Sentiment Analysis - 2

Text Analytics

Ramaseshan Ramachandran

Definitions

Sentiment Analysis	The process of assigning a polarity, positive, negative or neutral, to an unstructured text	Lexicon-based approach Supervised classification approach - statistical and machine learning
Semantic Orientation	Measuring subjectivity and opinion from an unstructured text	Identifying polarity and strength of words and/or phrases
Opinion Segment	Capturing individual sentiment/ segment from a compound sentence	It is a beautiful phone case but it is also hard to remove
Explicit aspect	The feature appears in the sentence	It is a very cute ***case***
Implicit aspect	The feature does not appear in the text	Arrived broken and very flimsy ***package and case?***
Subjectivity	Expressing desires, beliefs, proclamations, preferences, etc	Don't believe that these screen protectors have glue in them

Classification of Emotions

- Emotion: Most of living react in the same way to something happening outside or inside them
  - Examples: *anger, sadness, joy, fear, shame, pride, elation, desperation*
- Mood Definition: A diffuse affect state characterized by a change in subjective feeling, often without apparent cause
  - Intensity: Low intensity but relatively long duration
  - Examples: *Cheerful, gloomy, irritable, listless, depressed, buoyant*
- Interpersonal Stance: Affective stance taken toward another person in a specific interaction
  - Examples : Distant, cold, warm, supportive, contemptuous
- Attitudes: Relatively enduring, affectively colored beliefs, preferences, and predispositions
  - Components: *Beliefs, preferences, and predispositions*
  - Objects: Objects or persons

Sentiment as Text Classification

$$D = \{d_1, d_2, \cdots, d_n\}$$

$$C = \{c_1, c_2, \cdots, c_n\}$$

Computed class $c \in C$

Hand-coded Rules

- Hand coded
  - Rule $w \in W$ and $W \subseteq V \implies w \in V$
  - Accuracy is high
  - Defining V is expensive and domain specific

Supervised Learning

Positive

$$D = \{d_1, d_2, \cdots, d_n\}$$

Negative

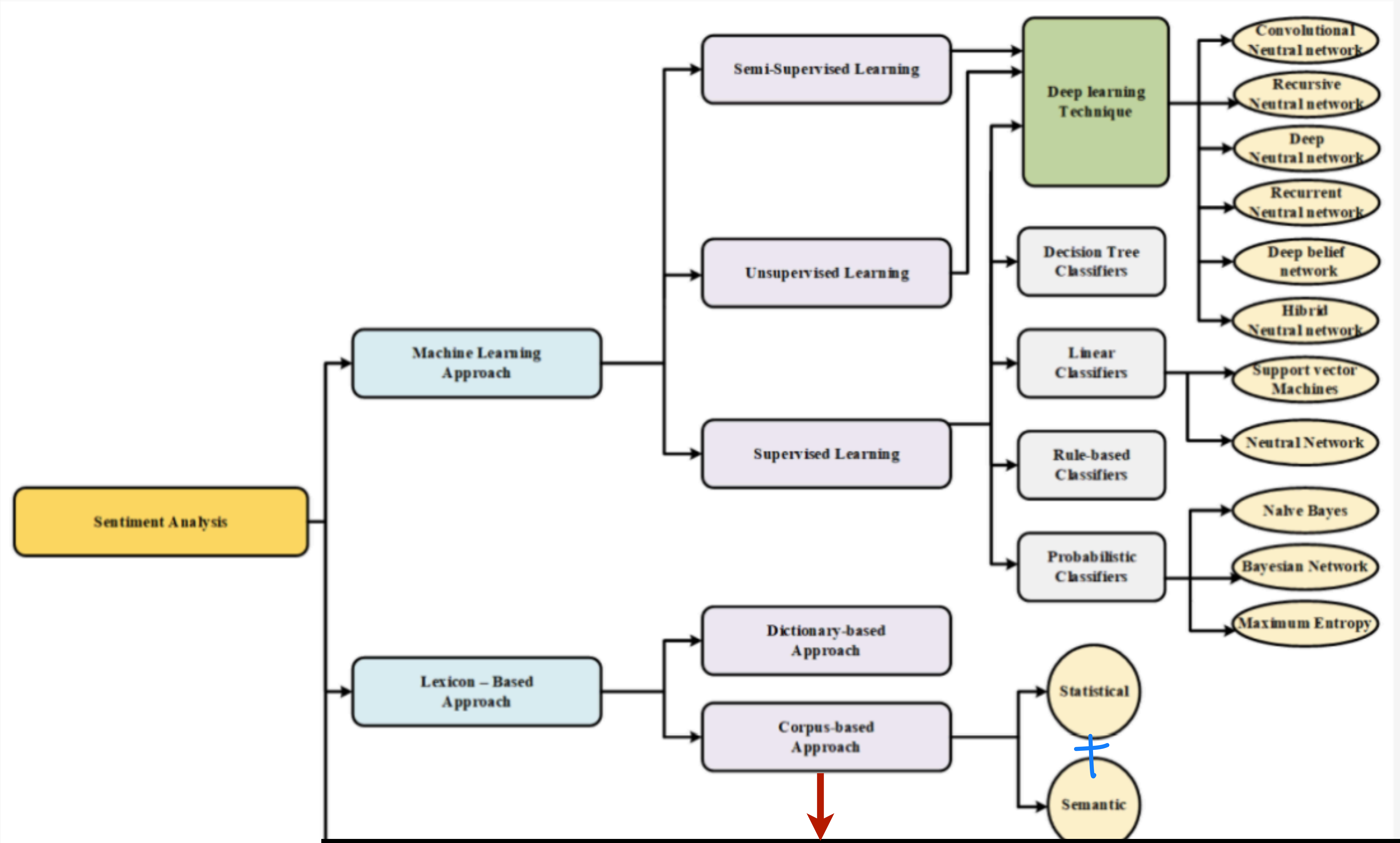
Undecided

$$C = \{c_1, c_2, \cdots, c_n\} \qquad T = \{(d_1, c_1), (d_2, c_2), \cdots, (d_n, c_n)\}$$

Model $\rightarrow f(w_i; \theta) \approx c_i$

Predicted class $c \in C$

Sentiment Classification Techniques



Syntax or cooccrence patterns are used. Word vectors can be used to identify similar words if a seed opinion word is provided

Source: Nhan Cach Dang María N. Moreno-García and Fernando De la Prieta , "Sentiment Analysis Based on Deep Learning: A Comparative Study", https://www.mdpi.com/2079-9292/9/3/483/htm

Sentiment Definition

Given a text find (t_i, a_{ij}, o_{ijk}) 1

It can be extended with more parameters when the opinion was created, who created, comparing entity, etc.

t_i is the target entity or product in the text and $t_i \in T$ where $T = \{t_1, t_2, t_3, \ldots, t_n\}$

a_{ij} is the set of aspects for the product t_i

O_{ijk} is the opinion expressed on the aspect a_{ij}. The scale of O_{ijk} could be one of the values such as positive, neutral or negative in the range $[a, b] = \{x \mid a \leq x \leq b\}$

We may include other parameters such as time, owner of the opinion, product version, platform where the opinion was recorded, country, etc.

Using 1, we can perform various analysis on the text

Sentiment Definition

I liked the **case** because it was cute, but the **studs** **fall off** easily and to **protect** a phone this would **not be recommended**. Buy if you just like it for **looks**

t_1 a_{11} o_{111} a_{12} o_{112}

Lexicon-based Model

- Dictionary used to classify the word based on sentiment lexicon
- Requires no training data, but is constructed from a generalizable, valence-based, human-curated gold standard[1]
- Libraries such as NLTK, Valence Aware Dictionary for sentiment Reasoning (VADER), SentiWordnet
- $S = \dfrac{number\ of\ positives - number\ of\ negatives}{total\ number\ of\ words}$

$$Sentiment = \begin{cases} positive, & if\ S = 1 \\ neutral, & if\ S = 0 \\ negative, & if\ S = -1 \end{cases}$$

[1] Hutto, C.J. & Gilbert, Eric. (2015). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Proceedings of the 8th International Conference on Weblogs and Social Media, ICWSM 2014.

WHAT IS VADER?

Valence Aware Dictionary and sEntiment Reasoner

- VADER is a rule-based sentiment analysis tool designed for social media and short text data.
- Handles slang, emojis, and informal language
- Classifies the word based on sentiment lexicon
- Sentiment scores are between -4 to +4
- Outputs Positive, Negative, Neutral, and Compound scores

WHAT IS VADER?

- Matches words to a sentiment lexicon
- Adjusts scores for
  - Intensity modifiers (e.g., 'very', 'slightly')
  - Negations (e.g., 'not', 'never')
  - Punctuation and capitalization
- Calculates sentiment proportions and compound score.

WHAT IS VADER?

- Simple and easy to use
- Pre-tuned; no training required
- Efficient for short and informal text
- Rule-based
- Sarcasm and irony are difficult to spot
- Less effective for domain-specific or long-form text.

Advantages

Limitations

VADER based Sentiment Analysis

- Requires no training data, but is constructed from a generalizable, valence-based, human-curated gold standard[1]
- Every lexicon has a score depending on its intensity [-4,4] -using 5 rules
  1. Punctuation mark
  2. Capitalisation
  3. Degree of modifiers/intensifiers
  4. Contrastive conjunction,
  5. Tri-gram before sentimentally loaded phrase
- The sentiment score is calculated by summing up the sentiment scores of each VADER-dictionary-listed word in the sentence

$$x = \sum_i s_{w_i}$$

- The ``compound`` score is computed by $\dfrac{x}{\sqrt{x^2 + \alpha}}$

[1] Hutto, C.J. & Gilbert, Eric. (2015). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Proceedings of the 8th International Conference on Weblogs and Social Media, ICWSM 2014.

Sample Python Code

```
import nltk
from nltk.sentiment import SentimentIntensityAnalyzer
nltk.download('vader_lexicon')
sia = SentimentIntensityAnalyzer()
texts = [
          '''I love this product! It's amazing and works perfectly.'''
        ]
for text in texts:
    sentiment = sia.polarity_scores(text)
    print(f"Text: {text}")
    print(f"Sentiment Scores: {sentiment}")
    print("-" * 50)
```

Sentiment Scores: {'neg': 0.0, 'neu': 0.286, 'pos': 0.714, 'compound': 0.9259}

Lexicon-based SA - Vader Results

It is a beautiful phone case but it is also hard to remove

compound: 0.2144, neg: 0.114, neu: 0.712, pos: 0.174,

It is a very cute case

compound: 0.5095, neg: 0.0, neu: 0.548, pos: 0.452,

Arrived broken and very flimsy

compound: -0.4767, neg: 0.437, neu: 0.563, pos: 0.0,

Don't believe that these screen protectors have glue in them

compound: 0.0, neg: 0.0, neu: 1.0, pos: 0.0,

I chose this case because it was beautiful

compound: 0.5994, neg: 0.0, neu: 0.606, pos: 0.394,

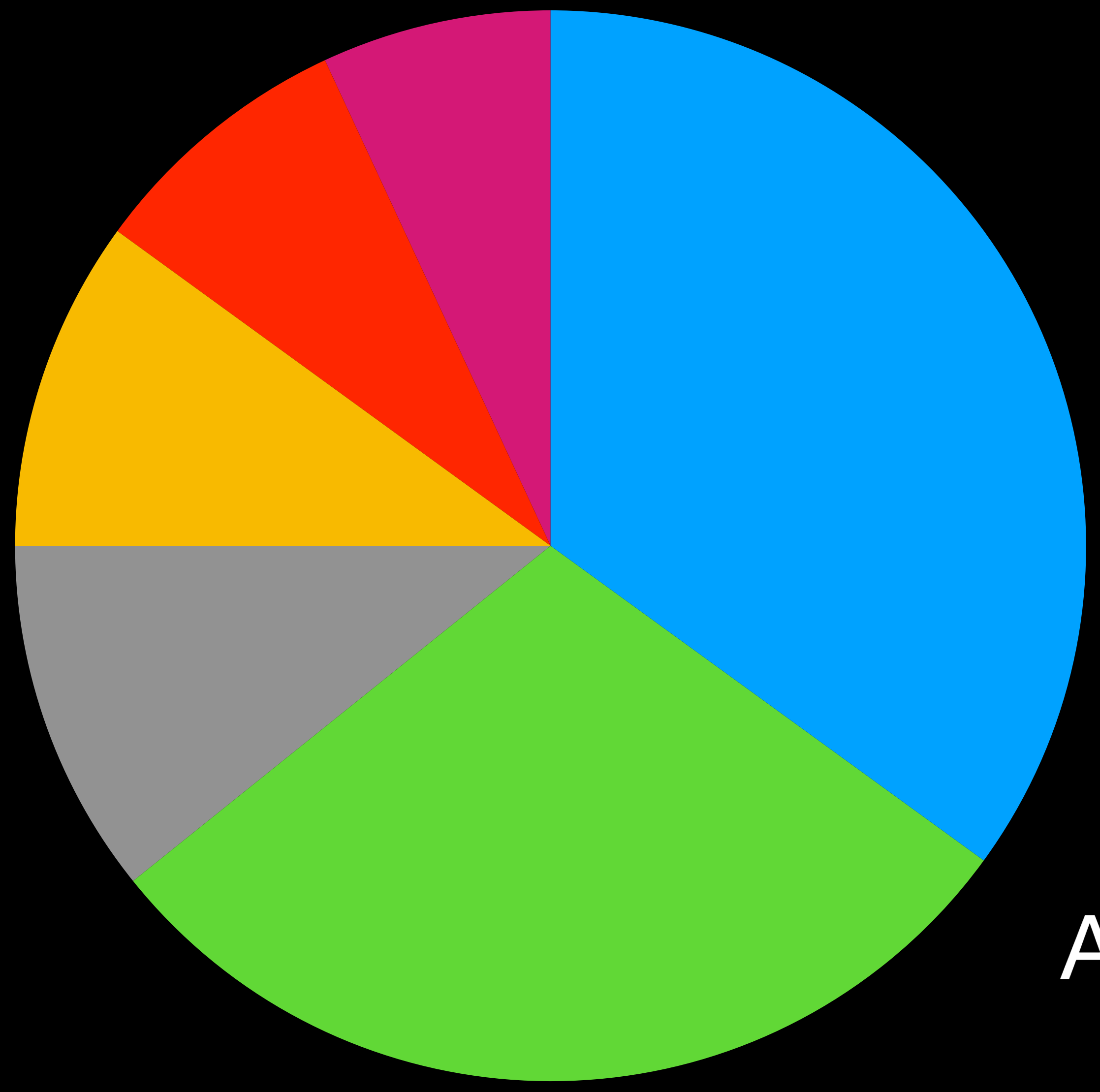
I liked it because it was cute, but the studs fall off easily and to protect a phone this would not be recommended. Buy if you just like it for looks

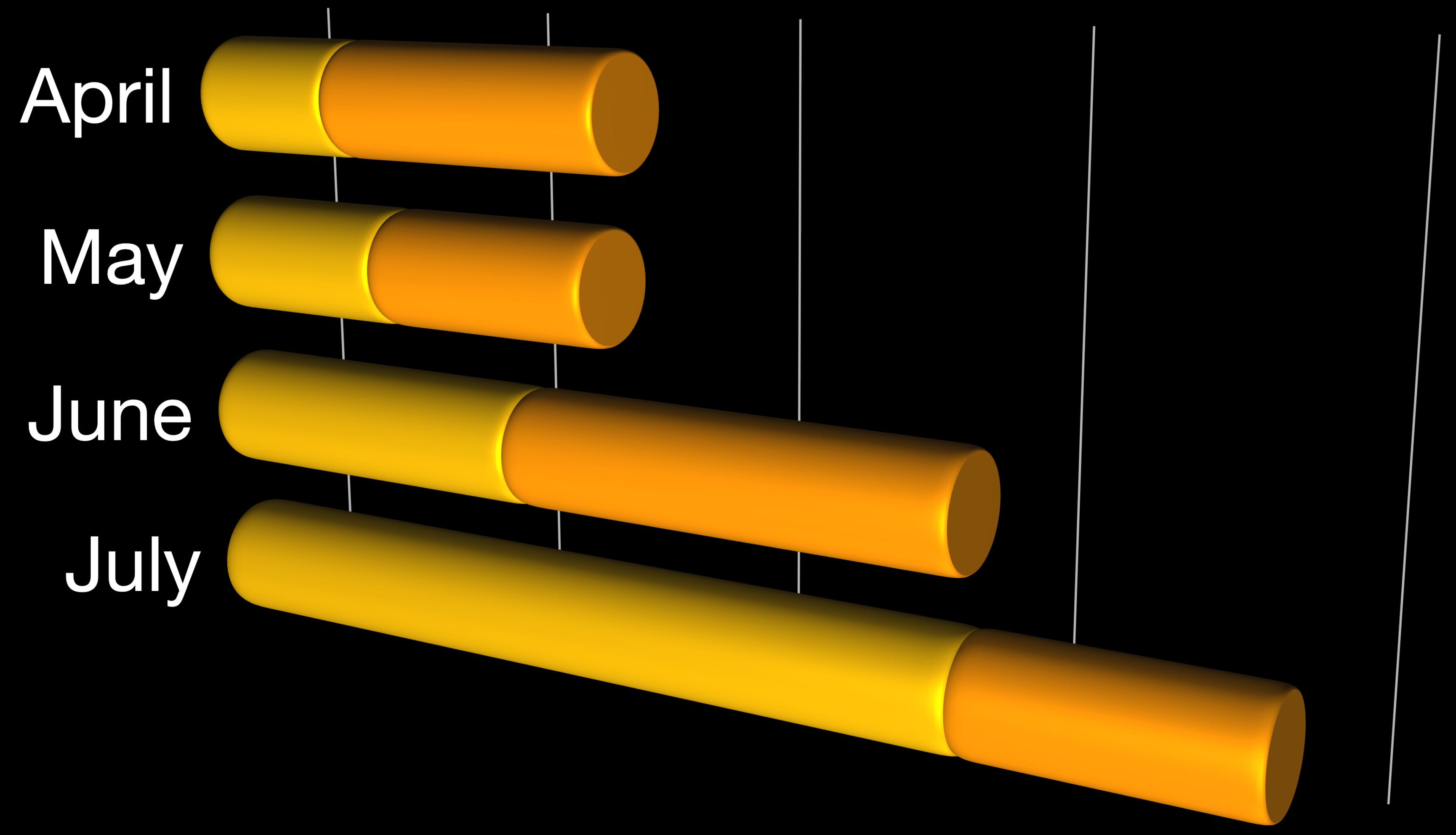
compound: 0.8948, neg: 0.049, neu: 0.597, pos: 0.354,

it works good but after few weeks it just stop working. the light just on but not charging

compound: -0.2144, neg: 0.135, neu: 0.771, pos: 0.094,

Visualisation of Sentiments





Naive Bayes Classifier

$$NBC = \arg\max_{c \in C} P(f_1, f_2, \cdots, f_n \mid c) P(c)$$

Likelihood

Prior

$f_1, f_2, \cdots, f_n$ - The document $\boldsymbol{d}$ is represented by the features

The location of the feature does not influence the classification.

Feature probabilities $P(f_i \mid c_k)$ are independent given the $\boldsymbol{c}_k$

$$P(\mathbf{f} \mid c_j) = P(f_1, f_2, \cdots, f_n \mid c_k) = \prod_{i=1}^{n} P(f_i \mid c_k)$$

Example

- Document - It is a beautiful phone case but it is also hard to remove
- Features - beautiful, hard

$$P(beatiful \mid +), P(beautiful \mid -) \quad c_i$$
$$P(hard \mid +), P(hard \mid -)$$
$$\ldots$$

$$P(+ \mid d) \propto \prod_{i=1}^{n} P(f_i \mid +) P(+)$$

Compute for $-$ also

$$P(- \mid \mathbf{f}) \propto \prod_{i=1}^{n} P(f_i \mid -) P(-)$$

Laplacce smoothing is applied for $f \notin V$

$$P(c_i) = \frac{Count(c_i)}{TotalCount}$$

Learn $P(f_1 \mid c_i) = \dfrac{Count(f_1, c_i)}{\sum_{f \in V} Count(\mathbf{f}, c_i)}$

Laplace smoothing

Learn $P(f_1 \mid c_i) = \dfrac{Count(f_1, c_i) + 1}{\sum_{f \in V} Count(\mathbf{f}, c_i) + |V|}$

Decision Rule

$$\hat{c} = \begin{cases} +, & \text{if } P(\mathbf{f}_{\text{test}} \mid y = +) P(y = +) > P(\mathbf{f}_{\text{test}} \mid y = -) P(y = -), \\ -, & \text{otherwise.} \end{cases}$$

A worked sentiment example with add-1 smoothing

	Cat	Documents
Training	-	just plain boring
	-	entirely predictable and lacks energy
	-	no surprises and very few laughs
	+	very powerful
	+	the most fun film of the summer
Test	?	predictable ~~with~~ no fun

1. Prior from training:

$$\hat{P}(c_j) = \frac{N_{c_j}}{N_{total}}$$

P(-) = 3/5

P(+) = 2/5

2. Drop "with"

3. Likelihoods from training:

$$p(w_i|c) = \frac{count(w_i, c) + 1}{(\sum_{w \in V} count(w, c)) + |V|}$$

$$P(\text{“predictable”}|-) = \frac{1+1}{14+20} \qquad P(\text{“predictable”}|+) = \frac{0+1}{9+20}$$

$$P(\text{“no”}|-) = \frac{1+1}{14+20} \qquad P(\text{“no”}|+) = \frac{0+1}{9+20}$$

$$P(\text{“fun”}|-) = \frac{0+1}{14+20} \qquad P(\text{“fun”}|+) = \frac{1+1}{9+20}$$

4. Scoring the test set:

$$\boxed{P(-)P(S|-) = \frac{3}{5} \times \frac{2 \times 2 \times 1}{34^3} = 6.1 \times 10^{-5}}$$

$$P(+)P(S|+) = \frac{2}{5} \times \frac{1 \times 1 \times 2}{29^3} = 3.2 \times 10^{-5}$$

Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition with language models, D. Jurafsky and J. H. Martin, 3Rd edition, 2025. Online manuscript released January 12, 2025

Find the sentiment using Naive Bayes Classification

$$\begin{bmatrix} word & positive & negative \\ I & 0.09 & 0.16 \\ love & 0.07 & 0.06 \\ to & 0.05 & 0.07 \\ fill & 0.29 & 0.06 \\ credit & 0.04 & 0.15 \\ card & 0.08 & 0.11 \\ application & 0.06 & 0.04 \end{bmatrix}$$

What class Naive Bayes classifier would assign to the sentence
"I do not like to fill in the application form"?

SA using RNN - I

Input - Word embeddings

RNN

$$\mathbf{h}_t = f(\mathbf{W}_h\mathbf{x}_t + \mathbf{U}_h\mathbf{h}_{t-1} + \mathbf{b}_h)$$

LSTM

Recurrent Layer

Forget Gate: $\mathbf{f}_t = \sigma(\mathbf{W}_f\mathbf{x}_t + \mathbf{U}f\mathbf{h}_{t-1} + \mathbf{b}_f)$

Input Gate: $\mathbf{i}_t = \sigma(\mathbf{W}_i\mathbf{x}_t + \mathbf{U}_i\mathbf{h}_{t-1} + \mathbf{b}_i)$

Outpput Gate: $\mathbf{o}_t = \sigma(\mathbf{W}_o\mathbf{x}_t + \mathbf{U}_o\mathbf{h}_{t-1} + \mathbf{b}_o)$

Cell state update: $\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tanh(\mathbf{W}_c\mathbf{x}_t + \mathbf{U}_c\mathbf{h}_{t-1} + \mathbf{b}_c)$

Hidden state: $\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t)$

$\mathbf{h}_{\text{final}} = \mathbf{h}_T$ - Aggregation of all hidden states

SA using RNN - II

Output Layer

$$P(+ \mid \mathbf{x}) = \sigma(\mathbf{W}_{\text{out}}\mathbf{h}_T + \mathbf{b}_{\text{out}})$$

$$P(c \mid \mathbf{x}) = \text{Softmax}(\mathbf{W}_{\text{out}}\mathbf{h}_{\text{final}} + \mathbf{b}_{\text{out}})$$

Loss

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(P_i) + (1 - y_i)\log(1 - P_i)\right] -$$ Binary cross-entropy for binay classification

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{i,j}\log(P_{i,j}) -$$ Categorical cross-entropy for multi-class

SA using RNN - III

Mapping Sentiments

High probability for class $c = +$: $P(+ \mid \mathbf{x}) \gg P(- \mid \mathbf{x})$

High probability for class $c = -$: $P(- \mid \mathbf{x}) \gg P(- \mid \mathbf{x})$

$$P(\text{neutral} \mid \mathbf{x}) \approx P(+ \mid \mathbf{x}) \approx (P - \mid \mathbf{x})$$

Aspect Identification

POS tags

demonstrative (that, those)

Tag	Description	Tag	Description
CC	Coordinating conjunction	RBR	Adverb, comparative
CD	Cardinal number	RBS	Adverb, superlative
DT	Determiner a, an, the, + 5 cat quantifiers (much, a bit, all, few, little)	RP	Particle
EX	Existential there	SYM	Symbol
FW	Foreign word	TO	to
IN	Preposition or subordinating conjunction	UH	Interjection goodbye
JJ	Adjective	VB	Verb, base form
JJR	Adjective, comparative	VBD	Verb, past tense
JJS	Adjective, superlative	VBG	Verb, gerund or present participle
LS	List item marker	VBN	Verb, past participle
MD	Modal could, will	VBP	Verb, non3rd person singular present
NN	Noun, singular	VBZ	Verb, 3rd person singular present
NNS	Noun, plural	WDT	Whdeterminer
NNP	Proper noun, singular	WP	Whpronoun
NNPS	Proper noun, plural’s	WP$	Possessive whpronoun
PDT	Predeterminer – occur b4 DT, all, both	WRB	Whadverb
POS	Possessive ending		
PRP	Personal pronoun hers, him …		
PRP$	Possessive pronoun – his, her		
RB	Adverb		

Little bit of Grammar :)

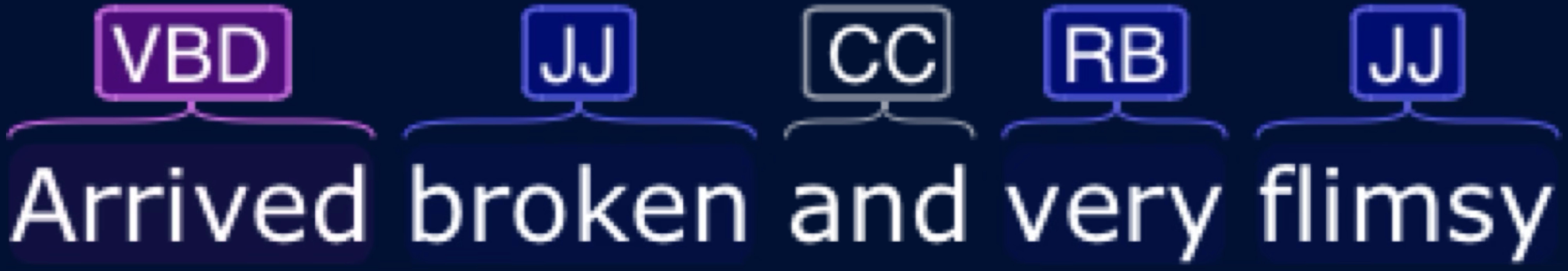
It is a beautiful phone case but it is also hard to remove

Little bit of Grammar

It is a very cute ***case***

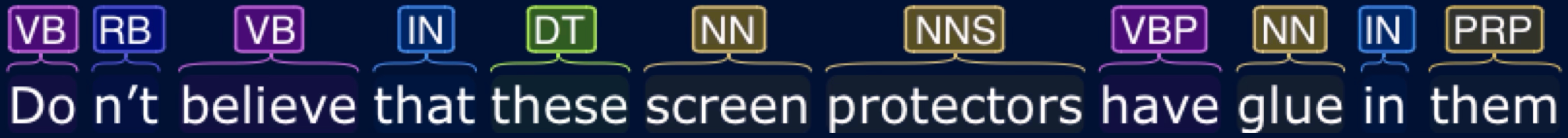
Little bit of Grammar

Arrived broken and very flimsy



Little bit of Grammar

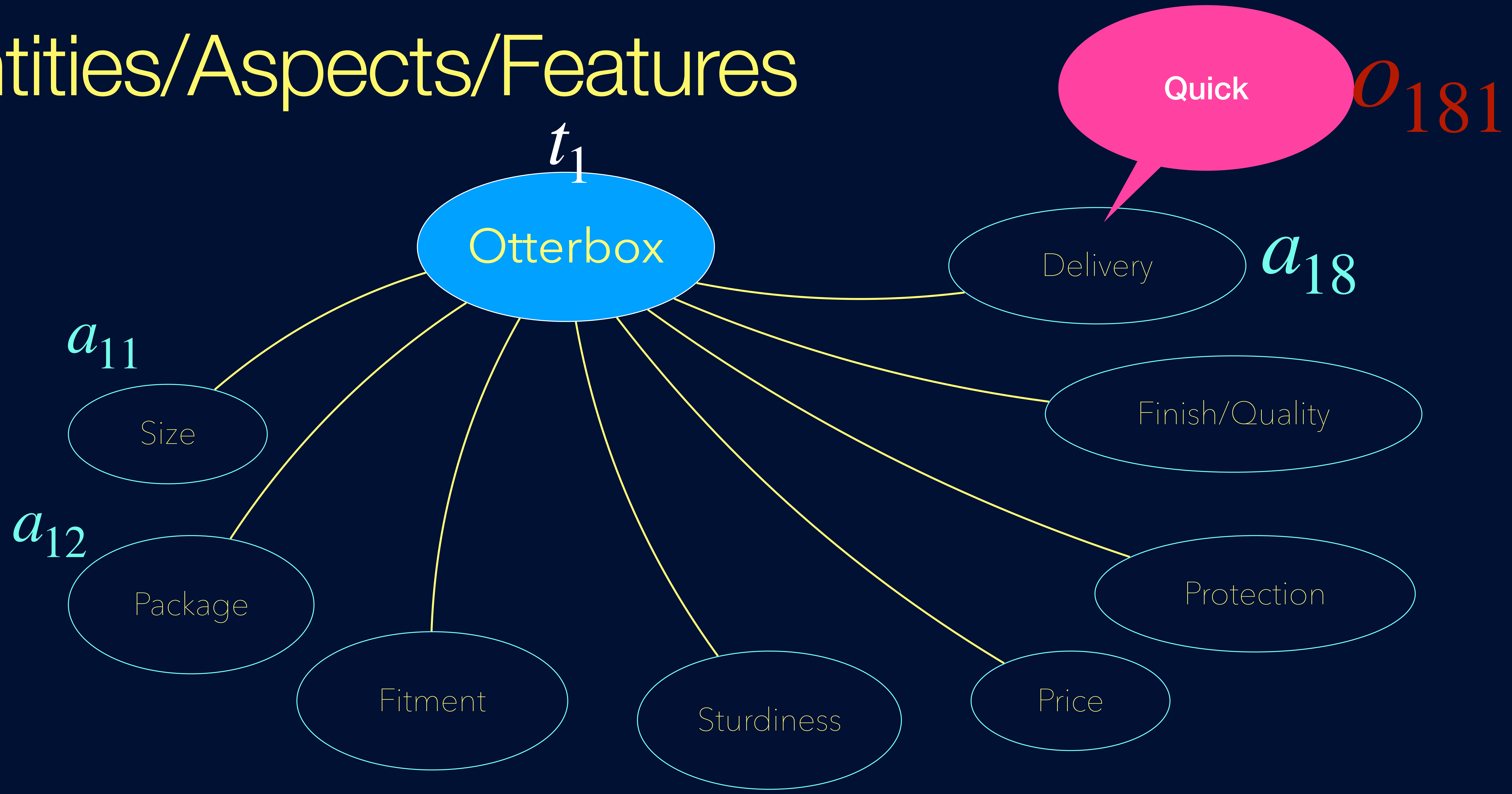
Don’t believe that these screen protectors have glue in them



Little bit of Grammar

I chose this case because it was beautiful

Entities/Aspects/Features



Sentiment/opinion can be expressed on each node

Explicit and Implicit Aspects

The quality of the cover is very good

I love,love,love it fits great so pretty and sturdy

this case is so cute it looks good on my white iphone its pretty good quality

This case may look very breakable but it is very sturdy

The battery life of this phone is excellent.

Another very good phone charger that does the job and has a very affordable price that anyone can afford to buy it

Was here right on time. Wall adapter what I expected and works well but dose Not Have Quick-Charge…

This one is heavier than the last model

It lasted an entire day without charging!

It's hard to read in sunlight

Using the sentiment words and context, implicit and explicit words can be detected using POS and dependency relationships.

Aspects and their sentiments

Looks even better in person. Be careful to not drop your phone so often because the rhinestones will fall off (duh). More of a decorative case than it is protective, but I will say that it fits perfectly and securely on my phone. Overall, very pleased with this purchase."

- Positive Aspects
  - Looks even better in person. (Explicit: Appearance)
  - It fits perfectly and securely. (Explicit: Functionality/Fit)
  - Overall, very pleased with this purchase. (Implicit: Satisfaction)
- Negative Aspects
  - Be careful to not drop your phone…(Implicit: Fragility)
  - Rhinestones will fall off (Explicit: Durability)
  - More of a decorative case than protective. (Mixed: Type/Protectiveness)

Explicit/Implicit Aspect Extraction

Looks even better in person. Be careful to not drop your phone so often because the rhinestones will fall off (duh). More of a decorative case than it is protective, but I will say that it fits perfectly and securely on my phone. Overall, very pleased with this purchase."

Rule for fits perfectly and securely?

[('not', 'RB'), ('drop', 'VB')]
ASPECT: not drop

[('decorative', 'JJ'), ('case', 'NN')]
ASPECT: decorative case

[('it', 'PRP'), ('is', 'VBZ'), ('protective', 'JJ')]

ASPECT: it is protective

[('very', 'RB'), ('pleased', 'JJ')]
ASPECT: very pleased ?

Patterns of POS tags for Aspect Mining

syntactic templates

	First Word	Second Word	Third Word (Not Extracted)
1.	JJ	NN or NNS	anything
2.	RB, RBR, or RBS	JJ	not NN nor NNS
3.	JJ	JJ	not NN nor NNS
4.	NN or NNS	JJ	not NN nor NNS
5.	RB, RBR, or RBS	VB, VBD, VBN, or VBG	anything

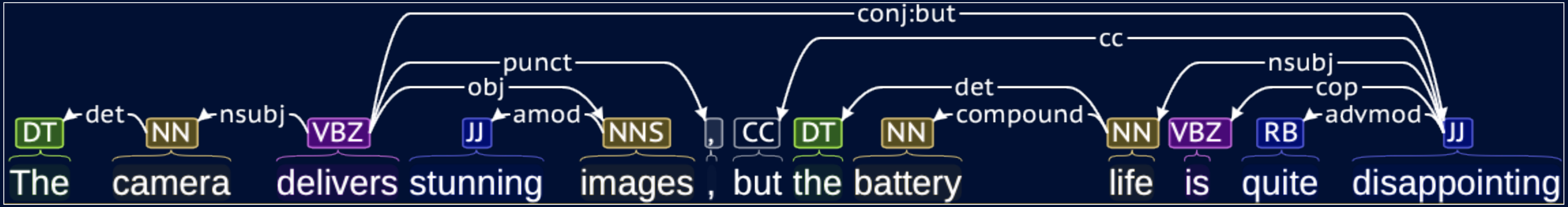
Frame your own rules

	First word	Second word	Third word
	NN	VBZ	RB or/and JJ

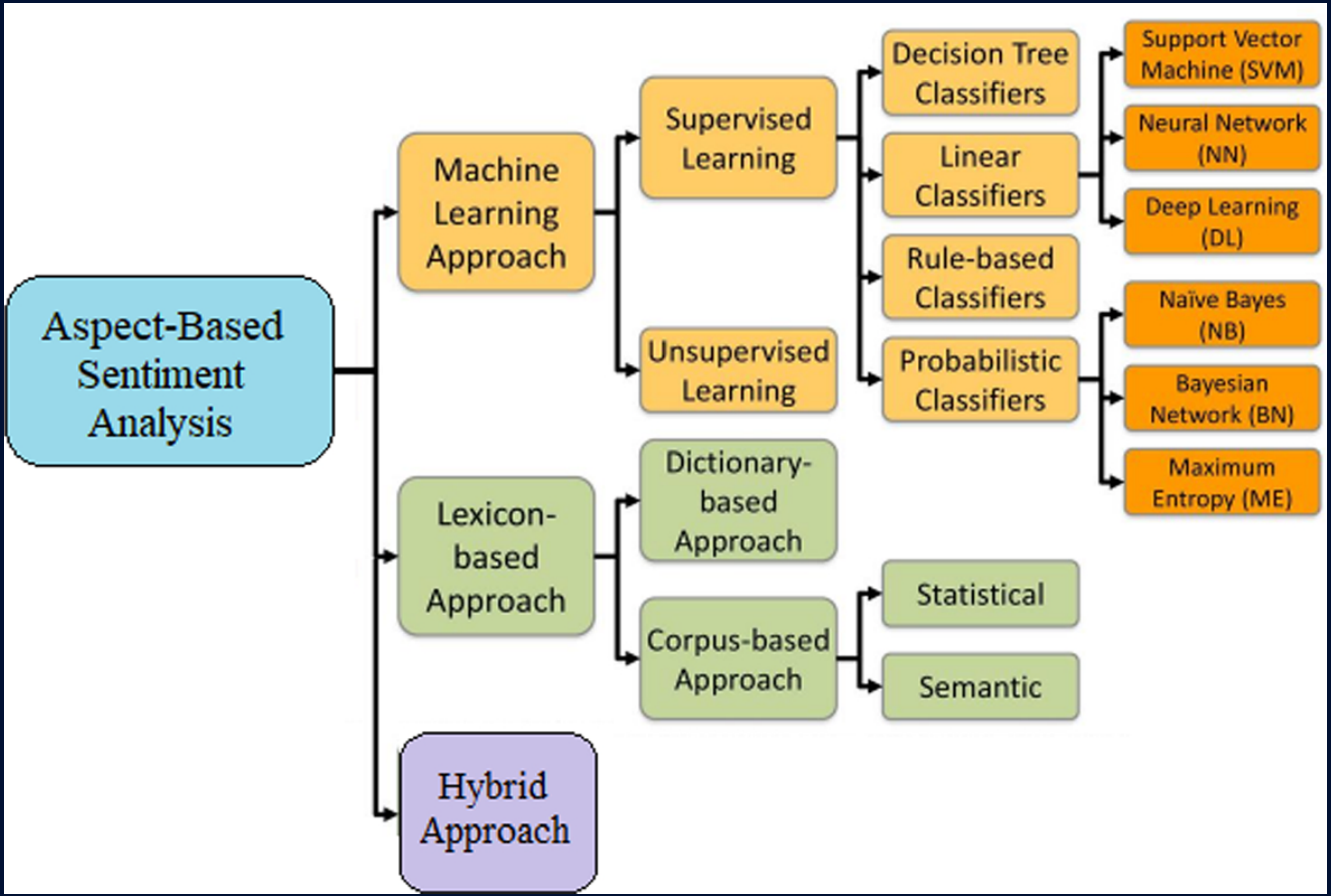
Reference: Peter D. Turnkey, "Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews", Proc. of the 40th Annual Meeting on the Association of Computationsl Linguistics (ACL), July 2002, https://arxiv.org/abs/cs/0212032

Aspect-based Sentiment Extraction (ABSA)

ABSA is a fine-grained sentiment analysis task that aims to identify for a given target (t_i) the aspect term (a_{ij}), and the opinion term (o_{ijk}) and its corresponding sentiment polarity (s_{ijkp}).


conj:but
cc
punct
nsubj
obj
det
cop
det
nsubj
amod
compound
advmod
DT
NN
VBZ
JJ
NNS
,
CC
DT
NN
NN
VBZ
RB
JJ
The
camera
delivers
stunning
images
,
but
the
battery
life
is
quite
disappointing

Different Approaches



Subtasks of ABSA

1. Aspect Term Extraction (ATE): This task identifies the focal points of sentiment in the text, such as “camera” and “battery life” in a smartphone review.
2. Opinion Term Extraction (OTE) aims to identify adjectives or adverbs expressing feelings or attitudes towards aspects, such as “stunning” for the camera or “disappointing” for battery life.
3. Aspect-Level Sentiment Classification (ALSC) categorizes the sentiment towards each aspect as positive, negative, or neutral
4. Aspect-Oriented Opinion Extraction (AOE) associates sentiments with their corresponding aspects, linking “stunning” to “camera.”
5. Aspect Extraction and Sentiment Classification (AESC): Aspects are tagged with their sentiment in one step 1
6. Aspect-Opinion Pair Extraction (AOPE)P: airs each aspect with its qualifying opinion, forming pairs like ("camera", “stunning")
7. Aspect Sentiment Triplet Extraction (ASTE): Combines aspects, opinions, and sentiments into a triplet

Aziz, K., Ji, D., Chakrabarti, P. et al. Unifying aspect-based sentiment analysis BERT and multi-layered graph convolutional networks for comprehensive sentiment dissection. Sci Rep 14, 14646 (2024). https://doi.org/10.1038/s41598-024-61886-7

Hang Yan, Junqi Dai, Tuo Ji, Xipeng Qiu, and Zheng Zhang. 2021. A Unified Generative Framework for Aspect-based Sentiment Analysis. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 2416–2429, Online. Association for Computational Linguistics.

Aspects using POS

Aspect Extraction using Rules

```
from nltk import pos_tag,
word_tokenize
from nltk.chunk import RegexpParser

text = '''
I liked the phone case because it was
cute, but the studs fall off easily
and to protect a phone this would not
be recommended. Buy if you just like
it for looks
'''
pos_tags =
pos_tag(word_tokenize(text))
```

```
grammar = r'''
    ASPECT: {<JJ>*<NN|NNS>+} (1)
        (2) {<NN|NNS>+<IN><NN|NNS>}
            {<NN|NNS><CC><NN|NNS>}
        (3) {<JJ><NN><CC><NN>}
        (4) {<VB><NN|NNS>}
    '''

parser = RegexpParser(grammar)
tree = parser.parse(pos_tags)
print(tree)
```

(1) Compound Noun Phrases

(2) Adjective + Noun + Conjunction + Noun

(3) Preposition + Noun

(4) Verb + Noun

Aspect Tree Format

```
(S
  I/PRP
  liked/VBD
  the/DT
  (ASPECT phone/NN case/NN)
  because/IN
  it/PRP
  was/VBD
  cute/JJ
  ,/,
  but/CC
  the/DT
  (ASPECT studs/JJ fall/NN)
  off/IN
  easily/RB
  and/CC
  to/TO
  protect/VB
  a/DT
  (ASPECT phone/NN)
  this/DT
  would/MD
  not/RB
  be/VB
  recommended/VBN
  ./.
  Buy/VB
  if/IN
  you/PRP
  just/RB
  like/IN
  it/PRP
  for/IN
  (ASPECT looks/NNS))
```

Extracted Aspects: ['phone case', 'studs fall', 'phone', 'looks']

BIO Format

- A tagging scheme used for sequence labeling tasks
  - Aspect Extraction
  - Named Entity Recognition (NER)
- Each word in a sentence is assigned a tag:
  - B -   Beginning of an entity
  - I   -   Inside the same entity
  - O - Outside any entity (not part of an entity)

The food was great, but the service was not gteat."
Tokens: ["The", "food", "was", "great", ",", "but", "the", "service", "was", "not", “great”, ”."]
BIO Tags: ["O", "B-Aspect", "O", "O", "O", "O", "O", "B-Aspect", "O", "O", “O”, “O”]