Introduction to Text Classification Using Python

Mohammad Fazeli

November 21, 2019

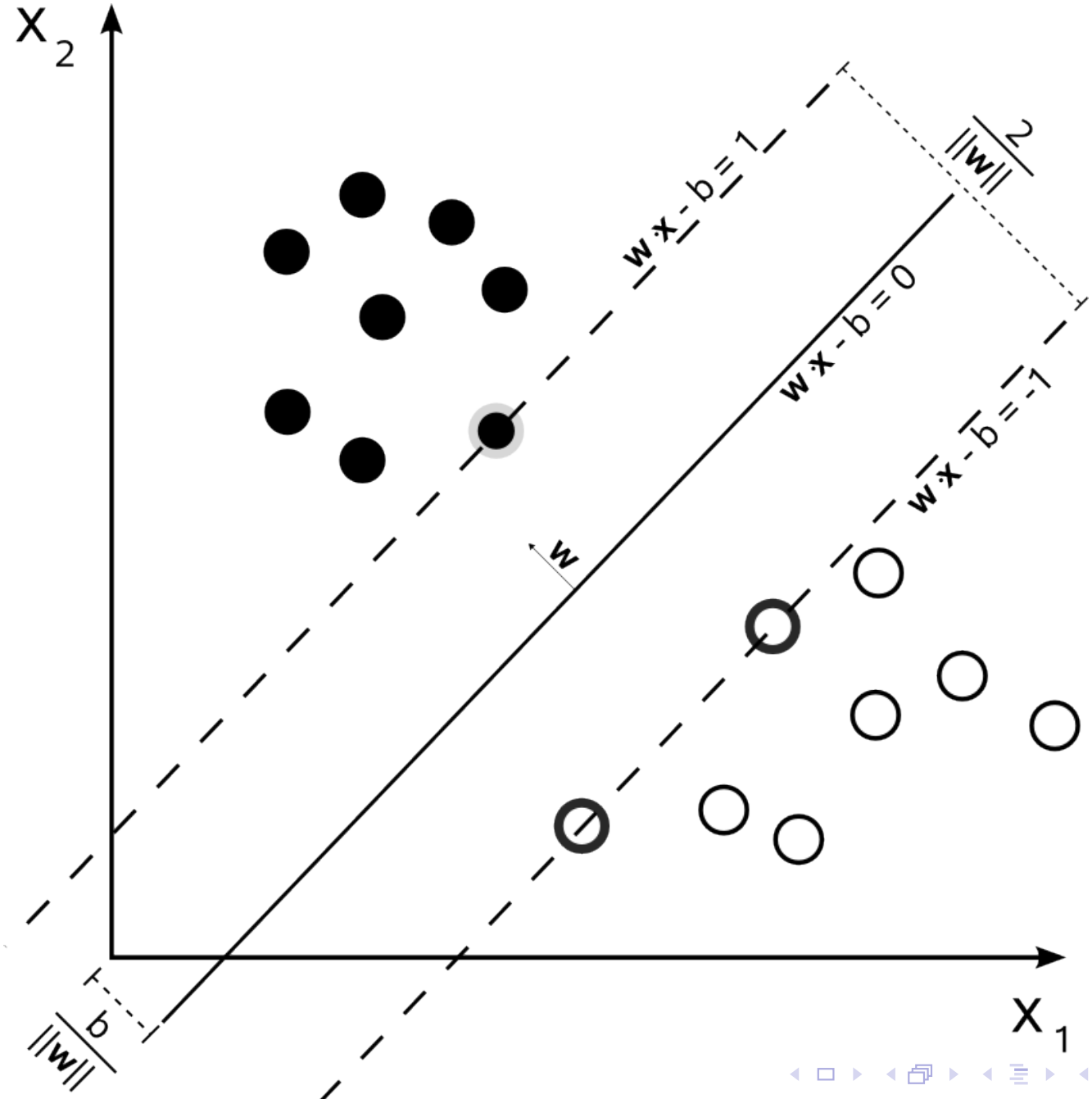
Why Classify Text?

- There are lots of it.
- Requires a lot of efforts.

What is Machine Learning?

- There are different types of ML.
- A good intuitive definition would be:
  - Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead. (wikipedia)

What is Machine Learning?



Why Use Machine Learning?

- The stuff is hard to program.
- Like spam detection(older SysAdmins problem now solved).

What We are Going to Do?

- Use machine learning to split texts to classes.
-

Stuff You Know

- Google Translate
- Chatbots like the following:

Hipmunk

Whole Foods Market

Madison Reed

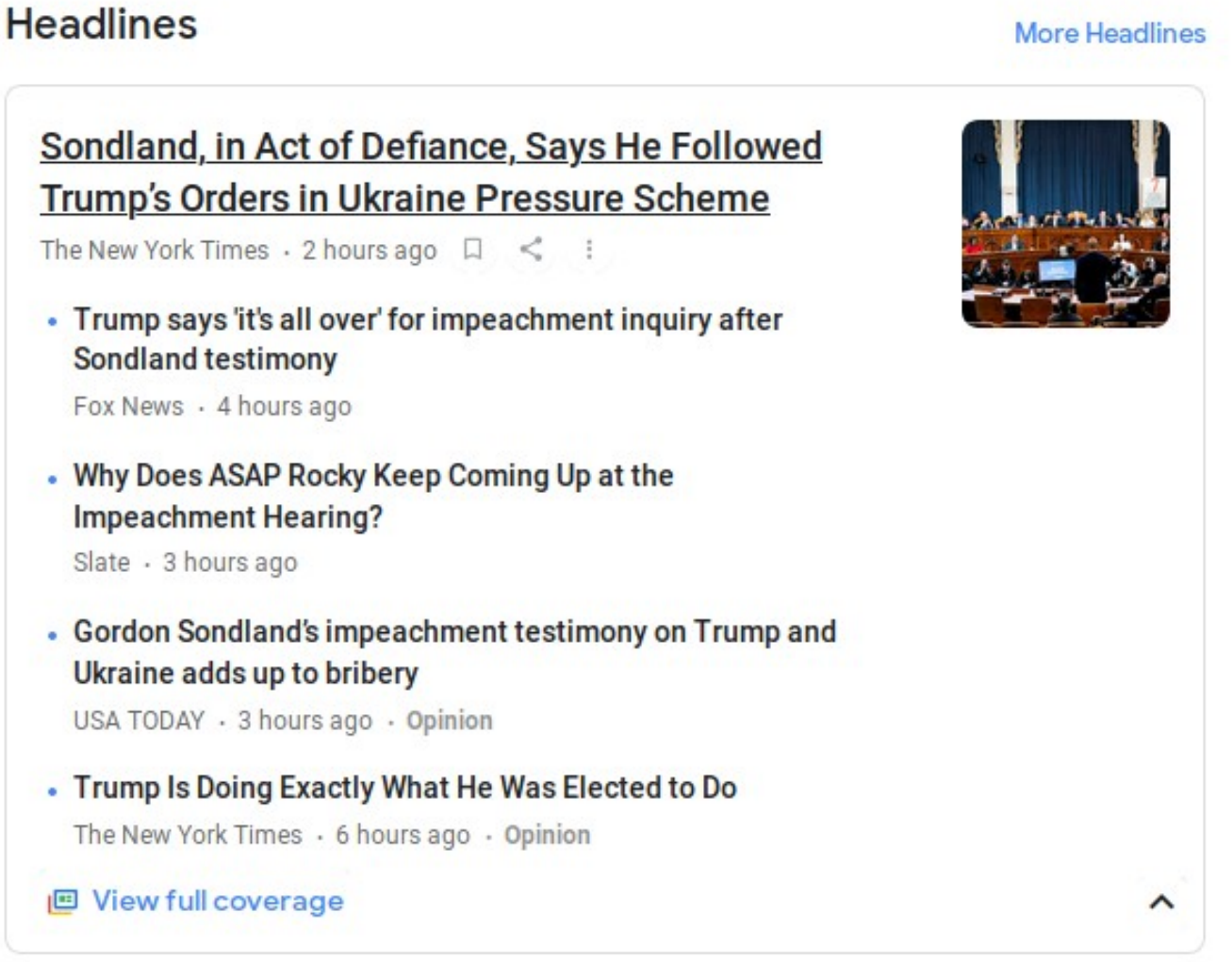
Duolingo

Amtrak

Domino's Pizza

Stuff You Know

- News Aggregators



Stuff You Know

- Spam filters

Stuff You Know

- Sentiment Analysis

Sentiment	Tweet mention
Positive	Maybe I'm mad but I'm now the proud owner of a potentially #bendy #iPhone6, it's so much bigger than the #4s
	Finally got to see an iPhone 6 today. Not revolutionary at all but it's absolutely gorgeous. (And I want one). #iPhone6
Negative	I'm not sure I want it. It's too big to fit in my back pocket! lol #iphone6
	I'm really disappointed with the #iPhone6. It took them 2 years to change the screen & size. Let down.

Lets Implement One for Persian

- The data-set is Hamshahri.
- Hamshahri with labels as categories.
- Target is demonstration for Persian.
- The million dollar question is:
- "How to represent text as points in R^n"

Let's formalize

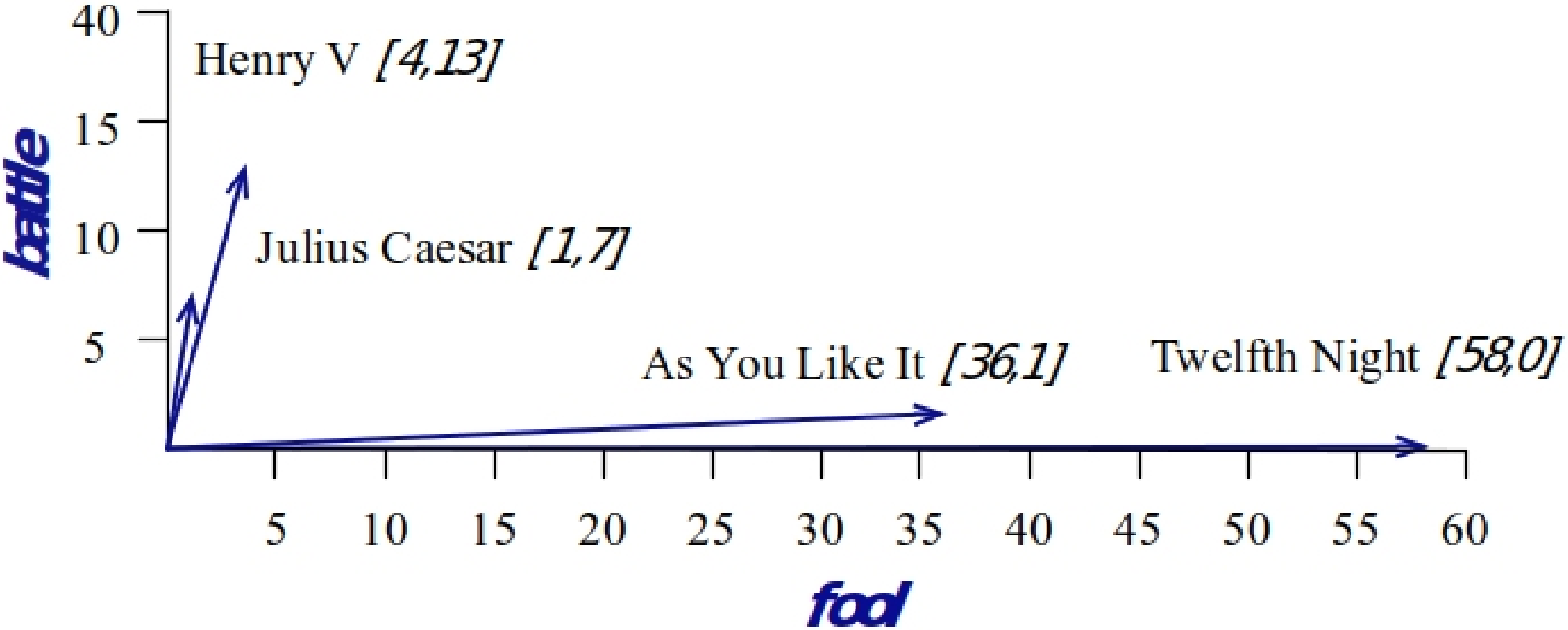
- Text as Vector in $R^{|V|}$
- Called an "embedding" because it's embedded into a space
- The standard way to represent meaning in NLP
- TF-IDF
  - A common baseline model
  - But works very good
  - Sparse vectors
  - Documents are represented by a simple function of the counts of nearby words

Term-document matrix

Each document is represented by a vector of words

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	1	8	15
soldier	2	2	12	36
fool	37	58	1	5
clown	5	117	0	0

Visualizing document vectors



Vectors are the basis of information retrieval

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	1	8	15
soldier	2	2	12	36
fool	37	58	1	5
clown	5	117	0	0

- Vectors are similar for the two comedies
- Different than the history
- Comedies have more fools and wit and fewer battles.

Reminders from linear algebra

$$\text{dot-product } v.w = \sum_{i=1}^{N} v_i \times w_i = v_1 \times w_1 + v_2 \times w_2 + \cdots + v_N \times w_N \tag{1}$$

$$\text{vector length } |v| = \sqrt{\sum_{i=1}^{N} v_i^2} \tag{2}$$

Cosine Distance

$$\text{cosine distance } d(v, w) = \frac{v.w}{|v| \times |w|} \tag{3}$$

-1: vectors point in opposite directions

+1: vectors point in same directions

0: vectors are orthogonal

Frequency is non-negative, so cosine range 0-1

But raw frequency is a bad representation

- Frequency is clearly useful; if sugar appears a lot near apricot, that's useful information
- But overly frequent words like the, it, or they are not very informative about the context
- Need a function that resolves this frequency paradox!

TF-IDF: combine two factors

- tf: term frequency. frequency count (usually log-transformed):

$$\mathrm{tf}_{t,d} = \begin{cases} 1 + \log_{10} \mathrm{count}(t,d) & \text{if } \mathrm{count}(t,d) > 0 \\ 0 & \text{otherwise} \end{cases}$$

- Idf: inverse document frequency: tf-

$$\mathrm{idf}_t = \log\left(\frac{N}{\mathrm{df}_t}\right)$$

Words like "the" or "good" have very low idf

- tf-idf value for word t in document d: $w_{t,d} = \mathrm{tf}_{t,d} \times \mathrm{idf}_t$