Natural Language Processing

Topic Models

Dirk Hovy

dirk.hovy@unibocconi.it

@dirk_hovy

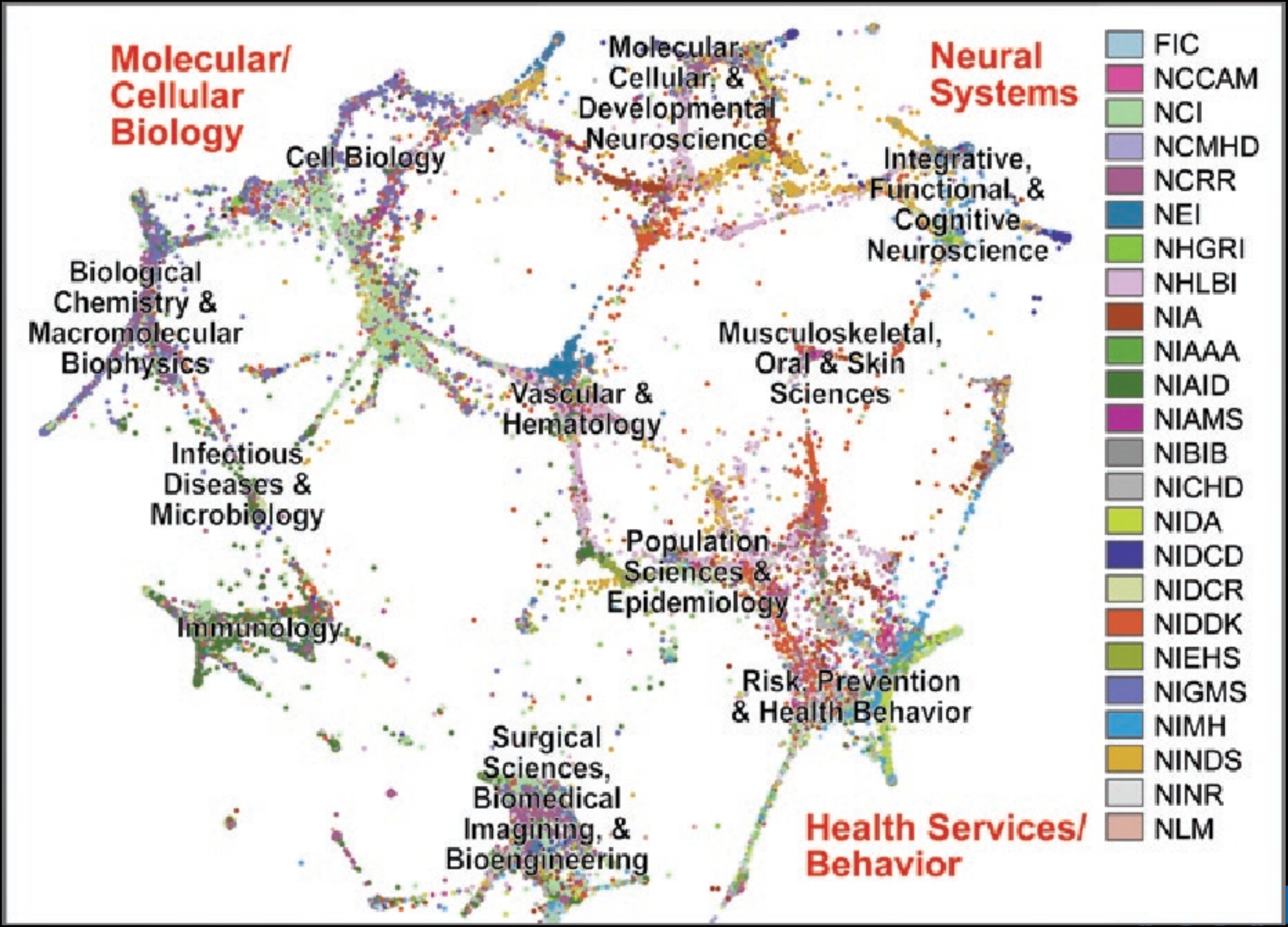
Bocconi

Goals for Today

- Understand what information **topic models** can and can not provide
- Learn about the **Latent Dirichlet Allocation (LDA)** model
- Understand the **parameters** influencing the output
- Learn about the **Structured Author Topic Model**
- Learn about **evaluation** criteria

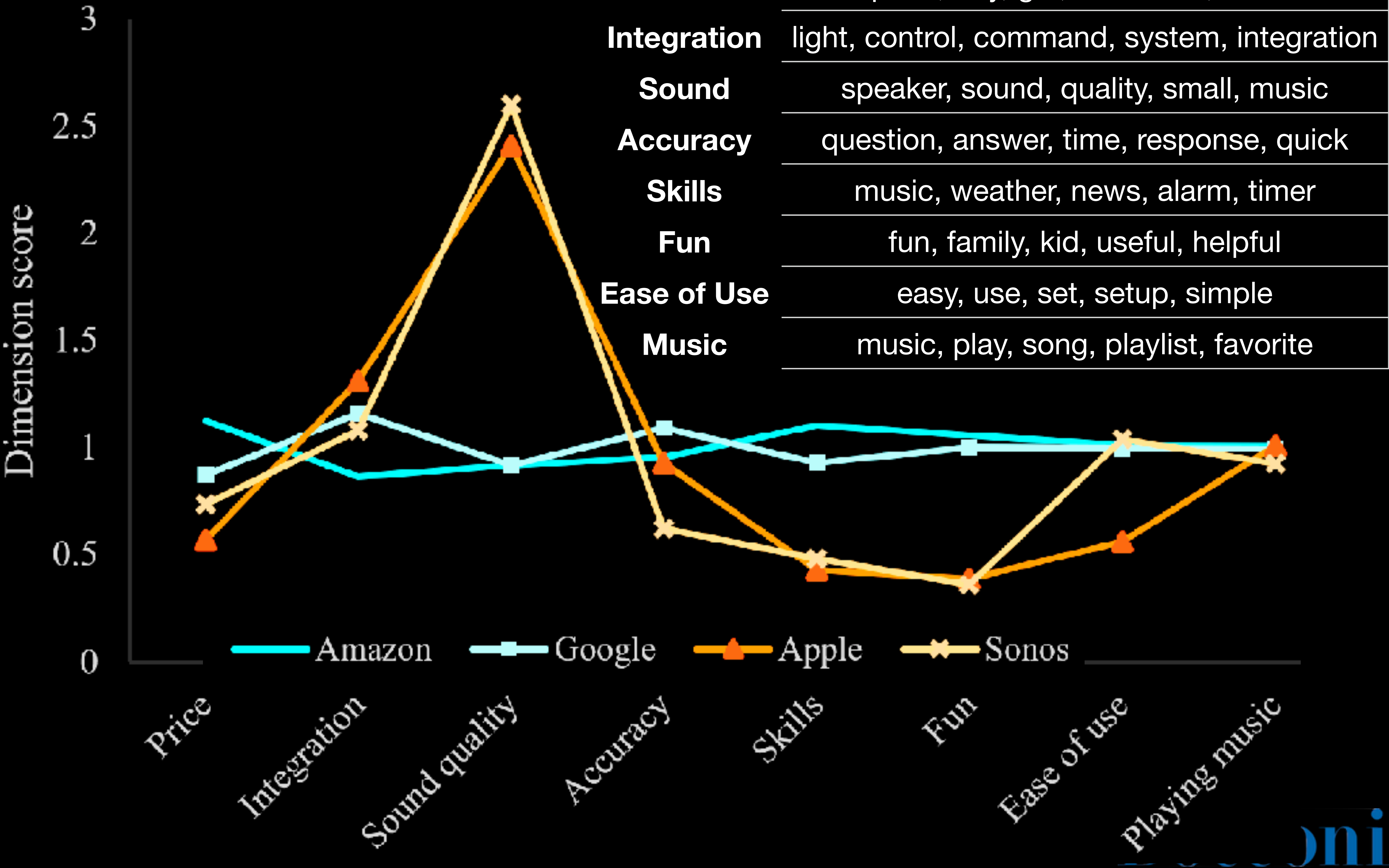
Talley et al., 2011

What Gets Funded?



Nguyen & Hovy (2019)

What do People Want in Smart Devices?



Price	price, buy, gift, christmas, worth
Integration	light, control, command, system, integration
Sound	speaker, sound, quality, small, music
Accuracy	question, answer, time, response, quick
Skills	music, weather, news, alarm, timer
Fun	fun, family, kid, useful, helpful
Ease of Use	easy, use, set, setup, simple
Music	music, play, song, playlist, favorite

Topics are Word Lists

TOPIC OR NOT?

- "pasta, pizza, wine, sauce, spaghetti"
- "BLEU, Bert, encoder, decoder, transformer"

SOME DOMAIN KNOWLEDGE REQUIRED…

How to use Topic Models

CORPUS

- preprocess

MODEL

- find best #topics
- find best parameters
- check output

DESCRIPTORS

[pasta, pizza, wine, sauce, spaghetti]

- choose top 5 words

TOPICS

FOOD

- name

Preprocessing

Preprocessing

- Be aggressive:
  - lemmatization,
  - stopwords,
  - replace numbers/user names,
  - join collocations
  - use TFIDF
- use minimum document frequency 10, 20, 50, or even 100
- use maximum document frequency 50% – 10%

Pre-processing steps

```
<div id="text">I've been in New York
in 2011, but didn’t like it. I
preferred Los Angeles.</div>
```

GOAL: MINIMIZE VARIATION

Pre-processing steps

- Remove formatting (e.g. HTML)
- Segment sentences
- Tokenize words
- Normalize words
  - numbers
  - lemmas vs. stems
- Remove unwanted words
  - stopwords
  - content words (use POS tagging!)
- join collocations

I've been in New York in 2011, but didn’t like it. I preferred Los Angeles.

Pre-processing steps

- Remove formatting (e.g. HTML)
- Segment sentences
- Tokenize words
- Normalize words
  - numbers
  - lemmas vs. stems
- Remove unwanted words
  - stopwords
  - content words (use POS tagging!)
- join collocations

```
I've been in New York in
2011, but didn’t like
it.

I preferred Los Angeles.
```

Pre-processing steps

- Remove formatting (e.g. HTML)
- Segment sentences
- Tokenize words
- Normalize words
  - numbers
  - lemmas vs. stems
- Remove unwanted words
  - stopwords
  - content words (use POS tagging!)
- join collocations

```
I ’ve been in New York
in 2011 , but did n’t
like it .

I preferred Los
Angeles .
```

Pre-processing steps

- Remove formatting (e.g. HTML)
- Segment sentences
- Tokenize words
- Normalize words
  - numbers
  - lemmas vs. stems
- Remove unwanted words
  - stopwords
  - content words (use POS tagging!)
- join collocations

```
i ‘ve been in new york
in 0000 , but did n’t
like it .

i preferred los
angeles .
```

Pre-processing steps

- Remove formatting (e.g. HTML)
- Segment sentences
- Tokenize words
- Normalize words
  - numbers
  - lemmas vs. stems
- Remove unwanted words
  - stopwords
  - content words (use POS tagging!)
- join collocations

```
i have be in new york in
0000 , but do not like
it .

i prefer los angeles .
```

Pre-processing steps

- Remove formatting (e.g. HTML)
- Segment sentences
- Tokenize words
- Normalize words
  - numbers
  - lemmas vs. stems
- Remove unwanted words
  - stopwords
  - content words (use POS tagging!)
- join collocations

```
i new york 0000 , like .

i prefer los angeles .
```

Pre-processing steps

- Remove formatting (e.g. HTML)
- Segment sentences
- Tokenize words
- Normalize words
  - numbers
  - lemmas vs. stems
- Remove unwanted words
  - stopwords
  - content words (use POS tagging!)
- join collocations

```
new york 0000 like

prefer los angeles
```

CONTENT = (NOUN, VERB, NUM)

Pre-processing steps

- Remove formatting (e.g. HTML)
- Segment sentences
- Tokenize words
- Normalize words
  - numbers
  - lemmas vs. stems
- Remove unwanted words
  - stopwords
  - content words (use POS tagging!)
- join collocations

```
new_york 0000 like

prefer los_angeles
```

Pre-processing steps

```
<div id="text">I've been in New York
in 2011, but didn’t like it. I
preferred Los Angeles.</div>
```

MINIMIMAL VARIATION

"BAG OF WORDS"

```
new_york 0000 like

prefer los_angeles
```

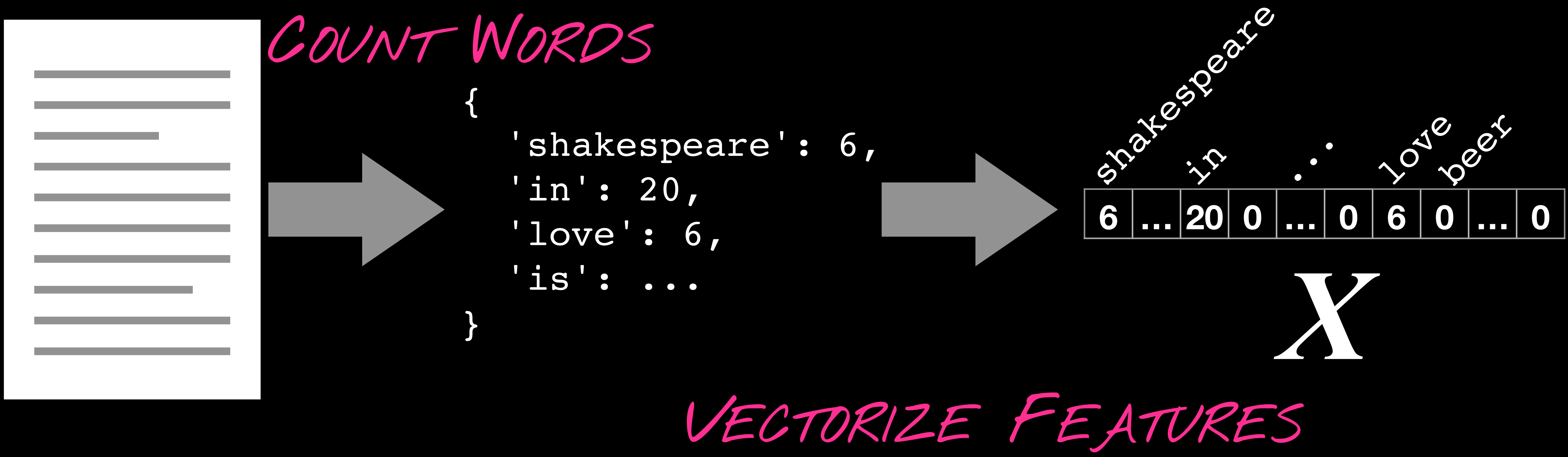
Representing Text

N-grams

"As Gregor Samsa awoke one morning from uneasy dreams, he found himself transformed in his bed into a gigantic insect-like creature."

Unigrams	`As, Gregor, Samsa, awoke, one, morning, from, uneasy, dreams, ...`
Bigrams	`As_Gregor, Gregor_Samsa, Samsa_awoke, awoke_one, one_morning, ...`
Trigrams	`As_Gregor_Samsa, Gregor_Samsa_awoke, Samsa_awoke_one, awoke_one_morning, ...`
4-grams	`As_Gregor_Samsa_awoke, Gregor_Samsa_awoke_one, Samsa_awoke_one_morning, ...`

Bags of words (BOW)



Count Words

```
{
   'shakespeare': 6,
   'in': 20,
   'love': 6,
   'is': ...
}
```

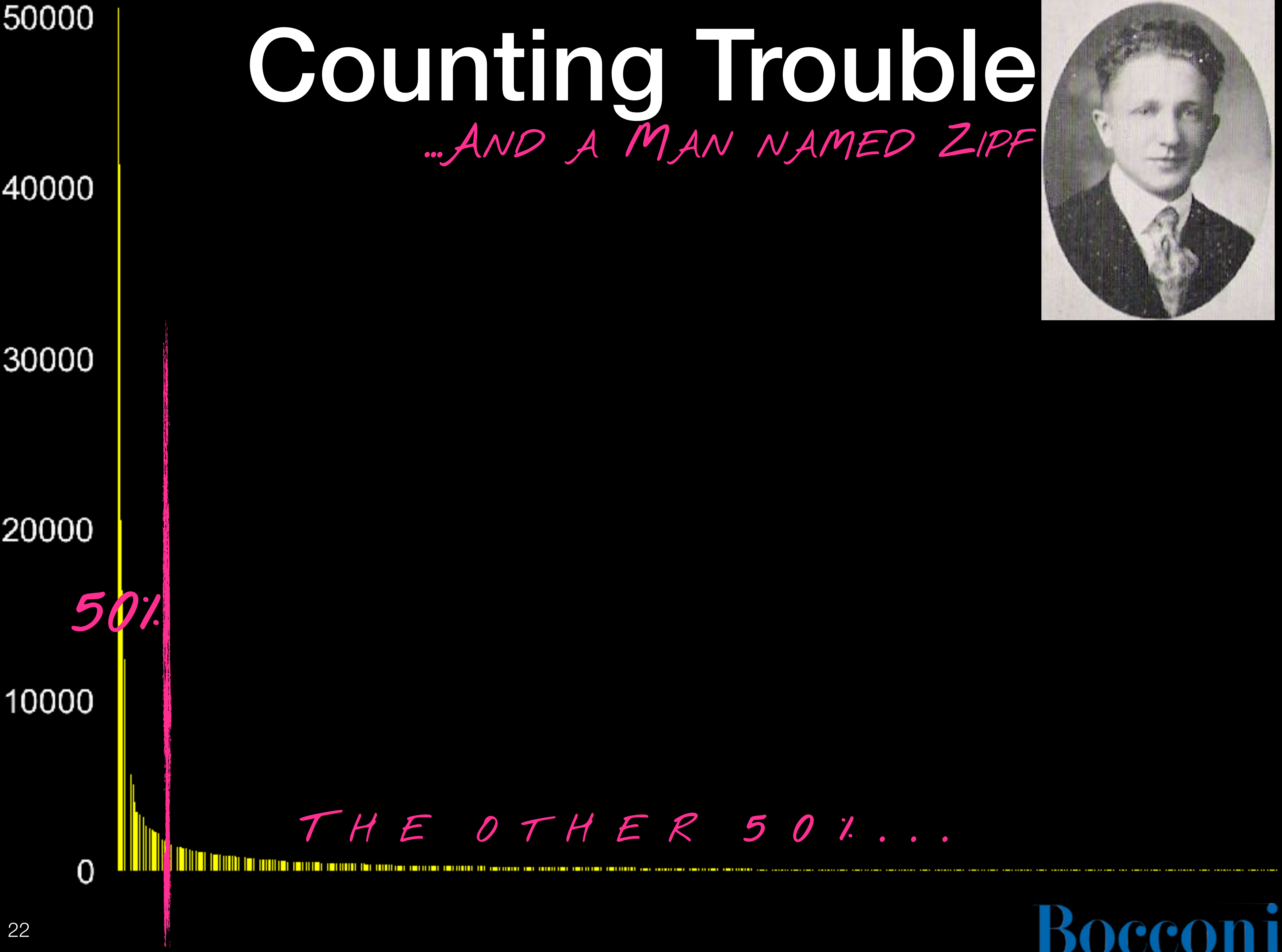
shakespeare		in				love	beer		
6	...	20	0	...	0	6	0	...	0

X

Vectorize Features

Counting Trouble

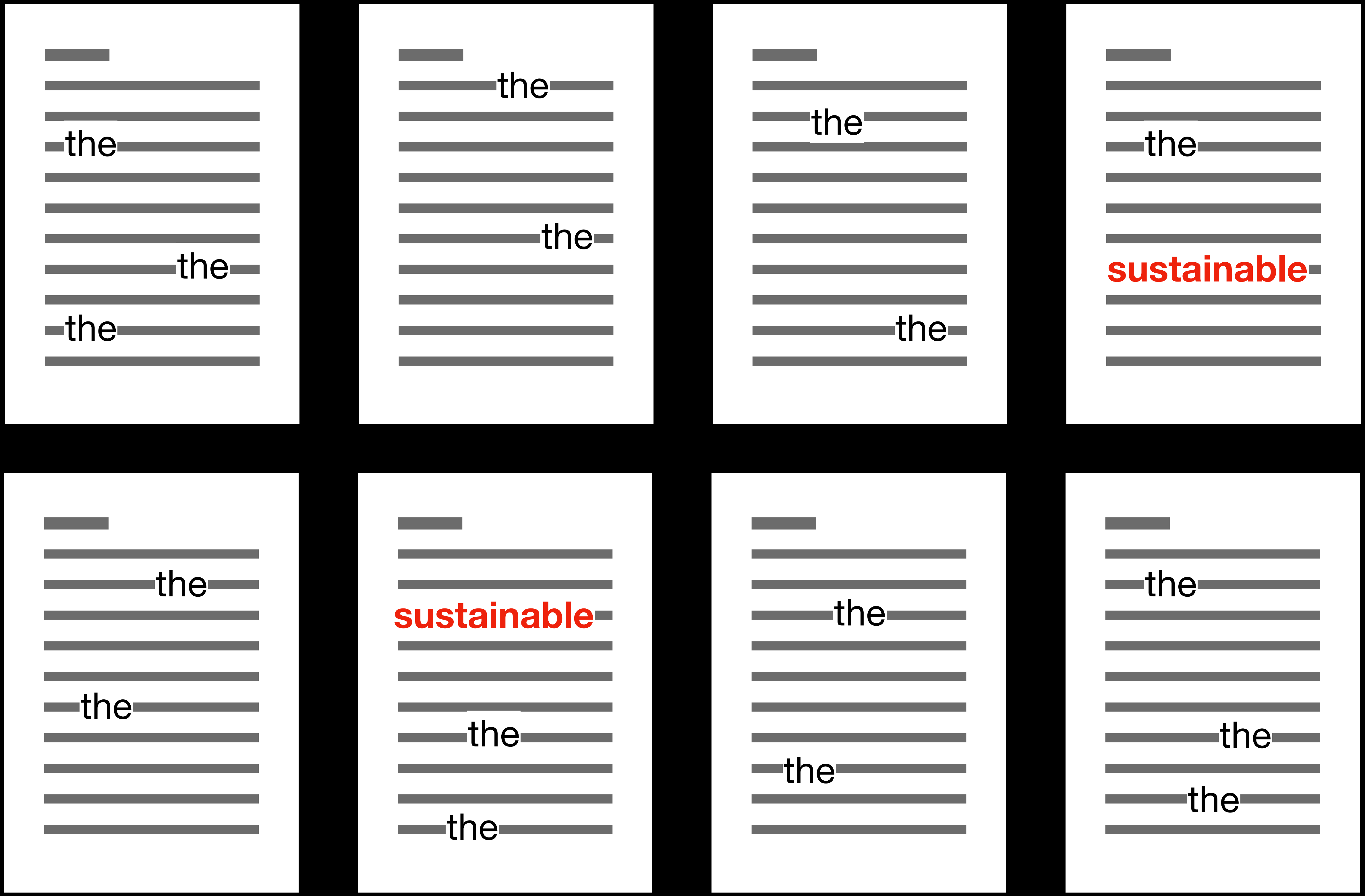
…AND A MAN NAMED ZIPF



Finding Important Words: TF-IDF

Bocconi

Some Words are Just More Interesting…

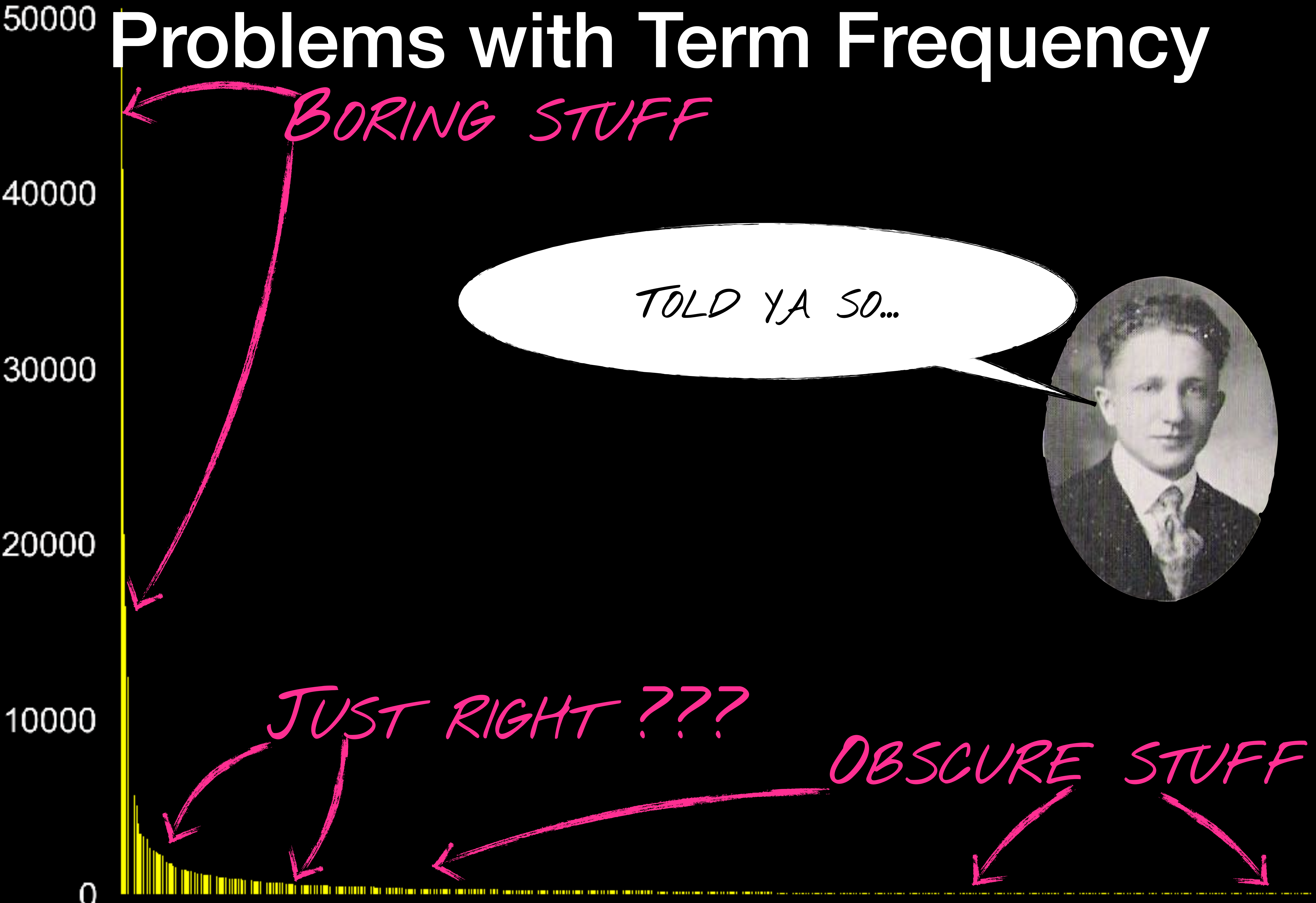


Karen Spärck Jones

1935–2007

- Became a teacher before starting CS career at Cambridge
- Laid the foundation for modern NLP, Google Search, text classification
- Campaigned for more women in CS
- Namesake of prestigious CS prize

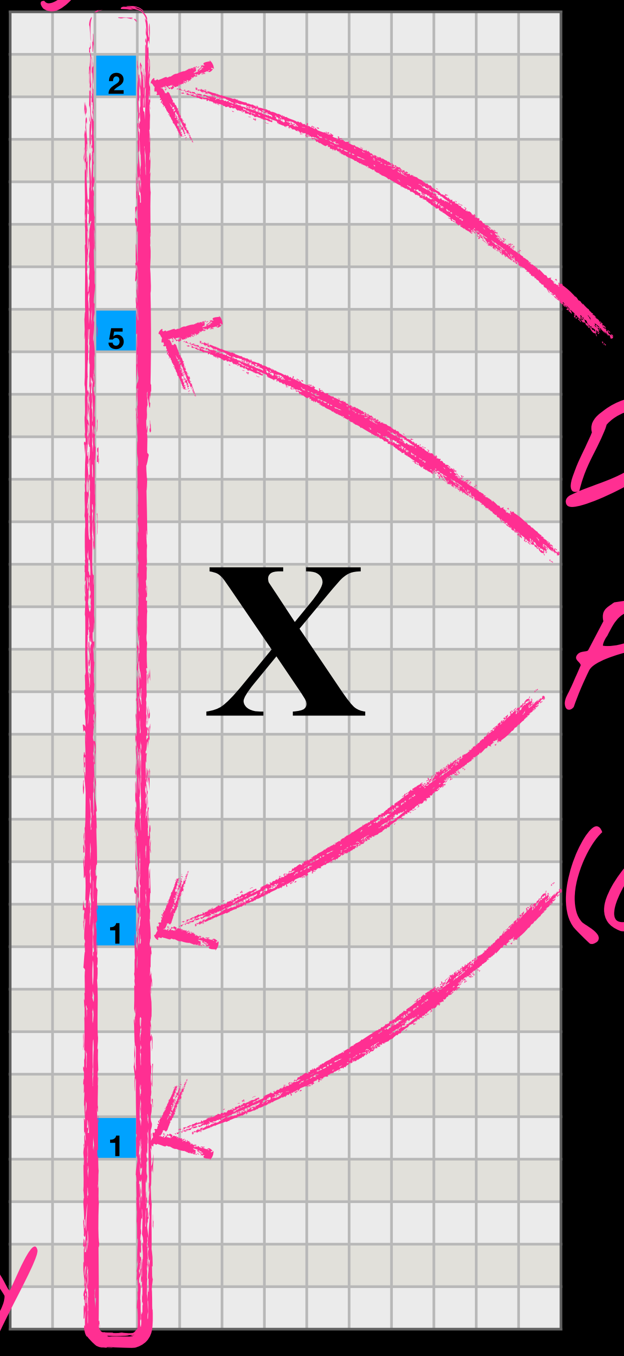
Problems with Term Frequency



Document and Term Frequency

$$IDF = \log \frac{N}{df(w)}$$

FEATURE

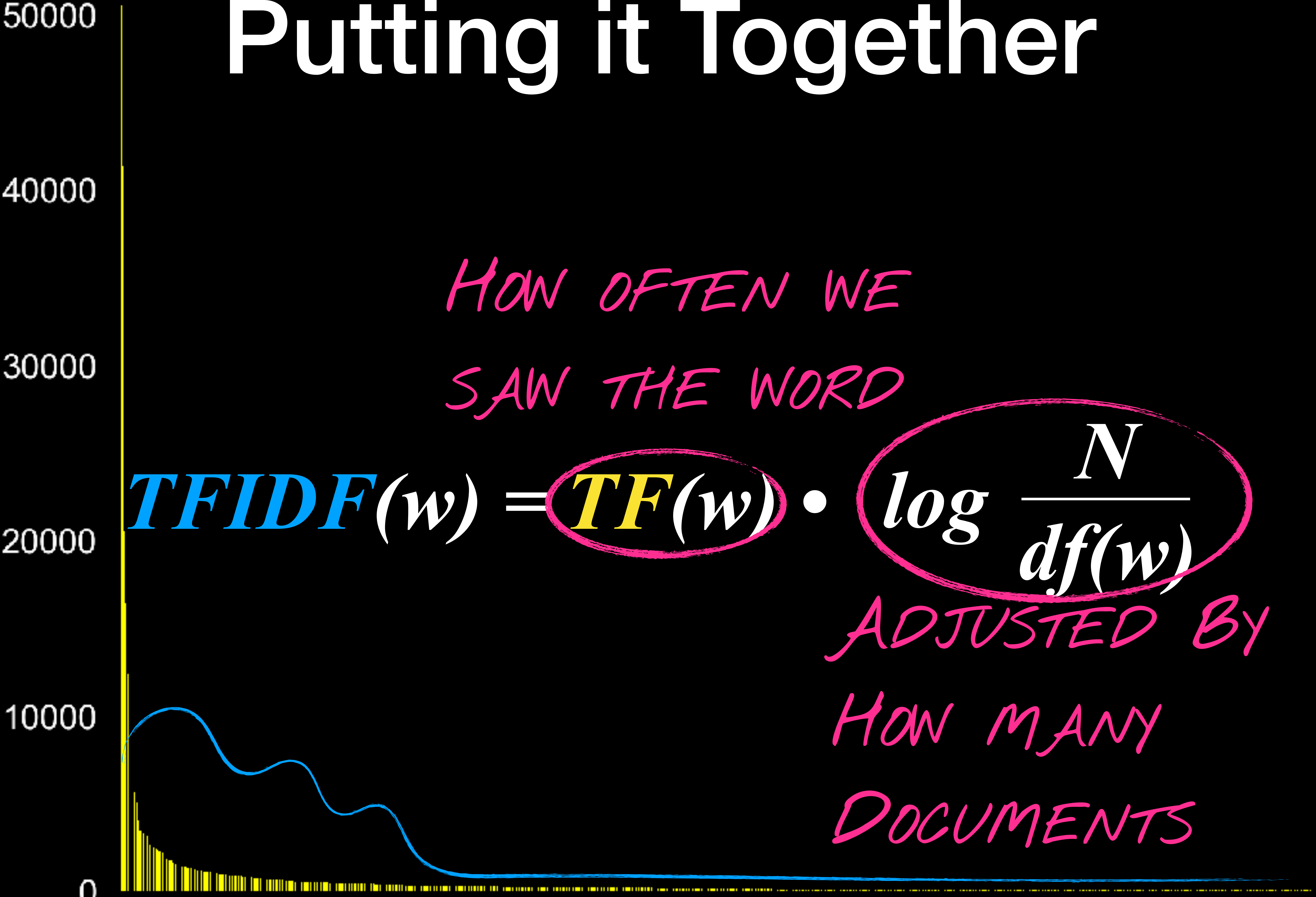


DOCUMENT FREQUENCY (COUNT): 4

TERM FREQUENCY (SUM): 9

TF

Putting it Together



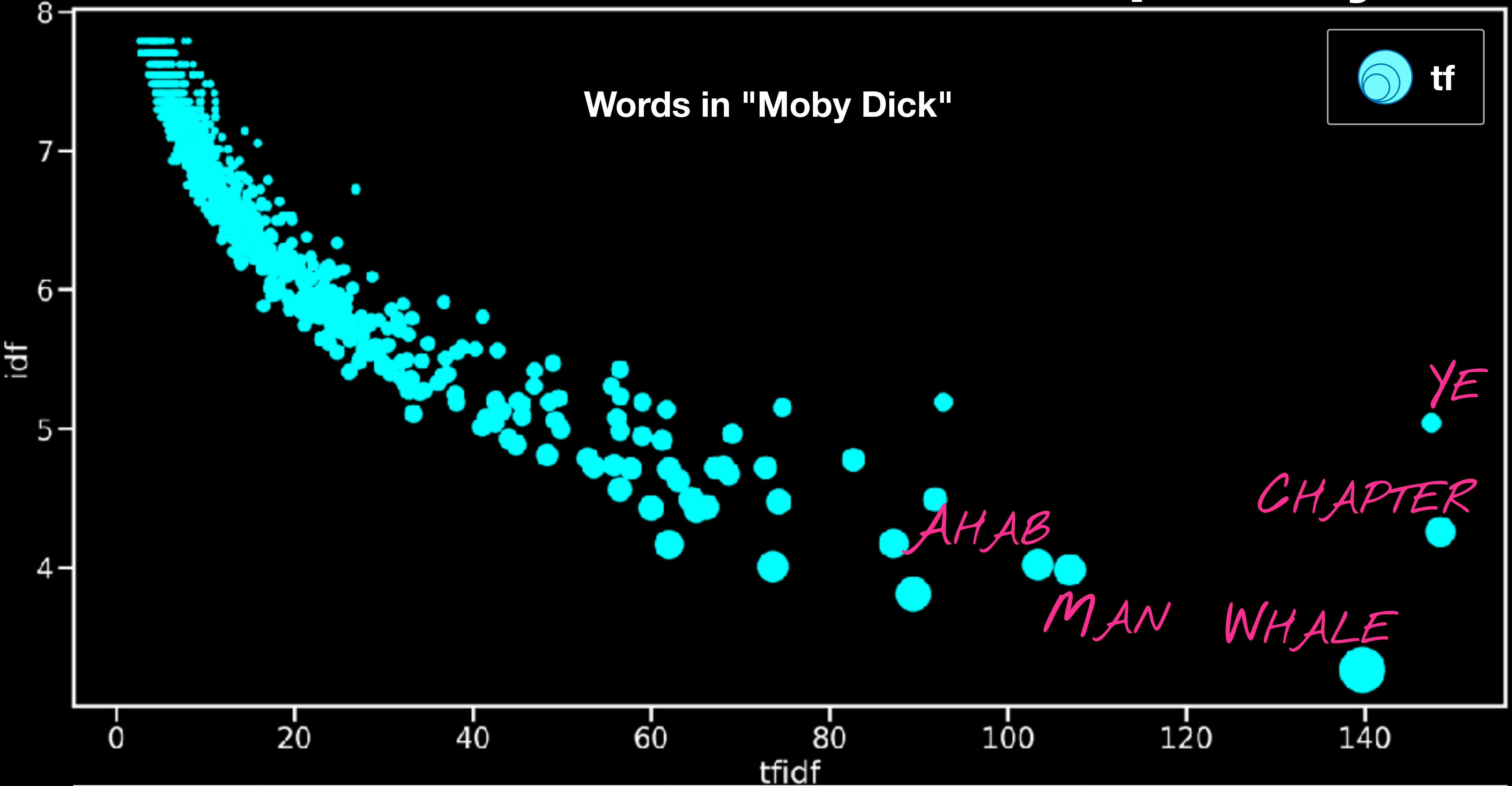
HOW OFTEN WE SAW THE WORD

$$TFIDF(w) = TF(w) \cdot \log \frac{N}{df(w)}$$

ADJUSTED BY HOW MANY DOCUMENTS

Bocconi

Document and Term Frequency

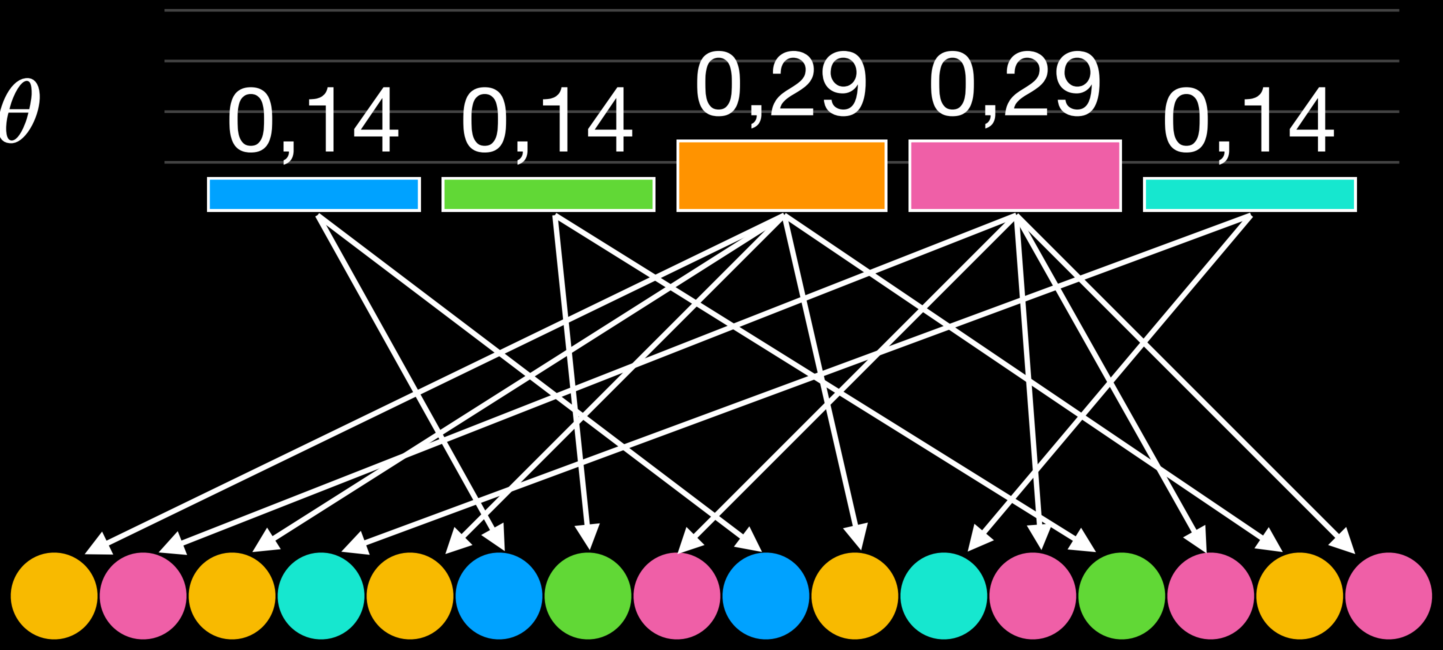


word	tf	idf	tfidf
ye	467	4.257380	148.497079
chapter	171	5.039475	147.504638
whale	1150	3.262357	139.755743
man	525	3.982412	106.932953
ahab	511	4.019453	103.357774

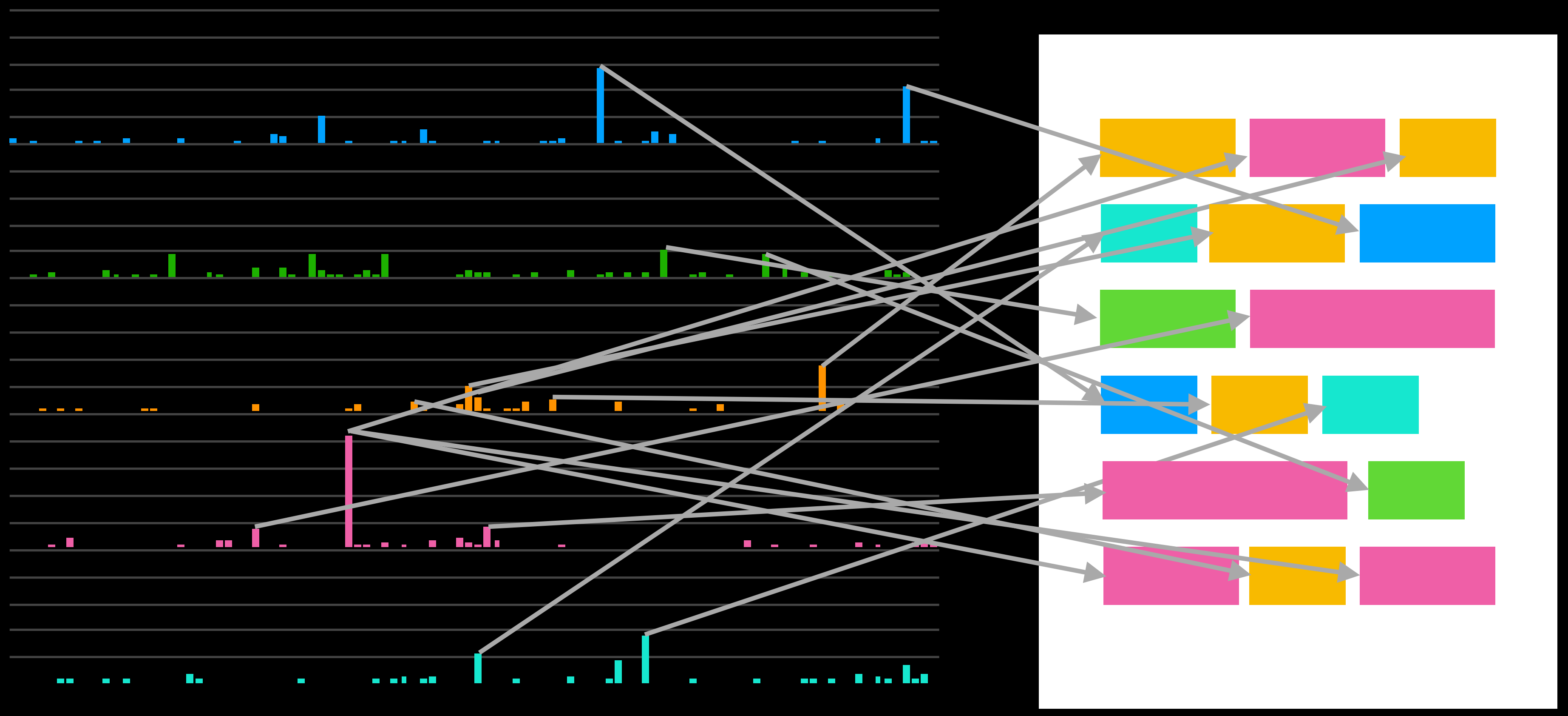
Latent Dirichlet Allocation

How to Generate Documents

- Draw a topic distribution θ

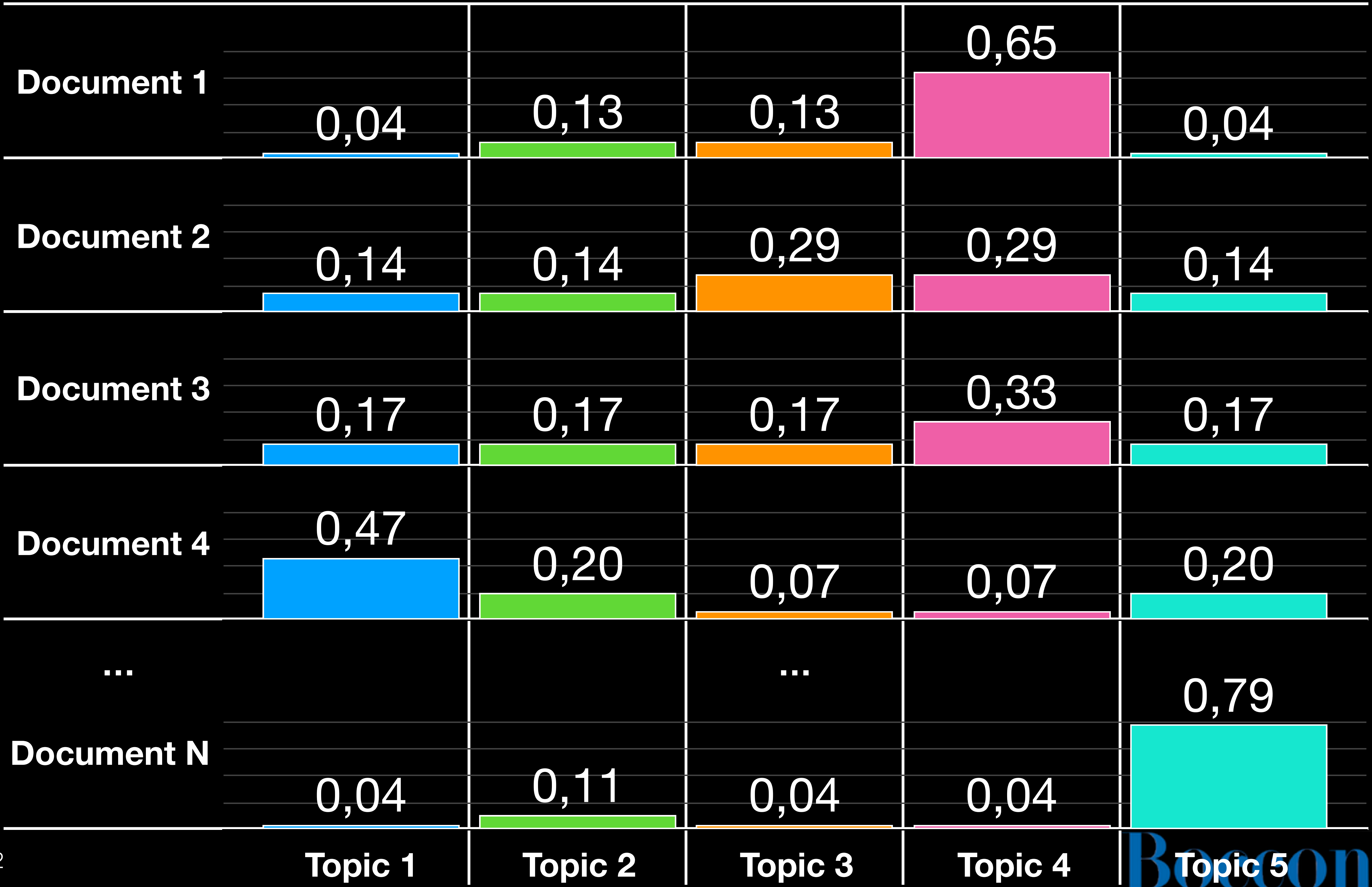


- For i in N:
  - Draw a topic from θ
  - Sample a word from the word distribution z



Topics per Document

$\theta = P(topic|document)$



	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
Document 1	0,04	0,13	0,13	0,65	0,04
Document 2	0,14	0,14	0,29	0,29	0,14
Document 3	0,17	0,17	0,17	0,33	0,17
Document 4	0,47	0,20	0,07	0,07	0,20
...			...		
Document N	0,04	0,11	0,04	0,04	0,79

Words per Topic

$z = P(word|topic)$

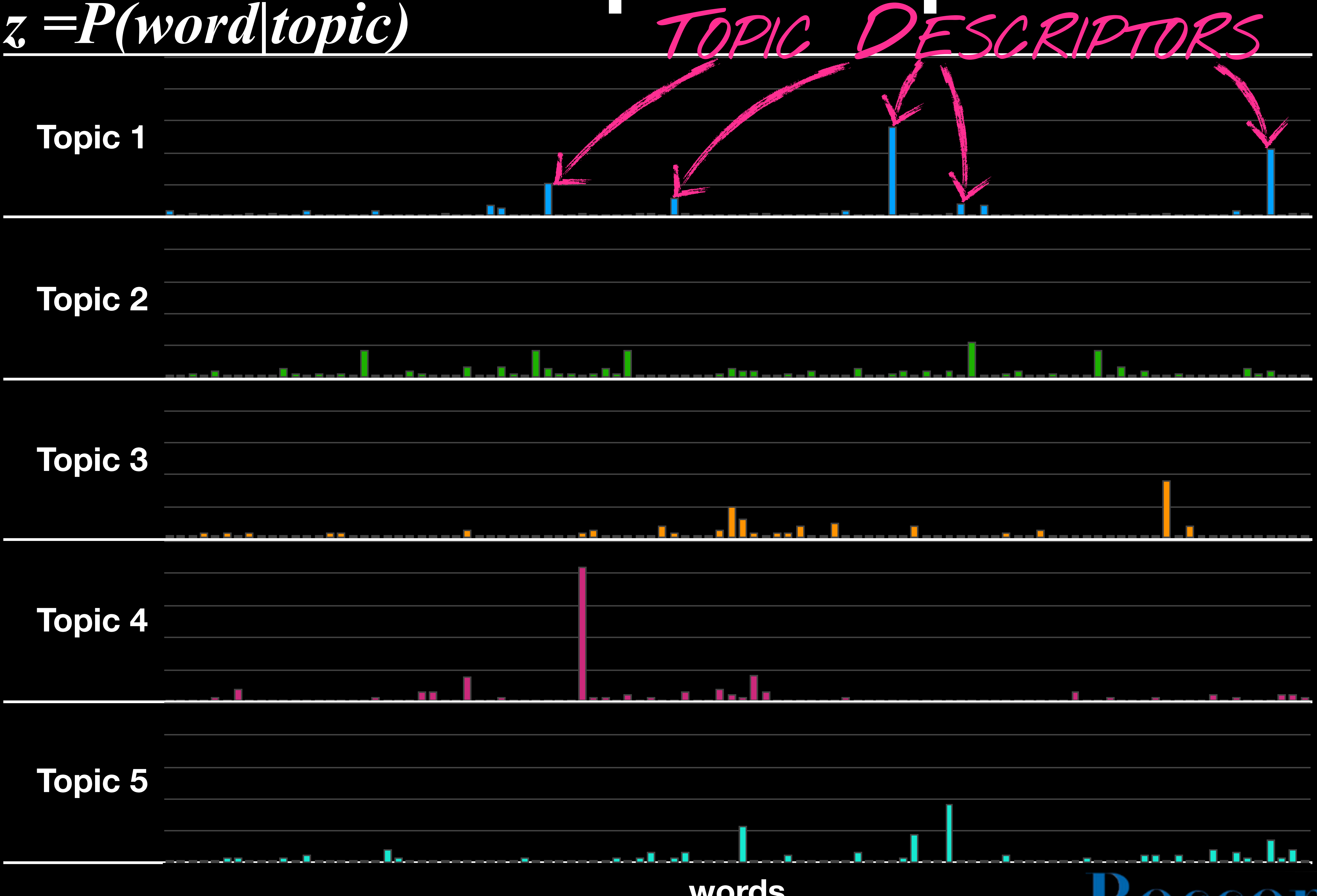
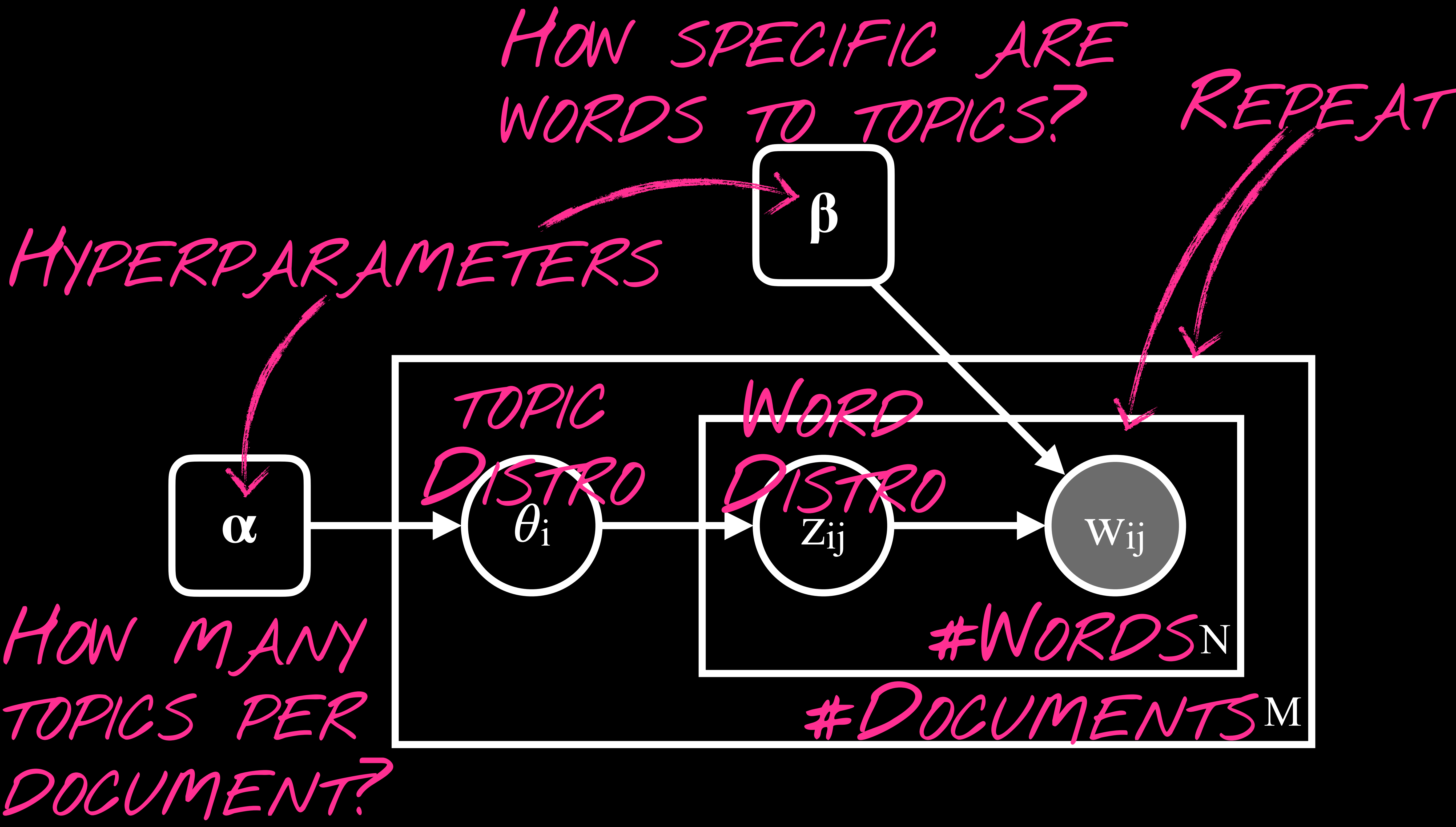


Plate Notation

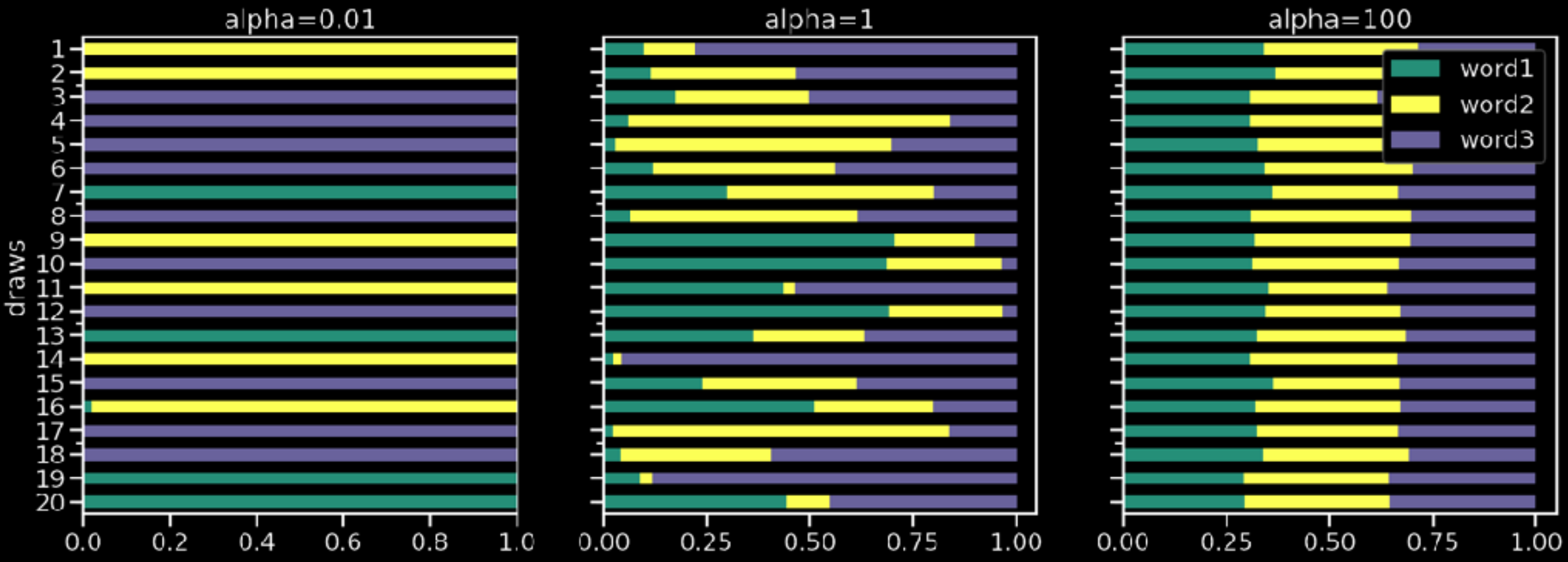


HOW SPECIFIC ARE WORDS TO TOPICS?

HOW MANY TOPICS PER DOCUMENT?

Dirichlet Distributions

"DISTRIBUTION GENERATOR"

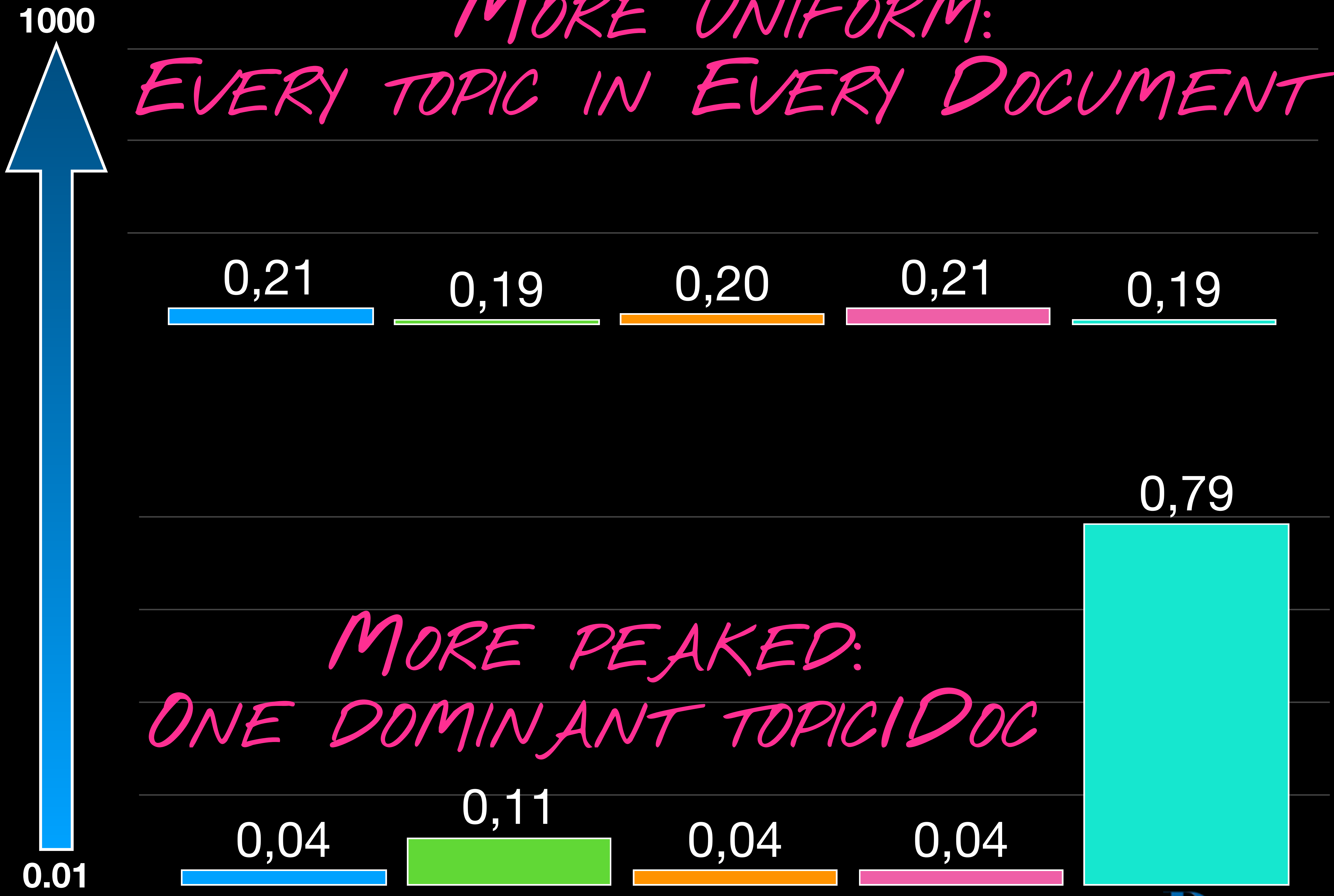


PEAKED

PARETO

UNIFORM

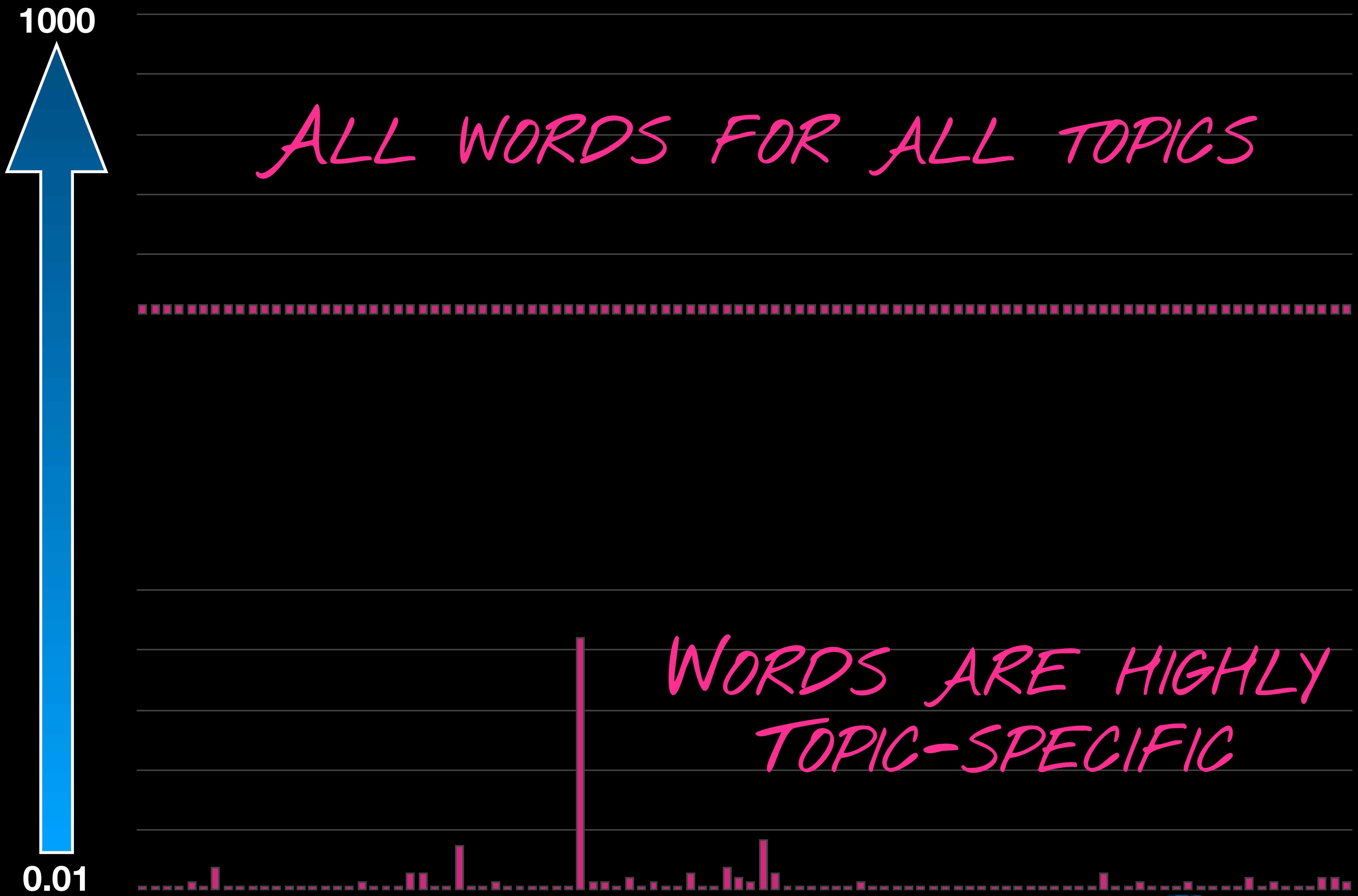
Parameters: α



MORE UNIFORM:
EVERY TOPIC IN EVERY DOCUMENT

MORE PEAKED:
ONE DOMINANT TOPIC/DOC

Parameters: β



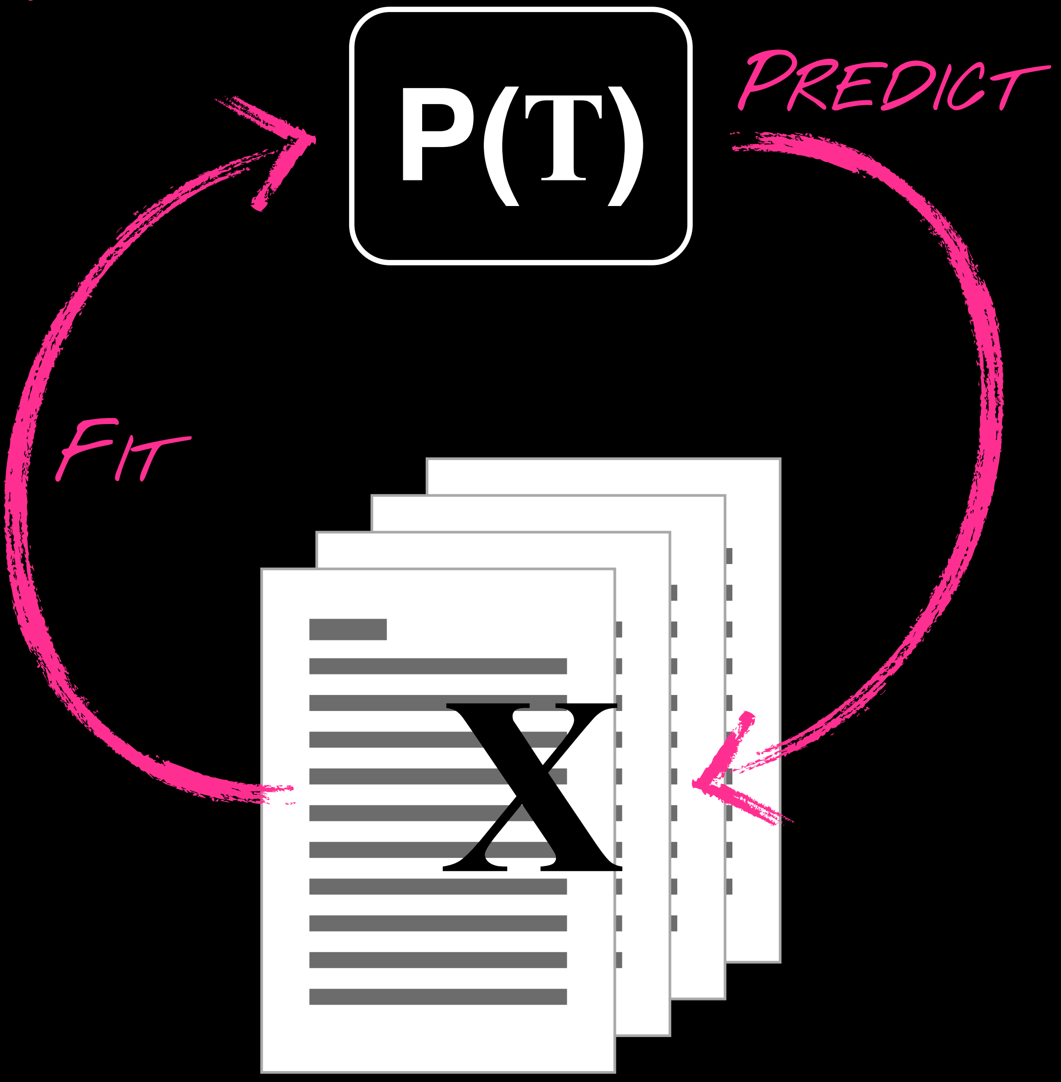
ALL WORDS FOR ALL TOPICS

WORDS ARE HIGHLY TOPIC-SPECIFIC

Training and Parameters

Evaluating LDA

Model-Inherent



$$=2^{-\sum_{x} p(x)\log p(x)}$$

Perplexity

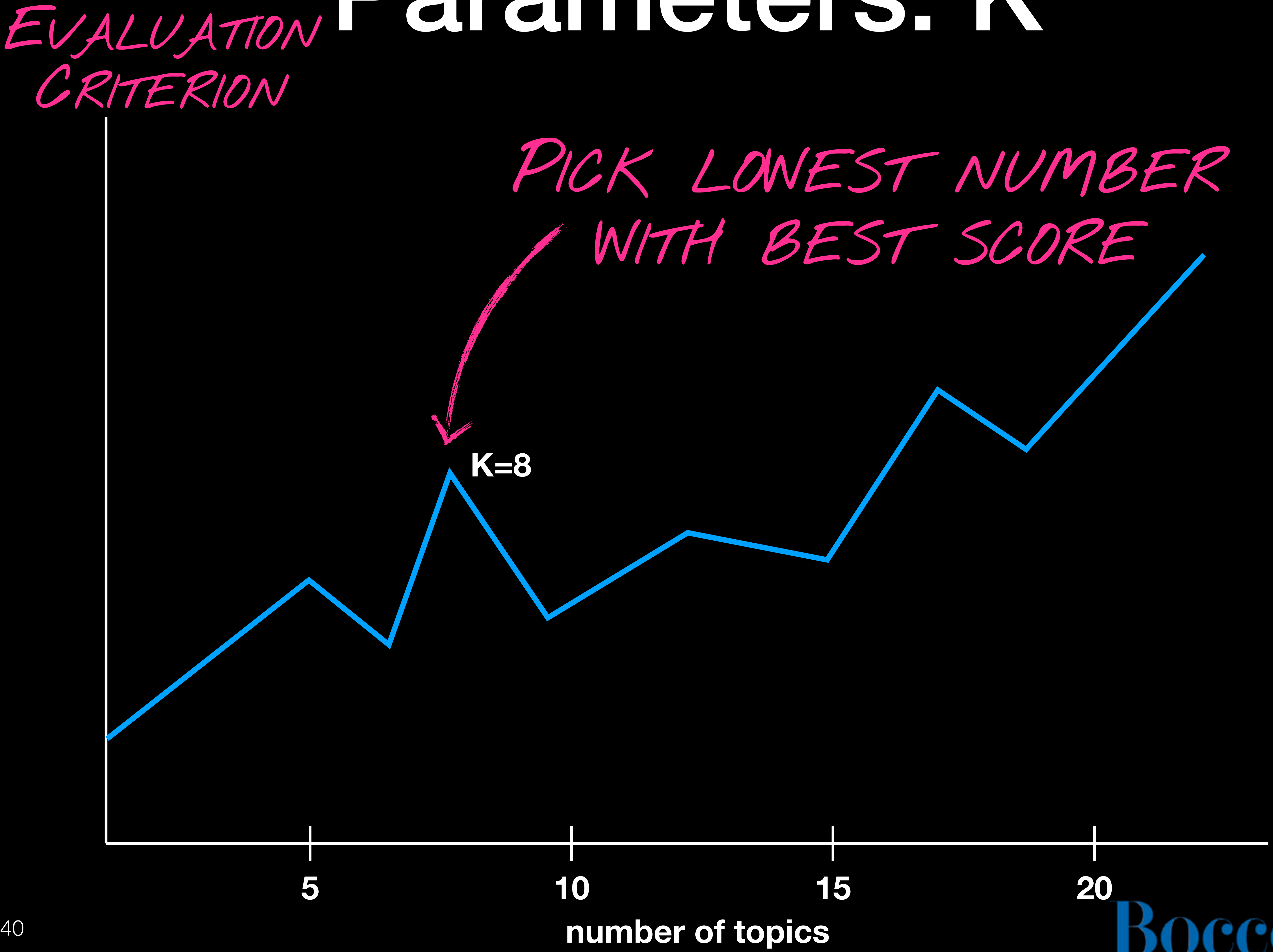
Content-Based

[apple, banana, pear, lime, orange]

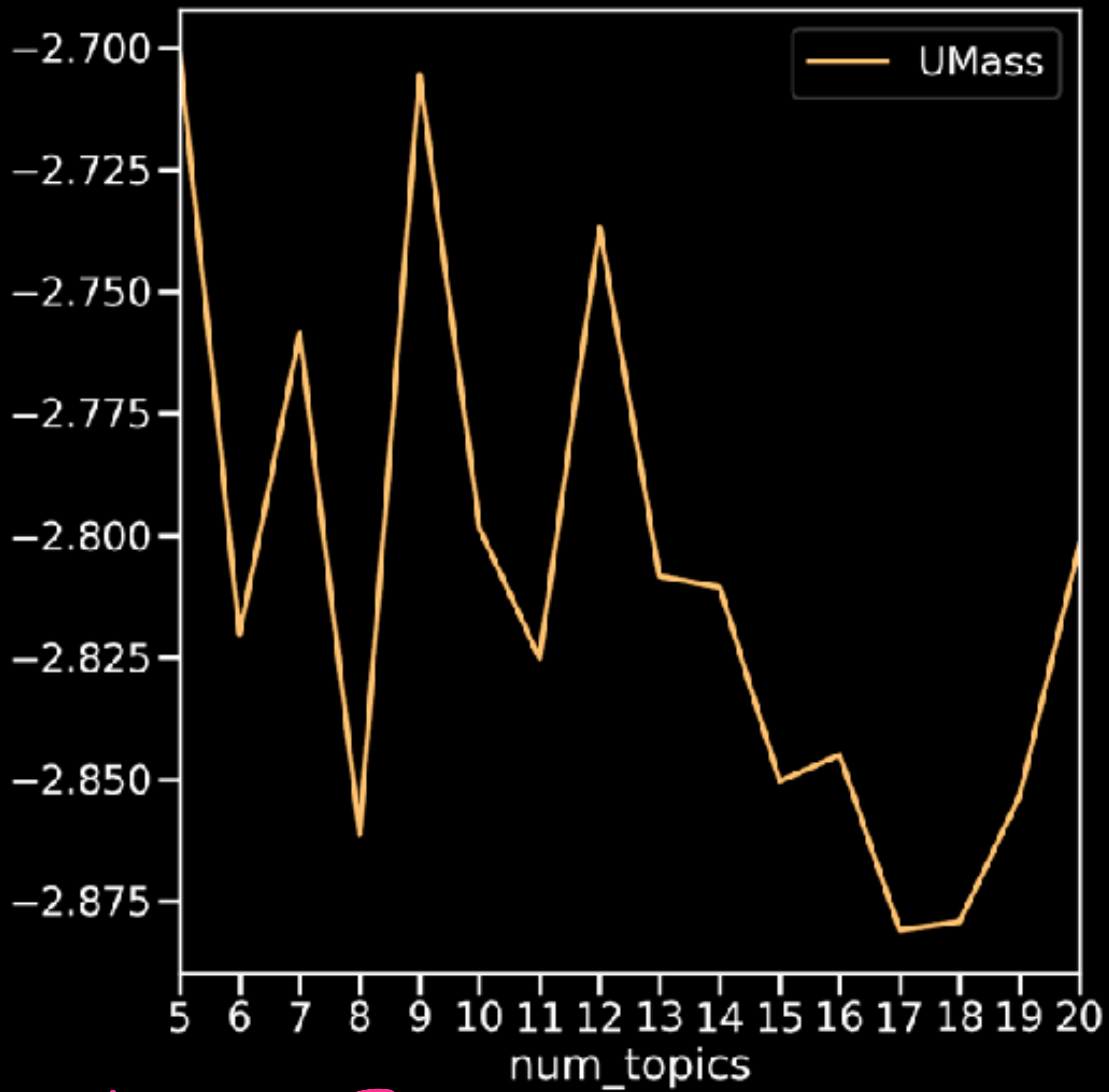
[apple, banana, **foot**, lime, orange]

Word Intrusion

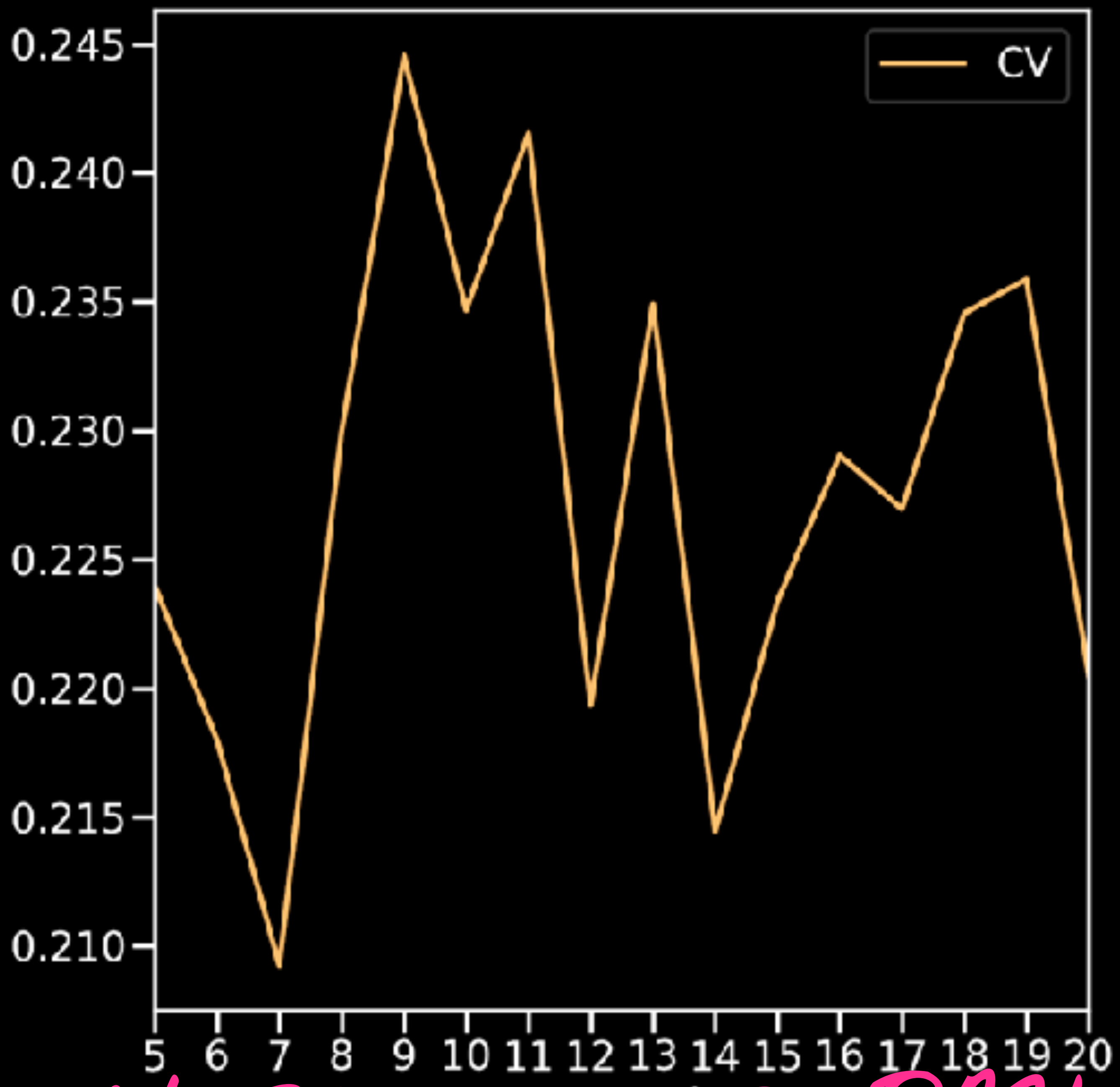
Parameters: K



Coherence Scores

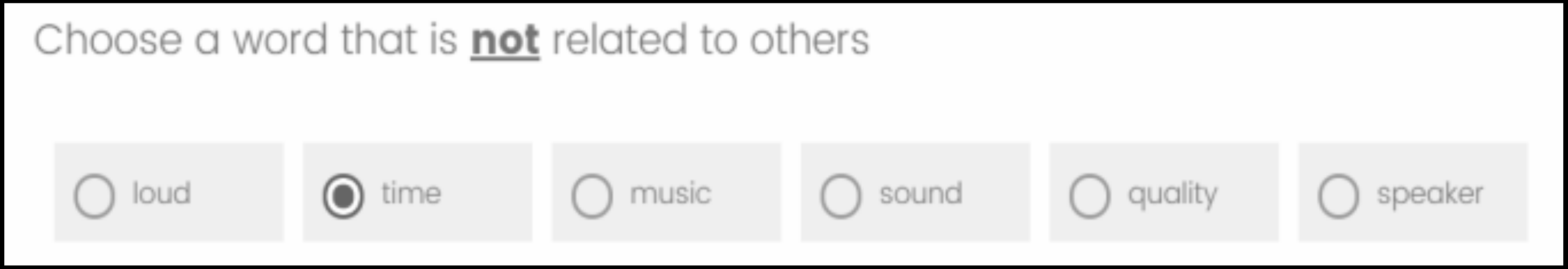


LOG PROB OF WORD CO-OCCURRENCES



NORMALIZED PMI AND COSINE SIMILARITY

Word and Topic Intrusion

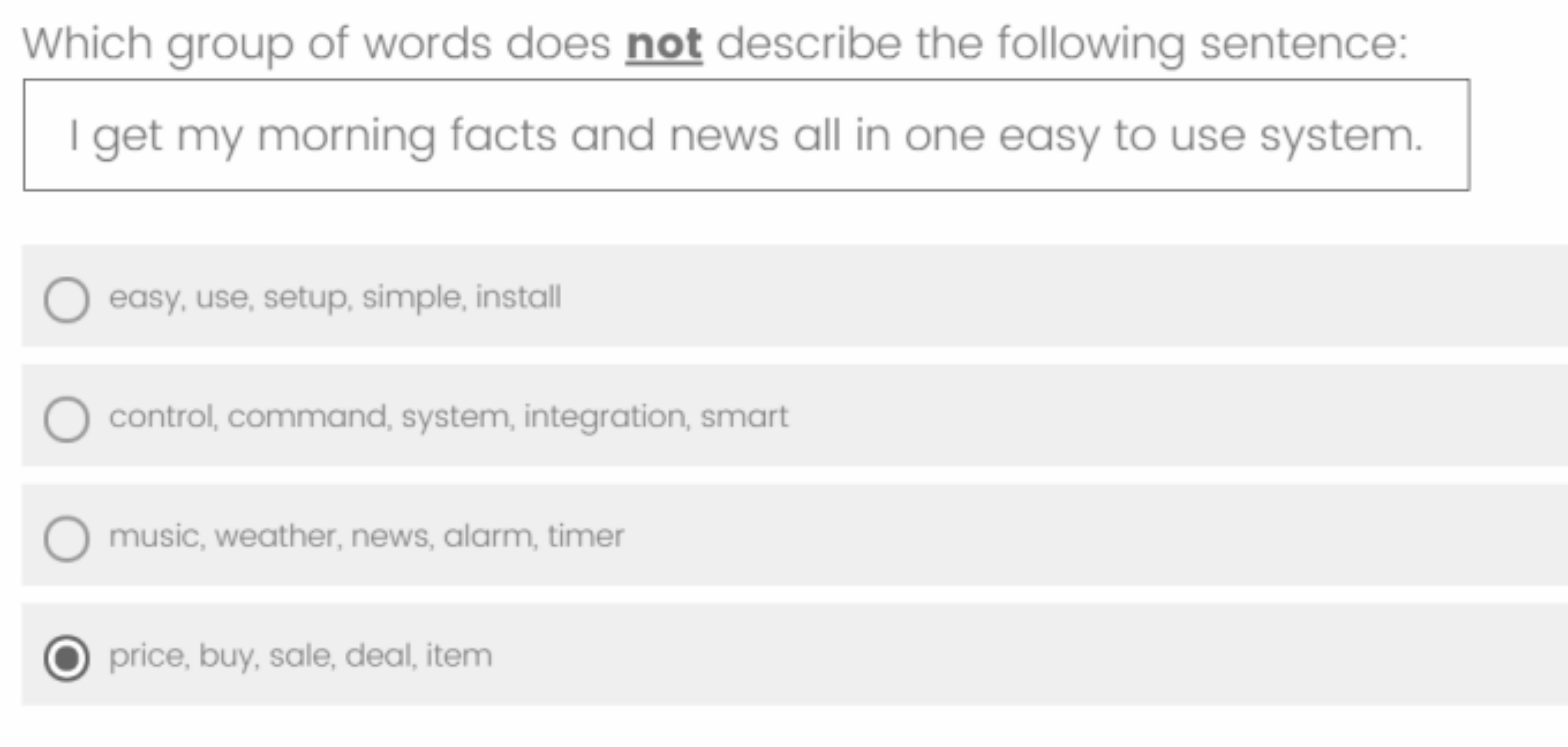


Choose a word that is **not** related to others

○ loud ● time ○ music ○ sound ○ quality ○ speaker

WORD INTRUSION

TOPIC INTRUSION



Which group of words does **not** describe the following sentence:

I get my morning facts and news all in one easy to use system.

○ easy, use, setup, simple, install

○ control, command, system, integration, smart

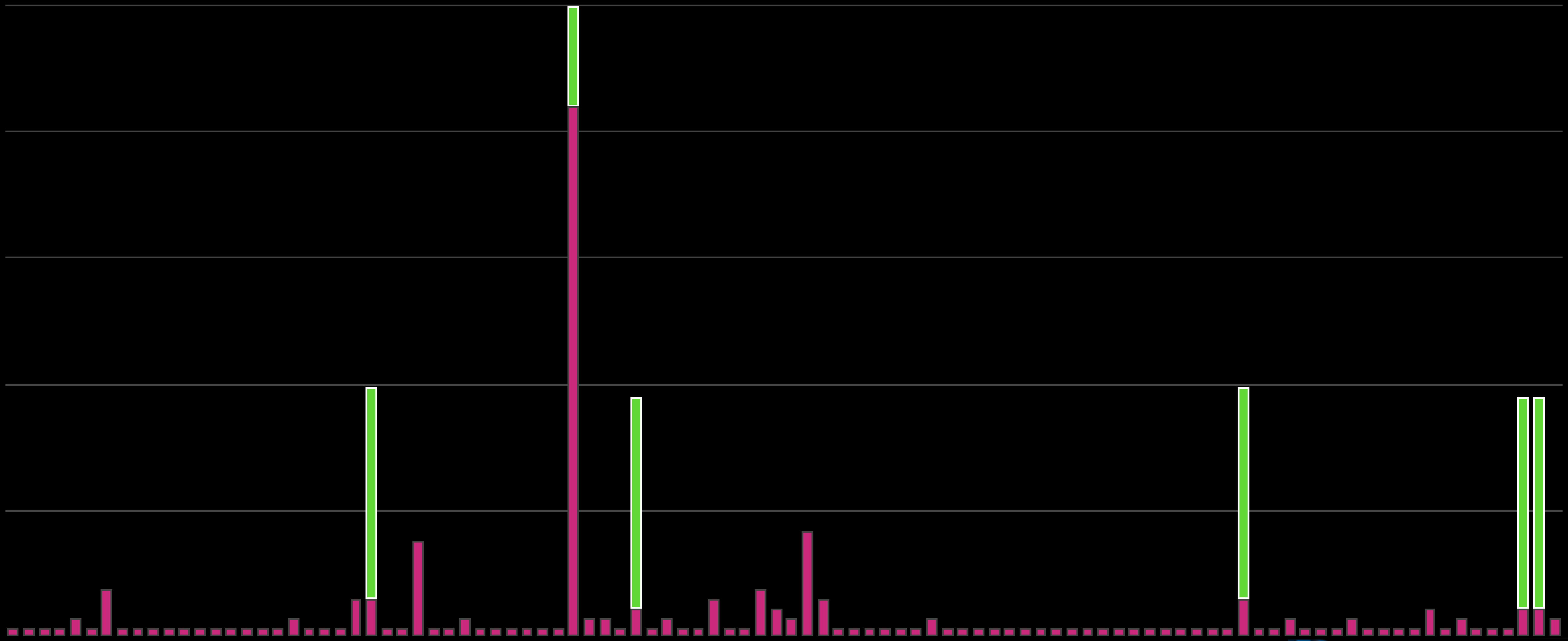
○ music, weather, news, alarm, timer

● price, buy, sale, deal, item

Slide credit: Hanh Nguyen

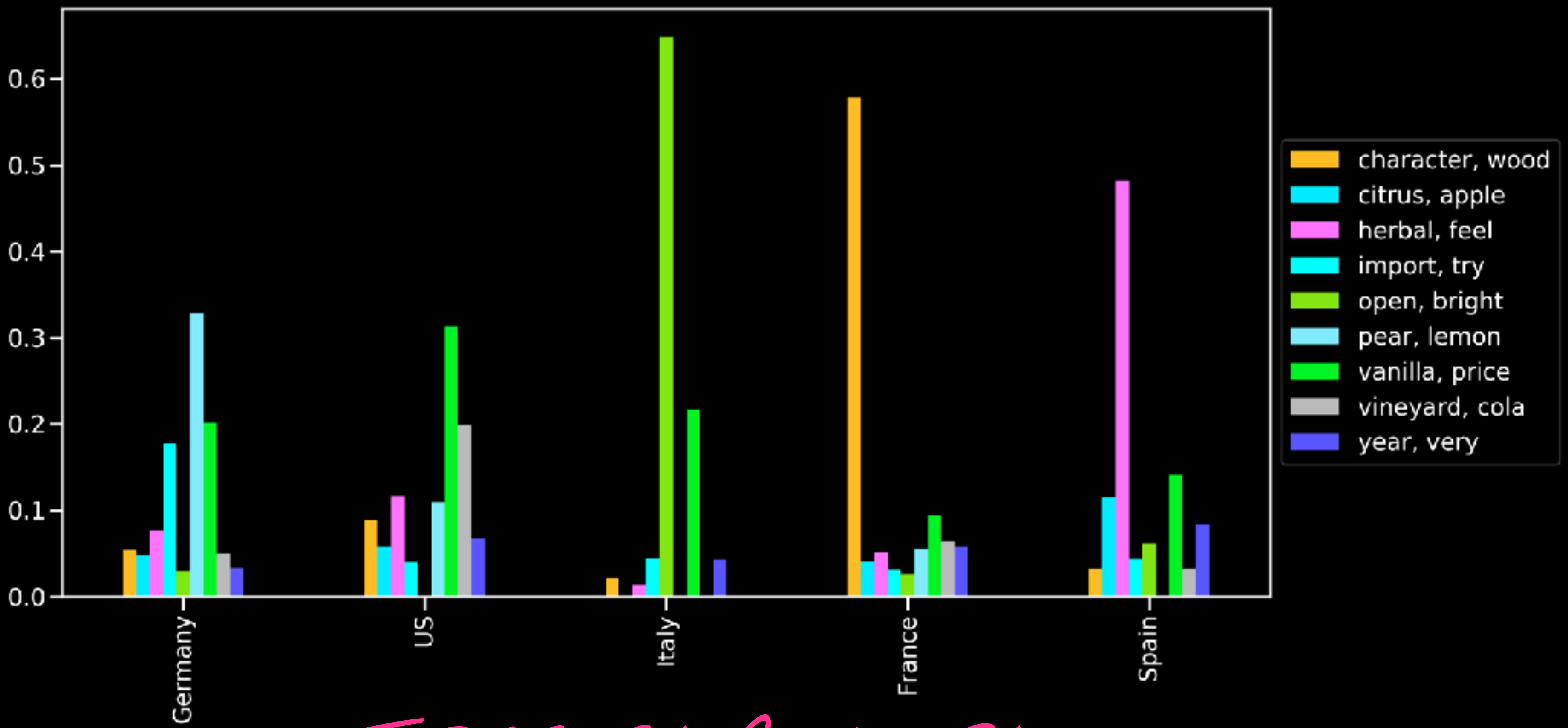
Adding Constraints

- Maybe we know which words go with a topic
- Fix some probabilities/add smoothing



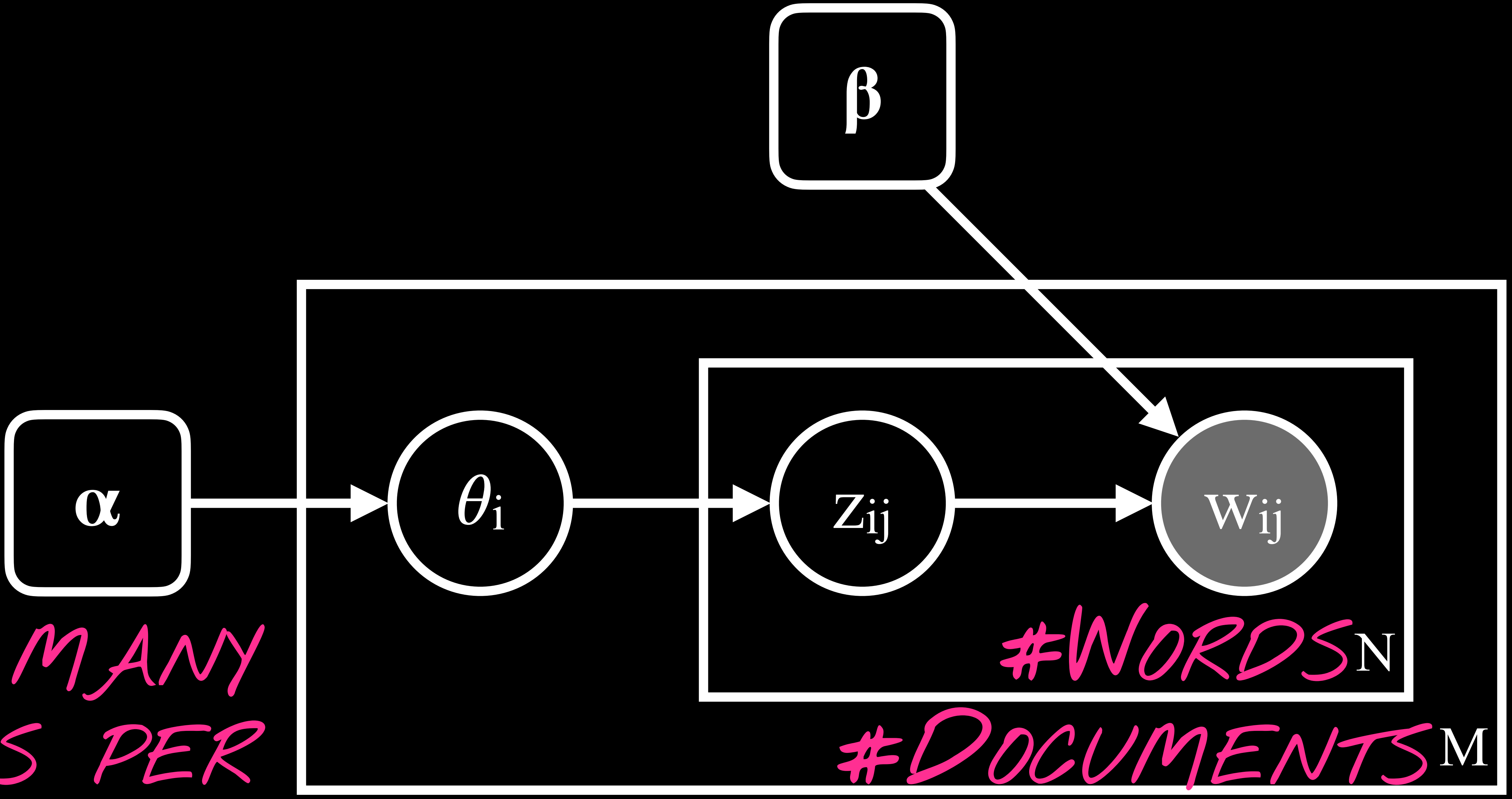
Author Topic Models

- Learn separate topic distribution for external factors



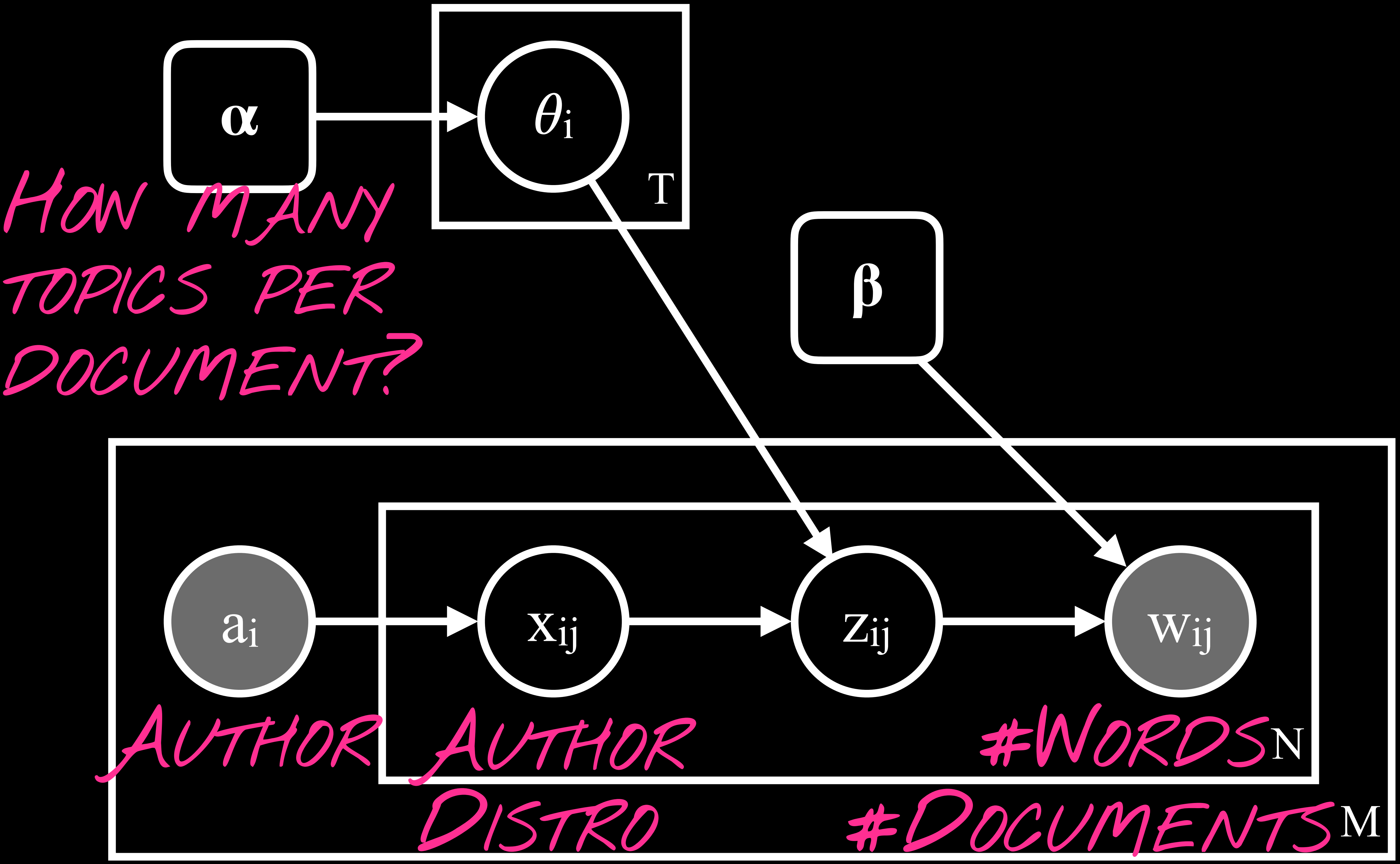
TOPICS BY COUNTRY

Plate Notation



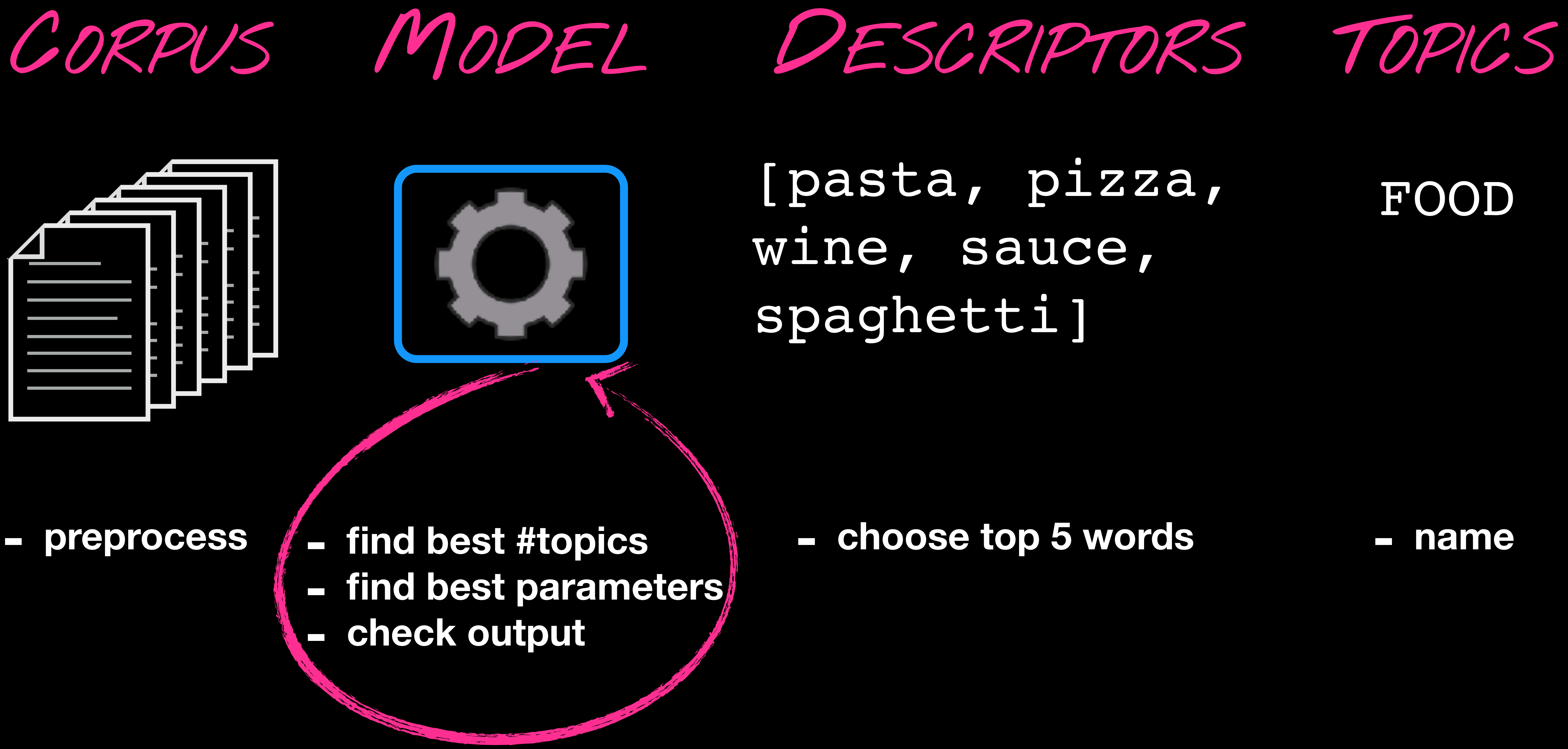
HOW MANY TOPICS PER DOCUMENT?

Plate Notation



Wrapping Up

How to use Topic Models



CORPUS

- preprocess

MODEL

- find best #topics
- find best parameters
- check output

DESCRIPTORS

[pasta, pizza, wine, sauce, spaghetti]

- choose top 5 words

TOPICS

FOOD

- name

Caveats!

Topic models ALWAYS need manual assessment, because:

- Random initialization: no two models are the same!
- More likely models ≠ more interpretable topics
- "Interpretable" is subjective
- Topics are not stable from run to run

NEVER USE TOPICS AS INPUT TO REGRESSION!

Take-Home Points

- **LDA** is one architecture for **topic models**
- Model document generation conditioned on latent topics
- Topic models are **stochastic**: each run is different
- **Preprocessing** and **parameters** influence performance
- Results need to be **interpreted**!
- We can introduce constraints through priors or labels

To Neural and Beyond

https://github.com/MilaNLProc/contextualized-topic-models

- Based on neural networks: better coherence
- cross-lingual: train in one language, use in others
- add supervision: use document labels (similar to author topics)

	Sentence	Topic
EN	Blackmore’s Night is a British/American traditional folk rock duo [...]	rock, band, bass, formed
IT	Blackmore’s Night sono la band fondatrice del renaissance rock [...]	rock, band, bass, formed
PT	Blackmore’s Night´e uma banda de folk rock de estilo [...]	rock, band, bass, formed
EN	Langton’s ant is a two-dimensional Turing machine with [...]	math, theory, space, numbers
FR	On nomme fourmi de Langton un automate cellulaire [...]	math, theory, space, numbers
DE	Die Ameise ist eine Turingmaschine mit einem [...]	math, theory, space, numbers