Natural Language Processing

AIGC 5501

Simple Language Models

Instructor: Ritwick Dutta
Email: ritwick.dutta@humber.ca

Probabilistic Language Models

NLP is all about WORDS

Explore available **Language Models**

What are they?
Why they’re important
N-gram models
[complications]

Ref: https://medium.com/swlh/language-modelling-with-nltk-20eac7e70853#d68a

What are Language Models?

Statistical models that **assign probabilities to sequences of words**

Also called **generative models** as it explicitly model the process of generating languages

Think of completing these sentence fragments.

➢ Word prediction task

- Once upon a…
- Far above Alan’s …
- Let’s go outside and take a…

Objective: Assign a probability to a sequence of word tokens

The cat sat on the mat

w_1 w_2 w_3 … w_n

Corpus

$$\mathrm{p}\,(w_1, w_2, \ldots, w_M)\,,\ \ with\ \ w_m \in V$$

- The set V is a **discrete vocabularv**

$$V\ = \{aardvark,\ abacus, \ldots, zither\}$$

Why Language Models are important?

- In many applications, the goal is to produce word sequences as output:
  - Machine translation
  - Speech recognition
  - Summarization
  - Dialogue systems
- Subcomponent computes the probability of the output text.
- Intent: generate texts that are more fluent (i.e. have a higher probability)

Examples

Speech Recognition

p(Mayenne loves ice cream) >>
p(Mayenne loves I scream)

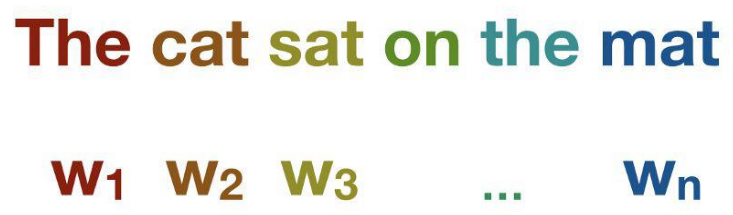
Machine Translation

El cafe negro me gusta mucho.

p(The coffee black me pleases much)
<< p(I love dark coffee)

Probabilistic Language Modeling

Goal: assign a probability to a sequence of words W (e.g. a sentence)



p(W) = p(w1,w2,w3,w4,w5, … ,wn)

Related task: compute the probability of the next word in a sequence, e.g.
p(w5 | w1, w2, w3, w4)

A model that computes either p(W) or p(wn| w1, w2,… wn-1) constitutes a language model.

Probability of a Word Sequence

p(w1 w2 … wn-1 wn)

Determine, e.g., **p(Mayenne ate my shoe)**

$$\frac{\text{count (“Mayenne ate my shoe”)}}{\text{total \# of utterances/sentences???}}$$

Estimate using a large corpus of text!

Unbiased Estimator, but…

- **Data intensive!!!!**
  - A small vocabulary for English has 10^5 ***word types***
  - Assume sentences of length 20…then 100,000^20 possible sentences.

- We'll need to rely on a **large corpus of text**

- **High variance**

  Even grammatical sentences will have probability zero if they have not occurred in the training data

Rely on the Chain Rule to Decompose

- $p(w_1, w_2, \ldots, w_{n-1}, w_n)$

$$\begin{aligned} P(w_1^n) &= P(w_1)P(w_2|w_1)P(w_3|w_1^2)\ldots P(w_n|w_1^{n-1}) \\ &= \prod_{k=1}^{n} P(w_k|w_1^{k-1}) \end{aligned}$$

p(Mayenne ate my shoe) = p(Mayenne) x p(ate|Mayenne) x p(my|Mayenne ate) x p(shoe|Mayenne ate my)

** calculated w.r.t. a particular corpus

N-gram Approximations

Markov assumption: probability of some future event (next word) depends only on a limited history of preceding events (previous words)

Use the previous N-1 words to predict the next word

1-gram: **unigram**

2-gram: **bigram**

3-gram: **trigram**

Unigram model ➔ $p(w_n \mid w_1^{n-1}) = p(w_n)$

Bigram model ➔ $p(w_n \mid w_1^{n-1}) = p(w_n \mid w_{n-1})$

Trigram model ➔ $p(w_n \mid w_1^{n-1}) = p(w_n \mid w_{n-2} w_{n-1})$

Example: p(lost|Not all those who wander are)

Approximate using a

- Unigram model → p(lost)
- Bigram model → p(lost|are)
- Trigram model → p(lost|wander are)

Bigram language model

So that's what we get for n = 2:

$$p(\mathbf{w}) = p(w_1)p(w_2|w_1)\dots p(w_k|w_{k-1})$$

Toy corpus:

This is the malt
That lay in the house that Jack built.

$$p(\text{this is the house}) = \overset{1/12}{p(\text{this})}\ \overset{1}{p(\text{is}|\ \text{this})}\ \overset{1}{p(\text{the}|\ \text{is})}\ \overset{1/2}{p(\text{house}|\ \text{the})}$$

“Accuracy” increases as N increases

Approach: Train various N-gram models and then use each generate random sentences.

<u>Sample Corpus</u>: Complete works of Shakespeare

- **Unigram**: Will rash been and by I the me loves gentle me not slavish page, the and hour; ill let
- **Bigram**: What means, sir. I confess she? Then all sorts, he is trim, captain.
- **Trigram**: Fly and will rid me these news of price. Therefore, the sadness of parting, as they say, ‘tis done.
- **Quadrigram**: They say all lovers swear more performance than they are wont to keep obliged faith unforfeited!

Counting Words in Corpora

- ❑ **Depends on tokenization**
  - Should we treat punctuation marks as words
  - Important for many NLP tasks
    - Grammar-checking, spelling error detection, author identification, part-of-speech tagging
  - Contractions
    - Isn’t vs. is n’t vs. isn ‘ t

- ❑ **Language-dependent**
  - Freundschaftsbezeigungen = demonstration of friendship

  - ❑ Decisions will have an effect on performance!

  - ❑ Typically, the goal is to reduce the vocabulary size.

  - ❑ And at the same time, we want to **preserve those distinctions/differences that matter** for the downstream applications.

- ❑ Depends also on **text normalization** – string transformations that remove distinctions irrelevant to downstream applications

Counting Words in Corpora

- ❑ **Capitalization**
  - Should They and they be treated as the same word?
    - For most statistical NLP applications, yes
    - Sometimes capitalization information is maintained as a feature
    - E.g. spelling error correction, part-of-speech tagging

- ❑ **Special fonts: Italics, bold**
  - Usually ignore

- ❑ **Inflected forms**
  - Should walks and walk be treated as the same word?
    - No…for most n-gram based systems

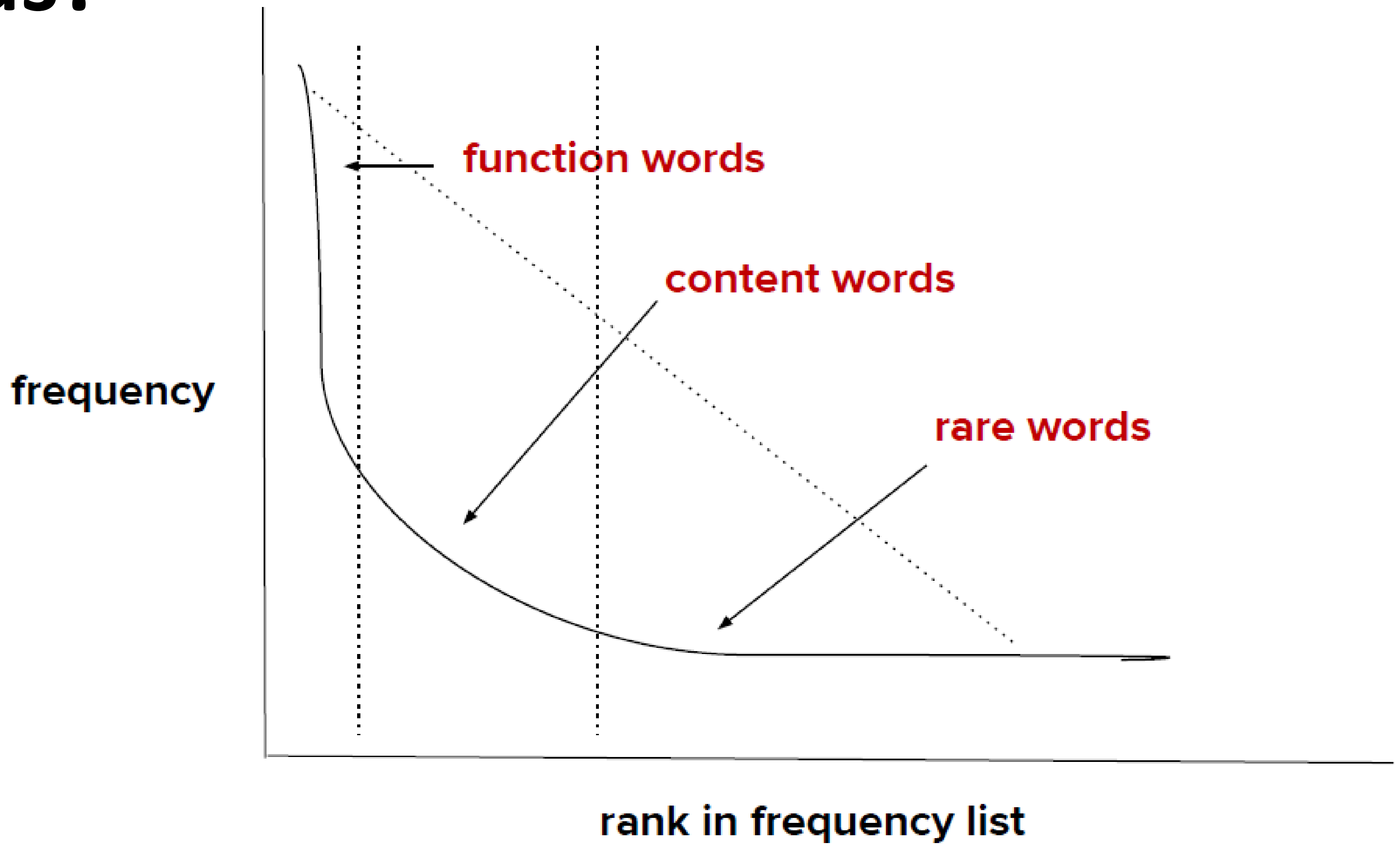
- ❑ **Spoken Language Corpora**
  - Utterances **don't usually have punctuation**, but they do have other phenomena that we might or might not want to treat as words, e.g. → <speech> I do uh main mainly business data processing
  - **Fragments**
  - **Filled pauses →** um and uh behave more like words, so most speech recognition systems treat them as such

Counting Words in Corpora

- **Need to distinguish the counting of**
  - word types
    - distinct words (vocabulary)
  - word tokens
    - the words in the “running” text (instances of the vocabulary items)

Example → All for one and one for all.
8 tokens (counting punctuation)
6 word types (assuming capitalized and uncapitalized versions of the same token are treated separately)
5 word types if capitalization is ignored

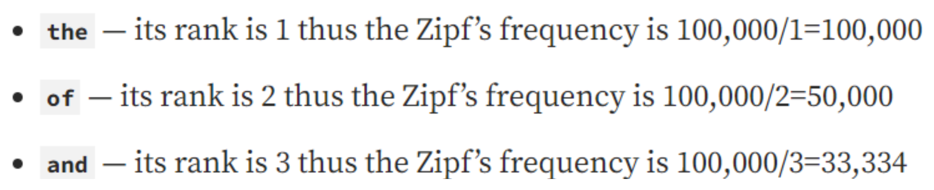
How are word types distributed within a corpus?



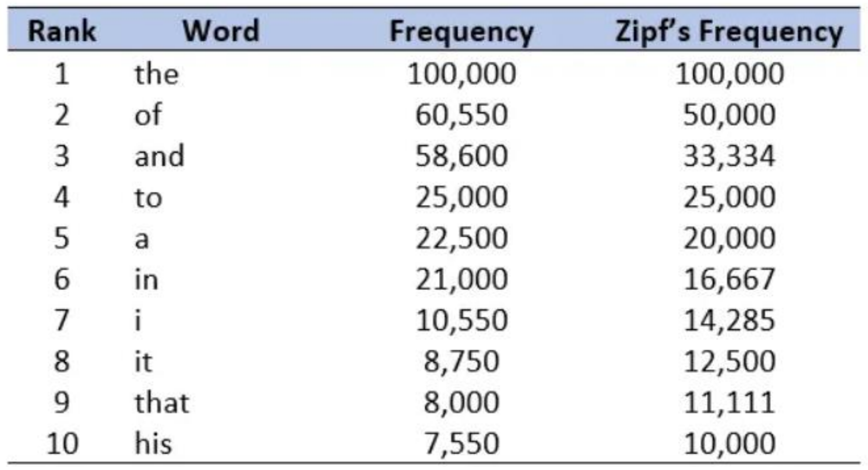
Some Statistical Properties of Text

Zipf's law reveals the connection between word frequencies and their ranks in a given document. The law states the most common word would be twice as much as the second word, three times the third word, four times the fourth word, and so on.

Example: Let's say we have an English document in a text file — for instance, a novel by Sherlock Holmes. Afterwards, **create a list of unique words** in the document and **count their occurrences** (frequency). Then, **sort this list descending** according to frequency.



- `the` — its rank is 1 thus the Zipf's frequency is 100,000/1=100,000
- `of` — its rank is 2 thus the Zipf's frequency is 100,000/2=50,000
- `and` — its rank is 3 thus the Zipf's frequency is 100,000/3=33,334



Rank	Word	Frequency	Zipf's Frequency
1	the	100,000	100,000
2	of	60,550	50,000
3	and	58,600	33,334
4	to	25,000	25,000
5	a	22,500	20,000
6	in	21,000	16,667
7	i	10,550	14,285
8	it	8,750	12,500
9	that	8,000	11,111
10	his	7,550	10,000

The frequency of occurrence f of a word type is inversely correlated with the rank r of that word type in an ordered frequency distribution

$$f \propto \frac{C}{r^{\alpha}}$$

C is a constant, which can be calculated using the Hurwitz Zeta function

r is the rank of a word type

α the value of the exponent characterizing the distribution. Typically, α is assumed to be approximately 1, but deviations have been reported in the literature

Zipf’s Law - Drawbacks

- ❑ For NLP, Zipf’s Law is useful as a rough description of the frequency distribution of words in human languages
  - Vast majority of word types in any corpus are RARE!!!!

- ❑ The most frequent words in one corpus may be rare words in another corpus
  - Example: “computer” in Association for Computing Machinery vs. National Geographic

- ❑ Each corpus has a different, fairly small working vocabulary

Text Processing / Classification Models

- Fake news classification
- Assigning subject categories, topics, or genres
- Authorship identification
- Age/gender identification
- Language identification
- Sentiment analysis
- …

Motivation: Problems with Probabilistic Models(Markov assumption)

Jack , thinking he was unwatched , very quickly helped <?> … to a huge slice of the birthday cake.

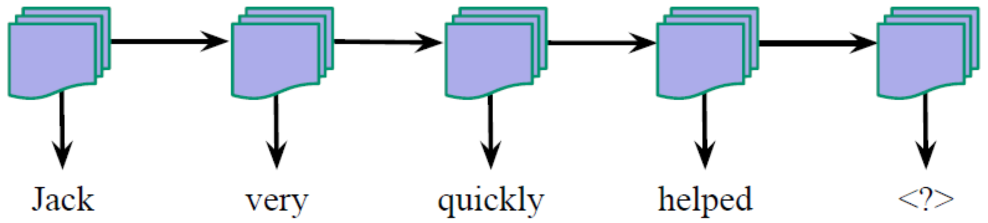
herself >> himself ?

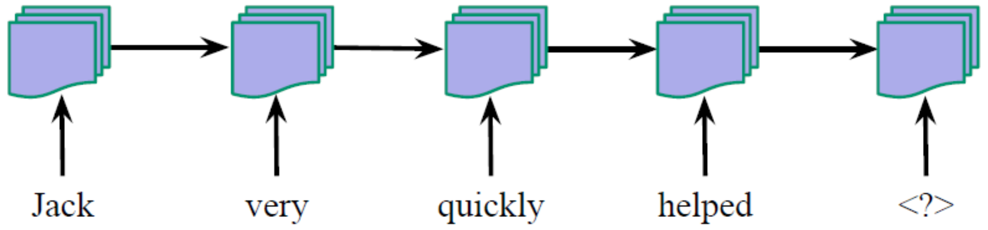
himself >> herself ?

Would need an 8- to 11-gram language model!!!!

Another option: word prediction as **classification**…

Using standard supervised machine learning techniques instead of n-gram-based LMs





Positive or negative review?

...zany characters and ***richly*** *applied satire, and some* ***great*** *plot twists*

It was ***pathetic****. The* ***worst*** *part about it was the boxing scenes...*

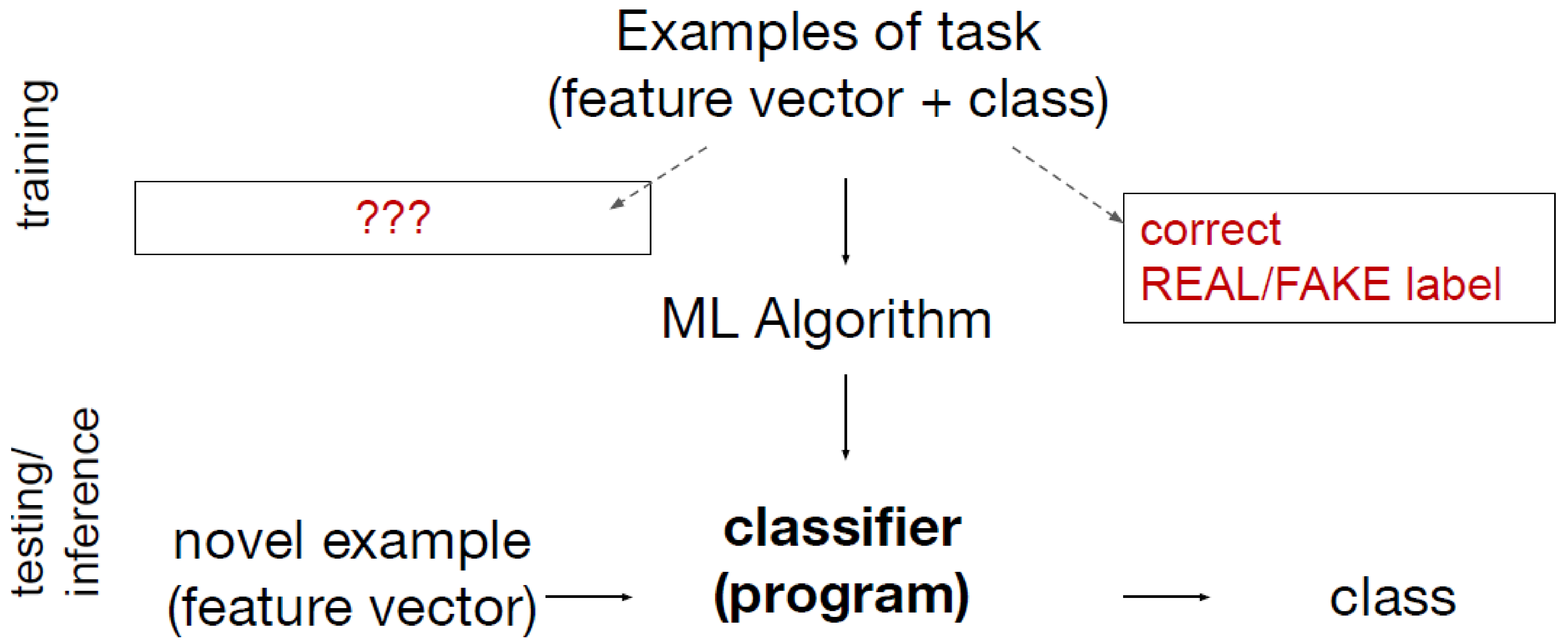
*...****awesome*** *caramel sauce and sweet toasty almonds. I* ***love*** *this place!*

*...****awful*** *pizza and* ***ridiculously*** *overpriced...*

Classification Methods: Hand-coded rules

- Rules based on combinations of words or other features
  - spam: black-list-address OR (“dollars” AND “you have been selected”)
- Accuracy can be **high**
  - If rules carefully refined by expert
- But building and maintaining these rules is expensive

Supervised ML framework



Bag of words (BoW)

Very good drama although it appeared to have a few blank areas leaving the viewers to fill in the action for themselves. I can imagine life being this way for someone who can neither read nor write. This film simply smacked of the real world: the wife who is suddenly the sole supporter, the live-in relatives and their quarrels, the troubled child who gets knocked up and then, typically, drops out of school, a jackass husband who takes the nest egg and buys beer with it. 2 thumbs up… very very very good movie.

although
Very
drama
good
blank
appeared
it
a
have
to
areas
few
viewers
the
leaving
fill
in

('the', 8),
(',', 5),
('very', 4),
('.', 4),
('who', 4),
('and', 3),
('good', 2),
('it', 2),
('to', 2),
('a', 2),
('for', 2),
('can', 2),
('this', 2),
('of', 2),
('drama', 1),
('although', 1),
('appeared', 1),
('have', 1),
('few', 1),
('blank', 1)
…..

Document / Text representation

- BOW (bag of words) representation is generally good for document-level sentiment classification

  - Expect to see enough words associated with true sentiment

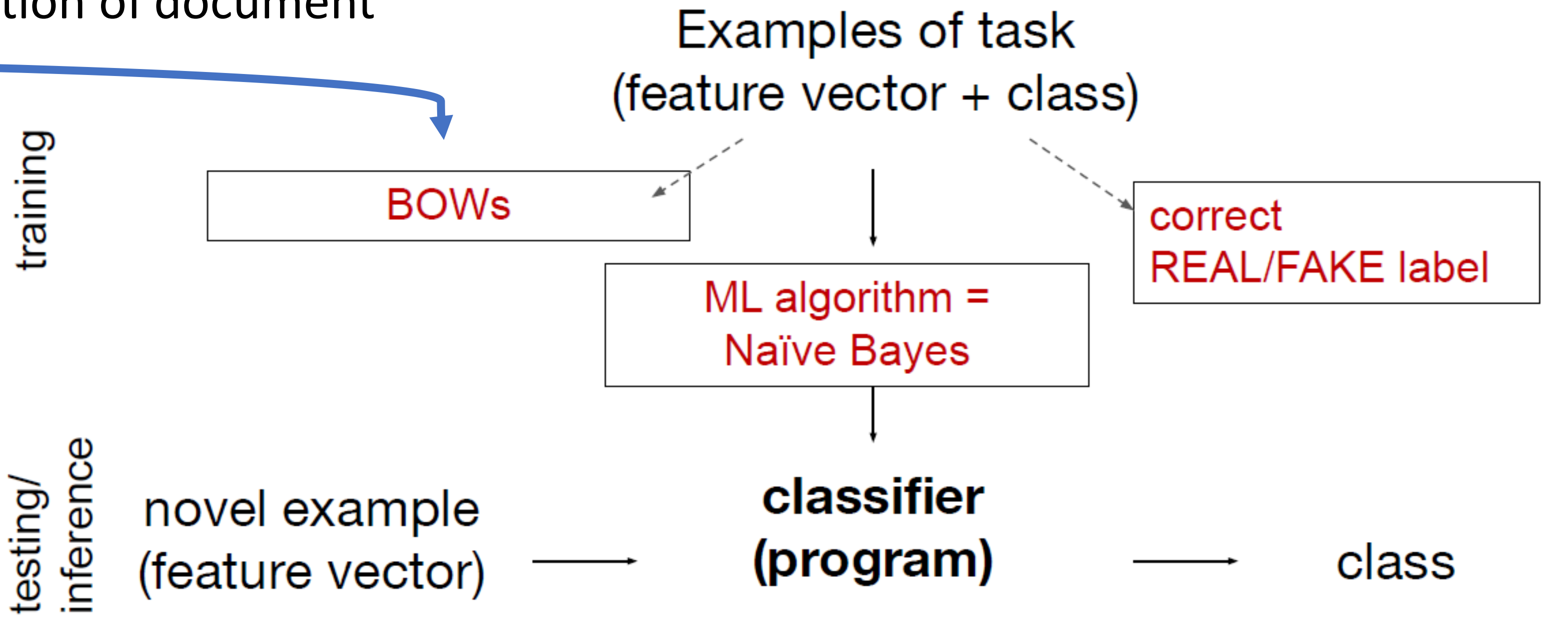
- For short "documents" (e.g. single sentence, social media posts), we will need additional features

Classification Methods: Supervised Machine Learning

- **Naive Bayes**
  - Logistic regression
  - Neural networks
  - k-Nearest Neighbors
  - …

Naive Bayes Intuition

- Simple ("naive") classification method based on Bayes rule
- Relies on very simple representation of document
  - **Bag of words**



Naive Bayes Classifier

Naive Bayes classifiers are a collection of classification algorithms based on **Bayes' Theorem**. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other.

The main two assumptions of Naive Bayes

Naive Bayes assumes that **each feature/variable** of the same class makes an:

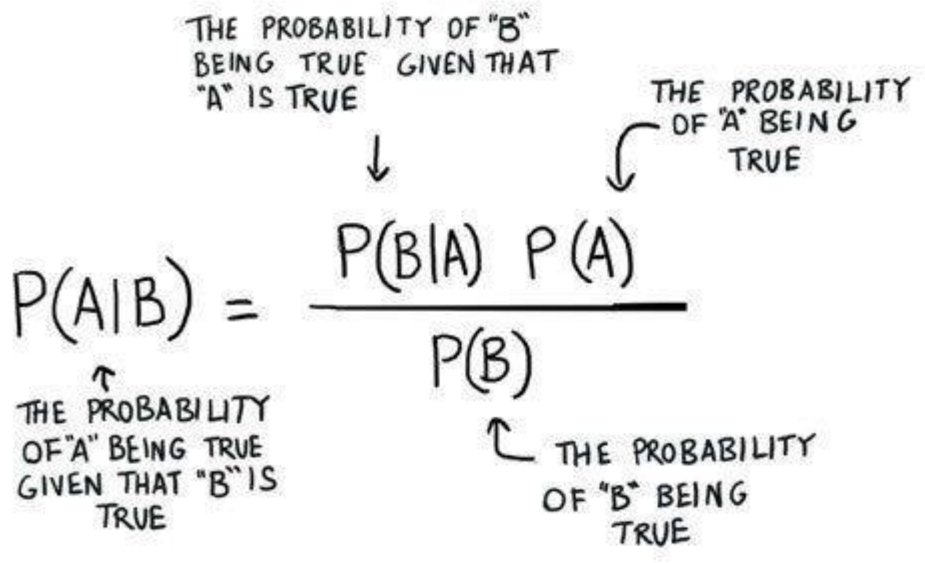
- **independent**
- **Equal contribution to the outcome.**

Side Note: The assumptions made by Naive Bayes are not generally correct in real-world situations.

In-fact, the independence assumption is often not met and this is why it is called "**Naive**" i.e. because it assumes something that might not be true.

The Bayes' Theorem

Bayes' Theorem finds the probability of an event occurring given the probability of another event that has already occurred.


THE PROBABILITY OF "B" BEING TRUE GIVEN THAT "A" IS TRUE
THE PROBABILITY OF "A" BEING TRUE
$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$
THE PROBABILITY OF "A" BEING TRUE GIVEN THAT "B" IS TRUE
THE PROBABILITY OF "B" BEING TRUE

where:

•**A** and **B** are called **events.**

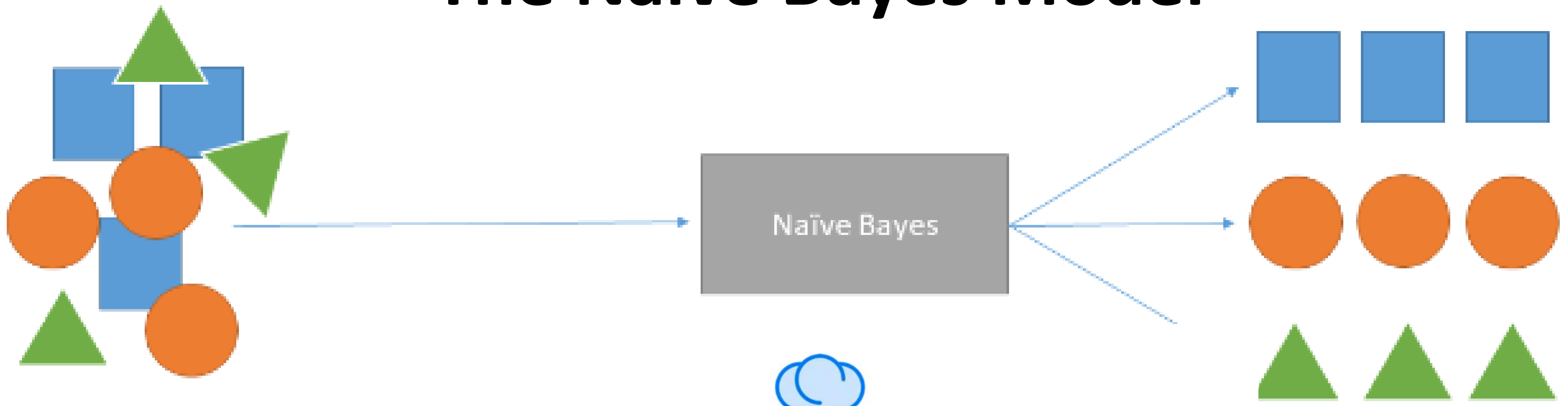
•P(A | B) is the probability of event A, given the event B is true (has occurred). Event B is also termed as **evidence**.

•P(A) is the **priori** of A (the prior independent probability, i.e. probability of event before evidence is seen).

•P(B | A) is the probability of B given event A, i.e. probability of event B after evidence A is seen.

$$\begin{aligned} A, B &= \text{events} \\ P(A \mid B) &= \text{probability of A given B is true} \\ P(B \mid A) &= \text{probability of } B \text{ given } A \text{ is true} \\ P(A), P(B) &= \text{the independent probabilities of A and B} \end{aligned}$$

The Naive Bayes Model



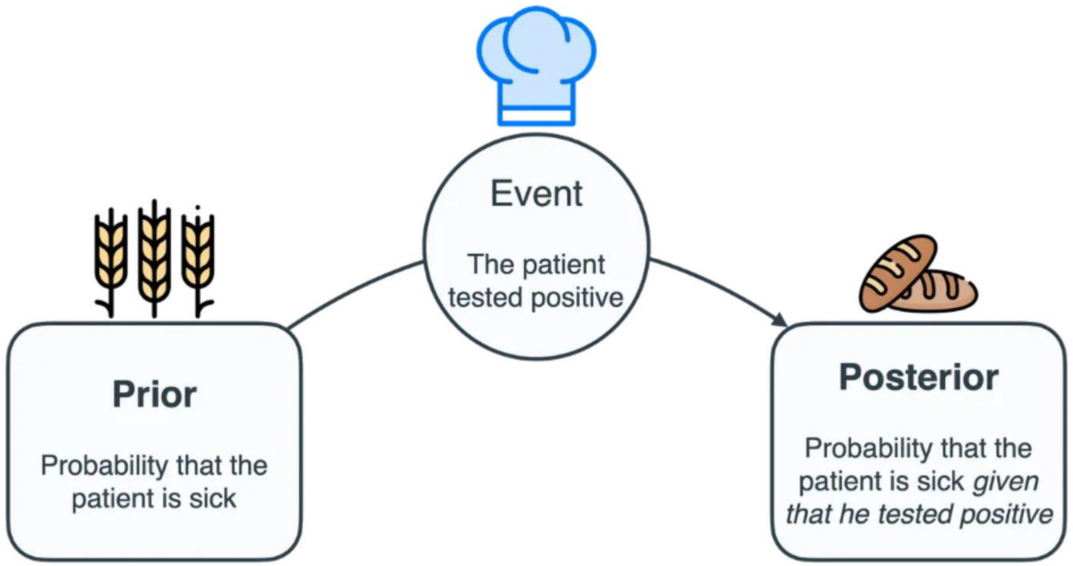
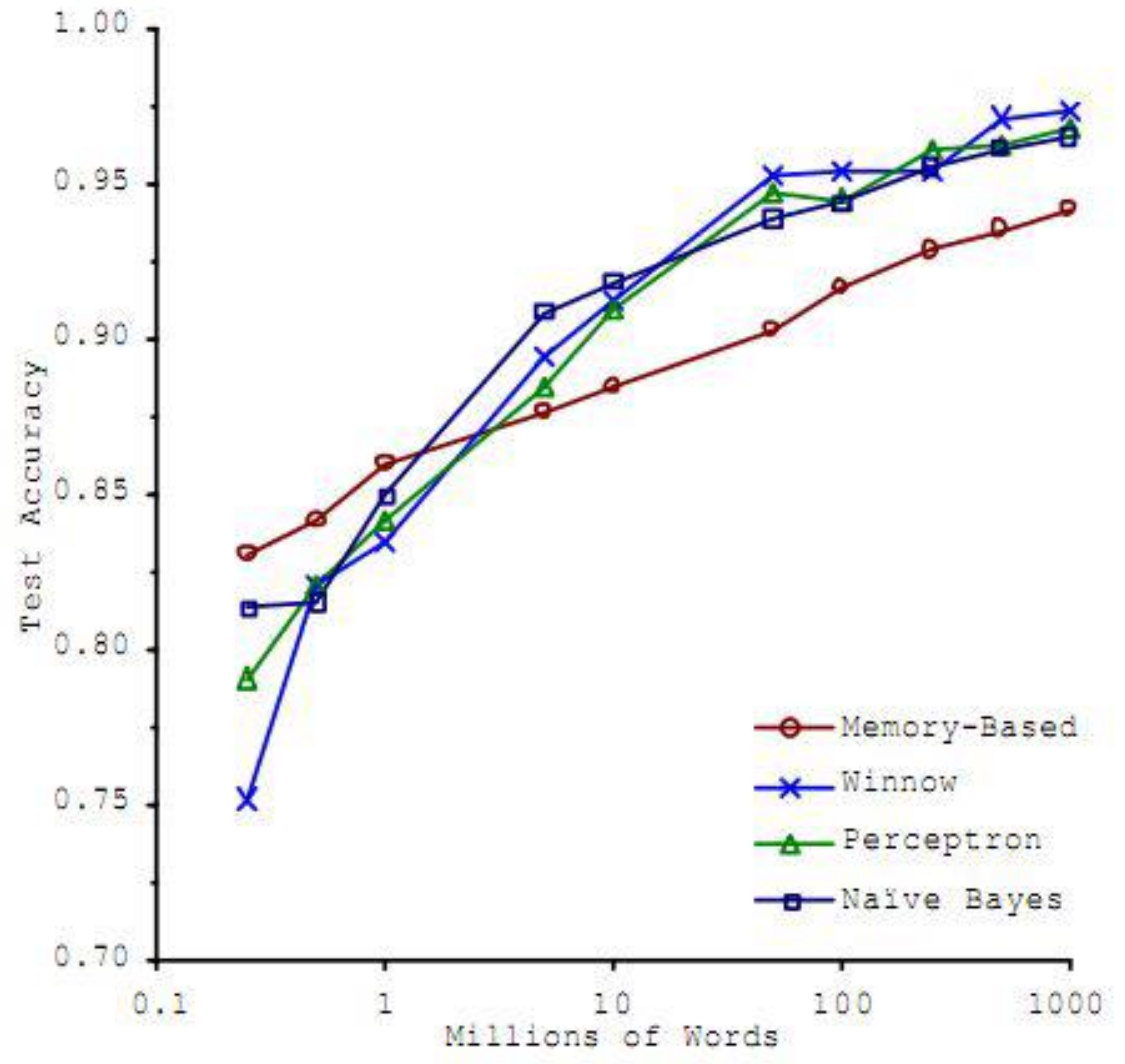

Event
The patient tested positive
Prior
Probability that the patient is sick
Posterior
Probability that the patient is sick *given that he tested positive*

Figure 3. The prior, the event, and the posterior. The prior is the 'raw' probability, namely, the probability we calculate when we know little. The event is the information that we obtain which helps us refine our calculation of the probability. The posterior is the 'cooked' probability, or the much more accurate probability that we calculate when we have more information.

Watch (5.57 mins): https://www.youtube.com/watch?v=okdxDL3-QyU

Which classifier to choose?

Accuracy as a function of data size



With enough data

– Classifier may not matter

Brill and Banko on spelling correction

Lab -2

Exercise 1: https://towardsdatascience.com/from-dataframe-to-n-grams-e34e29df3460

Exercise 2:

a) Find a new text dataset
b) Convert it into csv format
c) Redo the same exercise

Exercise 3: https://towardsdatascience.com/text-classification-using-naive-bayes-theory-a-working-example-2ef4b7eb7d5a