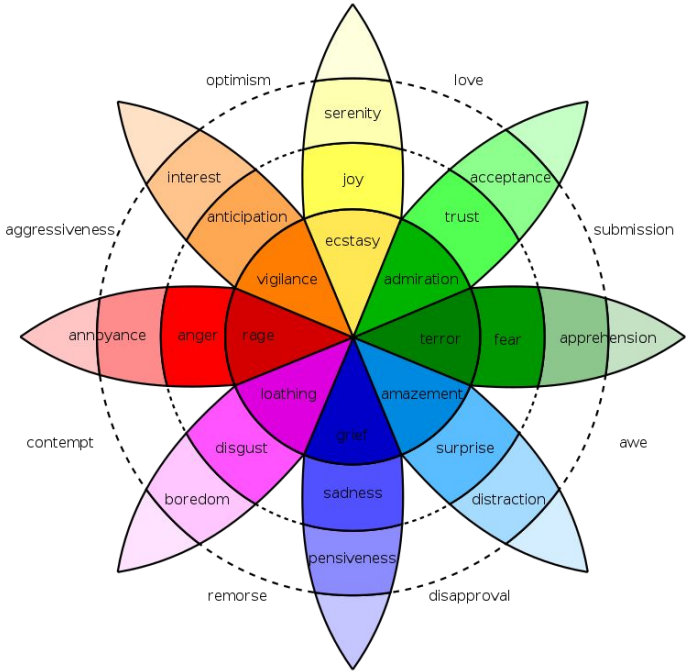
- Concepts
- Applications
- Tools
- Features
- Evaluation & Limitations
- Corpus Annotation

Concepts

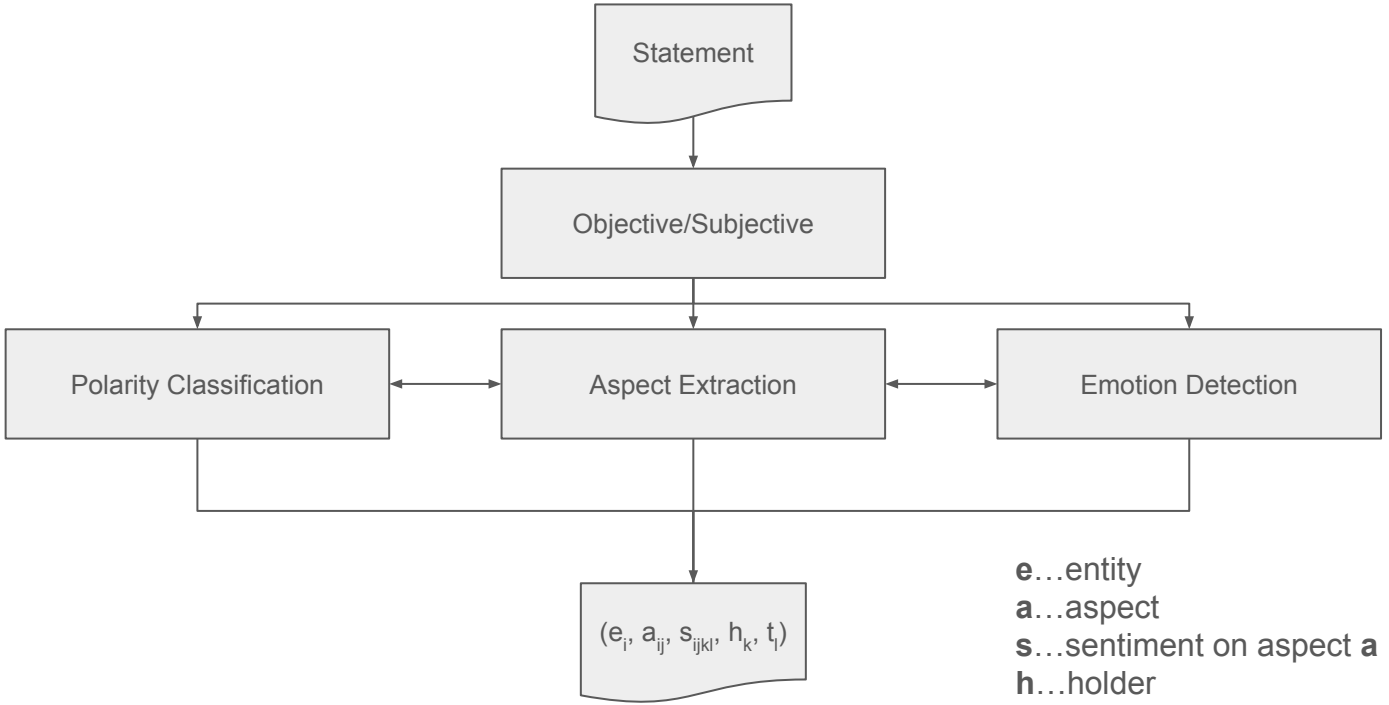
Concepts

- Sentiment analysis
  - = opinion mining
  - = sentiment detection
  - = polarity classification (?)
- Objective (factual) vs subjective (opinionated) statements
- Aspect-based opinion mining
- Emotion Detection
  - SentiSense
  - SenticNet

Robert Plutchik's wheel of emotions



Concepts



e…entity
a…aspect
s…sentiment on aspect **a**
h…holder
t ...time

"The **iPhone** is very good, but they still need to work on **battery life** and **security** issues", John said yesterday.

entity	aspect	sentiment	holder	time
iPhone	GENERAL	pos	John	2019-06-18
iPhone	security	neg	John	2019-06-18
iPhone	battery life	neg	John	2019-06-18

Applications

Applications

- Media monitoring:
  - Get customer feedback from online media
  - Measure the political sentiment of the public
  - Health*:
    - Pharmaceutical companies mine online health information to monitor patients' opinions on their products and services, and to obtain feedback on their performance and the consumers' satisfaction.
    - Secondary effects of drugs, alternative therapies, undocumented symptoms
- As an additional feature for other applications, e.g. stock price prediction
- Built-in feature in CRM tools

* journals.plos.org/plosone/article?id=10.1371/journal.pone.0207996

Applications

Health Sector

- Algorithmic Detection and Analysis of **Vaccine-Denialist** Sentiment Clusters in **Social Networks** (2019): arxiv.org/abs/1905.12908
- Understanding Perceptions and Attitudes in **Breast Cancer Discussions on Twitter** (2019): arxiv.org/abs/1905.12469
- Distinguishing Clinical Sentiment: The Importance of Domain Adaptation in **Psychiatric Patient Health Records** (2019): arxiv.org/abs/1904.03225
- "Hang in There": Lexical and Visual Analysis to Identify Posts Warranting Empathetic Responses (2019): arxiv.org/abs/1903.05210
- Natural Language Processing, Sentiment Analysis and Clinical Analytics (2019): arxiv.org/abs/1902.00679

Applications

Finance

- **Market Trend Prediction** using Sentiment Analysis: Lessons Learned and Paths Forward (2019): arxiv.org/abs/1903.05440
- Predicting the Effects of **News Sentiments on the Stock Market** (2018): arxiv.org/abs/1812.04199
- Leveraging Financial News for **Stock Trend Prediction** with Attention-Based Recurrent Neural Network (2018): arxiv.org/abs/1811.06173
- Using Stock Prices as Ground Truth in Sentiment Analysis to Generate Profitable Trading Signals (2018): arxiv.org/abs/1811.02886

Applications

Politics

- Topic-Specific Sentiment Analysis Can Help **Identify Political Ideology** (2018): arxiv.org/abs/1810.12897
- Sentiment Aggregate Functions for **Political Opinion Polling** using Microblog Streams (2016): arxiv.org/abs/1606.05242
- Learning from the News: Predicting Entity Popularity on Twitter (2016): arxiv.org/abs/1607.03057
- Dynamic Allocation of Crowd Contributions for Sentiment Analysis during the 2016 U.S. Presidential Election (2016): arxiv.org/abs/1608.08953

Approaches

- Simplest: count number of positive and negative terms
- Feature engineering: filter irrelevant language particles, enrich useful particles with further information
- All sorts of machine learners
  - Naive Bayes, SVM, Linear Regression
- Deep Learning
  - Sentiment analysis solved?

Tools

Out-of-the-box

- Python
  - VADER: Valence Aware Dictionary and sEntiment Reasoner), a lexicon and rule-based sentiment analysis tool
  - NLTK: has a VADER wrapper
  - TextBlob:
  - spacy: AFAIK no
- Outside of Python
  - R: sentimentr (dictionary-based, uses valence shifters)
  - Java: Stanford CoreNLP

Out-of-the-box

```
[ ]  1 statement = ("This movie doesn't care about cleverness, wit or any other kind "
     2              "of intelligent humor.")
```

NLTK

```
[ ]  1 from nltk.sentiment.vader import SentimentIntensityAnalyzer
     2 import nltk
     3 nltk.download('vader_lexicon')
```

```
[ ]  1 sa = SentimentIntensityAnalyzer()
     2 sa.polarity_scores(statement)
```

```
{'compound': -0.1338, 'neg': 0.265, 'neu': 0.497, 'pos': 0.239}
```

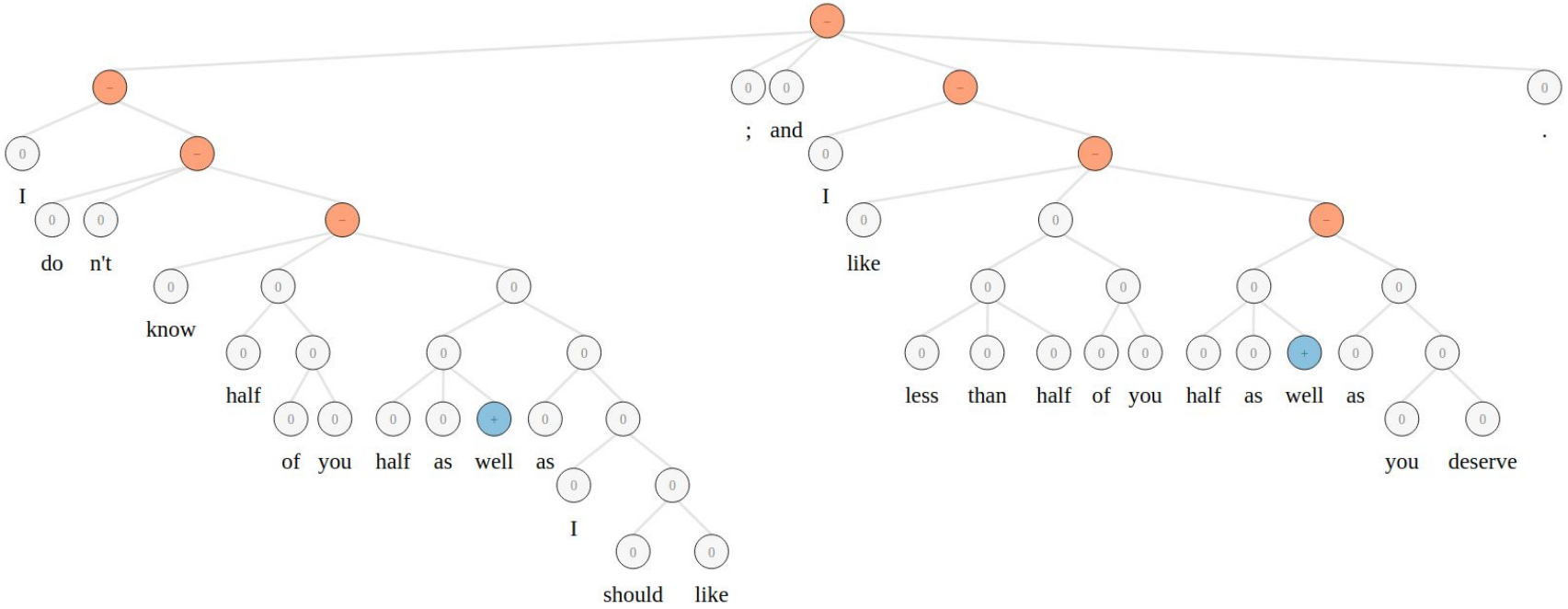
TextBlob

```
[ ]  1 from textblob import TextBlob
     2 TextBlob(statement).polarity
```

```
0.425
```

Let’s try it out...

→nlp.stanford.edu/sentiment/



Fun with Sentiment Analysis

→colab.research.google.com/drive/1umsBCj97qn07Sb76YO1N3H9ADnTFj8w_

Features

Features

- Bag-of-Words, e.g. counts or TF-IDF
- Stopword filtering
- Sentiment lexicons
  - General Inquirer, Opinion Lexicon, SentiSense and SentiStrength, SentiWordNet, SenticNet

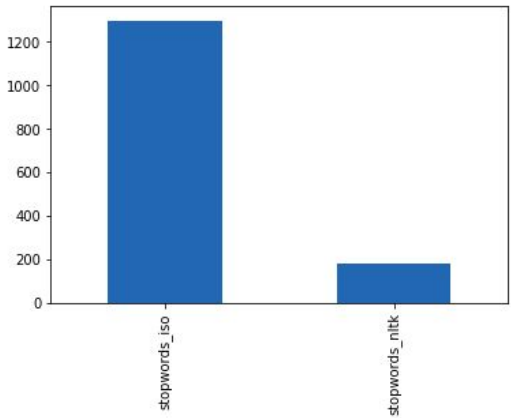
Stopwords

```
[ ]
from nltk.corpus import stopwords
import nltk
import pandas as pd
import requests

nltk.download('stopwords')
github_stopwords_en = ('https://raw.githubusercontent.com/stopwords-iso/'
                       'stopwords-en/master/stopwords-en.txt')

stopwords_iso = requests.get(github_stopwords_en).text.split('\n')
stopwords_nltk = set(stopwords.words("english"))
```

```
[ ]
print(f'Examples of stopwords_iso: {stopwords_iso[:5]}')
print(f'Examples of stopwords_nltk: {list(stopwords_nltk)[:5]}')
```

```
Examples of stopwords_iso: ["'ll", "'tis", "'twas", "'ve", '10']
Examples of stopwords_nltk: ['wouldn', 'the', 'and', 'by', 'she']
```

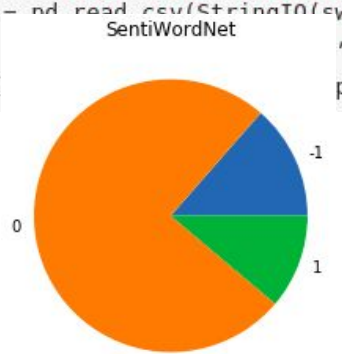
```
[ ]
stopwords = pd.Series({'stopwords_iso': len(stopwords_iso),
                       'stopwords_nltk': len(stopwords_nltk)})
stopwords.plot.bar()
```



ISO Stopwords: github.com/stopwords-iso

Sentiment Lexicons

```
[ ]
from io import StringIO
import pandas as pd
import requests


def get_compound(row):
    if row['PosScore'] == 0 and row['NegScor
        return 0
    return 1 if row['PosScore'] > row['NegSc

swn_text = requests.get('https://raw.githubu
                        'SentiWordNet/master
columns = ['POS', 'ID', 'PosScore', 'NegScor
df_swn = pd.read_csv(StringIO(swn_text.text)
                     , header=None

df_swn[                                  ply(get_compo
```




```
[ ]
df_swn.loc[100:110]
```

	POS	ID	PosScore	NegScore	SynsetTerms	Gloss	CompoundScore
100	a	21403.0	0.250	0.250	unobliging	NaN	-1
101	a	21592.0	0.125	0.250	uncooperative	NaN	-1
102	a	21766.0	0.500	0.000	accurate	NaN	1
103	a	22219.0	0.625	0.000	faithful	NaN	1
104	a	22437.0	0.625	0.250	dead-on	NaN	1
105	a	22680.0	0.500	0.125	high-fidelity	NaN	1
106	a	22852.0	0.500	0.000	surgical	NaN	1
107	a	22962.0	0.000	0.000	straight	NaN	0
108	a	23120.0	0.125	0.000	true	NaN	1
109	a	23278.0	0.875	0.000	veracious	NaN	1
110	a	23383.0	0.125	0.625	inaccurate	NaN	-1

SentiWordNet on Github: github.com/aesuli/SentiWordNet/blob/master/data/SentiWordNet_3.0.0.txt

Features

- Filter by adjectives, nouns, verbs and adverbs
- Contextual valence shifters: negation, intensification
  - NOT good/NOT bad
  - "VERY good", "EXTREMELY good"
  - "I REALLY love the weather." vs. "I HARDLY love the weather."
- Phrases/N-Grams
- Semantic Expansion: WordNet
- Emoticons
- Word Embeddings

PoS-tag Filtering

```
[ ]
from nltk import pos_tag
from nltk import word_tokenize
import nltk
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')


pos_tags_noun = {'NN', 'NNS', 'NNP', 'NNPS'}
pos_tags_adj = {'JJ', 'JJR', 'JJS'}
```

```
[ ]
sentence = 'The quick brown fox jumps over the lazy dog.'
tokens = word_tokenize(sentence)
tagged = pos_tag(tokens)
tagged[:4]
```

```
[('The', 'DT'), ('quick', 'JJ'), ('brown', 'NN'), ('fox', 'NN')]
```

```
[ ]
[token[0] for token in tagged if token[1] in pos_tags_noun]
```

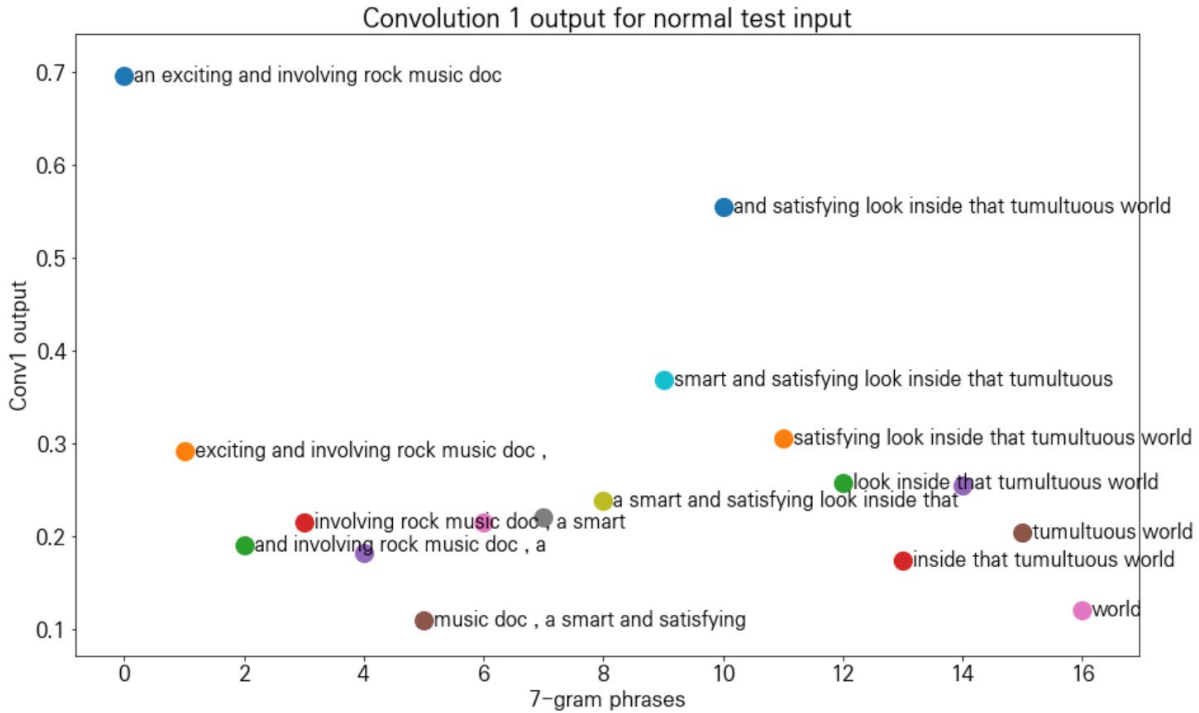
```
['brown', 'fox', 'dog']
```

```
[ ]
[token[0] for token in tagged if token[1] in pos_tags_adj]
```

```
['quick', 'lazy']
```

Penn Treebank Tag Set: www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html

N-Grams



Fentaw, H. W. & Kim, T.-H. **Design and Investigation of Capsule Networks for Sentence Classification**. Applied Sciences, 2019, 9.

WordNet

```
[ ]
from nltk.corpus import wordnet as wn
import nltk

nltk.download('wordnet')
```

```
[ ]
print(f"Synsets: {wn.synsets('sentiment')}")
print(f"Definition: {wn.synset('opinion.n.01').definition()}")
print(f"Hypernyms: {wn.synset('opinion.n.01').hypernyms()}")
print(f"Hyponyms: {wn.synset('opinion.n.01').hyponyms()[:2]}")
```

```
Synsets: [Synset('sentiment.n.01'), Synset('opinion.n.01')]
Definition: a personal belief or judgment that is not founded on proof or certainty
Hypernyms: [Synset('belief.n.01')]
Hyponyms: [Synset('eyes.n.01'), Synset('idea.n.03')]
```

```
[ ]
print(f"Similary opinion<->sentiment:", wn.synset('opinion.n.01').path_similarity(wn.synset('sentiment.n.01')))
print(f"Similarity car<->truck:", wn.synset('car.n.01').path_similarity(wn.synset('truck.n.01')))
print(f"Similarity car<->bed:", wn.synset('car.n.01').path_similarity(wn.synset('cat.n.01')))
```

```
Similary opinion<->sentiment: 0.1
Similarity car<->truck: 0.3333333333333333
Similarity car<->bed: 0.05555555555555555
```

Features - An Evaluation

- Content-based features
  - Bag of words (BOW)
  - Word embeddings (W2V)
  - Concept embeddings (C2V): map text onto concepts from the UMLS Metathesaurus
- Domain-specific features
  - Extraction UMLS Semantic Types (ST)
- Positional features (Pst)
  - position of the sentence within the post, position of the post within the thread
- Network features (Net)
  - number of replies of the post, Is a primary question (1 if the sentence belongs to the first post in a conversation, and 0 otherwise)

Feature engineering for sentiment analysis in e-health forums: journals.plos.org/plosone/article?id=10.1371/journal.pone.0207996

Features - An Evaluation

- Sentiment-based features (SA)
  - Number of positive/negative words (General Inquirer], SentiSense, and SentiStrength)
  - Emotions expressed in the text (from SentiSense lexicon: Ambiguous, Anger, Calmness, Despair, Disgust, Anticipation, Fear, Hate, Hope, Joy, Like, Love, Sadness, and Surprise)
- Grammatical features (Gramm)
  - Verb tense
  - Part-of-speech
  - Negation
- Factuality (Fact)
  - Includes the factual/non-factual label manual classification as a feature

Feature engineering for sentiment analysis in e-health forums: journals.plos.org/plosone/article?id=10.1371/journal.pone.0207996

	SMO		RF		NB		Vote	
Feature	**Acc**	**F-1**	**Acc**	**F-1**	**Acc**	**F-1**	**Acc**	**F-1**
BOW	57,1	47,7	57,6	46,1	58,1	49,4	58,4	48,9
BOW + ST	60,9	60,1***	64,4	59,4	63,9	62,9	63,4	61,5
BOW + ST + Pst	61,8	59,7***	64,8	59,9	63,6	62,6	63,9	62,4
BOW + ST + Pst + Net	59,8	59,3***	64,7	59,9	63,5	62,7	62,9	61,3
BOW + ST + Pst + Net + SA	62,1	61,6***	64,2	59	65,3	64,5	**65,5**	**64,1**
BOW + ST + Pst + Net + SA + Gramm	56,6	47,1	57,2	55,5	57,6	49,2	59,3	52,4
BOW + ST + Pst + Net + SA + Gramm + Fact	57,1	49,2	56,7	55,3	57,5	49,7	59,7	53,6
W2V	65,9	65,1	62,8	56,5	57,5	46,4	66,5	63,5
W2V + ST	66,2	65,2	63,1	56,4	60,3	62,9	67	63,4
W2V + ST + Pst	66,5	65,9	61,3	53,8	60,6	62,6	67,3	63
W2V + ST + Pst + Net	66,1	65,4	62,3	55,2	59,9	62,7	67,4	63,4
W2V + ST + Pst + Net + SA	68,4	66,5	62,6	55,7	61,9	64,5	**68,7**	**65,3**
W2V + ST + Pst + Net + SA + Gramm	66	65,8	62,9	56,5	62,6	49,2	68,2	65,9
W2V + ST + Pst + Net + SA + Gramm + Fact	64,7	64,5	63,3	56,9	62,6	49,7	68,5	66,4
C2V	62,6	60,8	61,2	55	57,4	45,6	63,8	59,6
C2V + ST	65	65,2**	63,1	56,4	63,9	53,8	65	61,4
C2V + ST + Pst	66,5	65,9**	61,3	53,8	63,7	54,4	65,3	62
C2V + ST + Pst + Net	66,1	65,4**	62,3	55,2	63,5	54,1	66,4	63,4
C2V + ST + Pst + Net + SA	**67,8**	**67,2****	62,6	55,7	65,3	57,6	67,7	65,3
C2V + ST + Pst + Net + SA + Gramm	62,4	61,6*	63,1	56,5	57,6	58,8	64,8	62
C2V + ST + Pst + Net + SA + Gramm + Fact	63,2	62,4*	63,3	57	57,5	59,5	65,3	62,5

https://doi.org/10.1371/journal.pone.0207996.t008

Feature engineering for sentiment analysis in e-health forums: journals.plos.org/plosone/article?id=10.1371/journal.pone.0207996

Features with Deep Learning

```
[ ]
```

```python
import re
from nltk.stem import WordNetLemmatizer
from nltk.corpus import stopwords

stop_words = set(stopwords.words("english"))
lemmatizer = WordNetLemmatizer()


def clean_text(text):
    try:
        text = re.sub(r'[^\w\s]','',text, re.UNICODE)
    except:
        print(text)
    text = text.lower()
    text = [lemmatizer.lemmatize(token) for token in text.split(" ")]
    text = [lemmatizer.lemmatize(token, "v") for token in text]
    text = [word for word in text if not word in stop_words]
    text = " ".join(text)
    return text
```

Kaggle IMDB solution: www.kaggle.com/nilanml/imdb-review-deep-model-94-89-accuracy

Evaluation and Limitations

Evaluation

- Recall
  - measure of completeness: TP/(TP+FN)
- Precision
  - measure of correctness: TP/(TP+FP)
- F-measure
  - harmonic mean of recall & precision: (2*recall*precision)/(recall+precision)
- Accuracy
  - proportion of true results among in all cases: (TP+TN)/(TP+TN+FP+FN)
- Cross-validation

Evaluation

```
[ ]
```

```python
from sklearn.metrics import classification_report
y_true = [1, 1, -1, 1, -1, -1, 1]
y_pred = [1, -1, -1, 1, 1, -1, 1]

print(classification_report(y_true, y_pred))
```

```
              precision    recall  f1-score   support

          -1       0.67      0.67      0.67         3
           1       0.75      0.75      0.75         4

    accuracy                           0.71         7
   macro avg       0.71      0.71      0.71         7
weighted avg       0.71      0.71      0.71         7
```

Limits

- Context, word ambiguity
  - This movie‘s plot is unpredictable.
  - The brakes of this car are unpredictable.
- Stilistic means: sarcasm, irony, cynicism, metaphors
- Domain-dependency
- Colloquial language (e.g. in social media)
  - “I luv u”
  - NLTK TweetTokenizer()

Corpus Annotation

Corpus Annotation

- Choose level of granularity
- Split corpus:
  - Set for annotation guidelines
  - Set for main annotation task
- Derive guidelines on small set
- Select annotators: level of expertise required?
- Annotators annotate on large set using the guidelines

Corpus Annotation

- Calculate inter-rater agreement
  - Cohen's kappa: for 2 raters
  - Fleiss' kappa: more than 2 raters

< 0: Agreement below chance
0.01-0.2: Small agreement
0.21-0.4: Fair agreement
0.41-0.6: Moderate agreement
0.61-0.8: Substantial agreement
0.81-0.99: Almost perfect agreement

```
[ ]  1 from sklearn.metrics import cohen_kappa_score
```

```
[ ]  1 rater_1 = [1, 1, 1, 0, 0, 1, 1, 0, 0, 1]
     2 rater_2 = [0, 1, 1, 0, 0, 1, 1, 0, 0, 1]
     3
     4 cohen_kappa_score(rater_1, rater_2)
```

```
0.8
```

Corpus Annotation

1. If at least two annotators assigned the same label to the sentence, then such label was finally assigned.
2. If each of the three annotators assigned a different label to a sentence, then a fourth annotator was asked to select the final label.
3. If a sentence was not labeled by at least two annotators, we preserve the sentence for readiness and labeled it as NOT_LABELED.

Corpus Annotation

	Experience	Opinion	Fact
Allergies	86%	69%	65%
Crohn	88%	65%	72%
Breast cancer	77%	70%	79%
Average	84%	68%	72%

https://doi.org/10.1371/journal.pone.0207996.t005

	Positive	Negative	Neutral
Allergies	79%	76%	71%
Crohn	68%	68%	79%
Breast cancer	77%	70%	79%
Average	75%	71%	76%

https://doi.org/10.1371/journal.pone.0207996.t004

Feature engineering for sentiment analysis in e-health forums: journals.plos.org/plosone/article?id=10.1371/journal.pone.0207996

Resources

Resources

- Tools:
  - NLTK (Natural Language ToolKit)
    - www.nltk.org
  - spacy.io
  - TextBlob: textblob.readthedocs.io/en/dev/
  - VADER (Valence Aware Dictionary and sEntiment Reasoner);
    - github.com/cjhutto/vaderSentiment
- Hubs:
  - English:
    - github.com/xiamx/awesome-sentiment-analysis
  - German Sentiment Analysis:
    - sites.google.com/site/iggsahome
    - Offensive language: projects.fzai.h-da.de/iggsa/resources-tools-and-literature/

Resources

- Sentiment Lexicons:
  - SentiWordNet:
    - github.com/aesuli/SentiWordNet
  - Opinion Lexicon:
    - www.cs.uic.edu/~liub/FBS/sentiment-analysis.html
    - github.com/shekhargulati/sentiment-analysis-python/tree/master/opinion-lexicon-English
  - SenticNet: sentic.net
  - General Inquirer: www.wjh.harvard.edu/~inquirer/
  - SentiStrength: sentistrength.wlv.ac.uk/
  - SentiSense: nlp.uned.es/~jcalbornoz/SentiSense.html
- Stopwords:
  - github.com/stopwords-iso/
  - `from nltk.corpus import stopwords`

Thank you.

Emotion is a sum totaled by an adding machine of the mind.
Ayn Rand, *Atlas Shrugged.*