Natural Language Processing

github.com/adarsh0806/ODSC

Natural Language Processing

- Introduction to NLP
- Text Processing
- Basic NLP
- Advanced NLP

Introduction to NLP

Introduction to NLP

- Communication and Cognition
- Structured Languages
- Unstructured Text
- Applications and Challenges

Communication and Cognition

Language is…

- a medium of communication
- a vehicle for thinking and reasoning

Structured Languages

- Natural language lacks precisely defined structure

Structured Languages

- Mathematics:

$$y = 2x + 5$$

Structured Languages

- Formal Logic:

$$\text{Parent}(x, y) \land \text{Parent}(x, z) \rightarrow \text{Sibling}(y, z)$$

Structured Languages

- SQL:

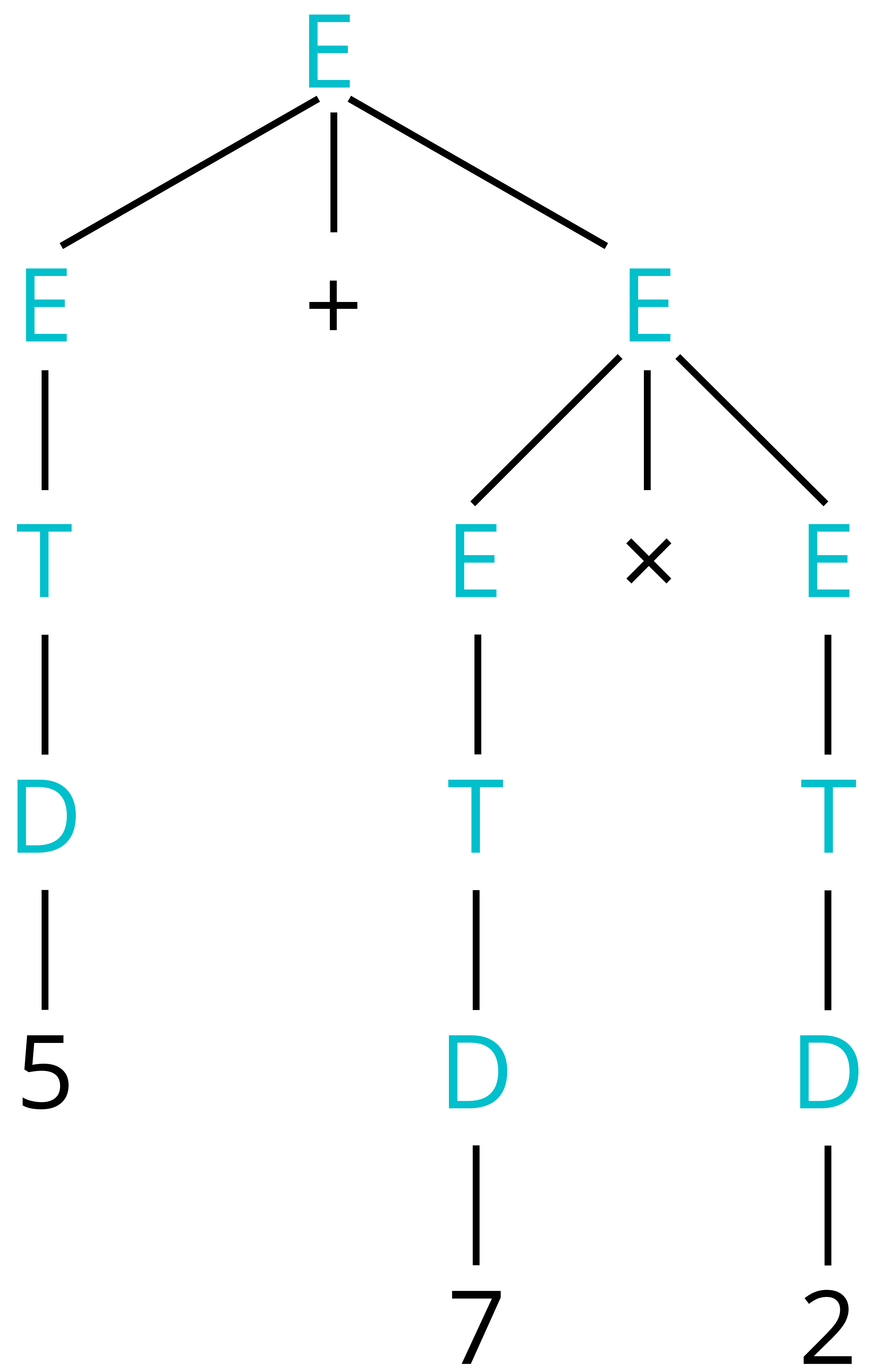
```
SELECT name, email
  FROM users
  WHERE name LIKE 'A%';
```

Grammar

- Arithmetic (single digit):

  $E \rightarrow E+E \mid E-E \mid E\times E \mid E\div E \mid (E) \mid D$

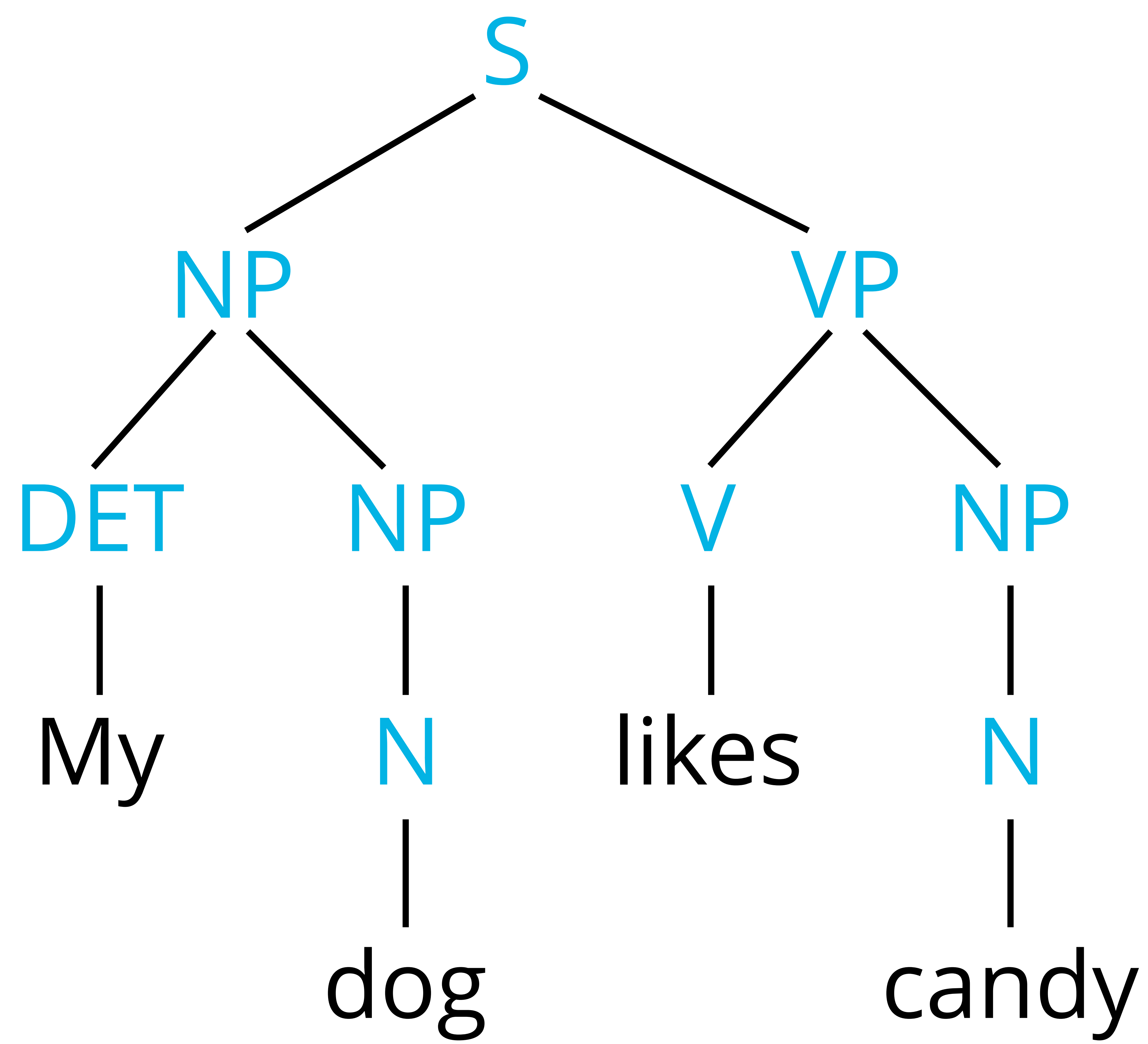
  $D \rightarrow 0 \mid 1 \mid 2 \mid \ldots \mid 9$



Grammar

- English sentences (limited):

  S → NP VP
  NP → N | DET NP | ADJ NP
  VP → V | V NP
  …



noun

"Because he was so small, Stuart was often hard to find around the house."

verb

– *Stuart Little*, E.B. White

Unstructured Text

the quick brown fox jumps over the lazy dog

Unstructured Text

jumps

the

fox

brown

over

dog

lazy

quick

the

Unstructured Text

jumps

the

fox

brown

over

dog

+

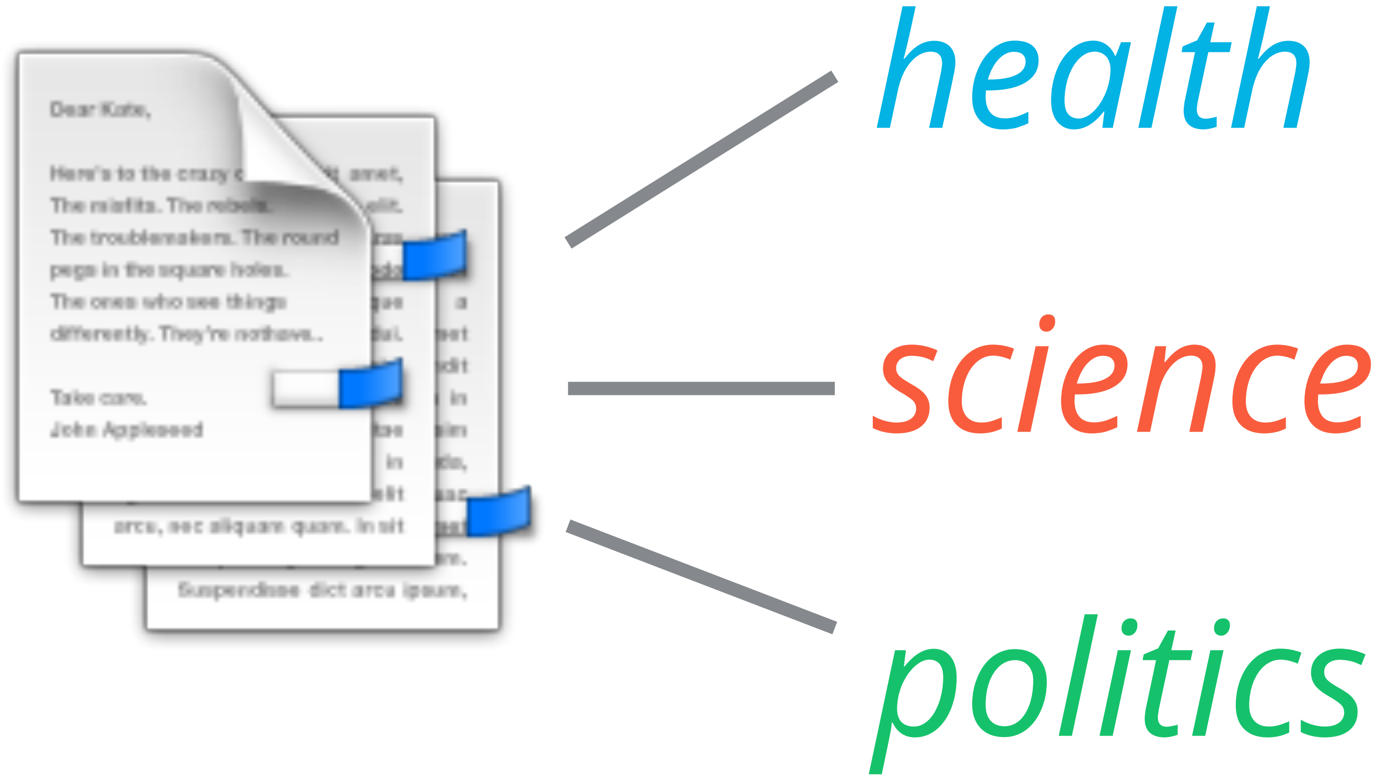
quick

lazy

-

the

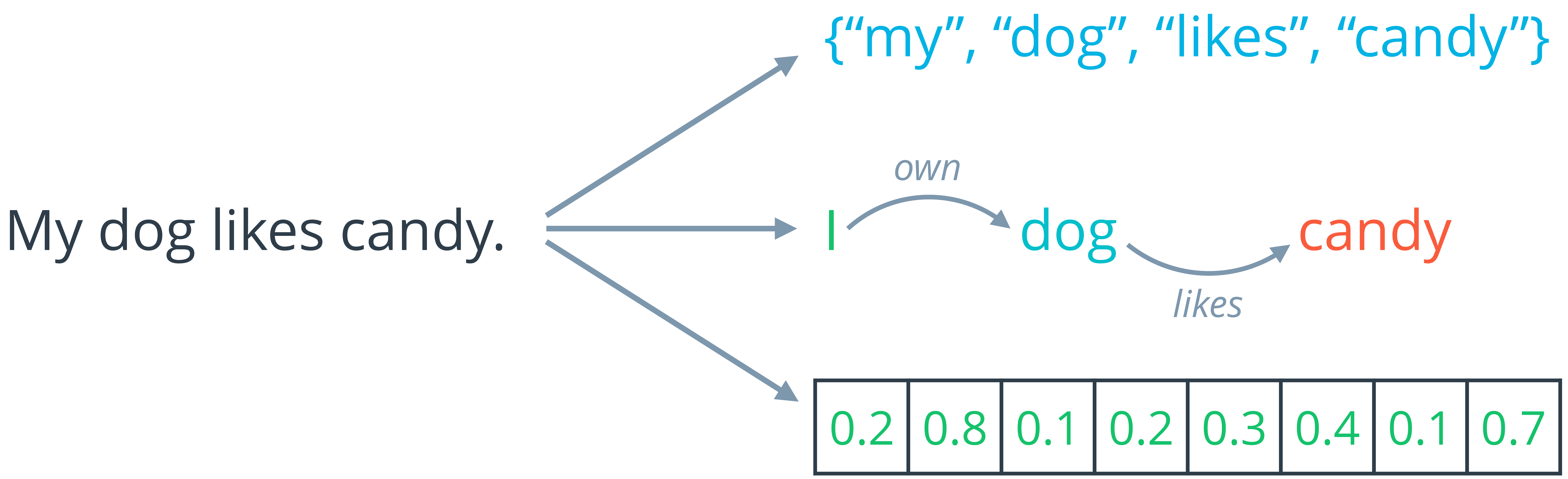
Applications



what time is it?

¿que hora es?

Challenges: Representation



Challenges: Temporal Sequence

I want to buy a gallon of milk

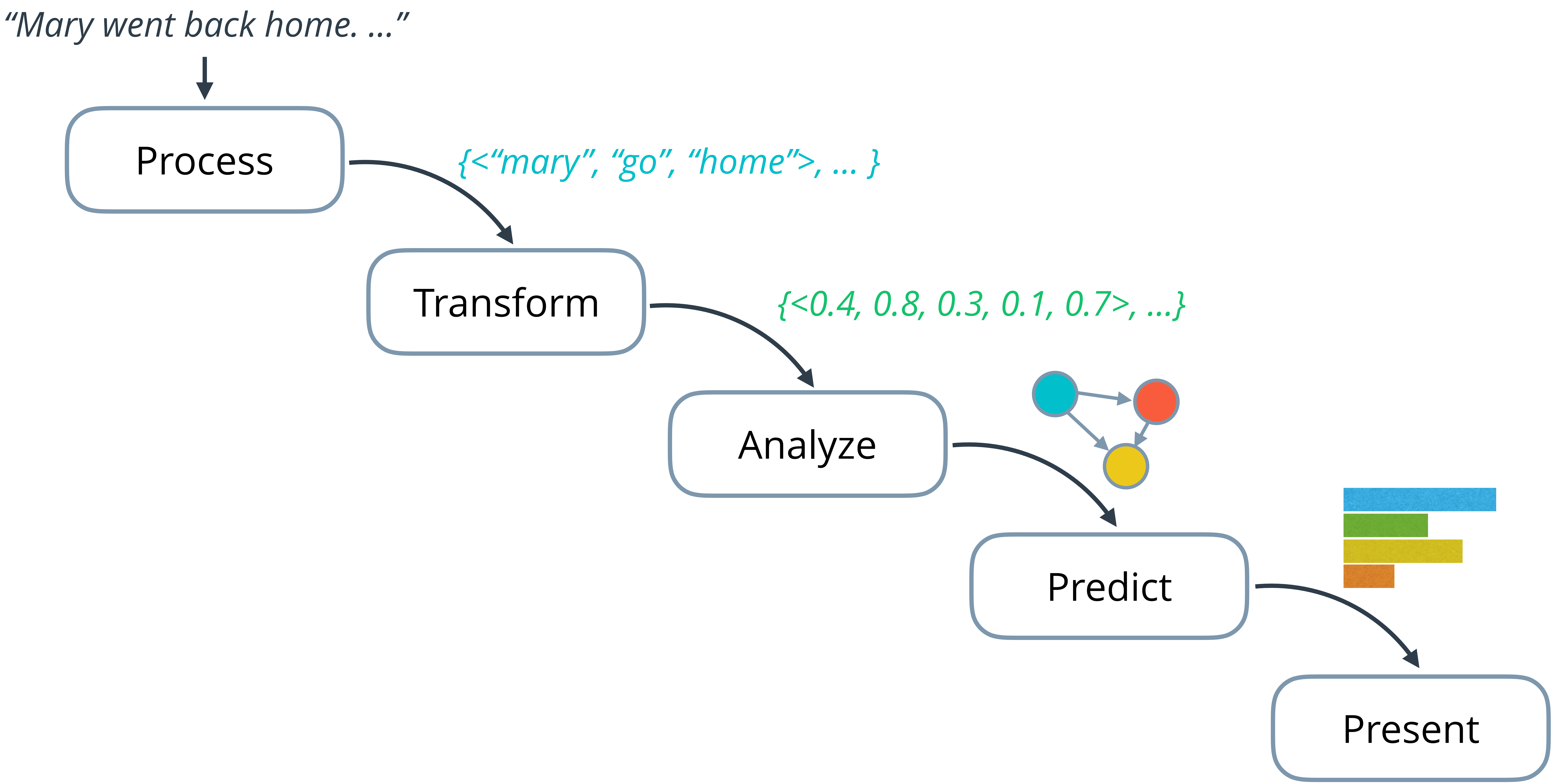
water

petrol

Challenges: Context

The old Welshman came home toward daylight, spattered with candle-grease, smeared with clay, and almost worn out. He found Huck still in the bed that had been provided for him, and delirious with fever. The physicians were all at the cave, so the Widow Douglas came and took charge of the patient.

—*The Adventures of Tom Sawyer*, Mark Twain


“Mary went back home. …”
Process
{<“mary”, “go”, “home”>, … }
Transform
{<0.4, 0.8, 0.3, 0.1, 0.7>, …}
Analyze
Predict
Present

Text Processing

Text Processing

- Tokenization
- Stop Word Removal
- Stemming and Lemmatization

Tokenization

“Jack and Jill went up the hill” → <“jack”, “and”, “jill”,

“went”, “up”, “the”, “hill”>

Tokenization

“No, she didn’t do it.”

<“no,”, “she”, “didn”,
“’”, “t”, “do” “it”, “.”>

<“no”, “she”, “didnt”,
“do”, “it”>

?

Tokenization

Big money behind big special effects tends to suggest a big story. Nope, not here. Instead this huge edifice is like one of those over huge luxury condos that're empty in every American town, pretending as if there's a local economy huge enough to support such.

—Rotten Tomatoes

<"big", "money", "behind", "big", "special", "effects", "tends", "to", "suggest", "big", "story", "nope", "not", "here", "instead", "this", "huge", "edifice", "is", "like", "one", "of", "those", "over", "huge", "luxury", "condos", "that", "re", "empty", "in", "every", "american", "town", "pretending", "as", "if", "there", "local", "economy", "huge", "enough", "to", "support", "such">

<"big", "money", "behind", "big", "special", "effects", "tends", "to", "suggest", "big", "story">

<"nope", "not", "here">

<"instead", "this", "huge", "edifice", "is", "like", "one", "of", "those", "over", "huge", "luxury", "condos", "that", "re", "empty", "in", "every", "american", "town", "pretending", "as", "if", "there", "local", "economy", "huge", "enough", "to", "support", "such">

?

Stop Word Removal

wristwatch invented 1904 Louis Cartier.

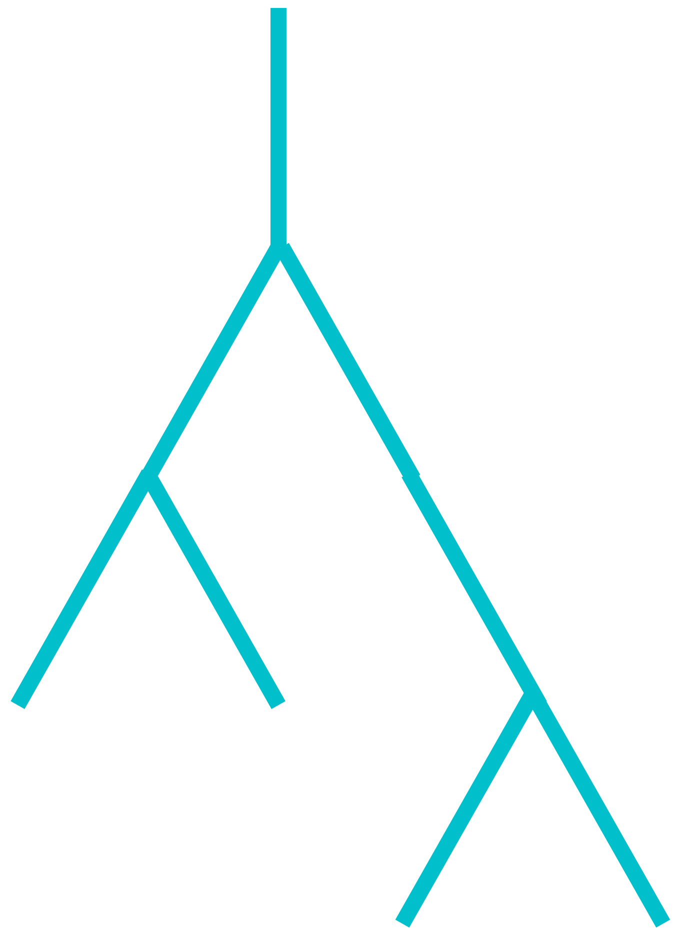
Stemming

branch~~ing~~

branch~~ed~~

branch~~es~~

branch

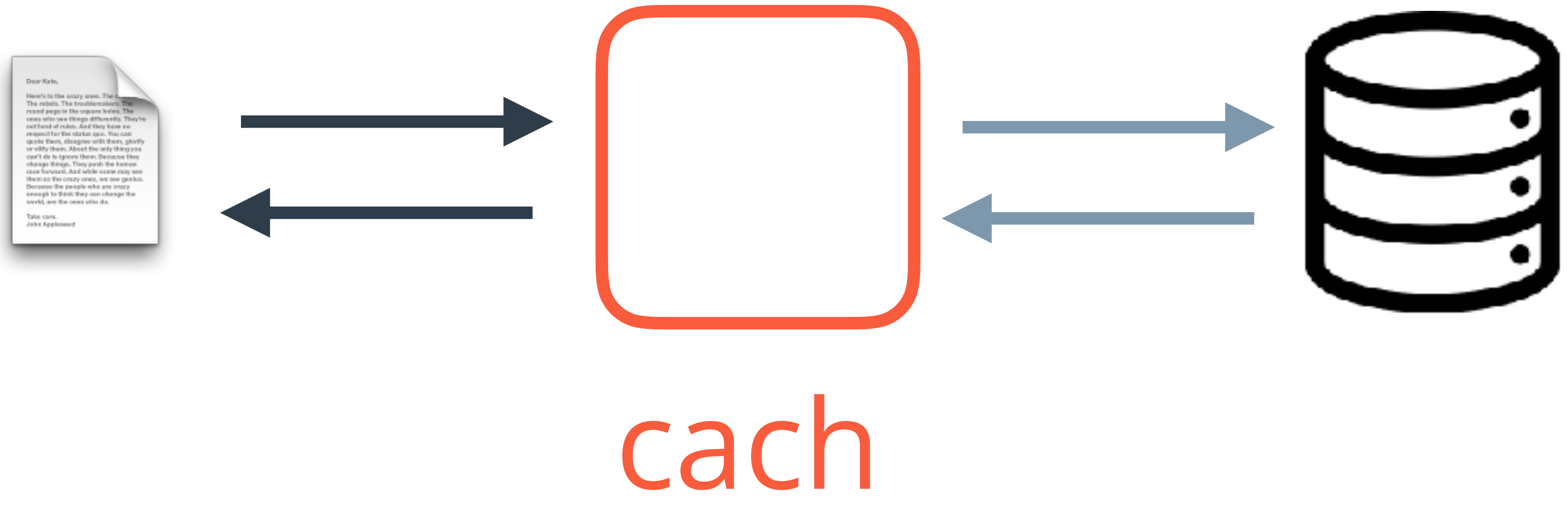


Stemming

cach~~ing~~

cach~~ed~~

cach~~es~~



Lemmatization

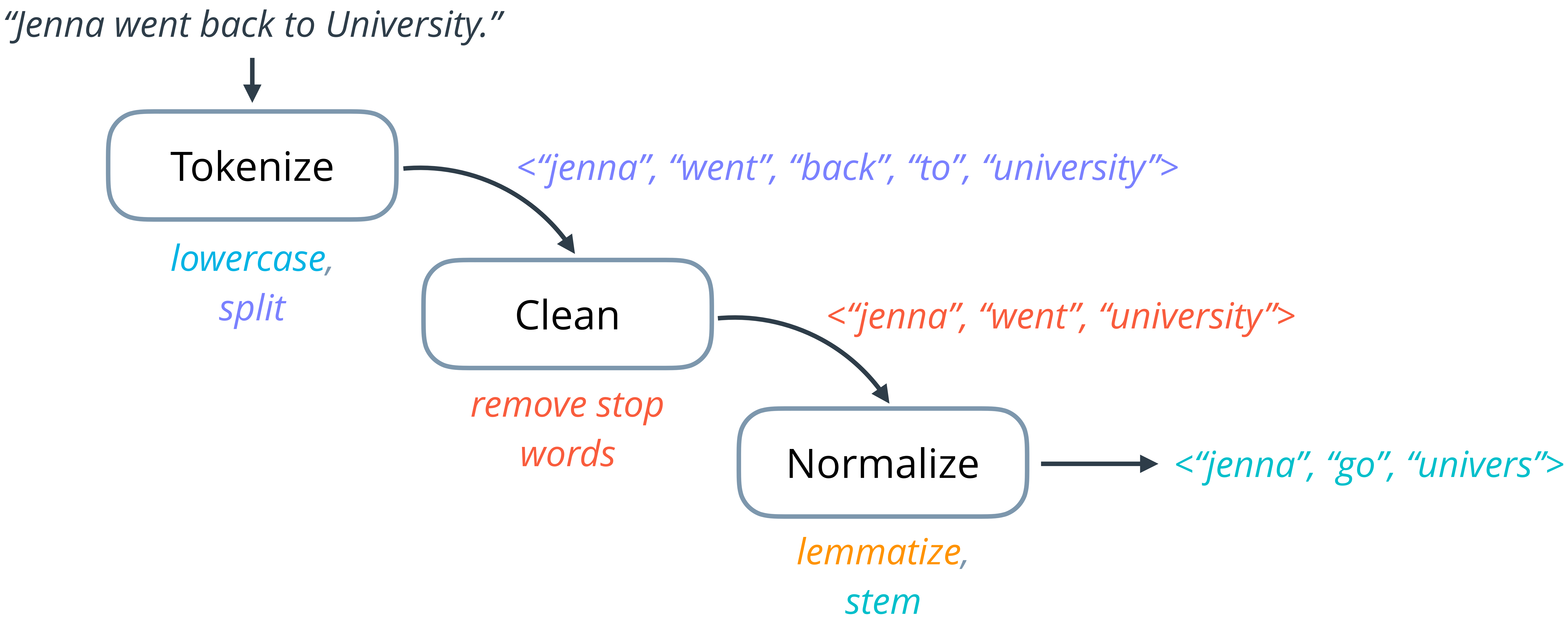
is

was

were

be

Text Processing Summary

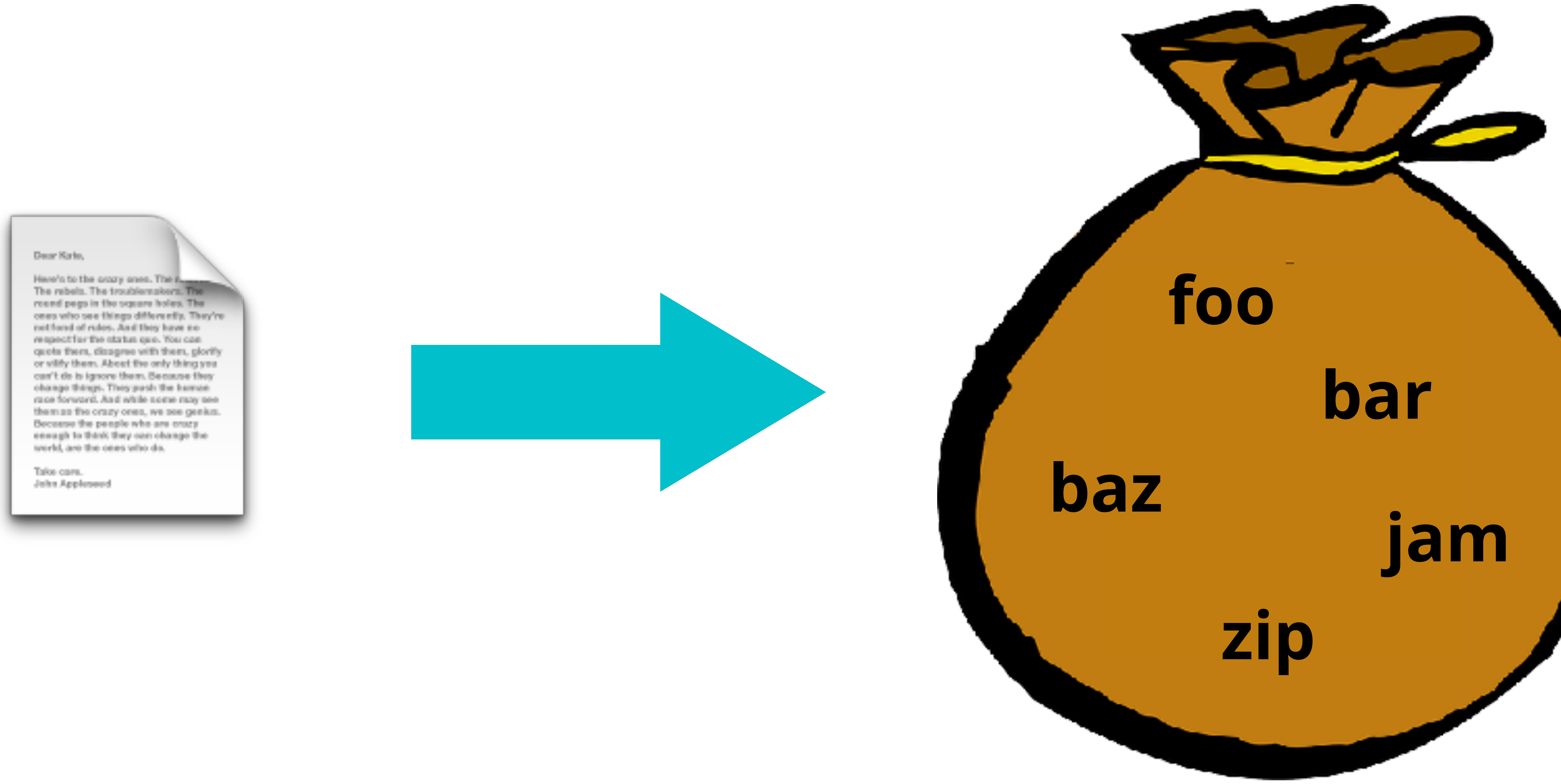


Basic NLP

Basic NLP

- Bag of Words Representation
- Document-Term Matrix
- Task: Document Classification

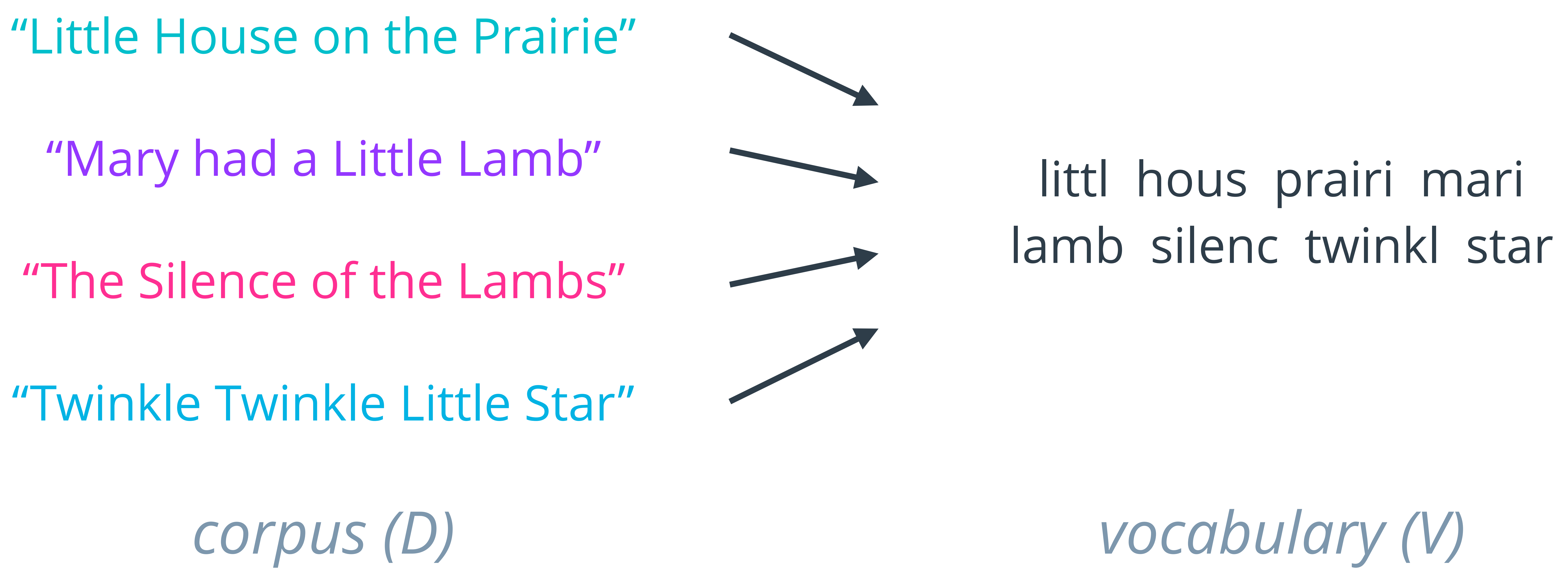
Bag of Words



Bag of Words

“Little House on the Prairie”	→	{“littl”, “hous”, “prairi”}
“Mary had a Little Lamb”	→	{“mari”, “littl”, “lamb”}
“The Silence of the Lambs”	→	{“silenc”, “lamb”}
“Twinkle Twinkle Little Star”	→	{“twinkl”, “littl”, “star”} ?

Bag of Words



"Little House on the Prairie"
"Mary had a Little Lamb"
"The Silence of the Lambs"
"Twinkle Twinkle Little Star"
littl hous prairi mari
lamb silenc twinkl star
corpus (D)
vocabulary (V)

Bag of Words

	littl	hous	prairi	mari	lamb	silenc	twinkl	star
"Little House on the Prairie"								
"Mary had a Little Lamb"								
"The Silence of the Lambs"								
"Twinkle Twinkle Little Star"								

Bag of Words
Document-Term Matrix

term frequency

	littl	hous	prairi	mari	lamb	silenc	twinkl	star
“Little House on the Prairie”	1	1	1	0	0	0	0	0
“Mary had a Little Lamb”	1	0	0	1	1	0	0	0
“The Silence of the Lambs”	0	0	0	0	1	1	0	0
“Twinkle Twinkle Little Star”	1	0	0	0	0	0	2	1

Document Similarity

	littl	hous	prairi	mari	lamb	silenc	twinkl	star
a "Little House on the Prairie"	1	1	1	0	0	0	0	0
b "Mary had a Little Lamb"	1	0	0	1	1	0	0	0

$$\mathbf{a}\cdot\mathbf{b} = \sum a_0 b_0 + a_1 b_1 + \ldots + a_n b_n = 1 + 0 + 0 + 0 + 0 + 0 + 0 + 0$$

dot product

Document Similarity

	littl	hous	prairi	mari	lamb	silenc	twinkl	star
a “Little House on the Prairie”	1	1	1	0	0	0	0	0
b “Mary had a Little Lamb”	1	0	0	1	1	0	0	0

$$\mathbf{a}\cdot\mathbf{b} = \sum a_0 b_0 + a_1 b_1 + \ldots + a_n b_n = 1$$

dot product

$$\cos(\theta) = \frac{\mathbf{a}\cdot\mathbf{b}}{\|\mathbf{a}\|\cdot\|\mathbf{b}\|} = \frac{1}{\sqrt{3}\times\sqrt{3}} = \frac{1}{3}$$

cosine similarity

Term Specificity

	littl	hous	prairi	mari	lamb	silenc	twinkl	star
"Little House on the Prairie"	1/3	1/1	1/1	0/1	0/2	0/1	0/1	0/1
"Mary had a Little Lamb"	1/3	0/1	0/1	1/1	1/2	0/1	0/1	0/1
"The Silence of the Lambs"	0/3	0/1	0/1	0/1	1/2	1/1	0/1	0/1
"Twinkle Twinkle Little Star"	1/3	0/1	0/1	0/1	0/2	0/1	2/1	1/1
document frequency	3	1	1	1	2	1	1	1

Term Specificity

	littl	hous	prairi	mari	lamb	silenc	twinkl	star
“Little House on the Prairie”	1/3	1	1	0	0	0	0	0
“Mary had a Little Lamb”	1/3	0	0	1	1/2	0	0	0
“The Silence of the Lambs”	0	0	0	0	1/2	1	0	0
“Twinkle Twinkle Little Star”	1/3	0	0	0	0	0	2	1

TF-IDF

$$\mathrm{tfidf}(t, d, D) = \mathrm{tf}(t, d) \cdot \mathrm{idf}(t, D)$$

term frequency

$$\mathrm{count}(t, d) / |\mathrm{d}|$$

inverse document frequency

$$\log(|D| / |\{\mathrm{d} \in \mathrm{D} : \mathrm{t} \in \mathrm{d}\}|)$$

Task: Document Classification

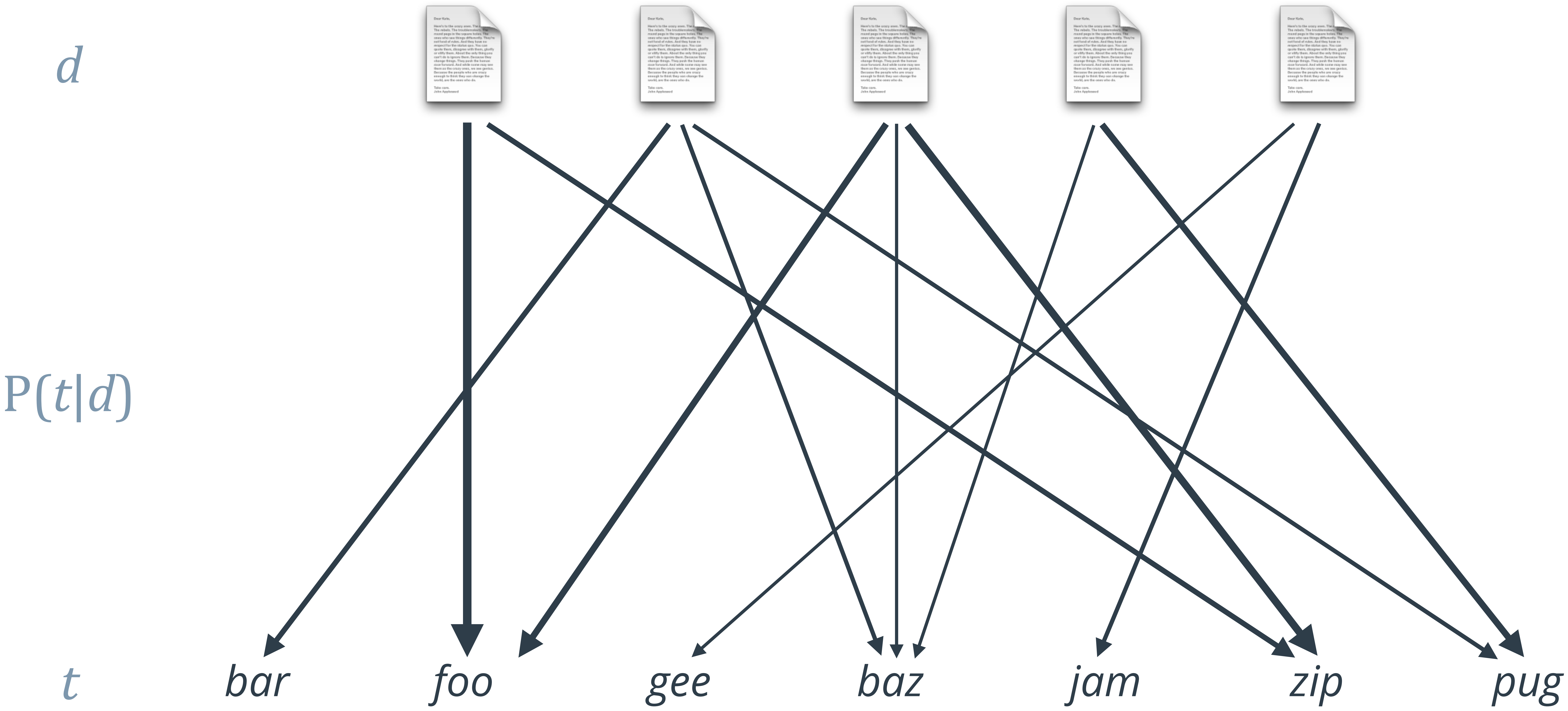
Spam Detection

Advanced NLP

Advanced NLP

- Latent Variables
- Task: Topic Modeling
- Word Embeddings

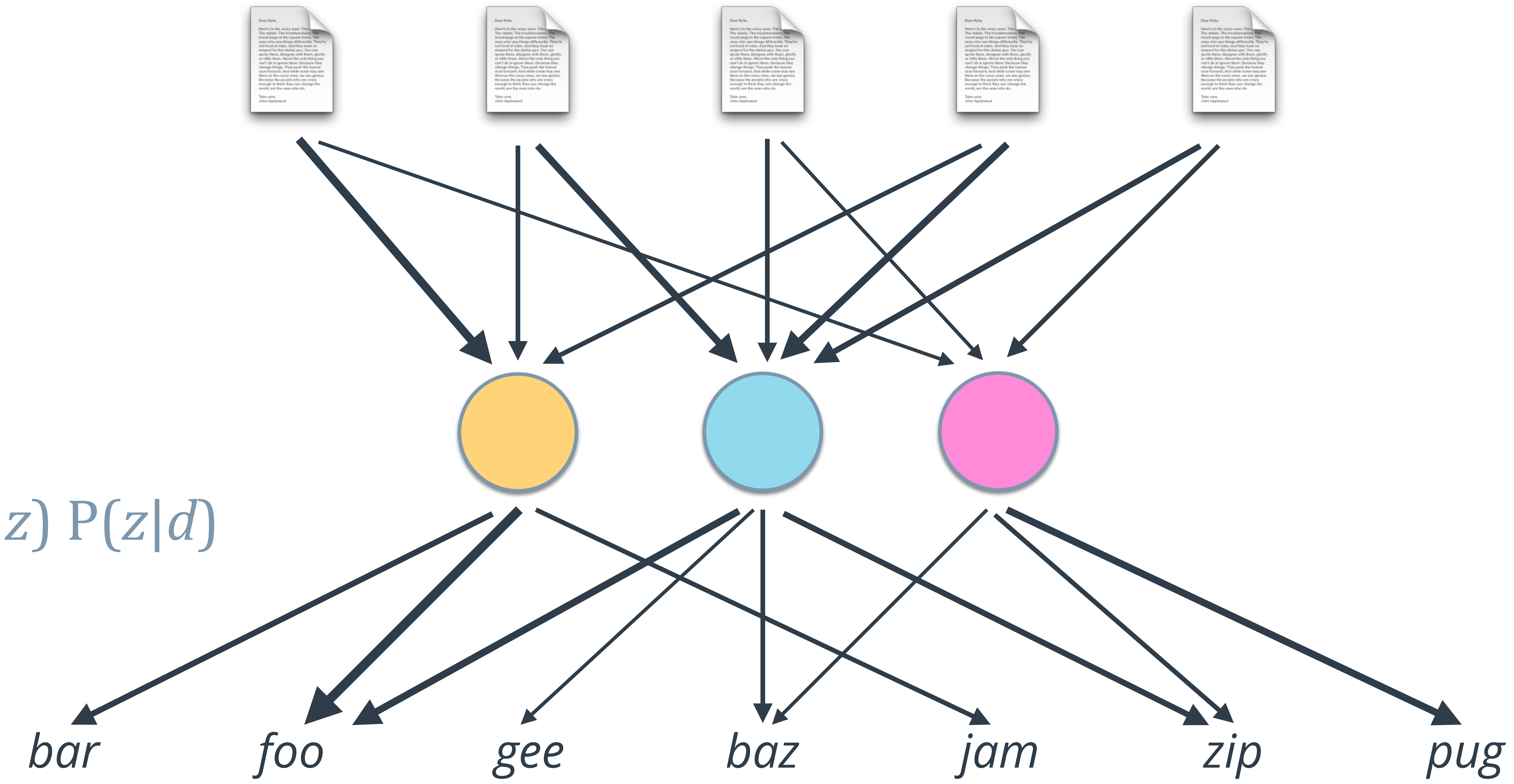
Bag of Words: Graphical Model



Latent Variables

d

$P(z|d)$



z

$$P(t|d) = \sum_z P(t|z)\, P(z|d)$$

$P(t|z)$

t

Missing Priors

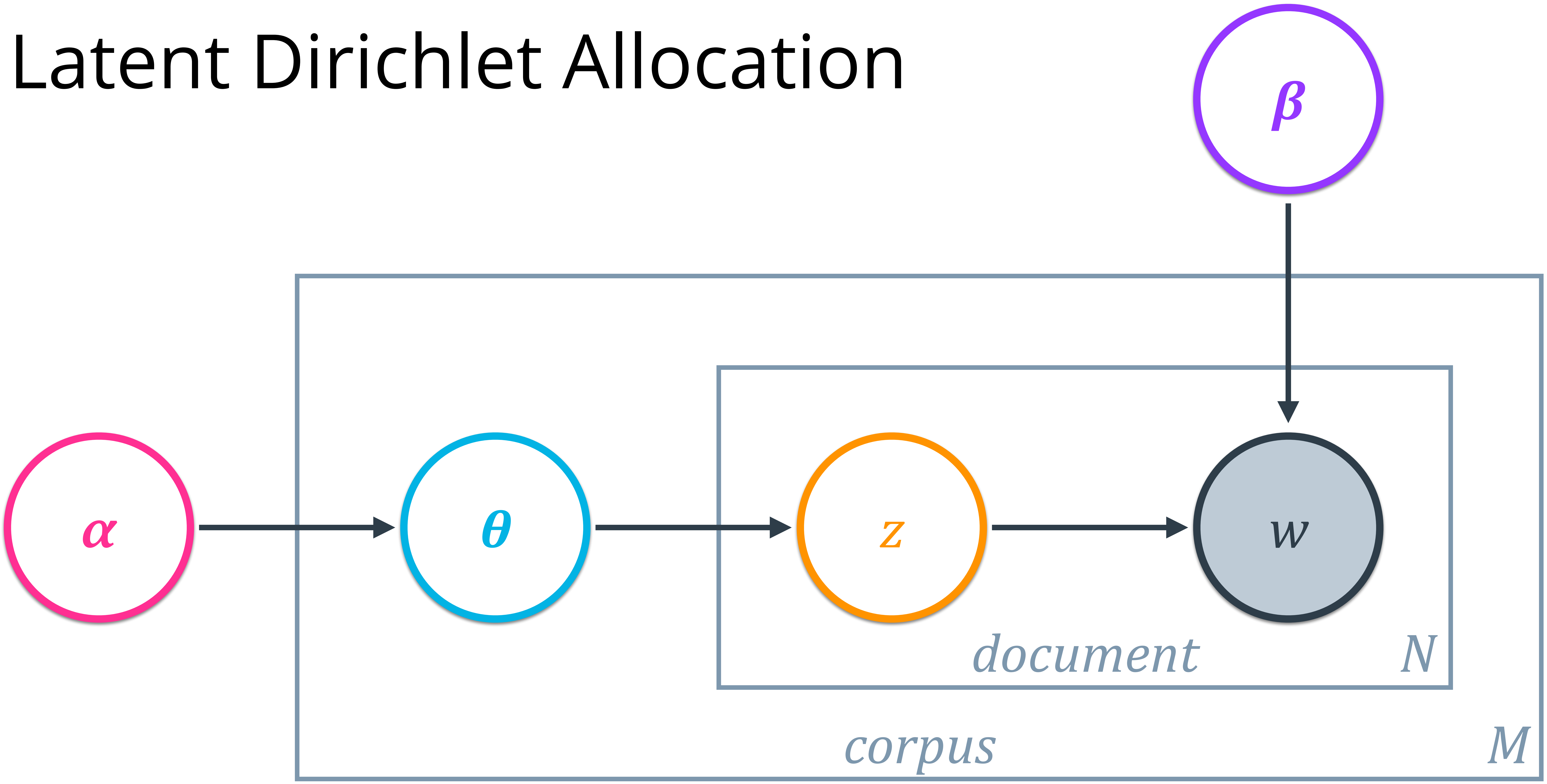
$\mathrm{P}(t|d) = \sum_z \mathrm{P}(t|z)\, \mathrm{P}(z|d)$ *conditional probabilities*

$\mathrm{P}(t, d) = ?$ $\mathrm{P}(t, z) = ?$ *joint probabilities*

$\mathrm{P}(d)$ $\mathrm{P}(\mathrm{z})$ *prior probabilities*

α β *Dirichlet distributions*

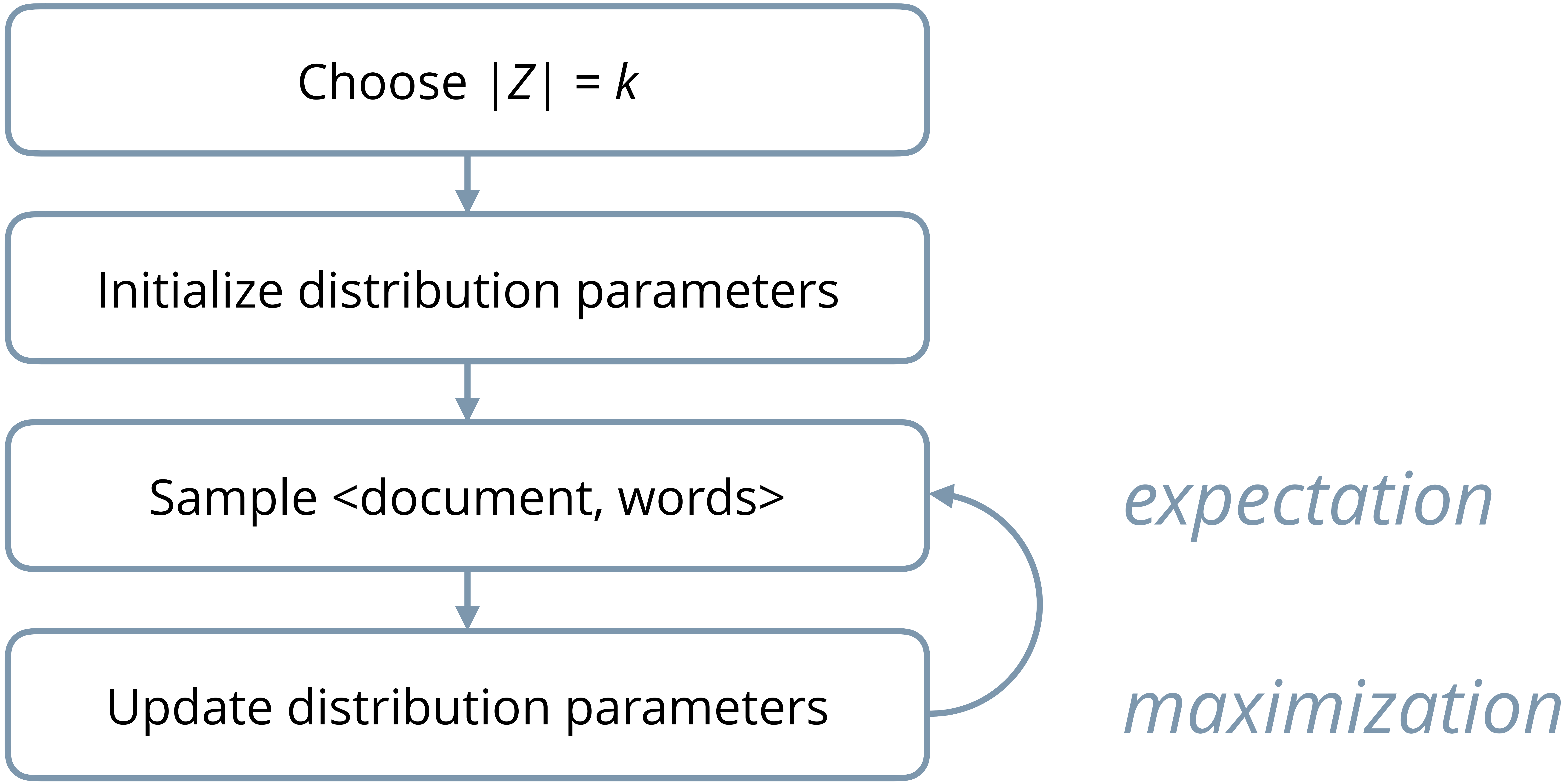
Latent Dirichlet Allocation



LDA: Use Cases

- Topic modeling, document categorization.
- Mixture of topics in a new document: $\mathrm{P}(\boldsymbol{z} \mid \boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\beta})$
- Generate collections of words with desired mixture.

LDA: Parameter Estimation



Task: Topic Modeling

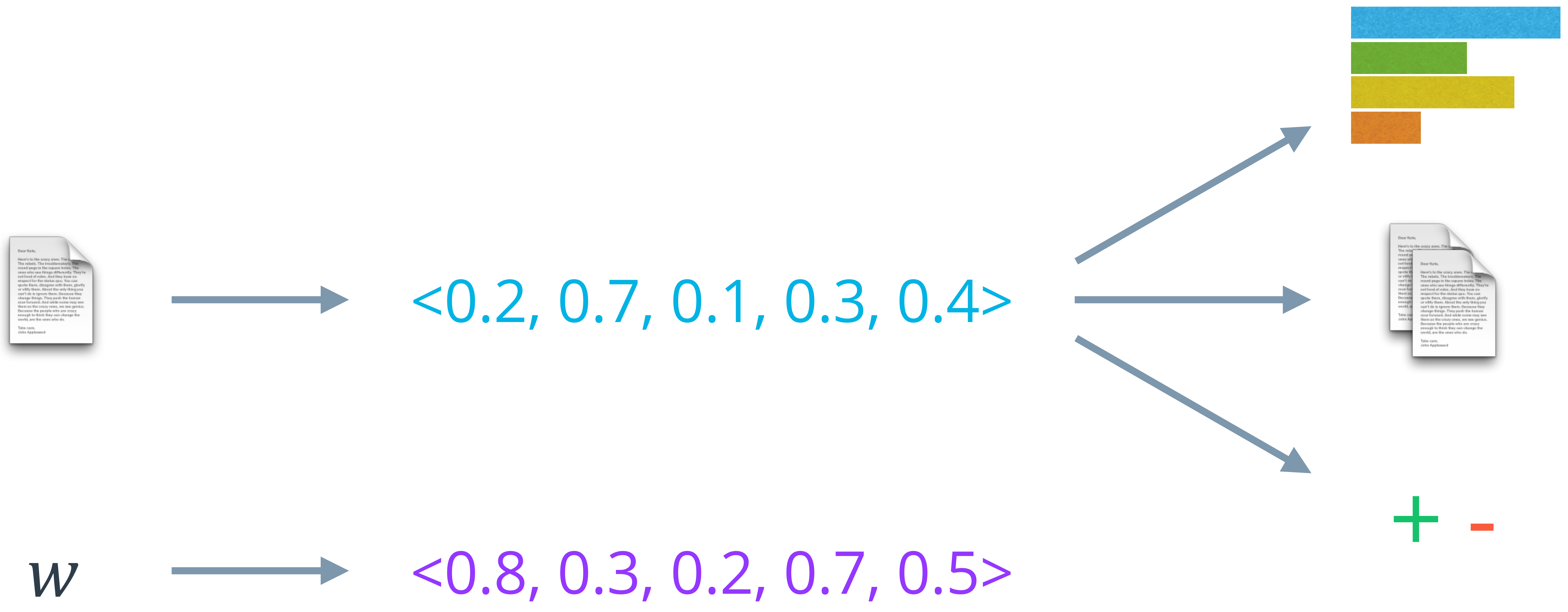
Categorize Newsgroups Data

LDA: Further Reading

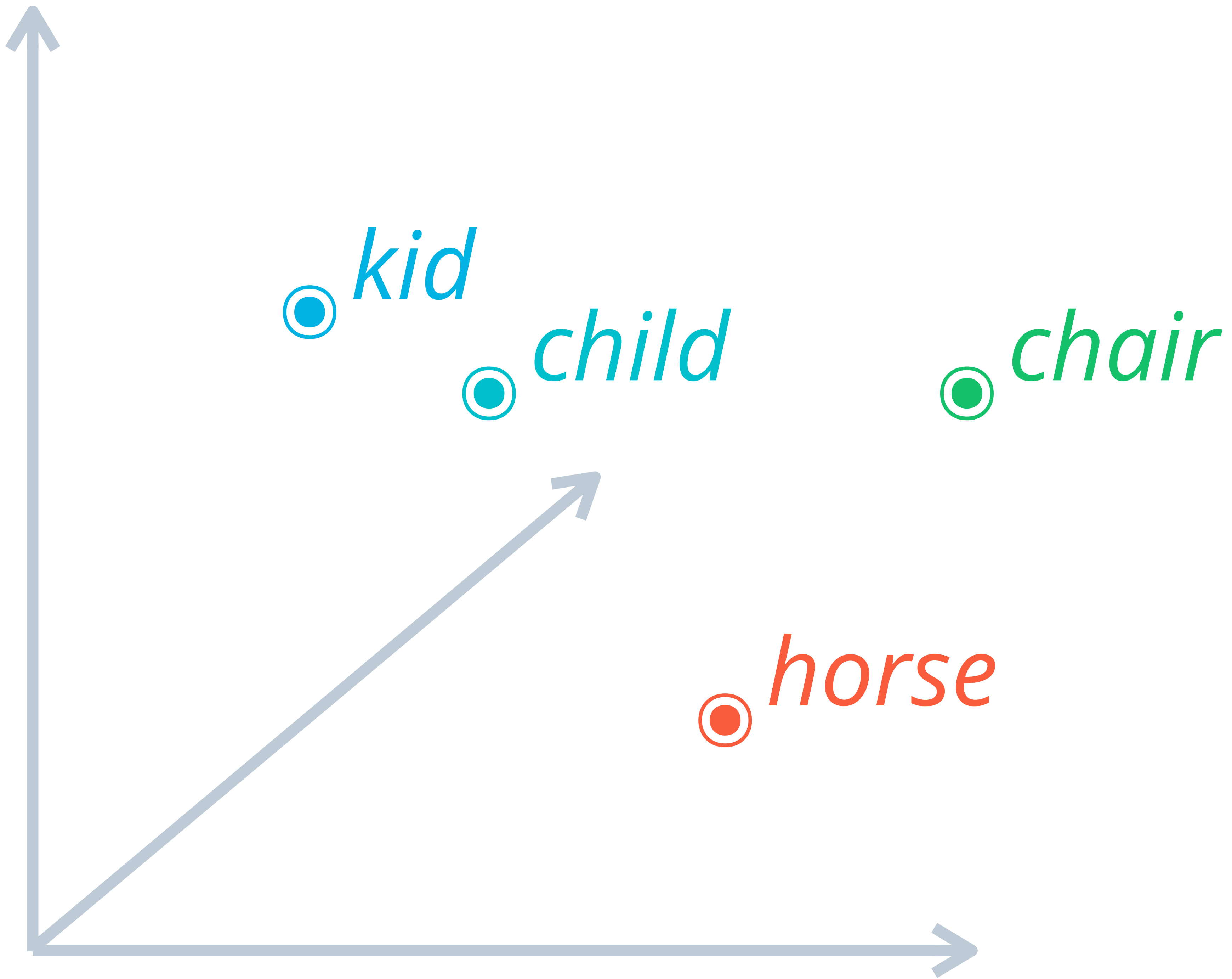
David Blei, Andrew Ng, Michael Jordan, 2003. Latent Dirichlet Allocation, In *Journal of Machine Learning Research*, vol. 3, pp. 993-102.

Thomas Boggs, 2014. Visualizing Dirichlet Distributions with matplotlib.

Document vs. Word Representations



Word Embeddings



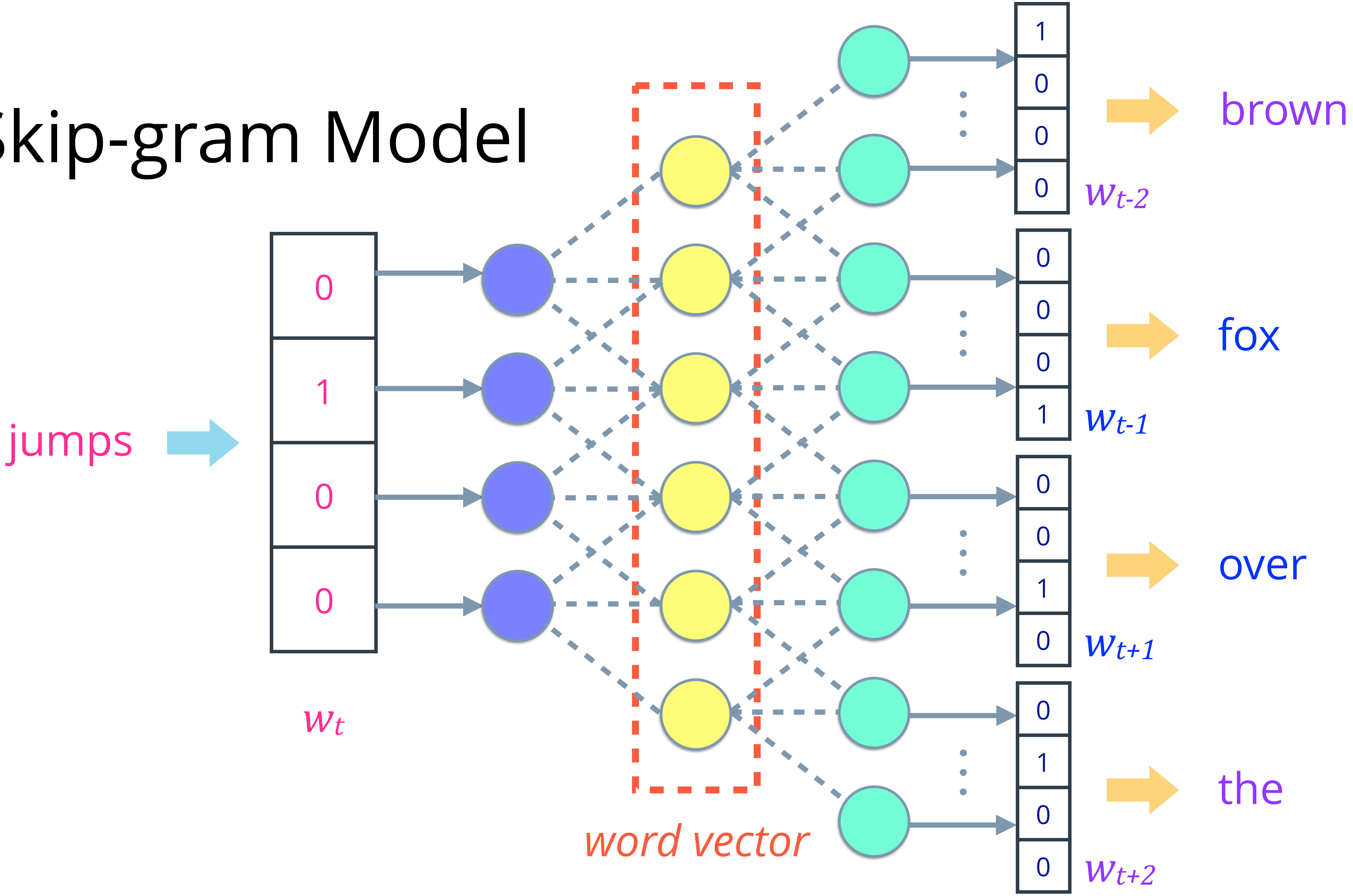
Word2Vec

Continuous Bag of Words (CBoW)

the quick brown fox **jumps** over the lazy dog

Continuous Skip-gram

Skip-gram Model



Word2Vec: Recap

- Robust, distributed representation.
- Vector size independent of vocabulary.
- Deep learning ready!

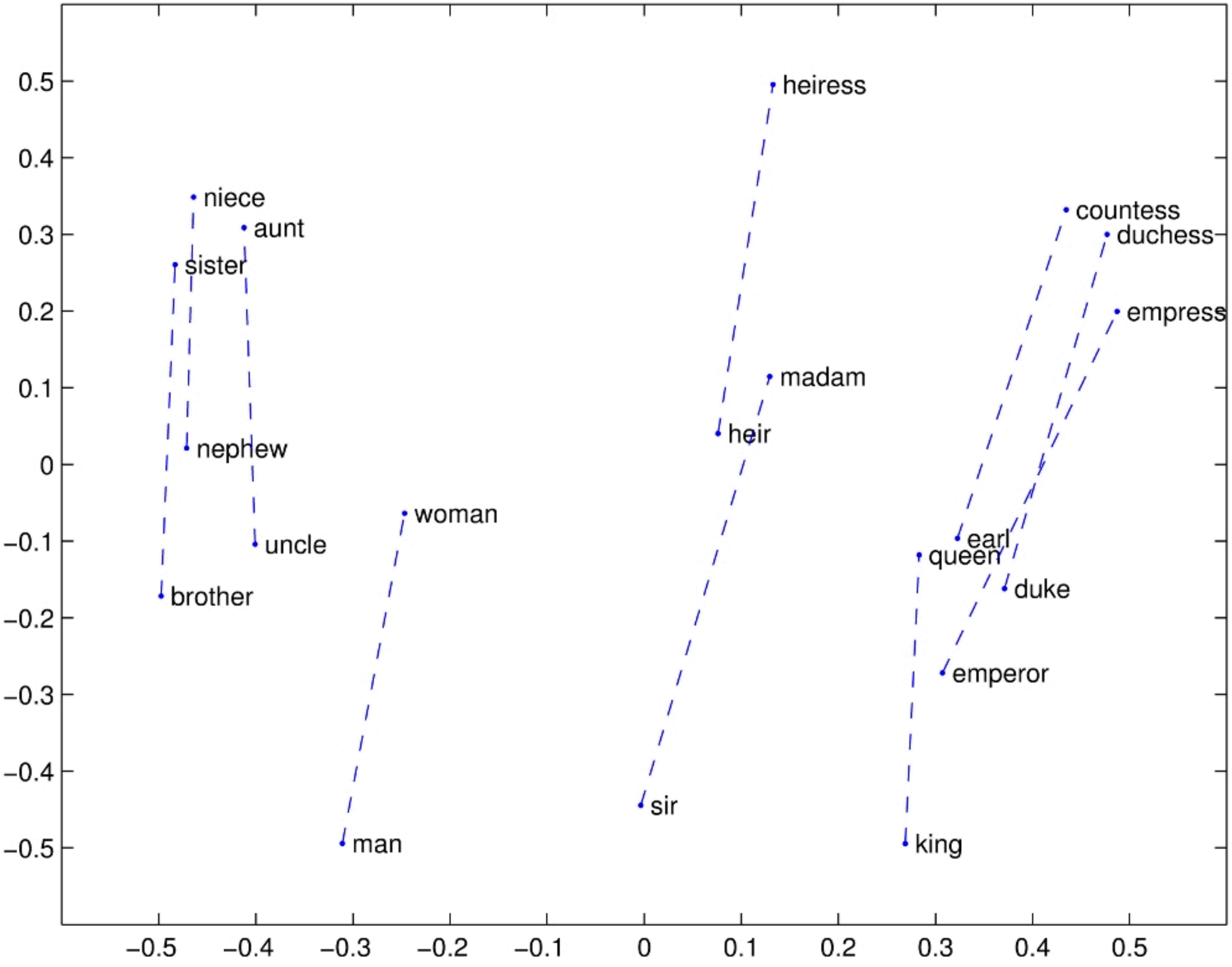
Word2Vec: Further Reading

Tomas Mikolov, et al., 2013. Distributed Representation of Words and Phrases and their Compositionality, In *Advances of Neural Information Processing Systems (NIPS)*, pp. 3111-3119.

Adrian Colyer, 2016. The amazing power of word vectors.

More Word Embeddings

- GloVe: Global Vectors for Word Representation
- t-SNE: t-Distributed Stochastic Neighbor Embedding



0.5
0.4
0.3
0.2
0.1
0
−0.1
−0.2
−0.3
−0.4
−0.5
−0.5
−0.4
−0.3
−0.2
−0.1
0
0.1
0.2
0.3
0.4
0.5
heiress
niece
aunt
sister
countess
duchess
empress
madam
heir
nephew
woman
uncle
earl
queen
duke
brother
emperor
sir
man
king

Workshop Summary

Text Processing

- Stop word removal, stemming, lemmatization

Basic NLP

- Bag-of-Words, TF-IDF, document classification

Advanced NLP

- LDA, topic modeling, word embeddings

What’s Next?

- Recurrent Neural Networks (RNNs)
- Long Short-Term Memory Networks (LSTMs)
- Visual Question-Answering

Adarsh Nair

Luis Serrano

Arpan Chakraborty

udacity.com/ai
udacity.com/ml