NLP with Classification and Vector Spaces

Week 2. Sentiment Analysis with Naïve Bayes

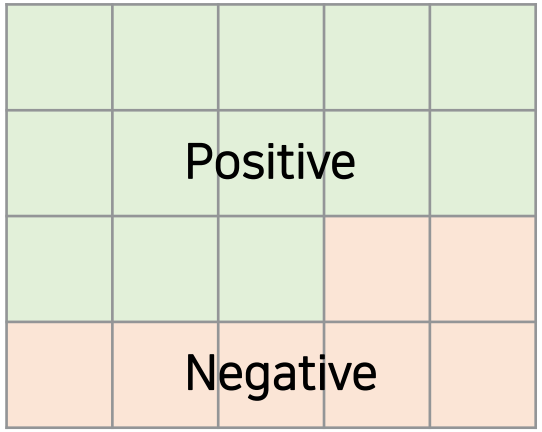
From ***Natural Language Processing Specialization*** lectures on coursera

SoYeop Yoo

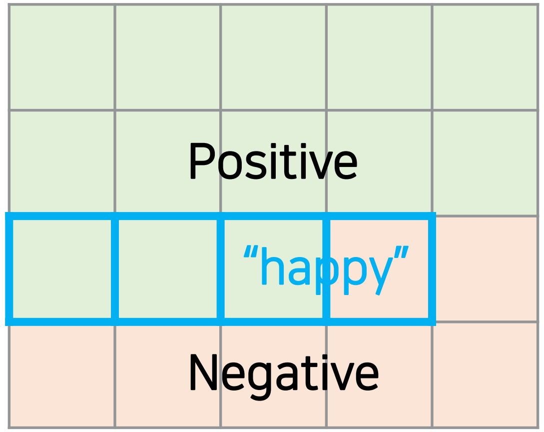
01 Probability and Bayes' Rule

Introduction

✱Corpus on tweets



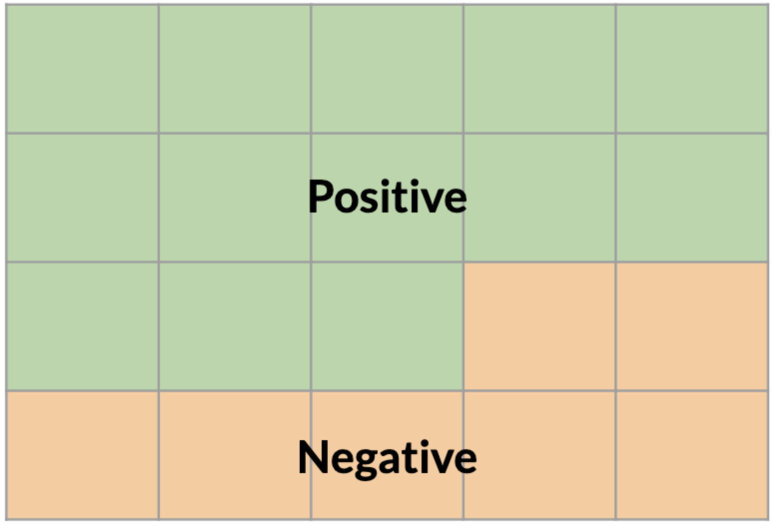
✱Tweets containing the word "happy"



01 Probability and Bayes' Rule

Probabilities

Corpus of tweets



A → Positive tweet

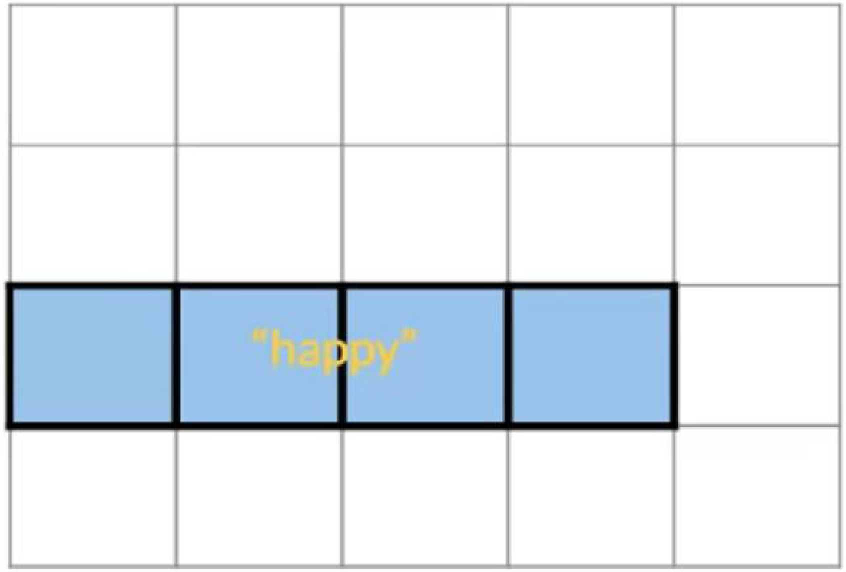
$P(A) = N_{pos} / N = 13 / 20 = 0.65$

$P(Negative) = 1 - P(Positive) = 0.35$

01 Probability and Bayes' Rule

Probabilities

Tweets containing the word "happy"



B → tweet contains "happy".

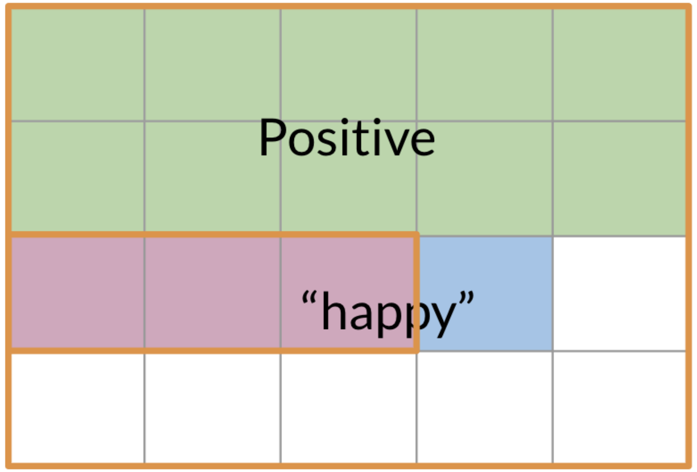
$P(B) = P(happy) = N_{happy} / N$

$P(B) = 4 / 20 = 0.2$

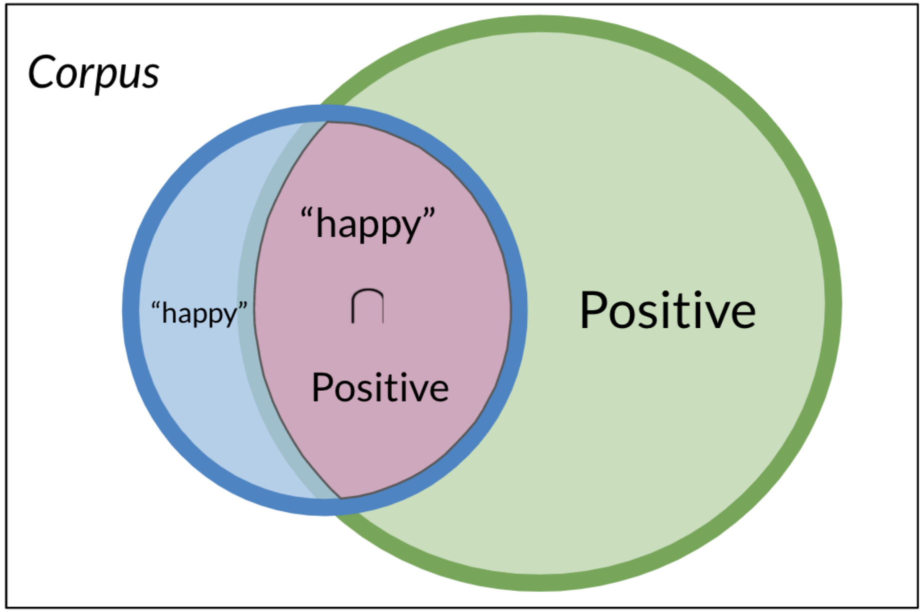
01 Probability and Bayes' Rule

Probability of the Intersection

- To compute the probability of 2 events happening
  - Ex. "happy" and "positive"



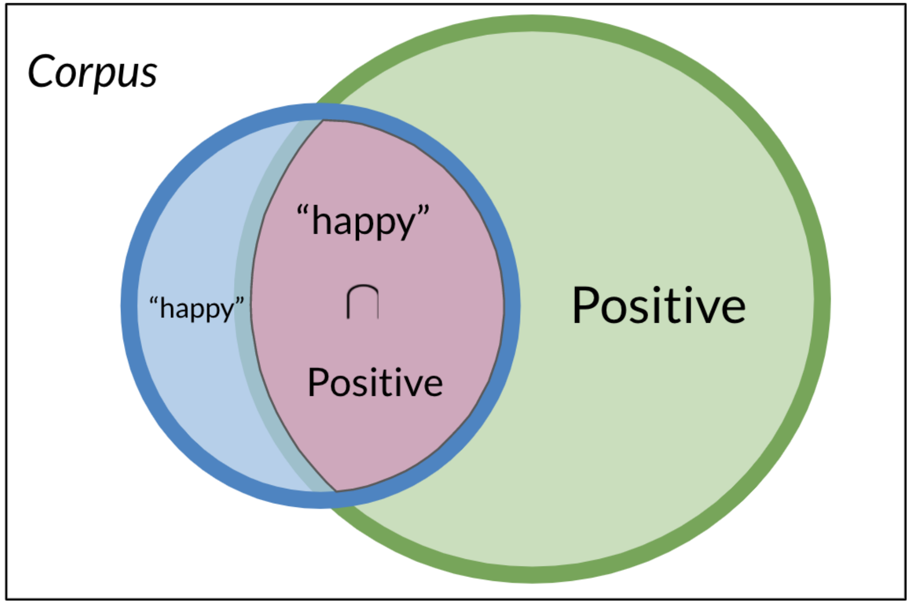
$$P(A \cap B) = P(A, B) = \frac{3}{20} = 0.15$$



02 Bayes' Rule

Conditional Probabilities

- Probability of B, given A happened
- Looking at the elements of set A, the chance that one also belongs to set B



$$P(\text{Positive} \mid \text{“happy”}) = \frac{P(\text{Positive} \cap \text{“happy”})}{P(\text{“happy”})}$$

02 Bayes' Rule

Bayes' Rule

$$P(\text{Positive} | \text{“happy”}) = \frac{\boxed{P(\text{Positive} \cap \text{“happy”})}}{P(\text{“happy”})}$$

$$P(\text{“happy”} | \text{Positive}) = \frac{\boxed{P(\text{“happy”} \cap \text{Positive})}}{P(\text{Positive})}$$

$$\boxed{P(\text{Positive} | \text{“happy”}) =} \boxed{P(\text{“happy”} | \text{Positive})} \times \boxed{\frac{P(\text{Positive})}{P(\text{“happy”})}}$$

02 Bayes' Rule

Bayes' Rule

- $P(X|Y) = \frac{P(Y|X)P(X)}{P(Y)}$

03 Naïve Bayes Introduction

Naïve Bayes for Sentiment Analysis

Positive tweets

I am happy because I am learning NLP

I am happy, not sad.

Negative tweets

I am sad, I am not learning NLP

I am sad, not happy

word	Pos	Neg
I	3	3
am	3	3
happy	2	1
because	1	0
learning	1	1
NLP	1	1
sad	1	2
not	1	2
N_{class}	13	12

03 Naïve Bayes Introduction

Naïve Bayes for Sentiment Analysis

- $P(w_i|class)$

word	Pos	Neg
I	3	3
am	3	3
happy	2	1
because	1	0
learning	1	1
NLP	1	1
sad	1	2
not	1	2
N_{class}	13	12

$$P(\mathrm{I}|Pos) = \frac{3}{13} \qquad P(\mathrm{I}|Neg) = \frac{3}{12}$$

⋮

word	Pos	Neg
I	0.24	0.25
am	0.24	0.25
happy	0.15	0.08
because	0.08	0
learning	0.08	0.08
NLP	0.08	0.08
sad	0.08	0.17
not	0.08	0.17

03 Naïve Bayes Introduction

Naïve Bayes for Sentiment Analysis

- Compute the likelihood score

Tweet: I am happy today; I am learning.

$$\prod_{i=1}^{m} \frac{P(w_i|pos)}{P(w_i|neg)} = \frac{0.14}{0.10} = 1.4 \quad > 1$$

$$\cancel{\frac{0.20}{0.20}} * \cancel{\frac{0.20}{0.20}} * \boxed{\frac{0.14}{0.10}} * \cancel{\frac{0.20}{0.20}} * \cancel{\frac{0.20}{0.20}} * \cancel{\frac{0.10}{0.10}}$$

word	Pos	Neg
I	0.20	0.20
am	0.20	0.20
happy	0.14	0.10
because	0.10	0.05
learning	0.10	0.10
NLP	0.10	0.10
sad	0.10	0.15
not	0.10	0.15

04 Laplacian Smoothing

Laplacian Smoothing

- The probability of a word given a class
  - $P(w_i|class) = \frac{freq(w_i,class)}{N_{class}}$ $\quad class \in \{Positive, Negative\}$
- To avoid $P(w_i|class) = 0$
  - $P(w_i|class) = \frac{freq(w_i,class)+1}{N_{class}+V}$
    - N_{class}: frequency of all words in class
    - V: number of unique words in vocabulary

04 Laplacian Smoothing

$P(w_i|class)$ with Laplacian Smoothing

word	Pos	Neg
I	3	3
am	3	3
happy	2	1
because	1	0
learning	1	1
NLP	1	1
sad	1	2
not	1	2
Nclass	13	12

$$P(\mathrm{I}|Pos) = \frac{3+1}{13+8} \qquad P(\mathrm{I}|Neg) = \frac{3+1}{12+8}$$

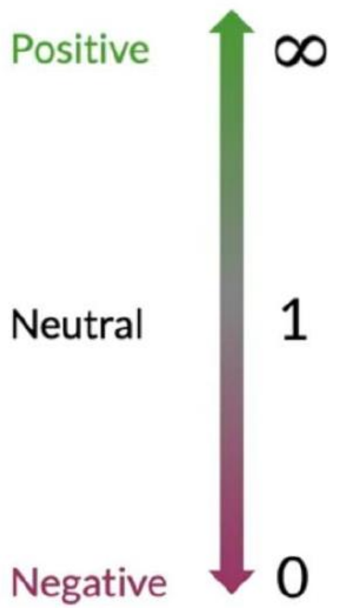
$$\vdots$$

word	Pos	Neg
I	0.19	0.20
am	0.19	0.20
happy	0.14	0.10
because	0.10	0.05
learning	0.10	0.10
NLP	0.10	0.10
sad	0.10	0.15
not	0.10	0.15
Sum	1	1

05 Log Likelihood

Ratio of Probabilities

- To compute the log likelihood, we need to get the ratios
- The higher the ratio, the more positive the word is



word	Pos	Neg	ratio
I	0.20	0.20	1
am	0.20	0.20	1
happy	0.14	0.10	1.4
because	0.10	0.10	1
learning	0.10	0.10	1
NLP	0.10	0.10	1
sad	0.10	0.15	0.6
not	0.10	0.15	0.6

$$\text{ratio}(w_i) = \frac{P(w_i \mid \text{Pos})}{P(w_i \mid \text{Neg})}$$

$$\approx \frac{freq(w_i, 1) + 1}{freq(w_i, 0) + 1}$$

05 Log Likelihood

Naïve Bayes' Inference

- $\boxed{\frac{P(pos)}{P(neg)}} \boxed{\prod_{i=1}^{m} \frac{P(w_i|pos)}{P(w_i|neg)}} > 1$

- A simple, fast, and powerful baseline
- A probabilistic model used for classification

05 Log Likelihood

Log Likelihood

- Products bring risk of underflow
  → to reduce the risk of numerical underflow
- $\log\left(\frac{P(pos)}{P(neg)}\prod_{i=1}^{m}\frac{P(w_i|pos)}{P(w_i|neg)}\right) \Rightarrow \underbrace{\log\frac{P(pos)}{P(neg)}}_{\text{log prior}} + \sum_{i=1}^{n}\underbrace{\log\frac{P(w_i|pos)}{P(w_i|neg)}}_{\text{log likelihood}}$

05 Log Likelihood

Summing the Lambdas

doc: I am happy because I am learning.

$$\lambda(w) = log\frac{P(w|pos)}{P(w|neg)}$$

$$\lambda(\text{happy}) = log\frac{0.09}{0.01} \approx 2.2$$

word	Pos	Neg	λ
I	0.05	0.05	0
am	0.04	0.04	0
happy	0.09	0.01	2.2
because	0.01	0.01	0
learning	0.03	0.01	1.1
NLP	0.02	0.02	0
sad	0.01	0.09	-2.2
not	0.02	0.03	-0.4

05 Log Likelihood

Inference with Log Likelihood

doc: I am happy because I am learning.

$$\sum_{i=1}^{m} log \frac{P(w_i|pos)}{P(w_i|neg)} = \sum_{i=1}^{m} \lambda(w_i)$$

log likelihood = 0 + 0 + 2.2 + 0 + 0 + 0 + 1.1 = **3.3**

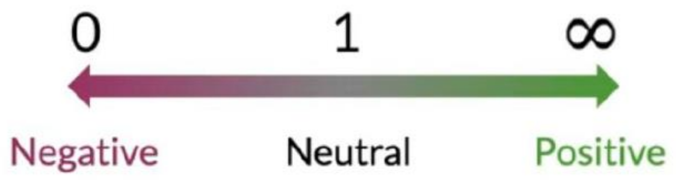
word	Pos	Neg	λ
I	0.05	0.05	0
am	0.04	0.04	0
happy	0.09	0.01	2.2
because	0.01	0.01	0
learning	0.03	0.01	1.1
NLP	0.02	0.02	0
sad	0.01	0.09	-2.2
not	0.02	0.03	-0.4

05 Log Likelihood

Summary of Log Likelihood

- It makes many things simpler and helps with numerical stability

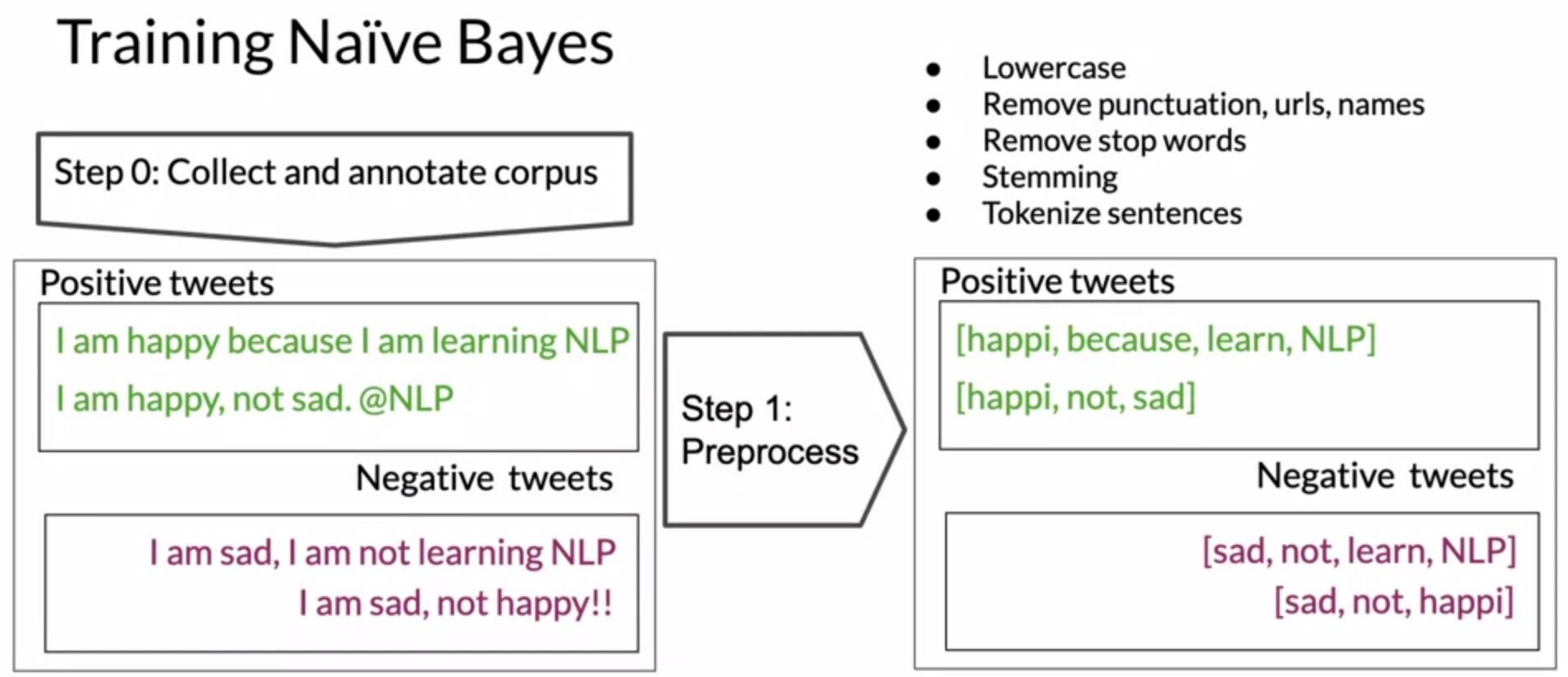
$$\prod_{i=1}^{m} \frac{P(w_i|pos)}{P(w_i|neg)} > 1$$



$$\sum_{i=1}^{m} log \frac{P(w_i|pos)}{P(w_i|neg)} > 0$$

06 Training Naïve Bayes

Step 0 & 1



06 Training Naïve Bayes

Step 2

Training Naïve Bayes

Positive tweets

[happi, because, learn, NLP]

[happi, not, sad]

Negative tweets

[sad, not, learn, NLP]

[sad, not, happi]

Step 2: Word count

freq(w, class)

word	Pos	Neg
happi	2	1
because	1	0
learn	1	1
NLP	1	1
sad	1	2
not	1	2
N_{class}	7	7

06 Training Naïve Bayes

Step 3 & 4

Training Naïve Bayes

freq(w, class)

word	Pos	Neg
happi	2	1
because	1	0
learn	1	1
NLP	1	1
sad	1	2
not	1	2
N_{class}	7	7

Step 3: $P(\mathrm{w}|\mathrm{class})$

$$V_{\mathrm{class}} = 6$$

$$\frac{\mathrm{freq}(\mathrm{w}, \mathrm{class}) + 1}{N_{\mathrm{class}} + V_{\mathrm{class}}}$$

$$\lambda(w) = log\frac{P(\mathrm{w}|\mathrm{pos})}{P(\mathrm{w}|\mathrm{neg})}$$

Step 4: Get lambda

word	Pos	Neg
happy	0.23	0.15
because	0.15	0.07
learning	0.08	0.08
NLP	0.08	0.08
sad	0.08	0.17
not	0.08	0.17

06 Training Naïve Bayes

Steps

0. Get or annotate a dataset with positive and negative tweets
1. Preprocess the tweets: $process\_tweet(tweet)$ → $[w_1, w_2, w_3, \cdots]$
2. Compute $freq(w, class)$
3. Get $P(w|pos), P(w|neg)$
4. Get $\lambda(w)$
5. Compute $\text{logprior} = \log(P(pos)/P(neg))$

07 Testing Naïve Bayes

Predict using Naïve Bayes

- log-likelihood dictionary $\lambda(w) = log\frac{P(w|pos)}{P(w|neg)}$
- $logprior = log\frac{D_{pos}}{D_{neg}} = 0$
- Tweet: [I, pass, the, NLP, interview]

$$score = -0.01 + 0.5 - 0.01 + 0 + logprior = 0.48$$

$$pred = score > 0$$

word	λ
I	-0.01
the	-0.01
happi	0.63
because	0.01
pass	0.5
NLP	0
sad	-0.75
not	-0.75

07 Testing Naïve Bayes

Testing Naïve Bayes

- Given $X_{val}, Y_{val}, \lambda, logprior$
  - $score = predict(X_{val}, \lambda, logprior)$
  - $pred = score > 0$

$$\begin{bmatrix} 0.5 \\ -1 \\ 1.3 \\ \vdots \\ score_m \end{bmatrix} > 0 = \begin{bmatrix} 0.5 > 0 \\ -1 > 0 \\ 1.3 > 0 \\ \vdots \\ score_m > 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ \vdots \\ pred_m \end{bmatrix}$$

- Accuracy
  - $\frac{1}{m}\sum_{i=1}^{m}(pred_i == Y_{val_i})$

$$\begin{bmatrix} 1 \\ 0 \\ 1 \\ \vdots \\ pred_m == Y_{val_i} \end{bmatrix}$$

08 Applications of Naïve Bayes

Applications

- Sentiment analysis
- Author identification
- Information retrieval
- Word disambiguation

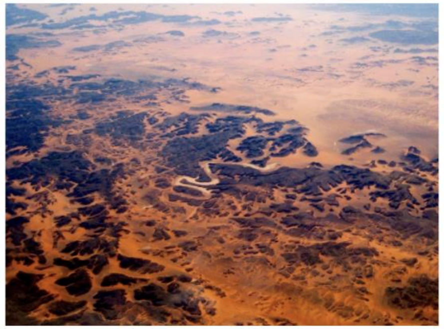
Naïve Bayes

- Simple, fast and robust

09 Naïve Bayes Assumptions

Independence

- Assume independence throughout



"It is sunny and hot in the Sahara desert."

"It's always cold and snowy in ___."

- Very difficult to guarantee → not true in NLP

09 Naïve Bayes Assumptions

Relative Frequencies in Corpus

- On Twitter
  - There are usually more positive tweets than negative ones
- Some “clean” datasets
  - Artificially balanced to have to the same amount of positive and negative tweets

- Affect the model

10 Error Analysis

Processing as a Source of Errors

- Removing punctuation

  Tweet: My beloved grandmother :(

  processed_tweet: [belov, grandmoth]

- Removing words

  Tweet: This is not good, because your attitude is not even close to being nice.

  processed_tweet: [good, attitude, close, nice]

10 Error Analysis

Processing as a Source of Errors

- Word order

Tweet: I am happy because I did not go.

Tweet: I am not happy because I did go.

10 Error Analysis

Adversarial attacks

- Sarcasm
- Irony
- Euphemisms

Tweet: This is a ridiculously powerful movie. The plot was gripping and I cried right through until the ending!

processed_tweet: [ridicul, power, movi, plot, grip, cry, end]