Text Analysis
TFIDF, REs, and Collocations

Dirk Hovy

dirk.hovy@unibocconi.it

@dirk_hovy

Bocconi

Today's Goals

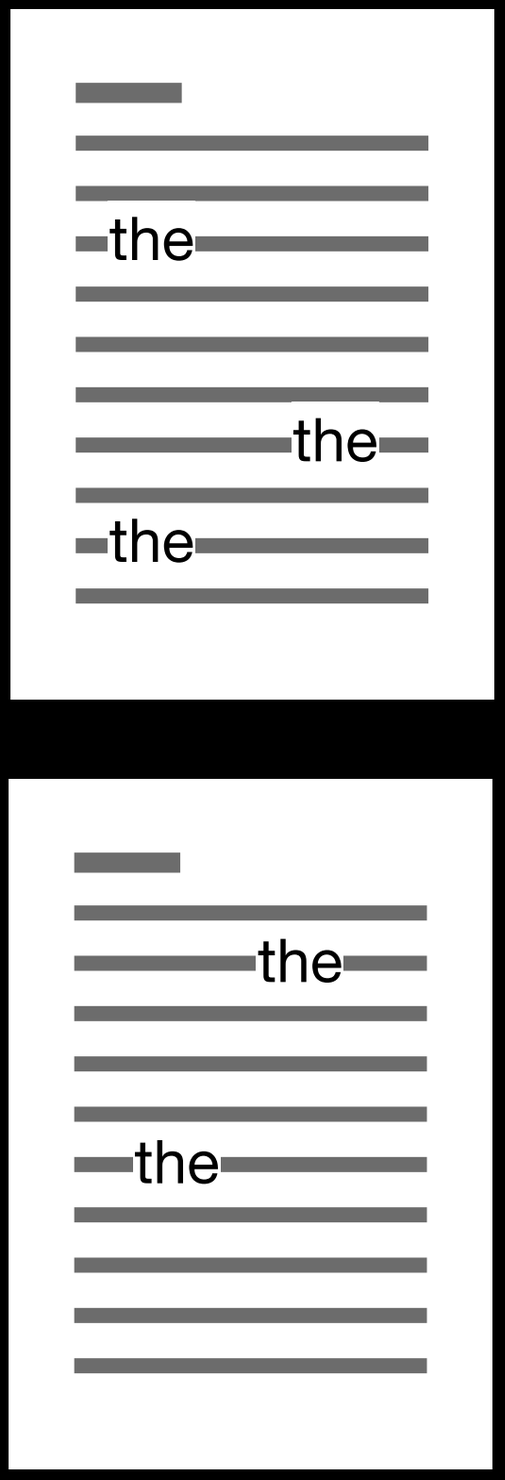
- Learn about forms of **TF-IDF** and its possibilities
- Know when (and when not) to use **Regular expressions**
- Understand **PMI** and see why it can help find collocations

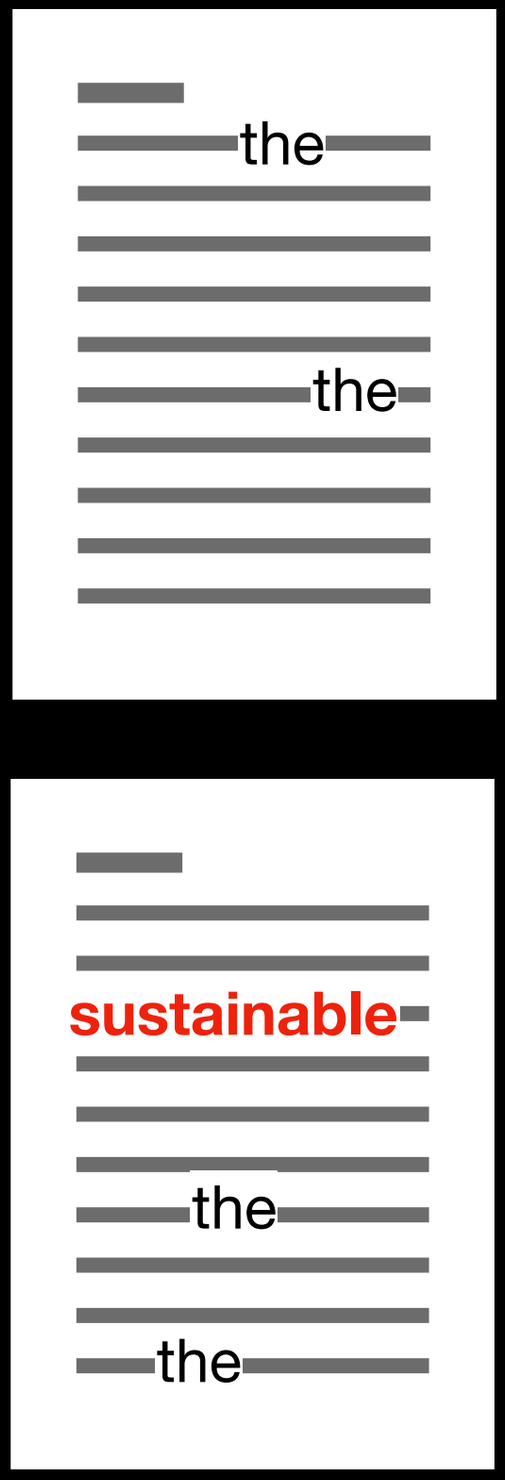
Finding Important Words:

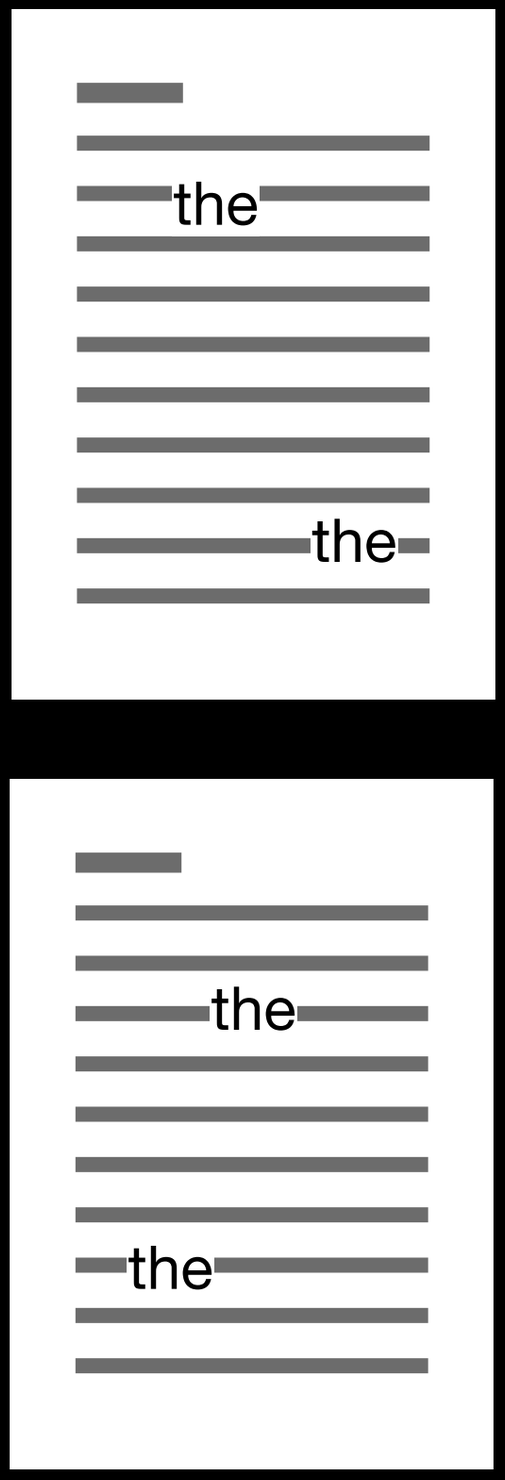
TF-IDF

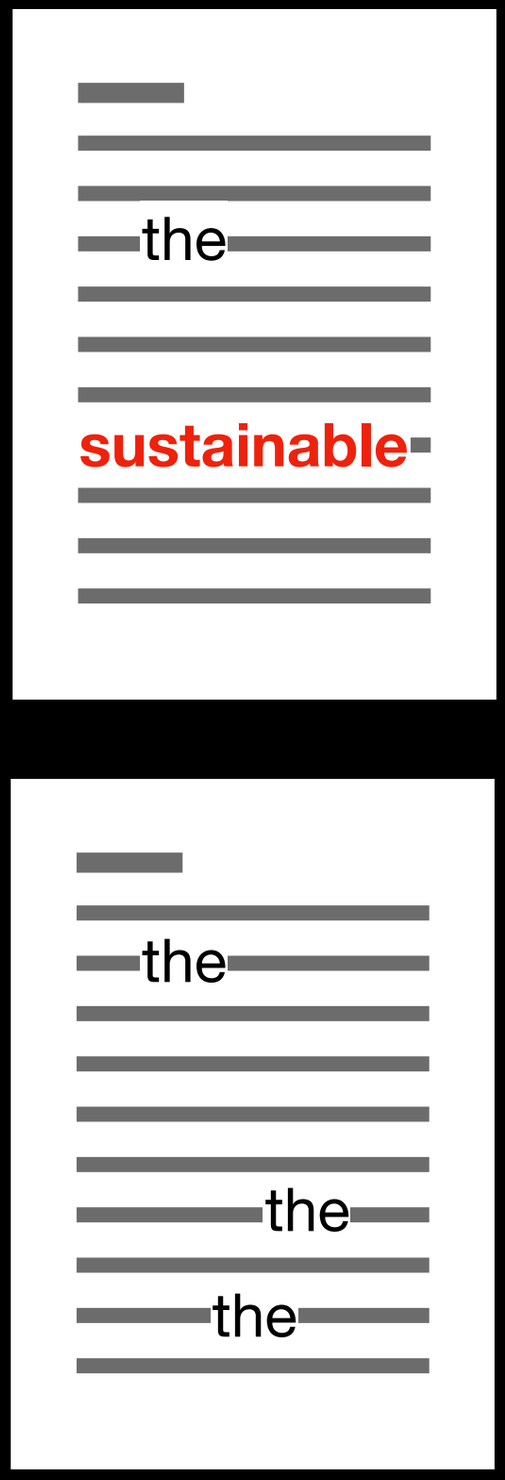
Bocconi

Some Words are Just More Interesting…









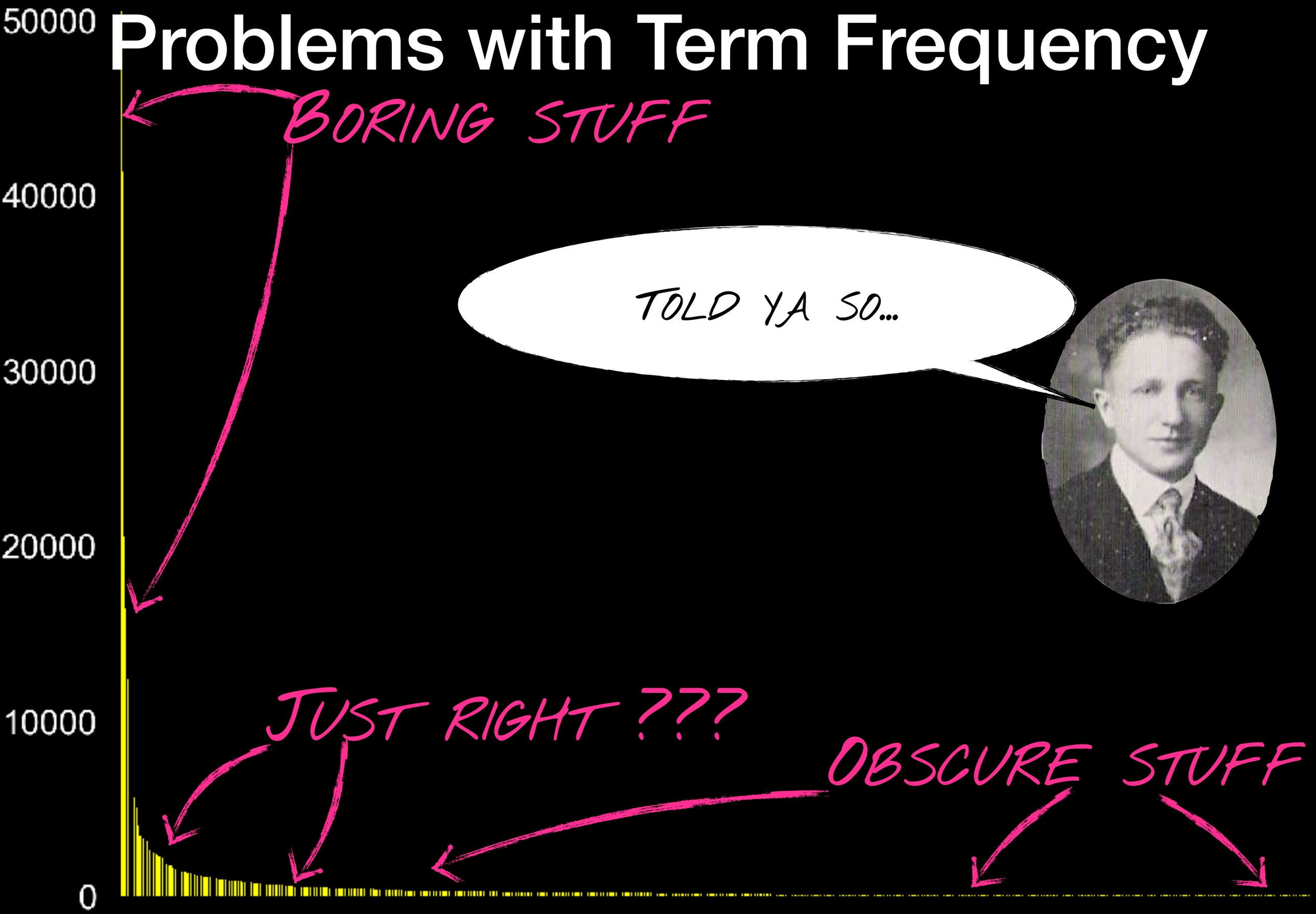
Bocconi

Karen Spärck Jones

1935–2007

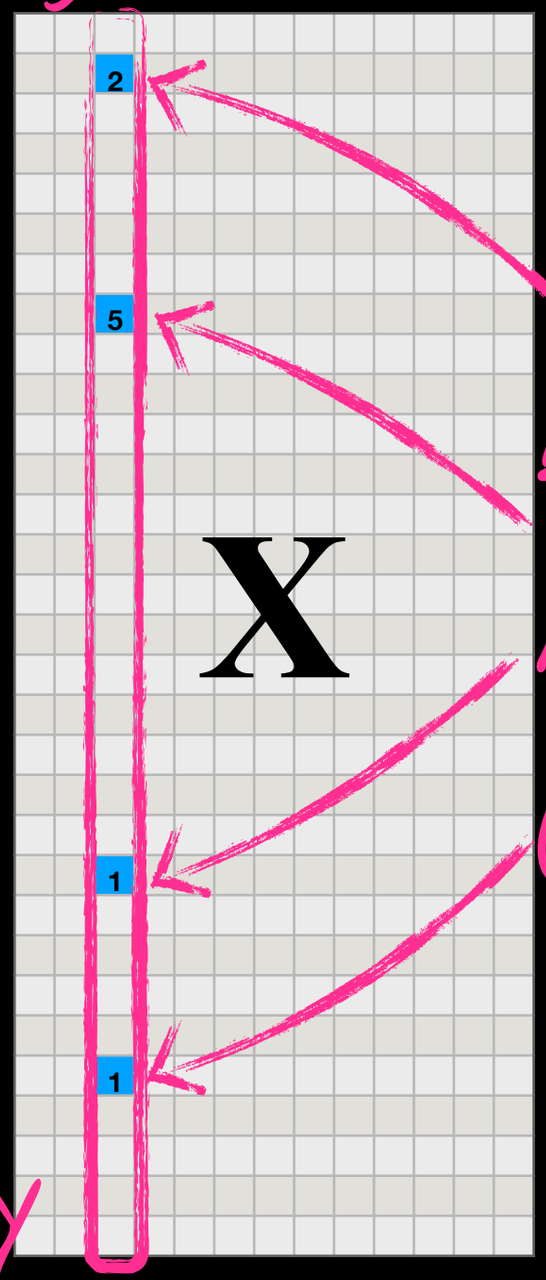
- Became a teacher before starting CS career at Cambridge
- Laid the foundation for modern NLP, Google Search, text classification
- Campaigned for more women in CS
- Namesake of prestigious CS prize

Problems with Term Frequency



Document and Term Frequency

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



Putting it Together

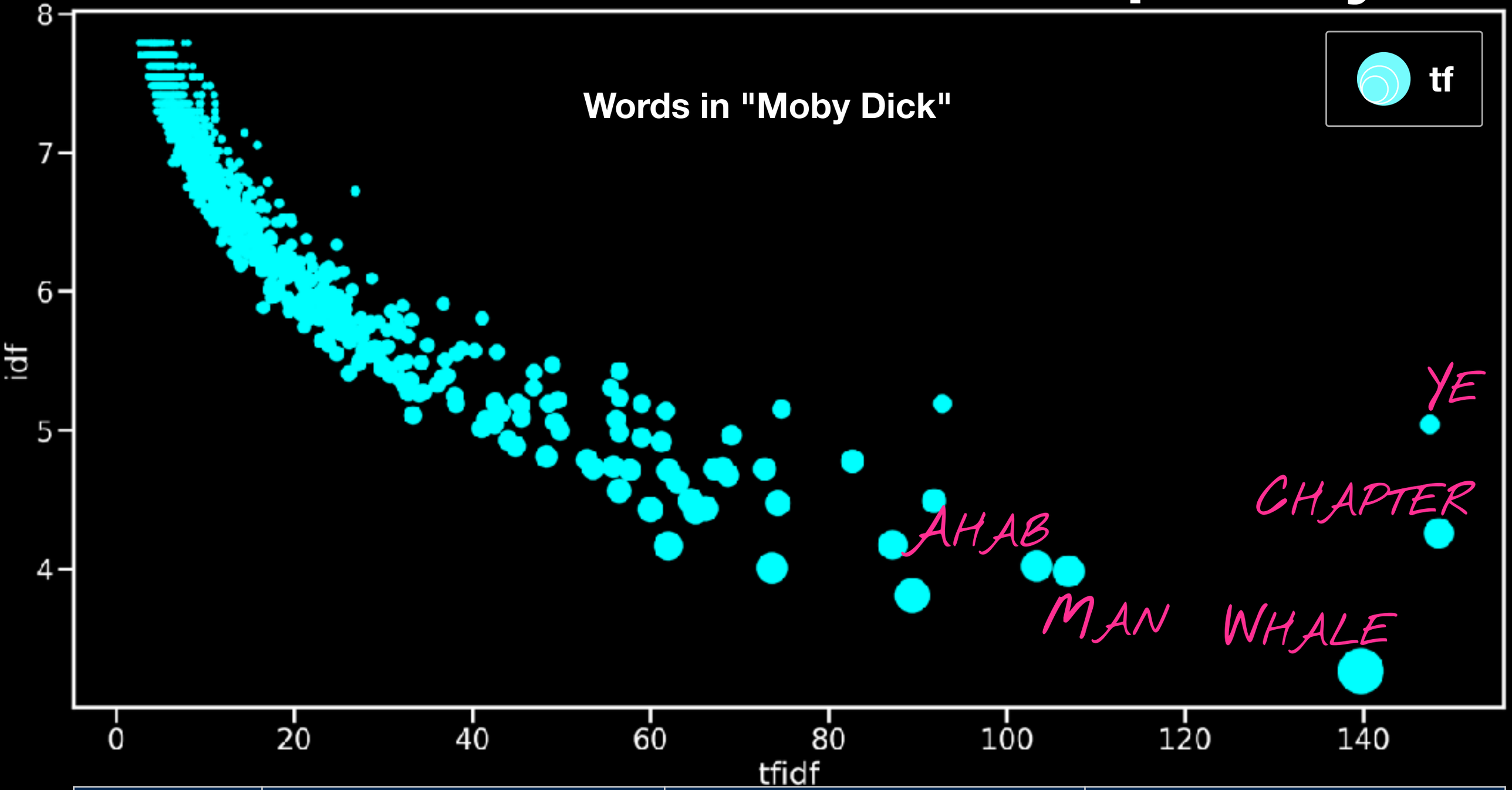
50000
40000
30000
20000
10000
0

How often we saw the word

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

adjusted by how many documents

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

Variants

	TF
binary	*1 if word in D, else 0*
raw	*c(word, D)*
relative	*c(word, D) / len(D)*
smooth	*log(c(word, D) + 1)*

	IDF
regular	$\log \frac{N}{df(word)}$
smooth	$\log \frac{N}{df(word)+1} + 1$

Flexible Matches:
Regular Expressions

Bocconi

http://xkcd.com/208/

The promise...

Is it an (Email) Address?

- notMyFault@webmail.com ✔
- smithie123@gmx ✗
- Free stuff@unibocconi.it ✗
- mark_my_words@hotmail;com ✗
- truthOrDare@webmail.in ✔
- look@me@twitter.com ✗
- how2GetAnts@aol.dfdsfgfdsgfd ✗

Bocconi

Simple Matching

sequence	Matches
e	any single occurrence of e
at	**at**, r**at**, m**at**, s**at**, c**at**, **at**tack, **at**tention, l**at**er

Quantifiers

	Means	Example	Matches
*	0 or more	`cooo*l`	`cool, cooool`
+	1 or more	`hello+`	`hello, helloo, hellooooooo`
?	0 or 1	`fr?og`	`fog, frog`

Special Characters

	Means	Example	Matches
.	any single character	`.el`	`eel, Nel, gel`
\n	newline character (line break)	`\n+`	One or more line breaks
\t	a tab stop	`\t+`	One or more tabs
\d	a single digit [0-9]	`B\d`	`B0, B1, ..., B9`
\D	a non-digit	`\D.t`	`' t, But, eat`
\w	any alphanumberic character	`\w\w\w`	`Top, WOO, ash, bee, ...`
\W	non-alphanumberic character		
\s	a whitespace character		
\S	a non-whitespace character		
****	"Escapes" special characters to match them	`.+` `\.com`	`abc.com, united.com`
^	the beginning of the input string	`^...`	First word in line
$	the end of the input string	`^\n$`	Empty line

Classes

	Means	Example	Matches
[abc]	Match any of a, b, c	`[bcrms]at`	`bat, cat, rat, mat, sat`
[^abc]	Match anything BUT a, b, c	`te[^ ]+s`	`tens, tests, teens, texts, terrors...`
[a-z]	Match any lowercase character	`[a-z][a-z]t`	`act, ant, not, ... wit`
[A-Z]	Match any uppercase character	`[A-Z]...`	`Ahab, Brit, In a, ..., York`
[0-9]	Match any digit	`DIN A[0-9]`	`DIN A0, DIN A1, ..., DIN A9`

Groups

	Means	Example	Matches
(abc)	Match abc	`.(ar).`	`hard, cart, fare, …`
(ab|c)	Match ab OR c	`(ab|c)ate`	`abate, cate`

Matching Addresses


```
^[A-Za-z0-9_\.-]+@[A-Za-z0-9_\.-]+\.[A-Za-z0-9_][A-Za-z0-9_]+$
```

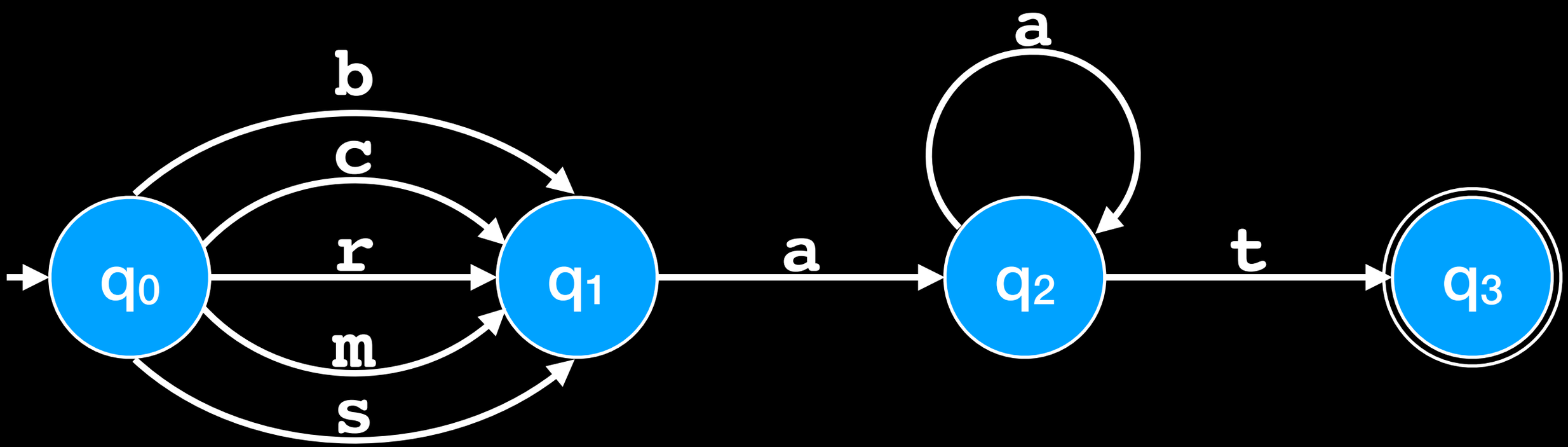

Bocconi

http://xkcd.com/1171/

A (W|w)ord of [Ww]arning

Bocconi

RegEx as Automata




```
[bcrms]a+t
```

Telling Neighbors: Pointwise Mutual Information

Bocconi

Some are not like the Others

frying pan

social media

and thought

New York

of the

FREQUENT BIGRAMS

COLLOCATIONS

Mutual Informativity

How well can we guess the Blank?

social _____ and _____

_____ media _____ the

Pointwise Mutual Information

CHANCE OF SEEING THEM TOGETHER

$$PMI(x, y) = log \frac{P(x, y)}{P(x)P(y)}$$

...SEEING EITHER

x	y	c(x)	c(y)	c(xy)	P(x)	P(y)	P(x, y)	PMI(x; y)
moby	**dick**	83	83	82	0.0003	0.0003	0.0003	**3.48**
captain	**ahab**	327	511	61	0.0013	0.0020	0.0002	**1.97**
white	**whale**	280	1150	106	0.0011	0.0045	0.0004	**1.93**
under	**the**	119	14175	45	0.0005	0.0553	0.0002	**0.83**
is	**a**	1690	4636	110	0.0066	0.0181	0.0004	**0.56**

c(X) = 256,149
c(XY) = 256,148

Wrapping up

Bocconi

Take home points

- **TF-IDF** finds "bursty" words: medium frequency overall, but concentrated in few documents
- **Regular expressions** allow us to search for flexible patterns
- **PMI** tells us how likely one word is to occur with/without another to find **collocations**