

94-775 Unstructured Data Analytics for Policy

Lecture 3: Wrap up basic text analysis,
co-occurrence analysis

Slides by George H. Chen

Today

1. Revisit last lecture's demo (basic text analysis) but using arrays/vectors instead of dictionaries/Counter objects

When we get to neural nets: array/vector representations are much more commonly used in practice than dictionaries/Counter objects

2. Talk about co-occurrence analysis

Array/Vector Representations

- To familiarize yourself with arrays/vectors, we'll extensively be using NumPy in this course

When we get to neural nets: we will use PyTorch, which is similar to coding using NumPy (but more complicated than NumPy!)

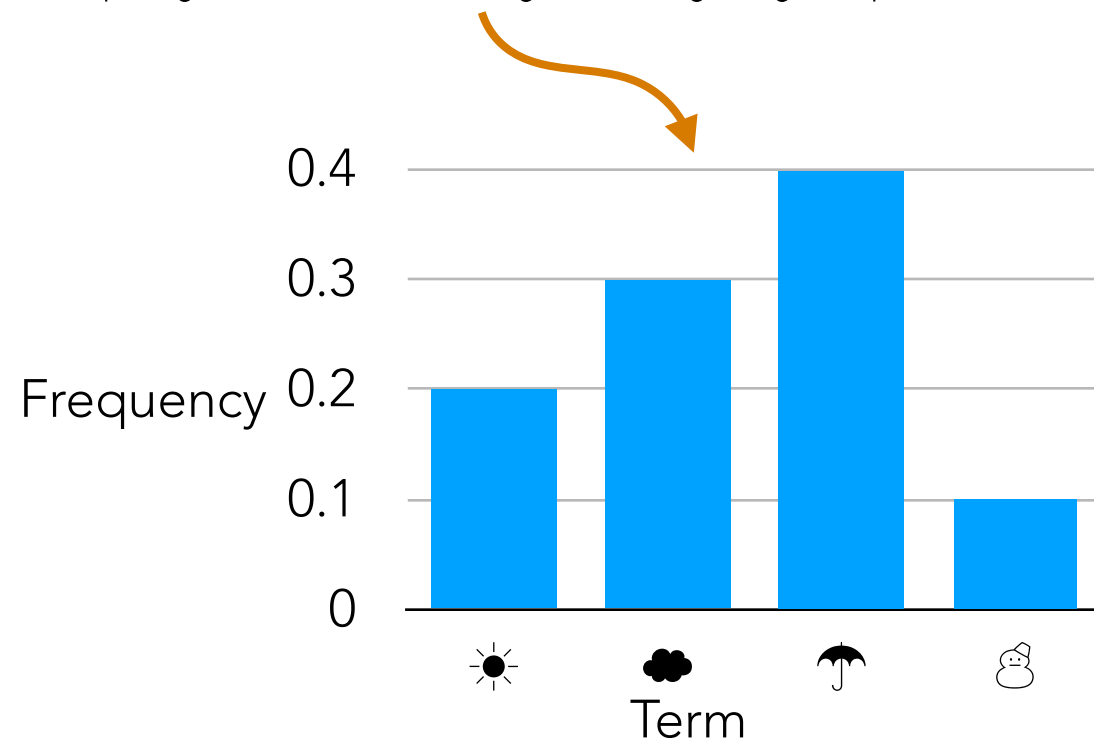
- We will *not* assume or expect you to know Pandas in this class
 - Making sure that you're fluent in NumPy is going to be much more beneficial for understanding PyTorch code than knowing Pandas
 - Whenever Pandas shows up in 94-775, it will be self-contained (so that you do not actually need to know Pandas)
 - Yes, you can use Pandas in HW if you want but we encourage you to become fluent in NumPy if you want to understand PyTorch code (that shows up in lectures at the end of the semester)

Pandas is great for working with 1D and 2D arrays but we'll regularly be working with 3D, 4D, 5D, etc arrays in PyTorch, which Pandas isn't great at

Recap: Basic Text Analysis

We represent each document as a histogram/probability distribution

Document: ☀️☔️☁️☁️☁️☔️☷☔️☔️☀️



☀️ ☁️ ☔️ ☷
→ [0.2 0.3 0.4 0.1]

This is a point in
4-dimensional space, $\mathbb{R}^4$

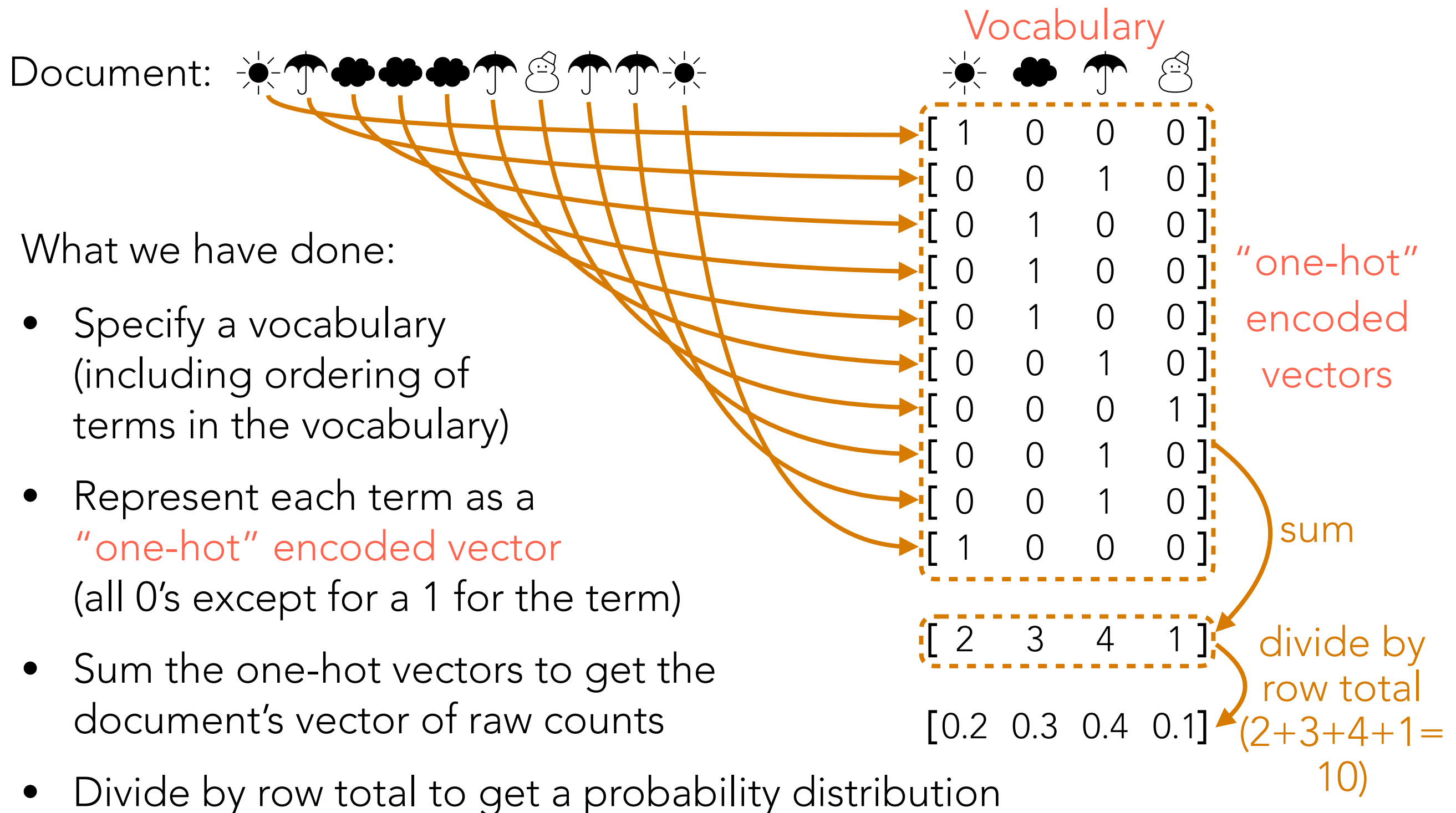
We refer to a vector representation of
the document as a feature vector

dimensions = number of terms

If there are lots of terms $\Rightarrow$ feature vectors are high-dimensional

Another Viewpoint

We represent each document as a histogram/probability distribution



Vector Version of Previous Code Demo

- We first build the vocabulary of the Wikipedia article (by finding all unique tokens present in the article)

Index: 0 1 2 3 1 4 5 6 7
The opioid epidemic or opioid crisis is the rapid
 8 9 6 10 11 12 13 14 15
 increase in the use of prescription and non-
 12 1 16 9 6 17 18
prescription opioid drugs in the United States ...

```
word_to_idx = {'The': 0,  
               'opioid': 1,  
               'epidemic': 2, ...}
```

maps each word to its index; this is *not* a term frequency table

```
vocab = ['The', 'opioid',  
         'epidemic', ...]
```

listed in the same order as the word indices

Index	Word
0	The
1	opioid
2	epidemic
⋮	⋮

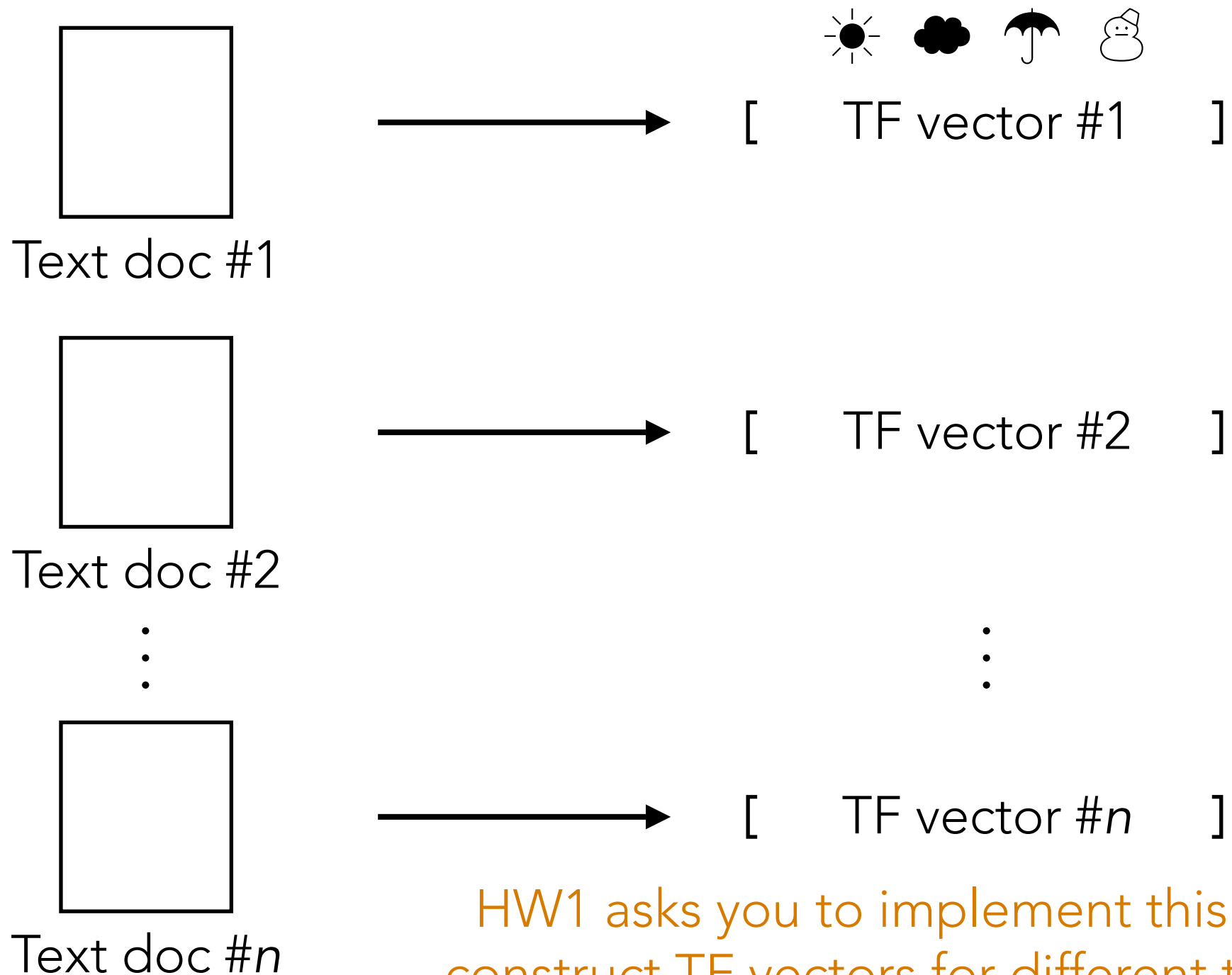
- We can then represent every word as a one-hot encoded vector
- We sum the one-hot encoded vectors to get a raw counts vector

Vector Version of Previous Code Demo

Demo

Multiple Documents

Choose a common vocabulary to
use across all documents



HW1 asks you to implement this idea to
construct TF vectors for different text docs
corresponding to job descriptions

Finding Possibly Related Entities with Co-occurrence Analysis

How to automatically figure out that "Lisa Su" and "AMD" are related?

TIME

 SIGN UP FOR OUR IDEAS NEWSLETTER POV

BUSINESS • THE LEADERSHIP BRIEF

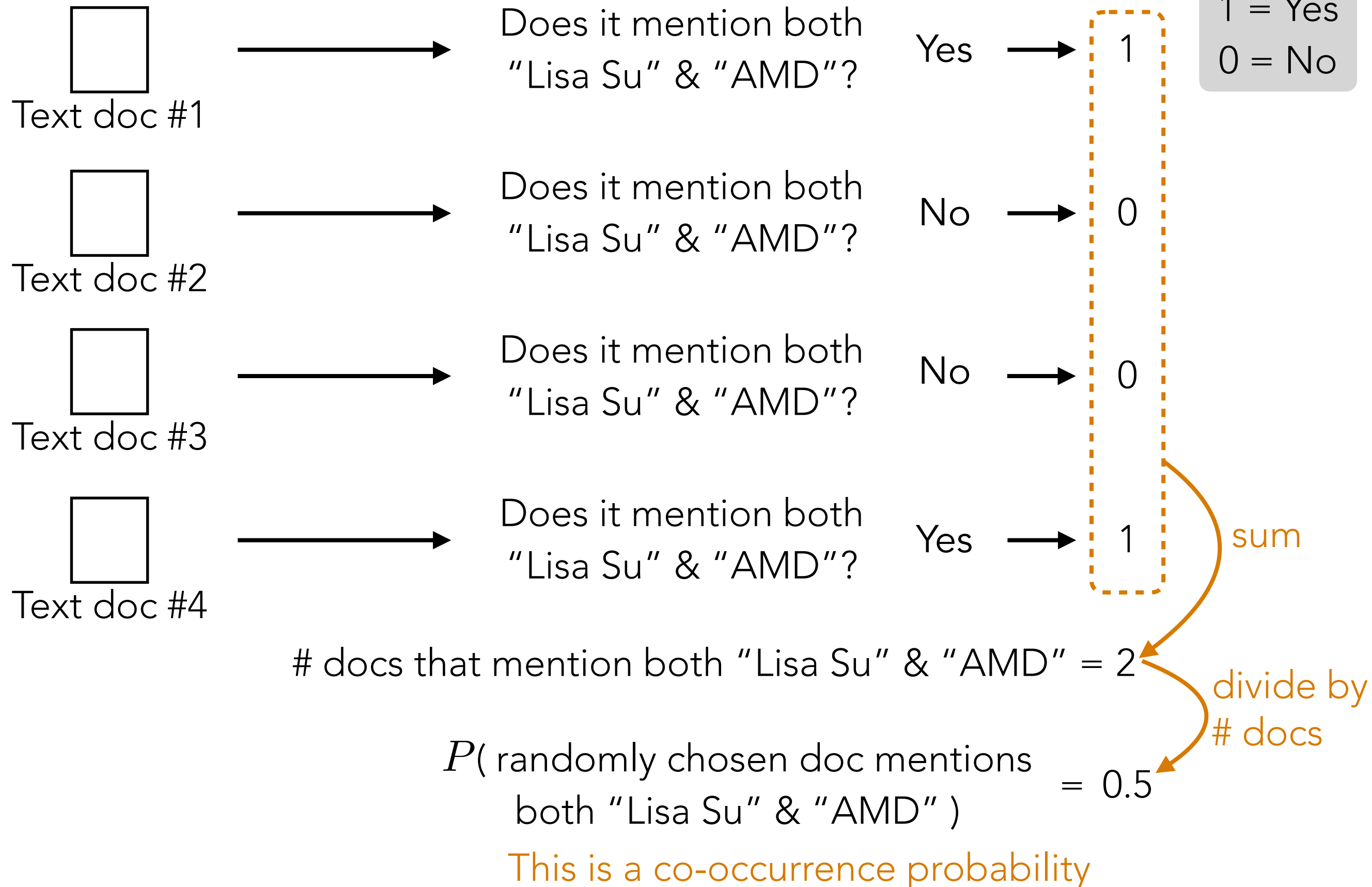
Lisa Su on AMD's Strategy for Growth and the Future of AI

11 MINUTE READ

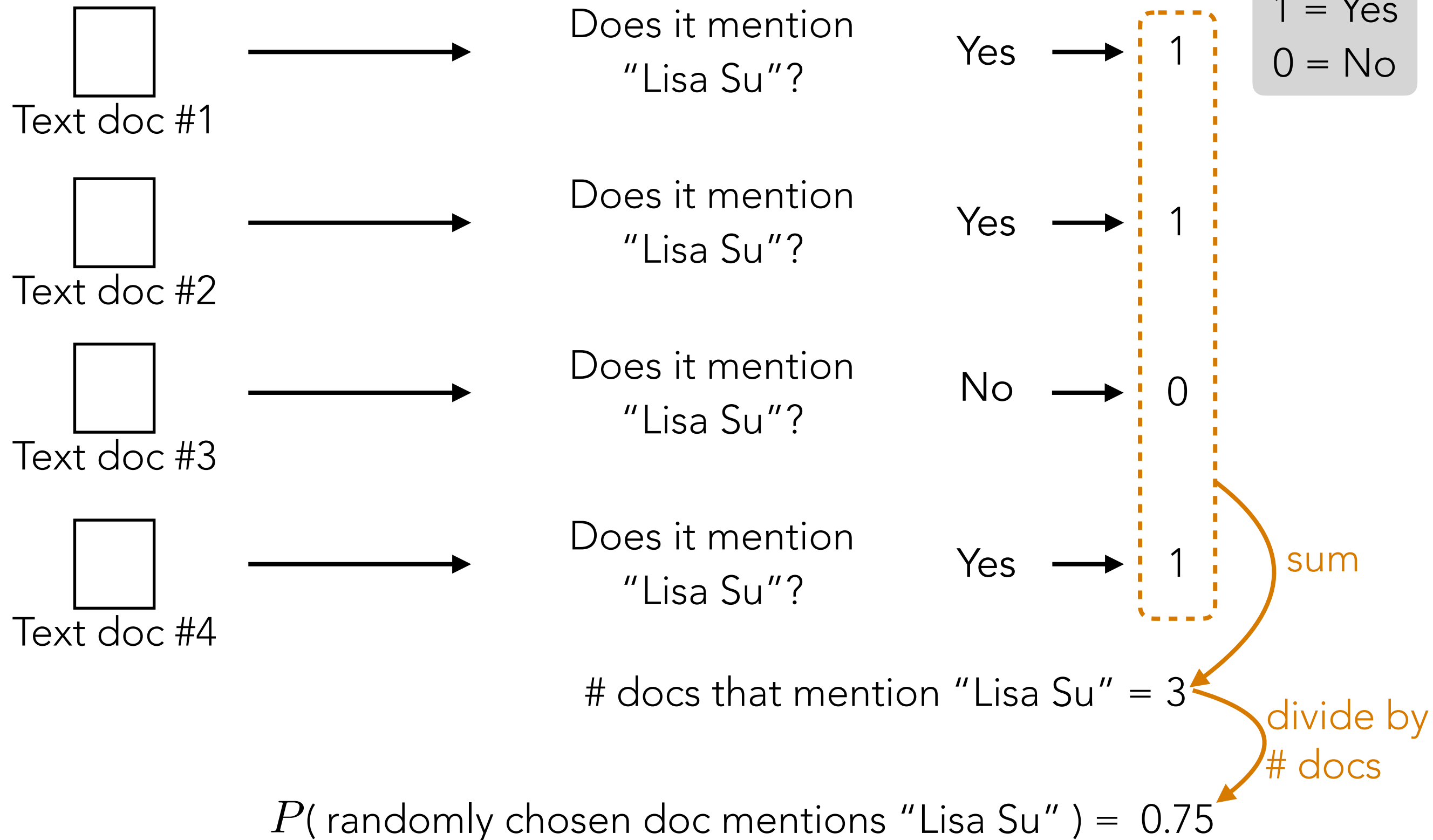


<https://time.com/7026241/lisa-su-amd-ceo-interview/>

Suppose that we have a dataset of 4 text docs (each is a news article)



Suppose that we have a dataset of 4 text docs (each is a news article)



This is not a co-occurrence probability; it is a marginal probability

Notation

$$P(\text{Lisa Su}, \text{AMD}) = P(\text{randomly chosen doc mentions both "Lisa Su" \& "AMD" })$$

$$P(A, B) = P(\text{randomly chosen doc mentions both } A \& B)$$

$$P(\text{Lisa Su}) = P(\text{randomly chosen doc mentions "Lisa Su" })$$

$$P(A) = P(\text{randomly chosen doc mentions } A)$$

Running Toy Example

Suppose that we keep track of 3 named entities that are people:

Lisa Su, Mark Zuckerberg, Sundar Pichai

Suppose that we keep track of 3 named entities that are companies:

Alphabet, AMD, Meta

Exhaustive list of every
person/company pair:

Lisa Su, Alphabet
Lisa Su, AMD
Lisa Su, Meta
Mark Zuckerberg, Alphabet
Mark Zuckerberg, AMD
Mark Zuckerberg, Meta
Sundar Pichai, Alphabet
Sundar Pichai, AMD
Sundar Pichai, Meta

→ Goal: rank these pairs
from “most interesting”
to “least interesting”

In practice: often
want to focus on most
interesting pairs

Need a numerical score
for “interesting”-ness

A Simple First Thing to Try

Just use co-occurrence probabilities!

Lisa Su, Alphabet

Lisa Su, AMD

Lisa Su, Meta

Mark Zuckerberg, Alphabet

Mark Zuckerberg, AMD

Mark Zuckerberg, Meta

Sundar Pichai, Alphabet

Sundar Pichai, AMD

Sundar Pichai, Meta



Compute co-occurrence probabilities
(one probability per pair)
& sort from largest to smallest

What could go wrong?!?

Co-occurrence Analysis

Demo

A Simple Fix: Re-weight Co-occurrence Probability

$$\frac{P(\text{Mark Zuckerberg}, \text{Meta})}{P(\text{Mark Zuckerberg})P(\text{Meta})}$$

Denominator larger $\Rightarrow$ overall ratio smaller

Idea: instead of using co-occurrence probability, use the above ratio

A Simple Fix: Re-weight Co-occurrence Probability

Probability of A (person) and B (company) co-occurring

$$\frac{P(A, B)}{P(A)P(B)}$$

if equal to 1

$\Rightarrow A, B$ are independent

Probability of A and B co-occurring *if they were independent*

Denominator larger $\Rightarrow$ overall ratio smaller

Idea: instead of using co-occurrence probability, use the above ratio

A Simple Fix: Re-weight Co-occurrence Probability

Probability of A (person) and B (company) co-occurring

$$\frac{P(A, B)}{P(A)P(B)}$$

if equal to 1

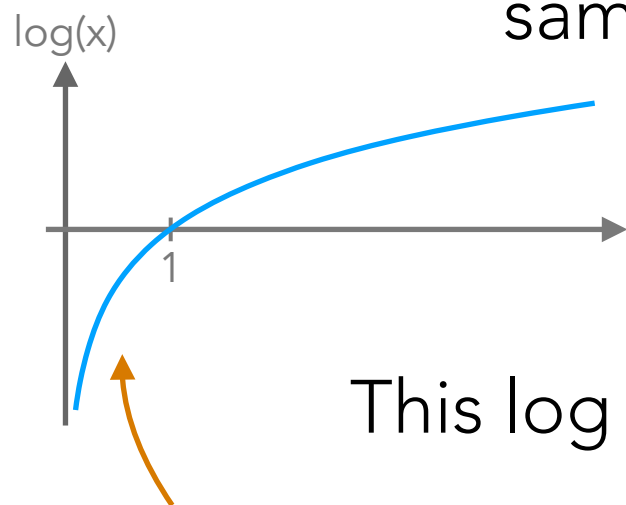
$\Rightarrow A, B$ are independent

Probability of A and B co-occurring *if they were independent*

Denominator larger $\Rightarrow$ overall ratio smaller

Idea: instead of using co-occurrence probability, use the above ratio

Ranking of person/company pairs using the above ratio remains the same if we instead use the *log* of the above ratio

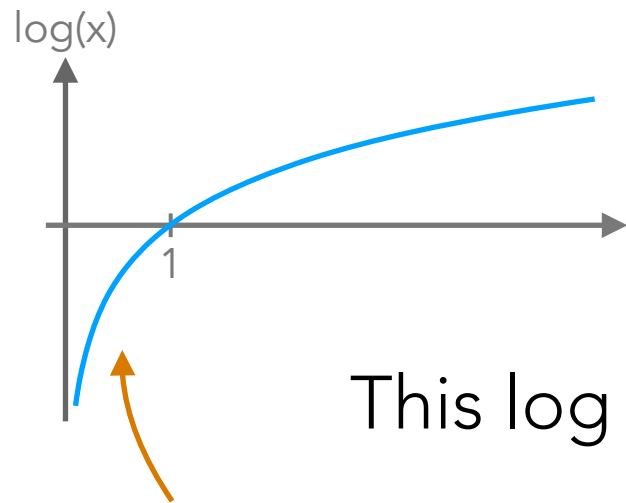


$$\text{PMI}(A, B) = \log \frac{P(A, B)}{P(A)P(B)}$$

This log ratio is called **pointwise mutual information (PMI)**

Reminder: this is what log looks like!

Pointwise Mutual Information (PMI)



$$\text{PMI}(A, B) = \log \frac{P(A, B)}{P(A)P(B)}$$

This log ratio is called **pointwise mutual information (PMI)**

Reminder: this is what log looks like!

What base should we use?

$$\log_2 \frac{P(A, B)}{P(A)P(B)} = \left(\frac{1}{\log 2} \right) \log \frac{P(A, B)}{P(A)P(B)}$$

$$\log_{10} \frac{P(A, B)}{P(A)P(B)} = \left(\frac{1}{\log 10} \right) \log \frac{P(A, B)}{P(A)P(B)}$$

Does not depend on A or B

If we use the same base to do all calculations,
which base we use does not affect the ranking of person-company pairs!

Co-occurrence Analysis

[Back to the demo](#)