Week 2
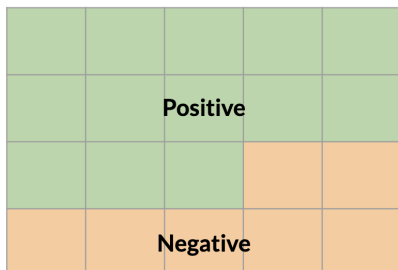
1. Probability and Bayes' Rule

You learned about probabilities and Bayes' rule.

Corpus of tweets
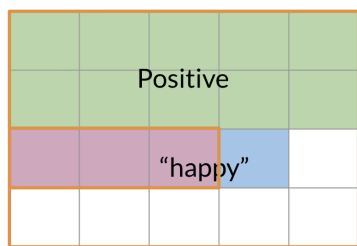

A → Positive tweet

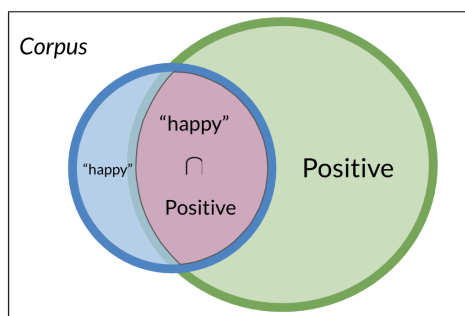$P(A) = N_{pos} / N = 13 / 20 = 0.65$

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

To calculate a probability of a certain event happening, you take the count of that specific event and you divide by the sum of all events. Furthermore, the sum of all probabilities has to equal 1.


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

To compute the probability of 2 events happening, like "happy" and "positive" in the picture above, you would be looking at the intersection, or overlap of events. In this case red and blue boxes overlap in 3 boxes. So the answer is $\frac{3}{20}$.

2. Bayes' Rule

Conditional probabilities help us reduce the sample search space. For example given a specific event already happened, i.e. we know the word is happy:
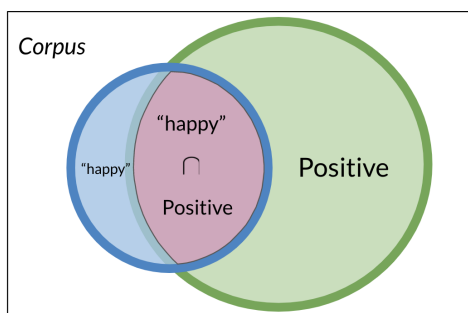
$$P(\text{Positive}|\text{"happy"}) =$$

$$\frac{P(\text{Positive} \cap \text{"happy"})}{P(\text{"happy"})}$$

Then you would only search in the blue circle above. The numerator will be the red part and the denominator will be the blue part. This leads us to conclude the following:

$$P(\text{Positive}|\text{"happy"}) = \frac{P(\text{Positive} \cap \text{"happy"})}{P(\text{"happy"})}$$

$$P(\text{"happy"}|\text{Positive}) = \frac{P(\text{"happy"} \cap \text{Positive})}{P(\text{Positive})}$$

Substituting the numerator in the right hand side of the first equation, you get the following:

$$P(\text{Positive}|\text{"happy"}) = P(\text{"happy"}|\text{Positive}) \times \frac{P(\text{Positive})}{P(\text{"happy"})}$$

Note that we multiplied by P(positive) to make sure we don't change anything. That concludes Bayes Rule which is defined as:

P(X|Y) = P(Y|X)P(X) / P(Y)

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

3. Naïve Bayes Introduction

- It is "naïve" because it assumes that all the features for classification are independent (which in reality may not be)

- Still works nicely for sentiment analysis

To build a classifier, we will first start by creating conditional probabilities given the following table:

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

- Note: The sum for "Neg" should also 13

This allows us compute the following table of probabilities:

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

Once you have the probabilities, you can compute the likelihood score as follows:

Tweet: I am happy today; I am learning.

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

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

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

- The product formulae is the Naïve Bayes inference condition rule for binary classification

- The 1.4 calculation is for the word "happy"

- Words that have occurred equally in Pos and Neg do not have any effect on even new (unseen) sentences, even if those words occur a lot of time in the new unseen sentence.

A score greater than 1 indicates that the class is positive, otherwise it is negative.

4. Laplacian Smoothing

- Laplacian smoothing to avoid $P(wi|class) = 0$ case

We usually compute the probability of a word given a class as follows:

$$P\left(\mathrm{w_i} \mid \text{ class }\right) = \frac{\text{freq}\left(\mathrm{w_i}, \text{ class }\right)}{\mathrm{N_{class}}} \quad \text{class } \in \{ \text{ Positive, Negative } \}$$

```
P(w_i | class ) = freq (w_i, class ) / N_class
class ∈ { Positive, Negative }
```

However, if a word does not appear in the training, then it automatically gets a probability of 0, to fix this we add smoothing as follows

$$P\left(\mathrm{w_i} \mid \text{ class}\right) = \frac{\text{freq}(\mathrm{w_i}, \text{ class })+1}{(\mathrm{N_{class}} +\mathrm{V})}$$

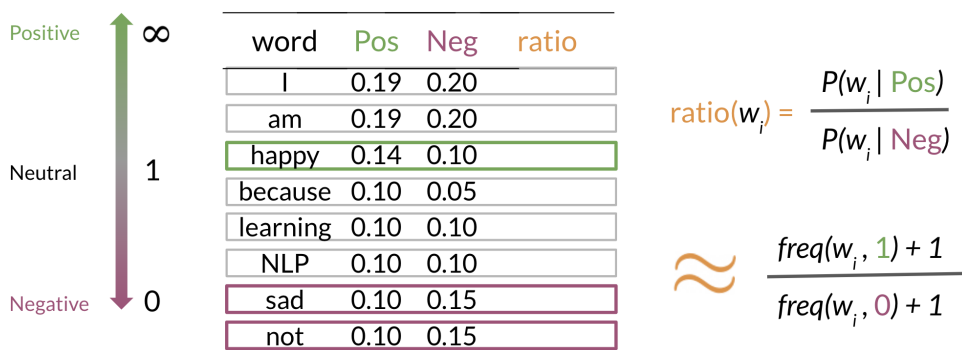```
P(w_i | class) = freq(w_i, class) + 1 / (N_class + V)
```

Note that we added a 1 in the numerator, and since there are V words to normalize, we add V in the denominator.

N_{class} : frequency of all words in class

V: number of unique words in vocabulary

## 5. Log Likelihood, Part 1

To compute the log likelihood, we need to get the ratios and use them to compute a score that will allow us to decide whether a tweet is positive or negative. The higher the ratio, the more positive the word is:

| 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 |

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

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

To do inference, you can compute the following:

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

```
P(pos) / P(neg) ∏ i=1 to m [ P(w_i | pos) / P(w_i | neg) ] > 1
```

As m gets larger, we can get numerical flow issues, so we introduce the log, which gives you the following equation:

$$\log \left(\frac{P(pos)}{P(neg)} \prod_{i=1}^{n} \frac{P(w_i \mid pos)}{P(w_i \mid neg)} \right) \Rightarrow \log \frac{P(pos)}{P(neg)} + \sum_{i=1}^{n} \log \frac{P(w_i \mid pos)}{P(w_i \mid neg)}$$

```
log [ P(pos) / P(neg) ∏ i=1 to n [P(w_i | pos) / P(w_i | neg) ] ] ⇒ log [ P(pos) /
```

The first component is called the log prior and the second component is the log likelihood. We further introduce λ as follows:

doc: I am happy because I am learning.

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

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

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

Having the λ dictionary will help a lot when doing inference.

5. Log Likelihood, Part 2

Once you computed the λ dictionary, it becomes straightforward to do inference:

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

As you can see above, since 3.3 > 0 , we will classify the document to be positive. If we got a negative number we would have classified it to the negative class.