TOPIC MODELING

By: METIS

METIS

AGENDA

- Topic Modeling Overview
- Matrix Factorization
  - Latent Semantic Analysis (LSA)
  - Non-Negative Matrix Factorization (NMF)
- Probabilistic Modeling
  - Latent Dirichlet Allocation (LDA)

TOPIC MODELING OVERVIEW

TOPIC MODELING

- The process of discovering “topics” that occur in a collection of documents

- Let’s go through two examples:
  1. Understand the concept of a “topic”
  2. Understand how this is a form of dimensionality reduction

Example #1: Topics

LET’S SAY WE HAVE 5 DOCUMENTS

- I like bananas and oranges.
- Frogs and fish live in ponds.
- Kittens and puppies are fluffy.
- For breakfast, I had a spinach and apple smoothie.
- My kitten loves kale.

Intuitively, what are the topics that you see here?

EXAMPLE TOPIC MODELING OUTPUT

- I like bananas and oranges. **100% Topic A**
- Frogs and fish live in ponds. **100% Topic B**
- Kittens and puppies are fluffy. **100% Topic B**
- For breakfast, I had a spinach and apple smoothie. **100% Topic A**
- My kitten loves kale. **60% Topic A**, **40% Topic B**

The model outputs the “topics” it finds.

The user needs to assign a name to the “topics”.

EXAMPLE TOPIC MODELING OUTPUT

- I like bananas and oranges. **100% Food**
- Frogs and fish live in ponds. **100% Topic B**
- Kittens and puppies are fluffy. **100% Topic B**
- For breakfast, I had a spinach and apple smoothie. **100% Food**
- My kitten loves kale. **60% Food**, **40% Topic B**

The model outputs the “topics” it finds.

The user needs to assign a name to the “topics”.

EXAMPLE TOPIC MODELING OUTPUT

- I like bananas and oranges. **100% Food**
- Frogs and fish live in ponds. **100% Animals**
- Kittens and puppies are fluffy. **100% Animals**
- For breakfast, I had a spinach and apple smoothie. **100% Food**
- My kitten loves kale. **60% Food**, **40% Animals**

The model outputs the "topics" it finds.

The user needs to assign a name to the "topics".

Example #2:
From Words —> Topics
(Dimensionality Reduction)

FROM WORDS (3D) —> TOPICS (2D)

- “I love my pet rabbit.”
- “That dish yesterday was amazing.”
- “She cooked the best rabbit dish ever.”
- “I gave leftovers of that dish to my pet, mr. rabbit”
- “Rabbits make messy pets.”
- “My rabbit growls when I pet her.”
- “He has five rabbits.”
- “I had this weird dish with fried rabbit.”
- “That’s my pet rabbit’s favorite dish.”

FROM WORDS (3D) —> TOPICS (2D)

- "I love my pet rabbit."
- "That dish yesterday was amazing."
- "She cooked the best rabbit dish ever."
- "I gave leftovers of that dish to my pet, mr. rabbit"
- "Rabbits make messy pets."
- "My rabbit growls when I pet her."
- "He has five rabbits."
- "I had this weird dish with fried rabbit."
- "That's my pet rabbit's favorite dish."

Let's clean this text a bit:
- Remove stop words
- Keep only nouns

FROM WORDS (3D) —> TOPICS (2D)

- “I love my **pet** **rabbit**.”
- “That **dish** yesterday was amazing.”
- “She cooked the best **rabbit** **dish** ever.”
- “I gave leftovers of that **dish** to my **pet**, mr. **rabbit**”
- “**Rabbits** make messy **pets**.”
- “My **rabbit** growls when I **pet** her.”
- “He has five **rabbits**.”
- “I had this weird **dish** with fried **rabbit**.”
- “That’s my **pet** **rabbit’s** favorite **dish**.”

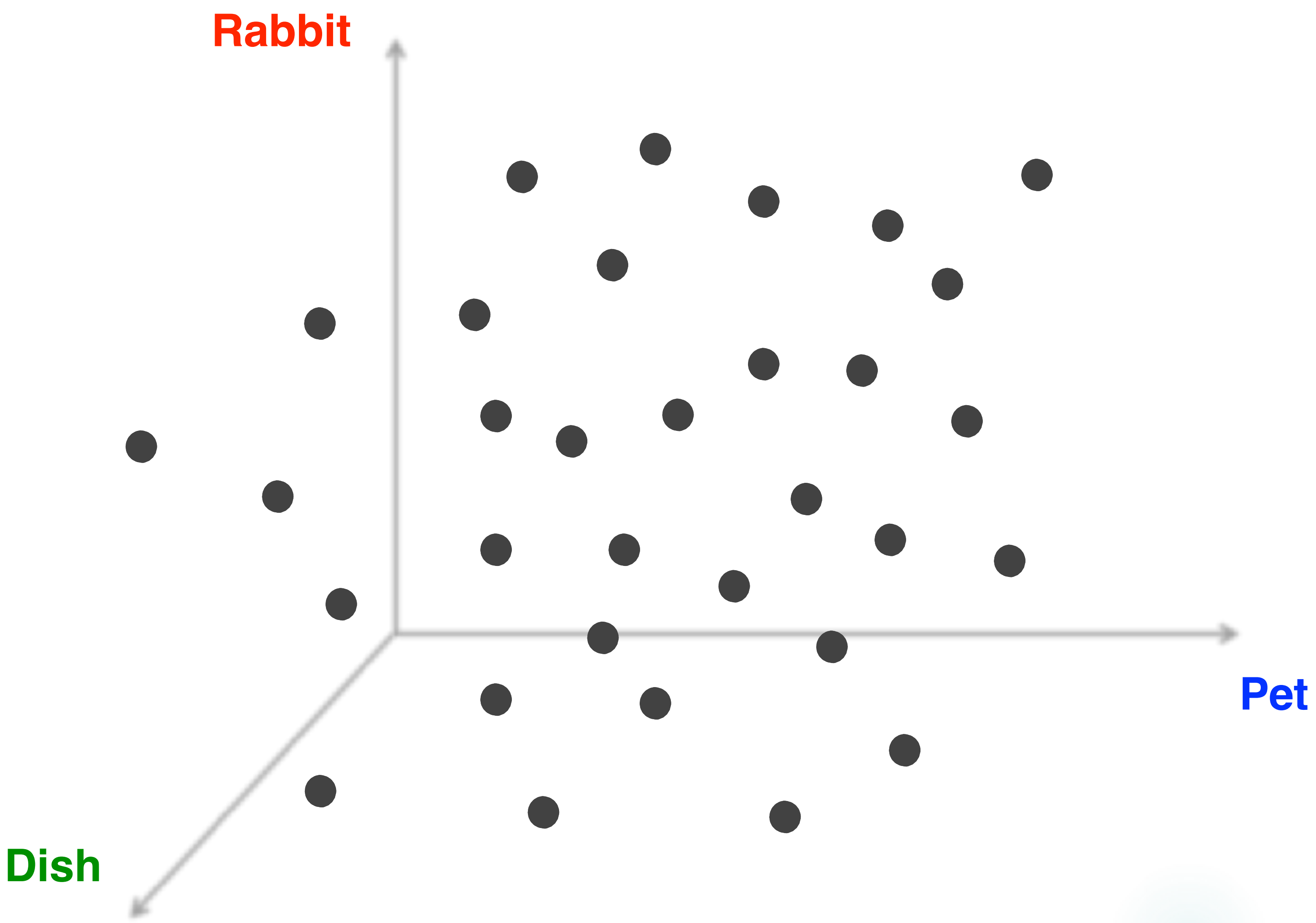
We end up with 3 features:
- Pet
- Rabbit
- Dish

FROM WORDS (3D) —> TOPICS (2D)

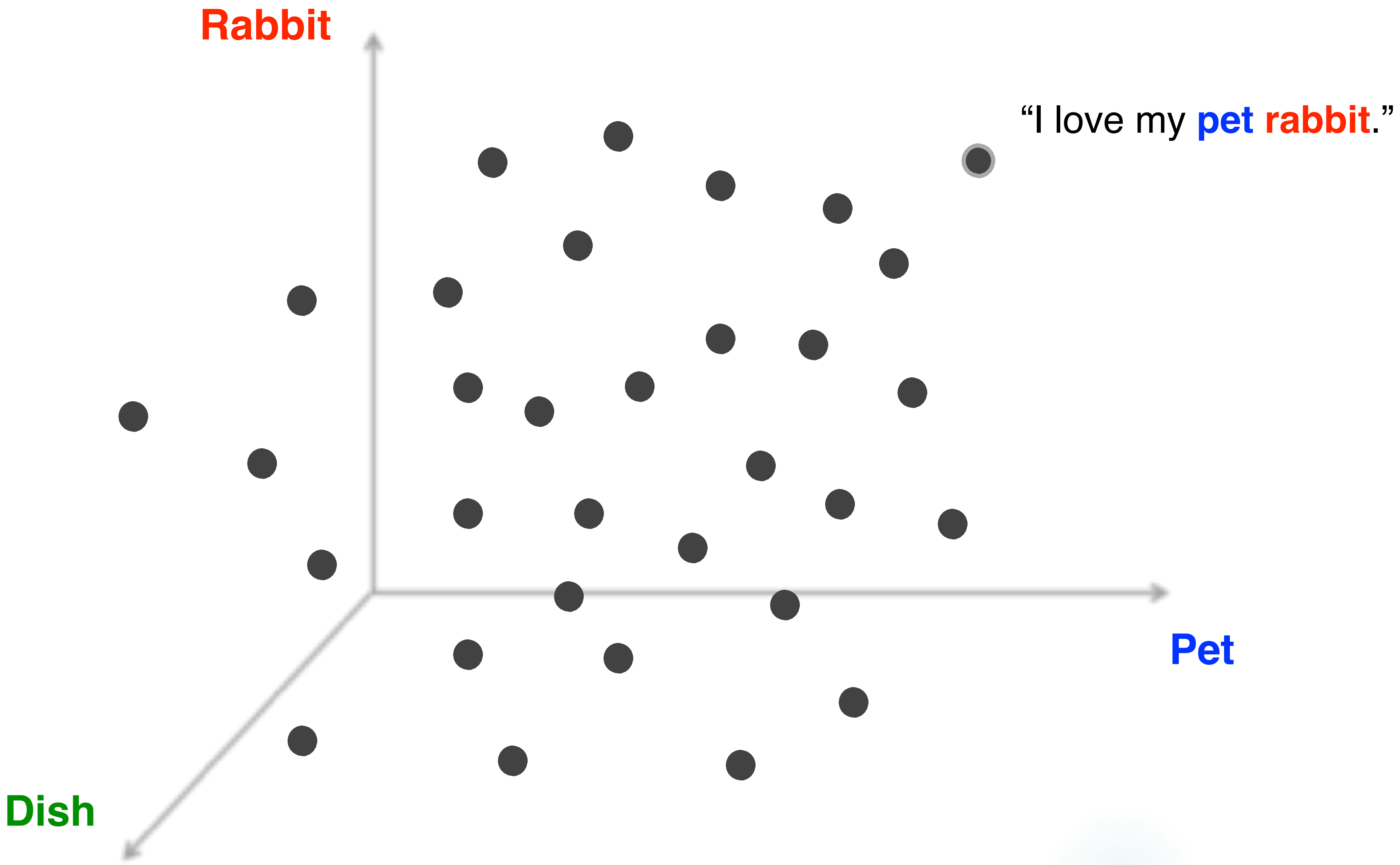
- “I love my **pet** **rabbit**.”
- “That **dish** yesterday was amazing.”
- “She cooked the best **rabbit** **dish** ever.”
- “I gave leftovers of that **dish** to my **pet**, mr. **rabbit**”
- “**Rabbits** make messy **pets**.”
- “My **rabbit** growls when I **pet** her.”
- “He has five **rabbits**.”
- “I had this weird **dish** with fried **rabbit**.”
- “That’s my **pet** **rabbit’s** favorite **dish**.”

Let’s move from the word space (3 features) to the topic space (2 features).

FROM WORDS (3D) —> TOPICS (2D)



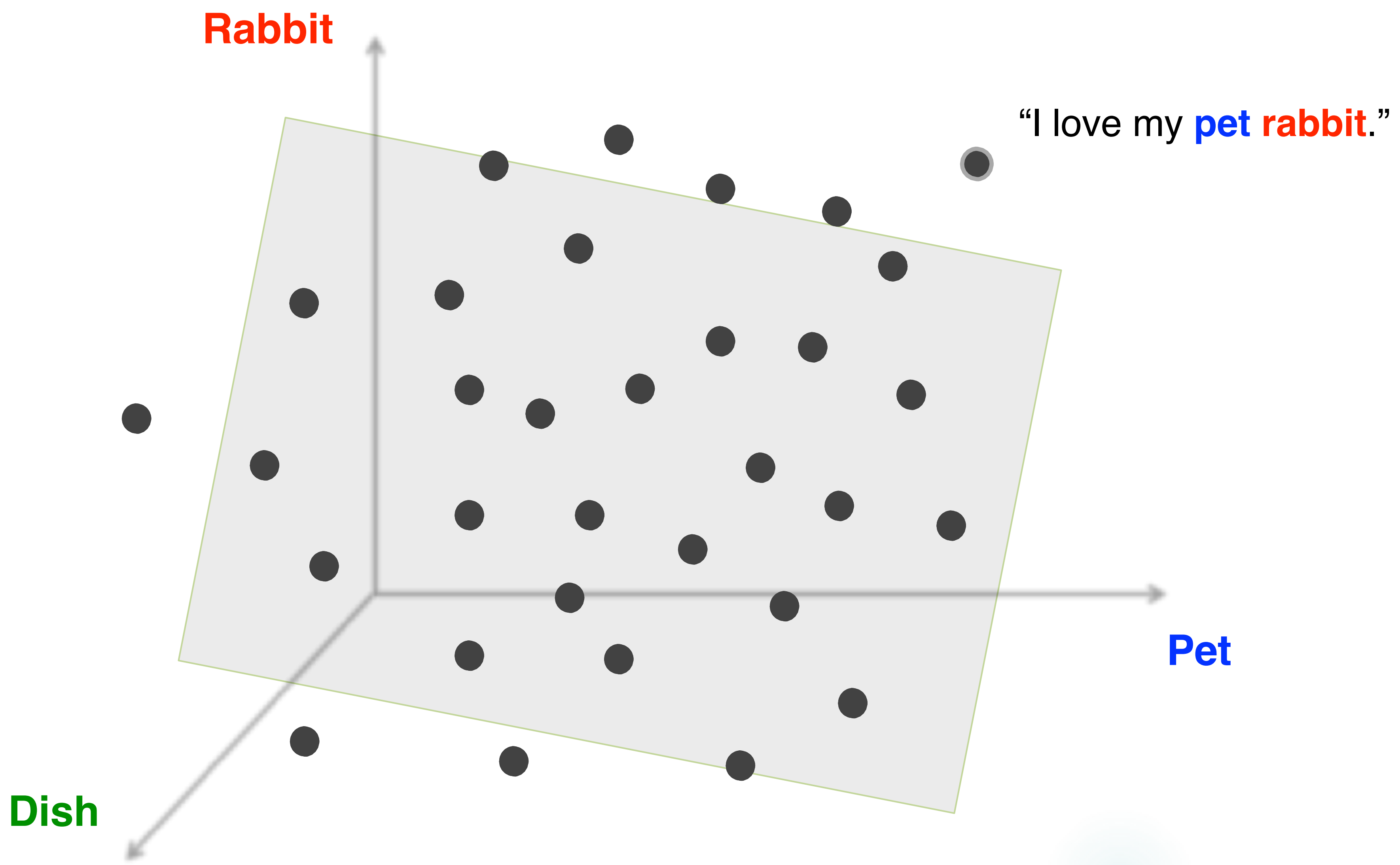
FROM WORDS (3D) —> TOPICS (2D)



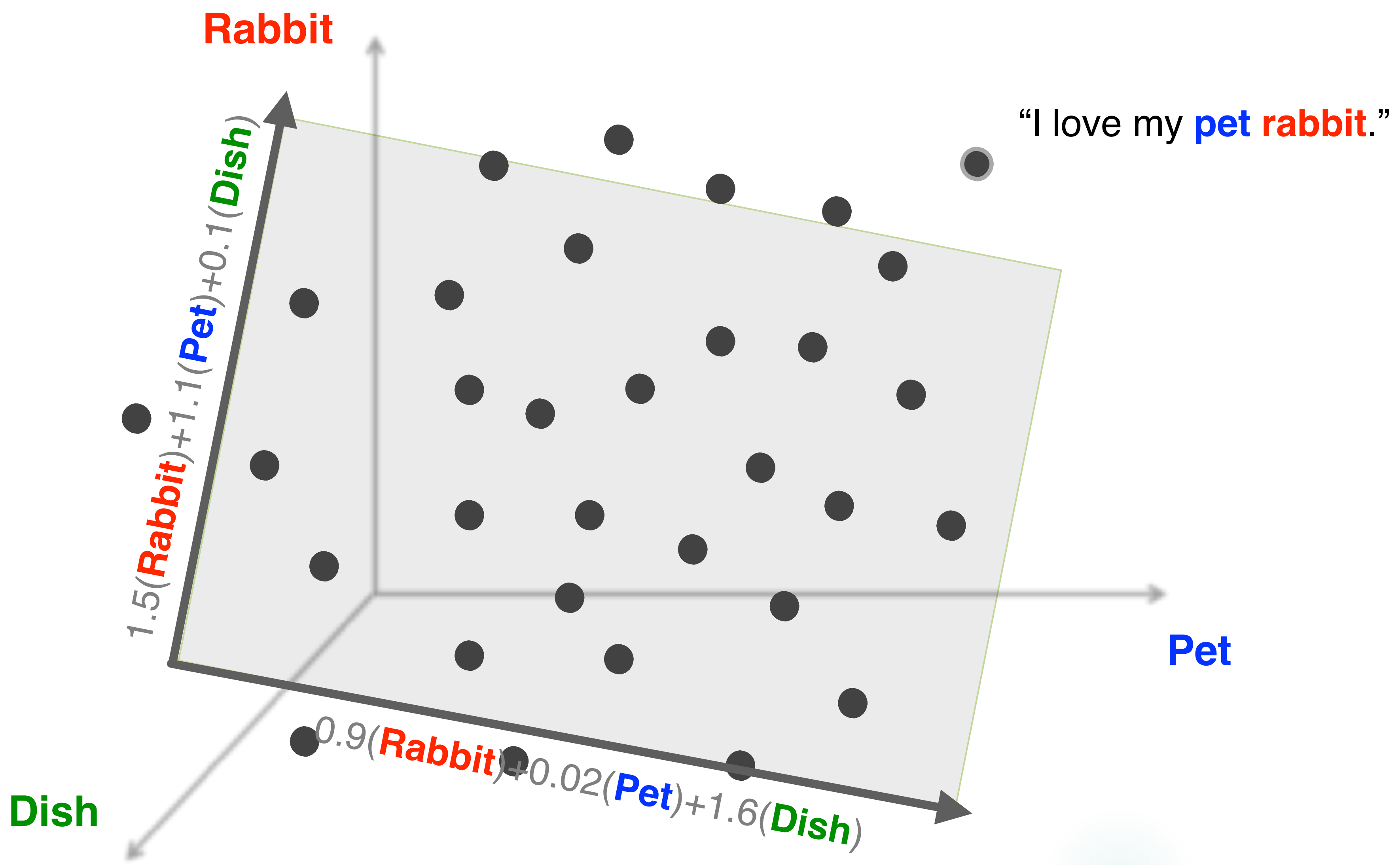
HOW TO GO FROM 3D —> 2D?

- **Feature Selection**
  - Start with 3 features: PET, RABBIT, DISH
  - Decide that DISH isn't an important feature
  - End with 2 features: PET, RABBIT
- **Feature Extraction**
  - Start with 3 features: PET, RABBIT, DISH
  - Create 2 new features that are combos of PET / RABBIT / DISH (example: PCA)
  - End with 2 features: Principal Component 1, Principal Component 2

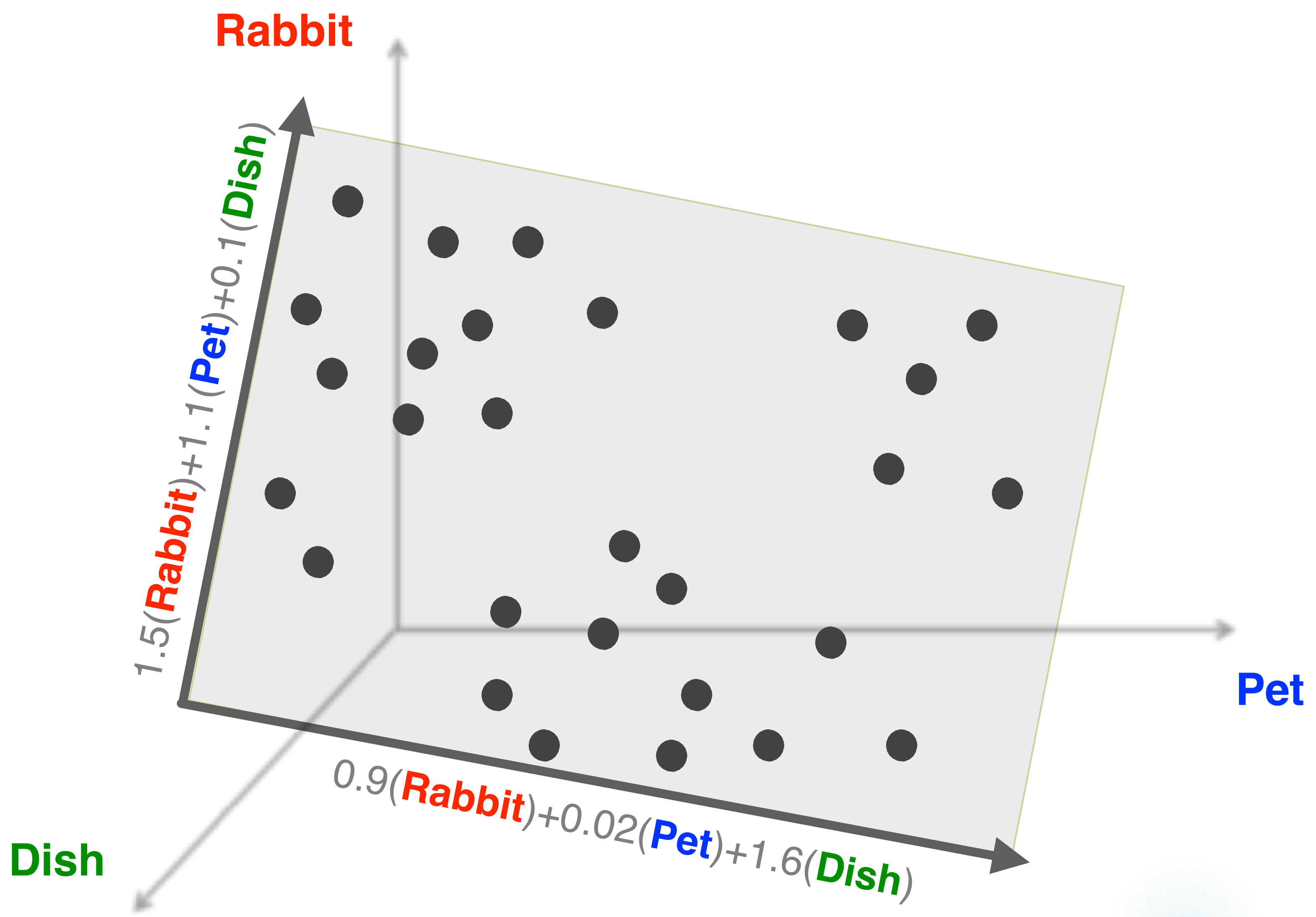
FEATURE EXTRACTION



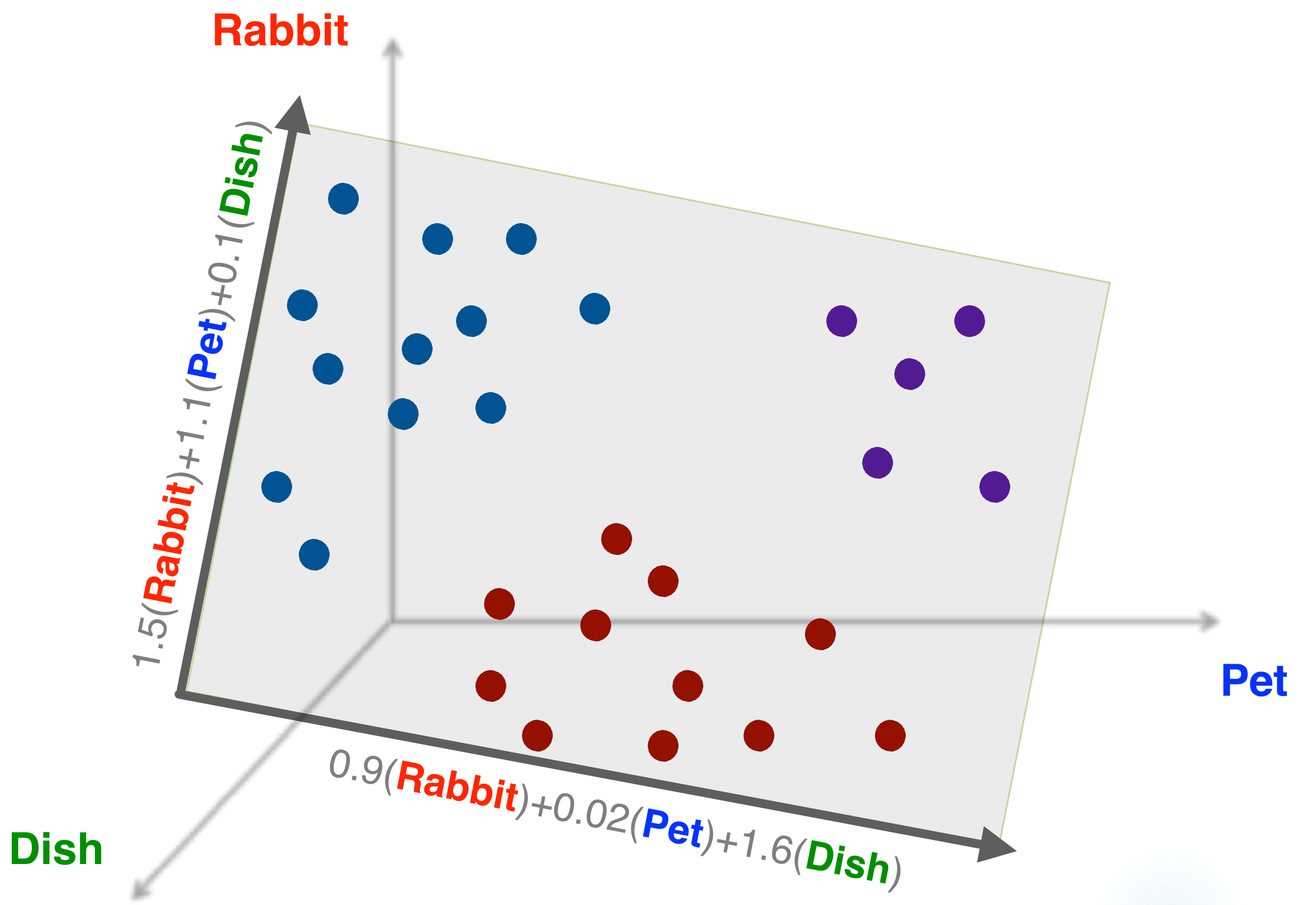
FEATURE EXTRACTION



FEATURE EXTRACTION



CLUSTERING IS EASIER IN 2D



LET'S LOOK AT THE CLUSTERS

"I love my pet rabbit."
"Rabbits make messy pets."
"My rabbit growls when I pet her."
"He has five rabbits."

"That dish yesterday was amazing."
"She cooked the best rabbit dish ever."
"I had this weird dish with fried rabbit."

"I gave leftovers of that dish to my pet, Mr. Rabbit"
"That's my pet rabbit's favorite dish."

LET'S LOOK AT THE CLUSTERS

Axis 1: 1.5(Rabbit) + 1.1 (Pet) + 0.1(Dish)
Axis 2: 0.9(Rabbit) + 0.02(Pet) + 1.6(Dish)

"I love my pet rabbit."
"Rabbits make messy pets."
"My rabbit growls when I pet her."
"He has five rabbits."

"That dish yesterday was amazing."
"She cooked the best rabbit dish ever."
"I had this weird dish with fried rabbit."

"I gave leftovers of that dish to my pet, Mr. Rabbit"
"That's my pet rabbit's favorite dish."

LET'S LOOK AT THE CLUSTERS

Axis 1: 1.5(Rabbit) + 1.1 (Pet) + 0.1(Dish)
Axis 2: 0.9(Rabbit) + 0.02(Pet) + 1.6(Dish)

Axis 1: High
Axis 2: Low

"I love my **pet rabbit**."
"**Rabbits** make messy **pets**."
"My **rabbit** growls when I **pet** her."
"He has five **rabbits**."

Axis 1: Low
Axis 2: High

"That **dish** yesterday was amazing."
"She cooked the best **rabbit dish** ever."
"I had this weird **dish** with fried **rabbit**."

Axis 1: High
Axis 2: High

"I gave leftovers of that **dish** to my **pet**, Mr. **Rabbit**"
"That's my **pet rabbit's** favorite **dish**."

LET'S LOOK AT THE CLUSTERS

Topic 1: 1.5(Rabbit) + 1.1 (Pet) + 0.1(Dish) <— pets, pet rabbits
Topic 2: 0.9(Rabbit) + 0.02(Pet) + 1.6(Dish) <— food, rabbit dishes

Topic 1: High
Topic 2: Low

"I love my **pet rabbit**."
"**Rabbits** make messy **pets**."
"My **rabbit** growls when I **pet** her."
"He has five **rabbits**."

Topic 1: Low
Topic 2: High

"That **dish** yesterday was amazing."
"She cooked the best **rabbit dish** ever."
"I had this weird **dish** with fried **rabbit**."

Topic 1: High
Topic 2: High

"I gave leftovers of that **dish** to my **pet**, Mr. **Rabbit**"
"That's my **pet rabbit's** favorite **dish**."

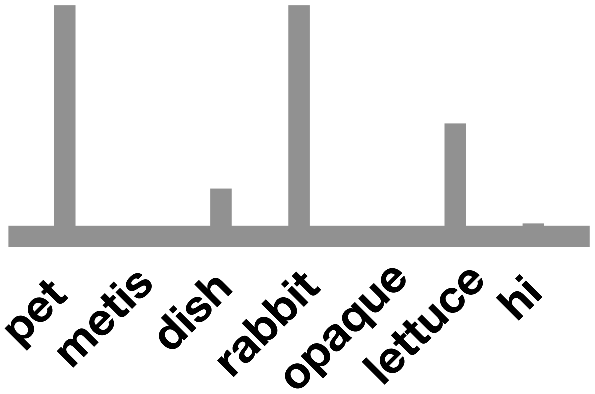
LET'S LOOK AT THE CLUSTERS

Topic modeling produces soft / fuzzy clusters. A document does not belong to a single cluster. Each document contains bits of each topic.

T1	T2	
87%	13%	"I love my **pet rabbit**."
88%	12%	"**Rabbits** make messy **pets**."
80%	20%	"My **rabbit** growls when I **pet** her."
66%	34%	"He has five **rabbits**."
2%	98%	"That **dish** yesterday was amazing."
16%	84%	"She cooked the best **rabbit dish** ever."
15%	85%	"I had this weird **dish** with fried **rabbit**."
47%	53%	"I gave leftovers of that **dish** to my **pet**, Mr. **Rabbit**"
42%	58%	"That's my **pet rabbit's** favorite **dish**."

WHAT IS A TOPIC?

A topic can be thought of as a **probability distribution of words**



Take the "pet rabbits" topic for example:

- **Words more likely to appear**: pet, rabbit, lettuce, cage, fluffy…
- **Words less likely to appear**: metis, dish, opaque, hi…

Topic 1: 1.5(Rabbit) + 1.1 (Pet) + 0.1(Dish) <— pets, pet rabbits
Topic 2: 0.9(Rabbit) + 0.02(Pet) + 1.6(Dish) <— food, rabbit dishes

Topic: probability distribution over all possible words

Word	Prob in [Topic 1]	Prob in [Topic 2]
pet	2.3×10^{-7}	1.2×10^{-10}
rabbit	7.9×10^{-7}	3.4×10^{-8}
dish	6.8×10^{-11}	4.5×10^{-7}
car	3.1×10^{-12}	1.8×10^{-12}
hello	8.3×10^{-9}	1.4×10^{-9}
the	7.4×10^{-4}	7.3×10^{-4}
love	5.4×10^{-8}	3.9×10^{-8}
affair	3.0×10^{-13}	2.1×10^{-13}
delicious	9.1×10^{-9}	9.8×10^{-8}

Topic 1: 1.5(Rabbit) + 1.1 (Pet) + 0.1(Dish) <— pets, pet rabbits
Topic 2: 0.9(Rabbit) + 0.02(Pet) + 1.6(Dish) <— food, rabbit dishes

Topic: probability distribution over all possible words

Word	**Prob in [Pets]**	**Prob in [Food]**
pet	2.3×10^{-7}	1.2×10^{-10}
rabbit	7.9×10^{-7}	3.4×10^{-8}
dish	6.8×10^{-11}	4.5×10^{-7}
car	3.1×10^{-12}	1.8×10^{-12}
hello	8.3×10^{-9}	1.4×10^{-9}
the	7.4×10^{-4}	7.3×10^{-4}
love	5.4×10^{-8}	3.9×10^{-8}
affair	3.0×10^{-13}	2.1×10^{-13}
delicious	9.1×10^{-9}	9.8×10^{-8}

Topic 1: 1.5(Rabbit) + 1.1 (Pet) + 0.1(Dish) <— pets, pet rabbits
Topic 2: 0.9(Rabbit) + 0.02(Pet) + 1.6(Dish) <— food, rabbit dishes

Topic: probability distribution over all possible words

Word	**Prob in [Pets]**	**Prob in [Food]**
pet	**2.3×10^{-7}**	1.2×10^{-10}
rabbit	**7.9×10^{-7}**	3.4×10^{-8}
dish	6.8×10^{-11}	4.5×10^{-7}
car	3.1×10^{-12}	1.8×10^{-12}
hello	8.3×10^{-9}	1.4×10^{-9}
the	**7.4×10^{-4}**	7.3×10^{-4}
love	5.4×10^{-8}	3.9×10^{-8}
affair	3.0×10^{-13}	2.1×10^{-13}
delicious	9.1×10^{-9}	9.8×10^{-8}

Topic 1: 1.5(Rabbit) + 1.1 (Pet) + 0.1(Dish) <— pets, pet rabbits
Topic 2: 0.9(Rabbit) + 0.02(Pet) + 1.6(Dish) <— food, rabbit dishes

Topic: probability distribution over all possible words

Word	Prob in [Pets]	Prob in [Food]
pet	2.3×10^{-7}	1.2×10^{-10}
rabbit	7.9×10^{-7}	3.4×10^{-8}
dish	6.8×10^{-11}	**4.5×10^{-7}**
car	3.1×10^{-12}	1.8×10^{-12}
hello	8.3×10^{-9}	1.4×10^{-9}
the	7.4×10^{-4}	**7.3×10^{-4}**
love	5.4×10^{-8}	3.9×10^{-8}
affair	3.0×10^{-13}	2.1×10^{-13}
delicious	9.1×10^{-9}	**9.8×10^{-8}**

TOPIC MODELING

- The process of discovering "topics" that occur in a collection of documents

- We went through two examples:
  1. Understand the concept of a "topic"
  2. Understand how this is a form of dimensionality reduction

Latent Semantic Analysis (LSA)
&
Non-Negative Matrix Factorization (NMF)

MATRIX FACTORIZATION APPROACHES

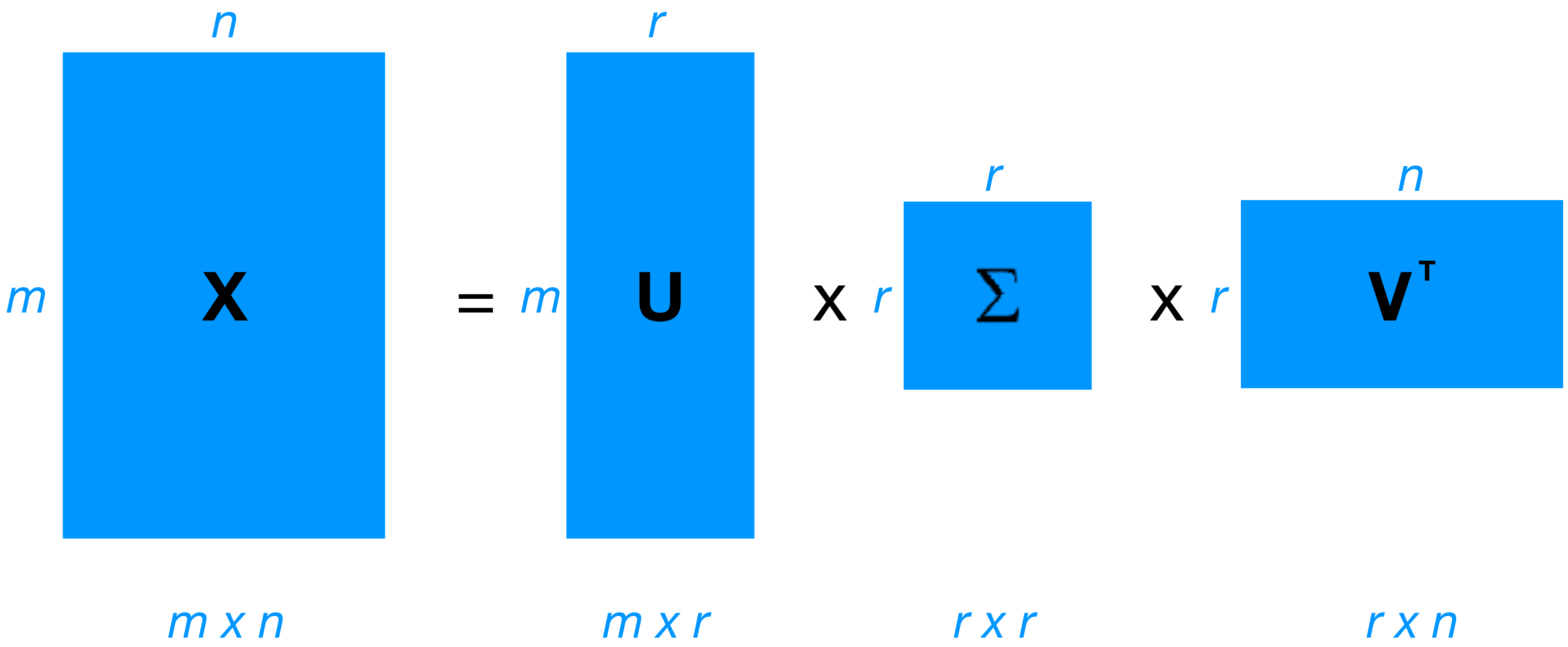
MATRIX FACTORIZATION

- This is a standard machine learning approach that can also be used for topic modeling

- We will review two techniques:
  - Latent Semantic Analysis (LSA)
  - Non-Negative Matrix Factorization (NMF)

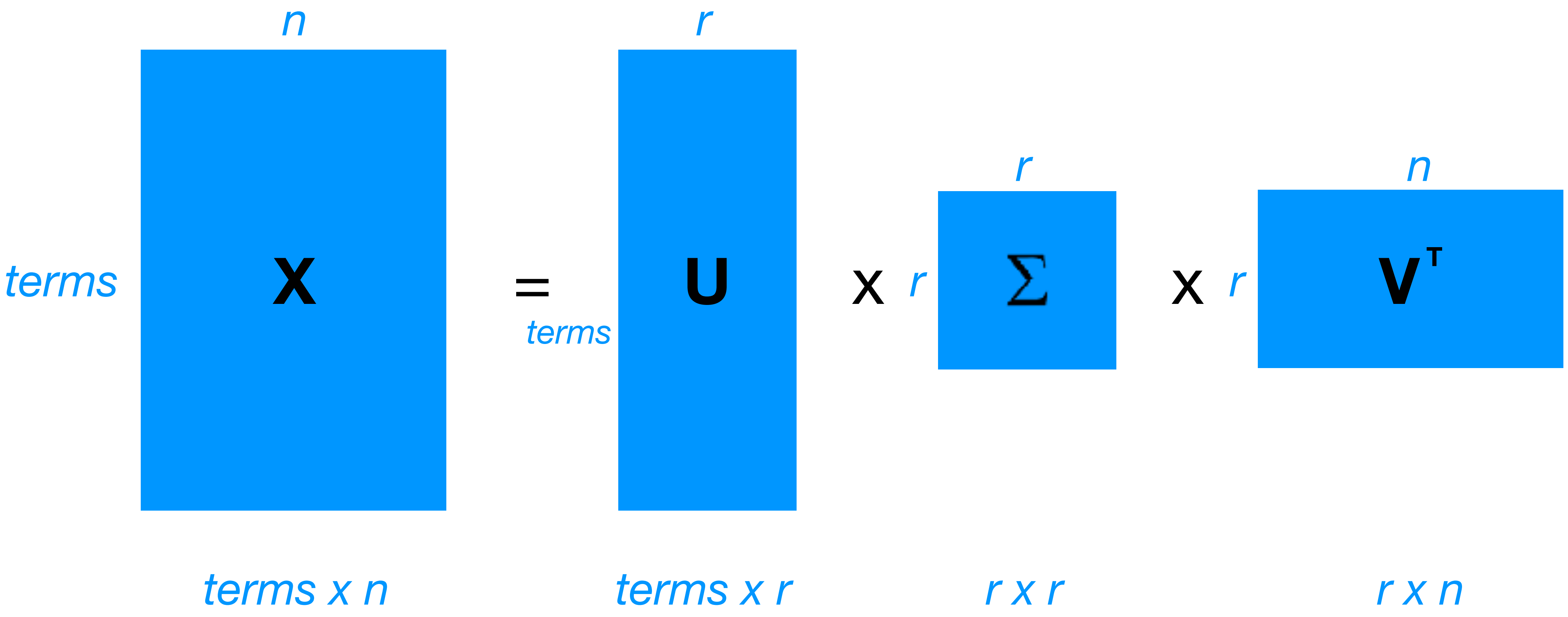
Singular Value Decomposition

$$X = U \Sigma V^T$$



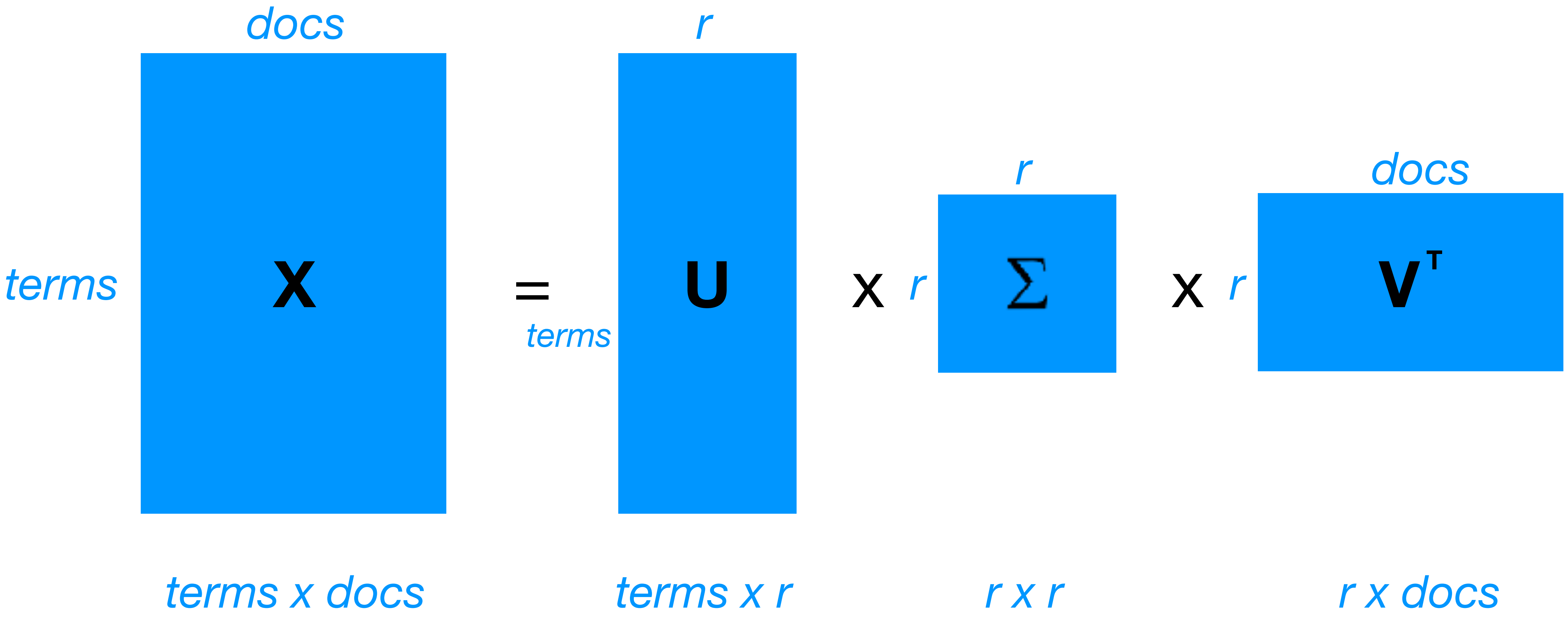
Singular Value Decomposition

$$X = U \Sigma V^T$$



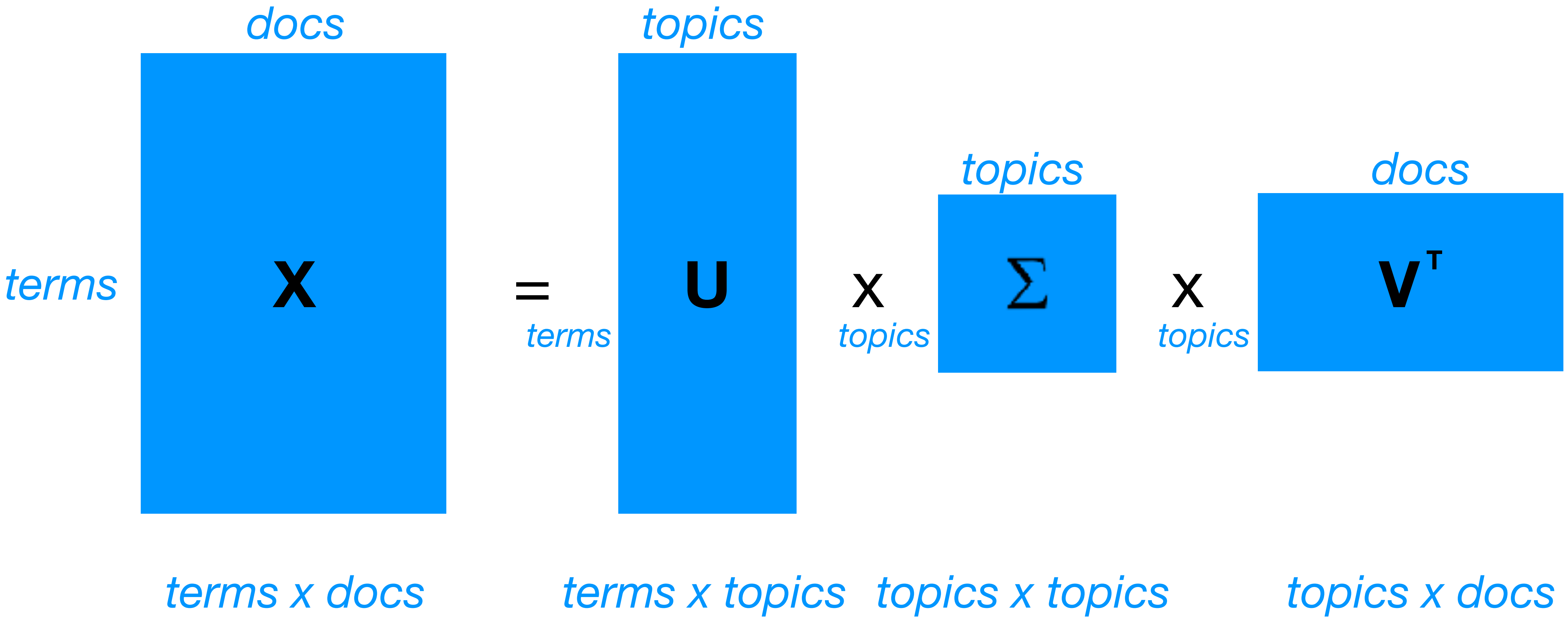
Singular Value Decomposition

$$\mathbf{X} = \mathbf{U}\,\Sigma\,\mathbf{V}^{T}$$



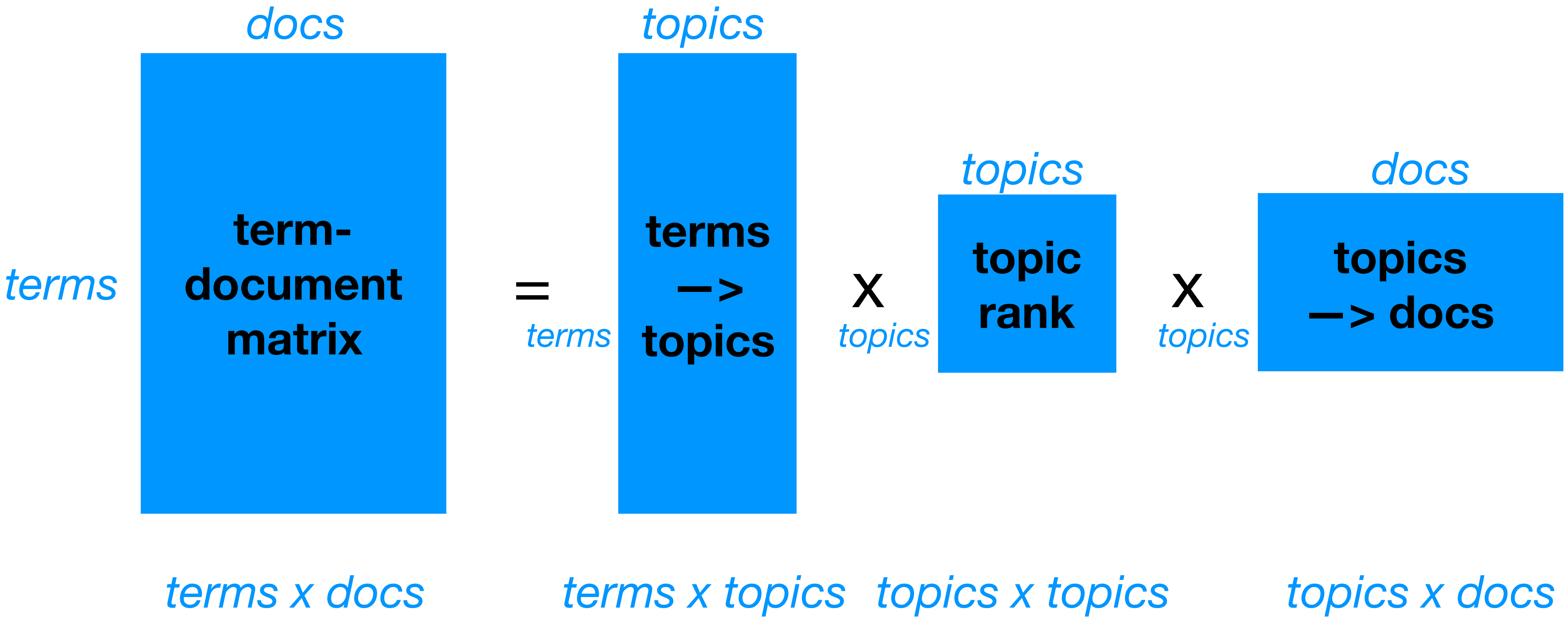
Singular Value Decomposition

$$X = U \Sigma V^T$$



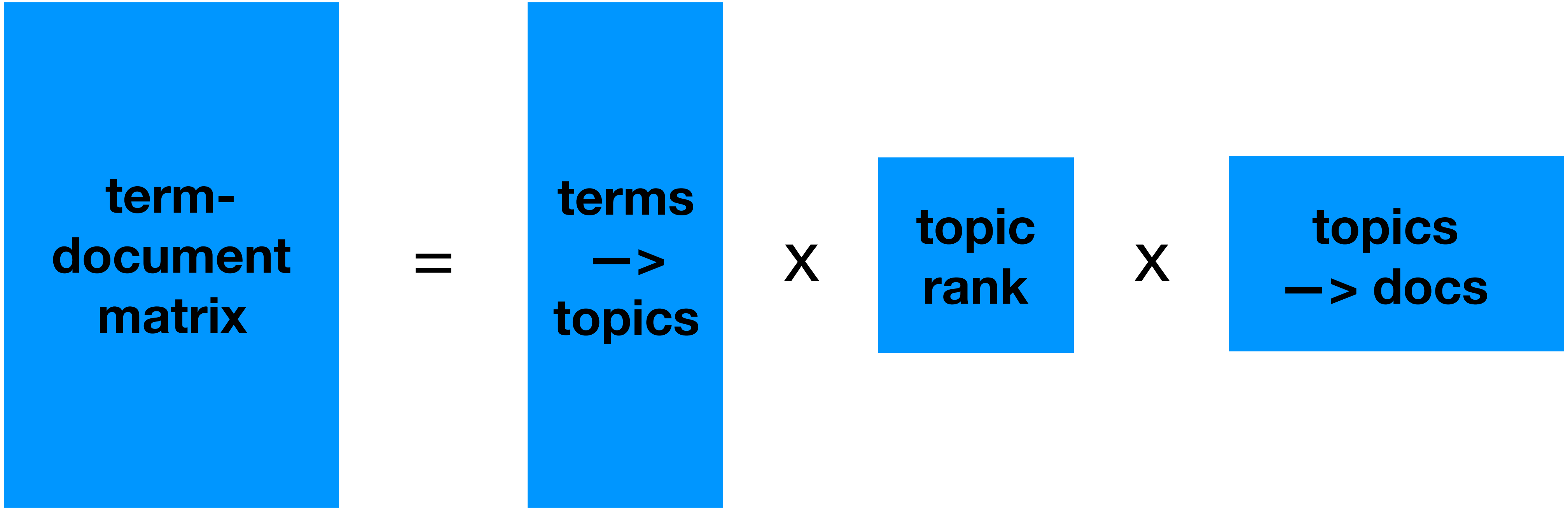
Latent Semantic Analysis

$$X = U \Sigma V^T$$



Latent Semantic Analysis

$$X = U \Sigma V^{T}$$



Decompose the document-term matrix to identify topics in the documents.

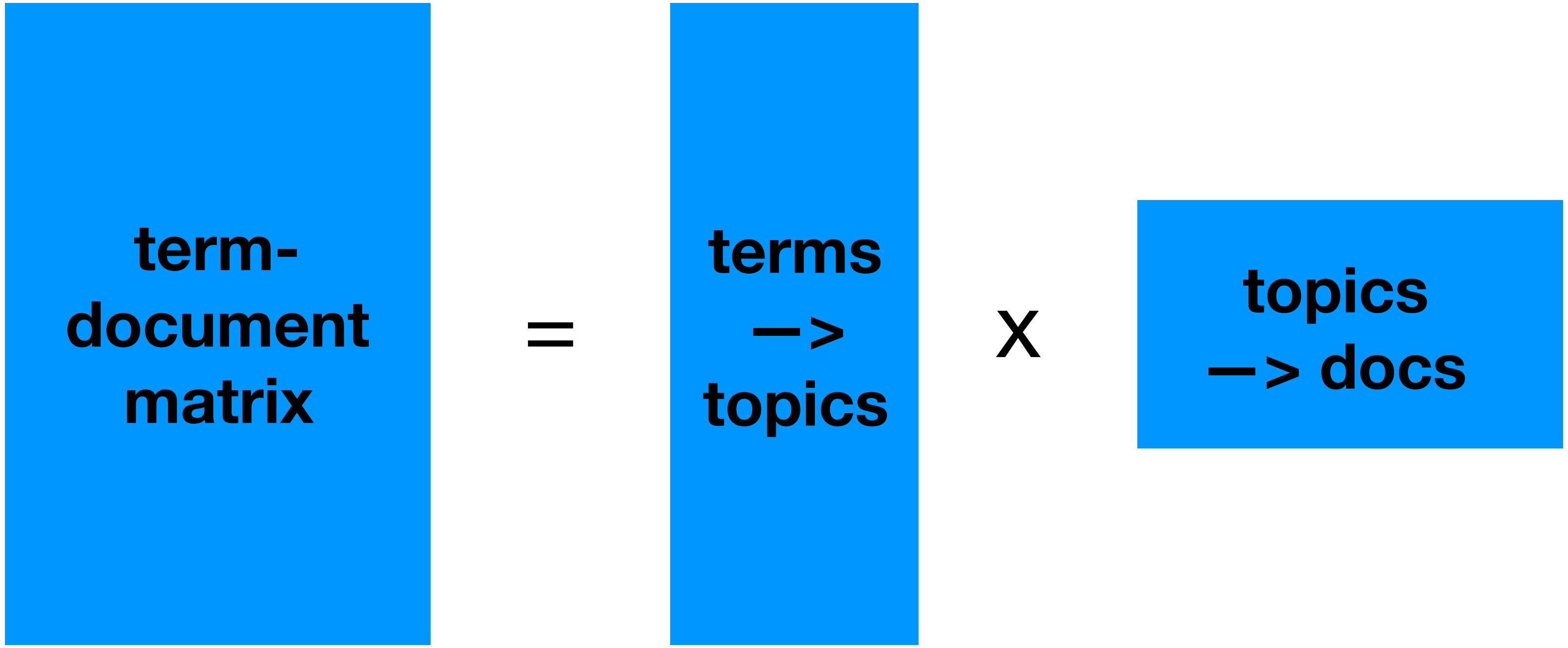
LSA CODE

- Go to the **Topic_Modeling_LSA_NMF.ipynb**

- **Input**: Count Vectorizer or TF-IDF Vectorizer
- **Parameters to Tune**:
  - Number of Topics
  - Text Preprocessing (stop words, min / max doc freq, parts of speech…)
- **Output**: U Matrix (terms —> topics) and V Matrix (documents —> topics)

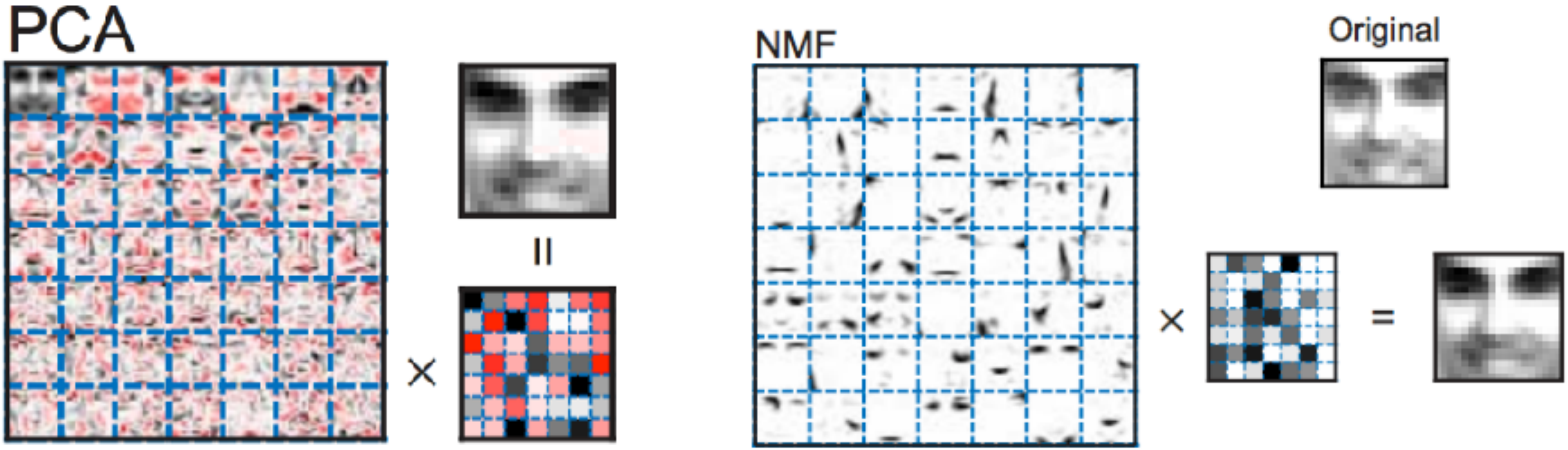
Non-Negative Matrix Factorization

$$V = W \times H$$



Same idea, but all three matrices must have only positive values.

WHY ONLY POSITIVE VALUES?



- Since NMF can never undo the application of a latent feature, it is much more careful about what it adds at each step. In some applications, this can make for more human interpretable latent features.
- Because NMF has the extra constraint of positive values, it will tend to lose more information when truncating. Also, NMF does not have to give orthogonal latent vectors.

NMF CODE

- Go to the **Topic_Modeling_LSA_NMF.ipynb**

- **Input**: Count Vectorizer or TF-IDF Vectorizer
- **Parameters to Tune**:
  - Number of Topics
  - Text Preprocessing (stop words, min / max doc freq, parts of speech…)
- **Output**: W Matrix (terms —> topics) and H Matrix (documents —> topics)

Latent Dirichlet Allocation (LDA)

PROBABILISTIC APPROACH

LATENT DIRICHLET ALLOCATION

- Latent: Hidden
- Dirichlet: Type of Probability Distribution

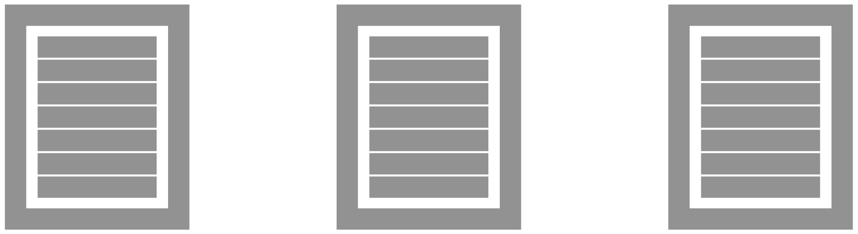
```
output = []
for _ in range(1000):
    output.append(np.random.dirichlet((1, 1, 1)))

print(np.mean(output, axis=0))
```

```
[ 0.3297311   0.33714122  0.33312768]
```

LATENT DIRICHLET ALLOCATION

Think in terms of probability distributions

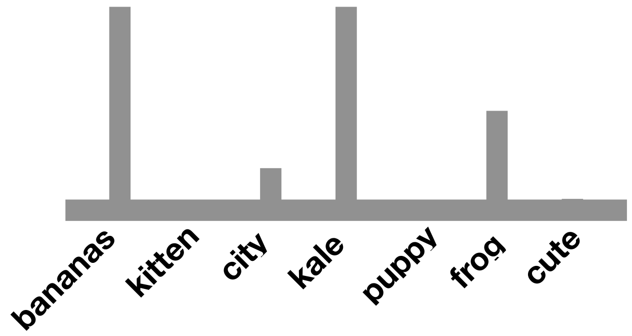
Every **document** consists of a distribution of **topics**



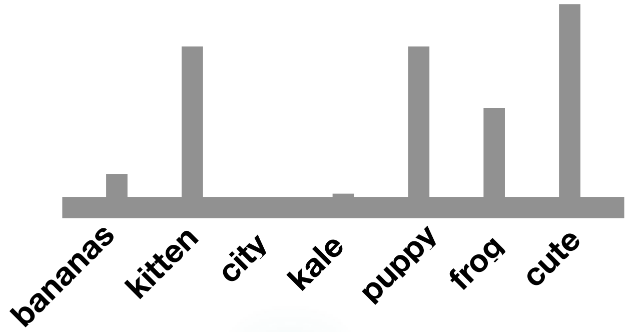
100% Topic A	100% Topic B	60% Topic A 40% Topic B

Every **topic** consists of a distribution of **words**

Topic: Food



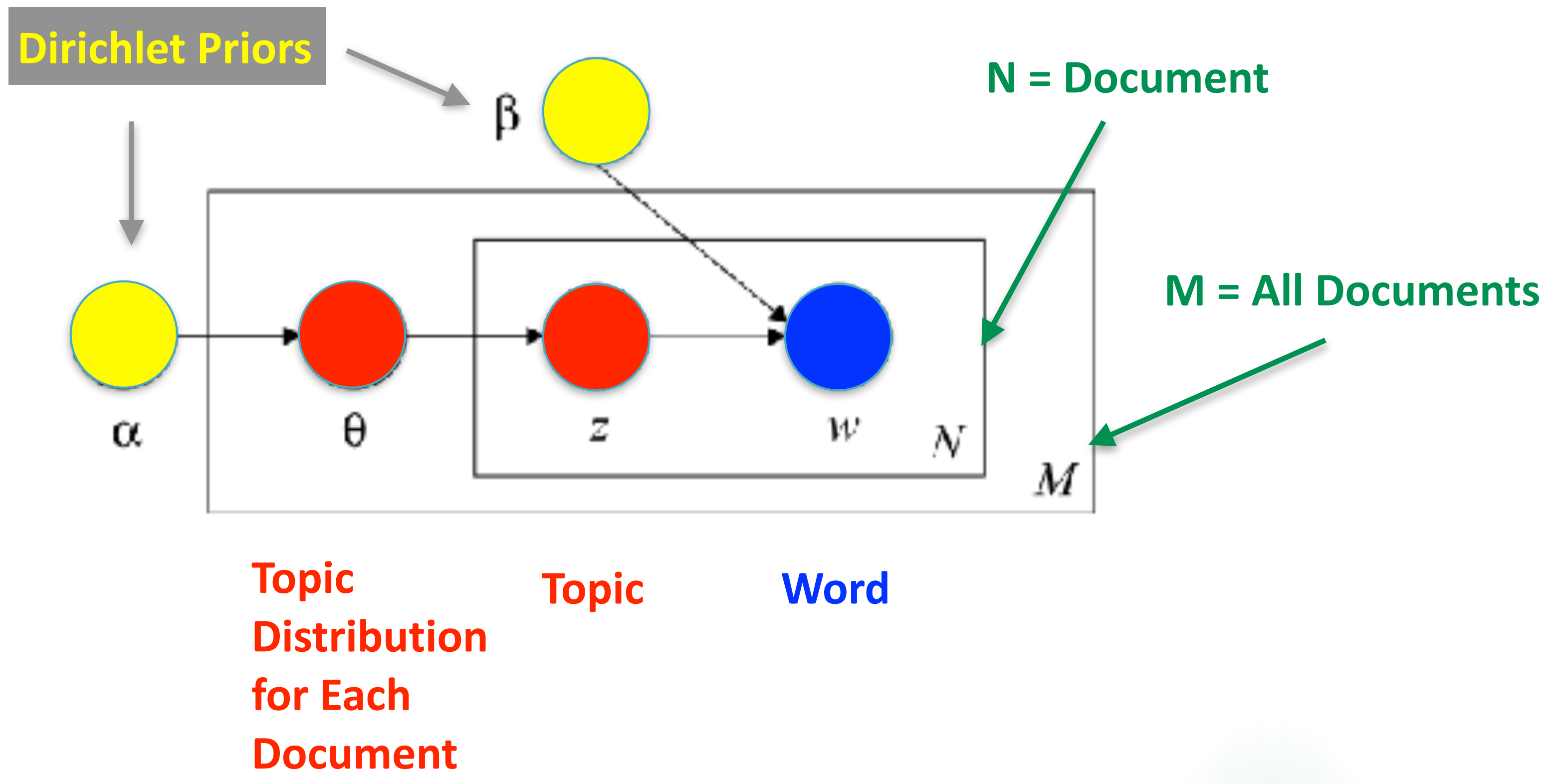
Topic: Animals



LATENT DIRICHLET ALLOCATION

alpha: per doc topic distribution
- high: each doc has lots of topics
- low: each doc has few topics

beta: per topic word distribution
- high: each topic has lots of words
- low: each topic has few words



How LDA Works

Perspective #1

Topic Modeling: LDA

Let's use an algorithm specifically developed to find topics.

Topic Modeling: LDA

Let’s use an algorithm specifically developed to find topics.

Model the process of writing

Empty page: I’ll write a document.

Topic Modeling: LDA

Let’s use an algorithm specifically developed to find topics.

Model the process of writing

Empty page: I’ll write a document.

First, I’ll decide what topics to write on.

Choose the topic distribution.

Topic Modeling: LDA

Let's use an algorithm specifically developed to find topics.

Model the process of writing

Empty page: I'll write a document.

First, I'll decide what topics to write on.
Choose the topic distribution.
Sex: 2%, Drugs: 33%, Rock'n Roll: 65%

Topic Modeling: LDA

Let's use an algorithm specifically developed to find topics.

Model the process of writing

Empty page: I'll write a document.

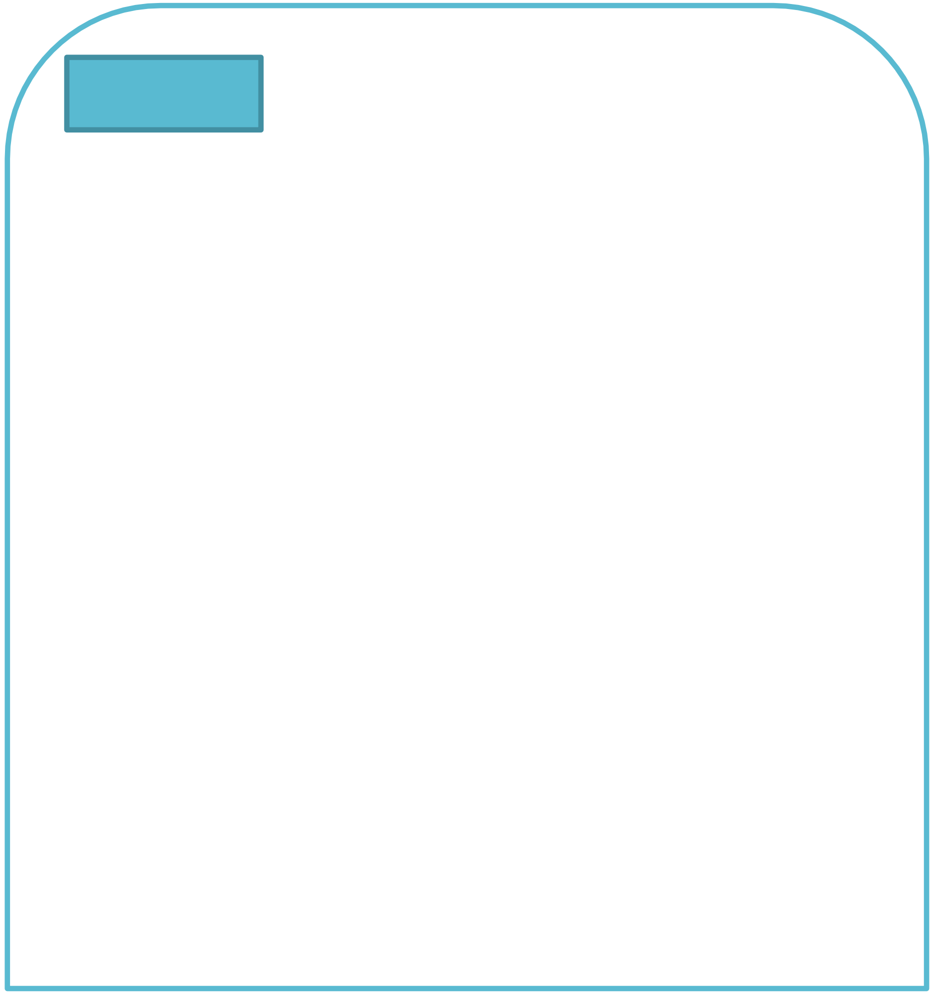
First, I'll decide what topics to write on.
Choose the topic distribution.
Sex: 2%, Drugs: 33%, Rock'n Roll: 65%

Ok. I'll write the document word by word (bag of words). First word!

Topic Modeling: LDA

Let's use an algorithm specifically developed to find topics.

Model the process of writing



Empty page: I'll write a document.

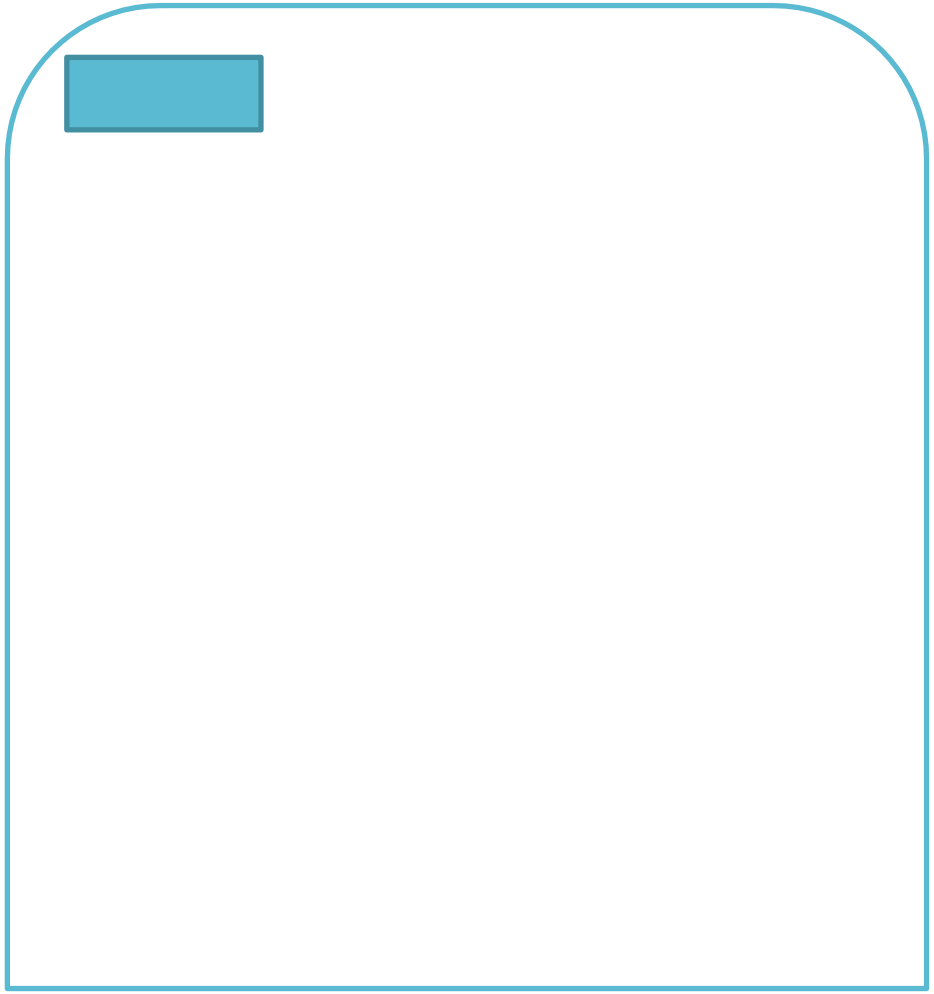
First, I'll decide what topics to write on.
Choose the topic distribution.
Sex: 2%, Drugs: 33%, Rock'n Roll: 65%

Ok. I'll write the document word by word (bag of words). First word!

Topic Modeling: LDA

Let's use an algorithm specifically developed to find topics.

Model the process of writing



Empty page: I'll write a document.

First, I'll decide what topics to write on.
Choose the topic distribution.
Sex: 2%, Drugs: 33%, Rock'n Roll: 65%

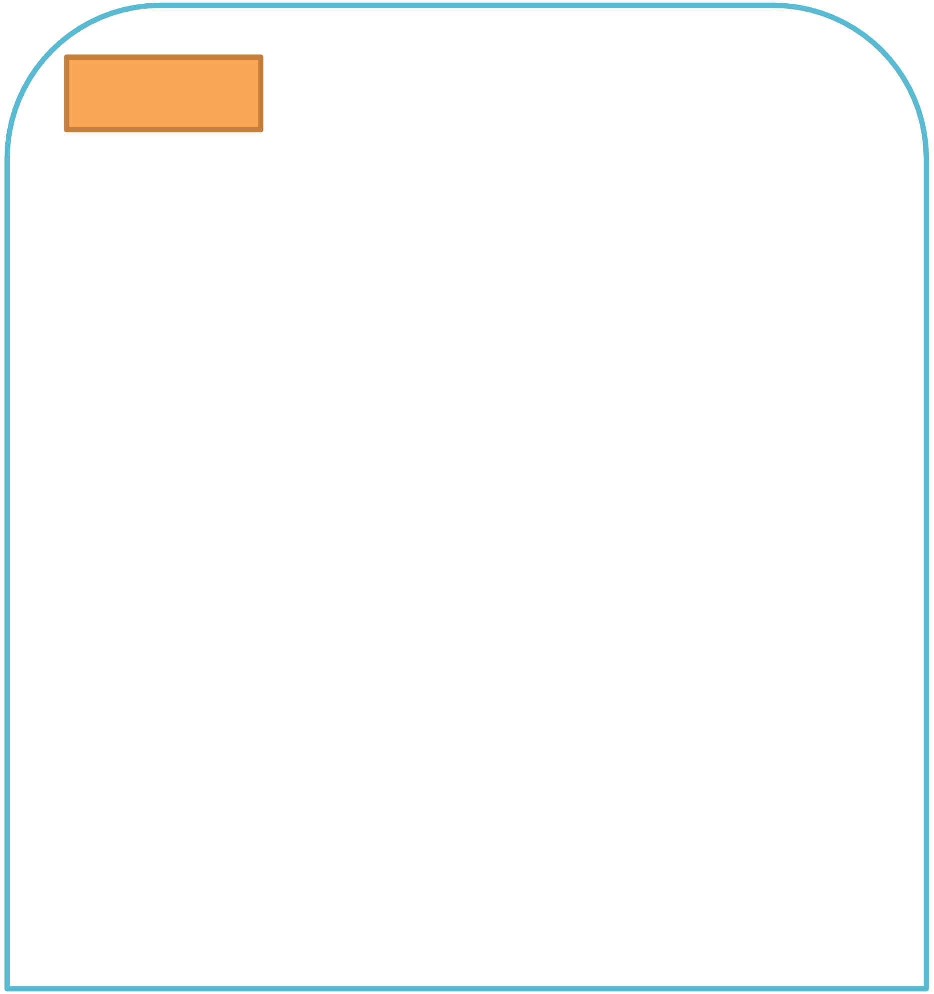
Ok. I'll write the document word by word (bag of words). First word!

Choose which topic this word will be about. Roll the dice, pick randomly from the topic distribution for the doc.

Topic Modeling: LDA

Let's use an algorithm specifically developed to find topics.

Model the process of writing



Empty page: I'll write a document.

First, I'll decide what topics to write on.
Choose the topic distribution.
Sex: 2%, Drugs: 33%, Rock'n Roll: 65%

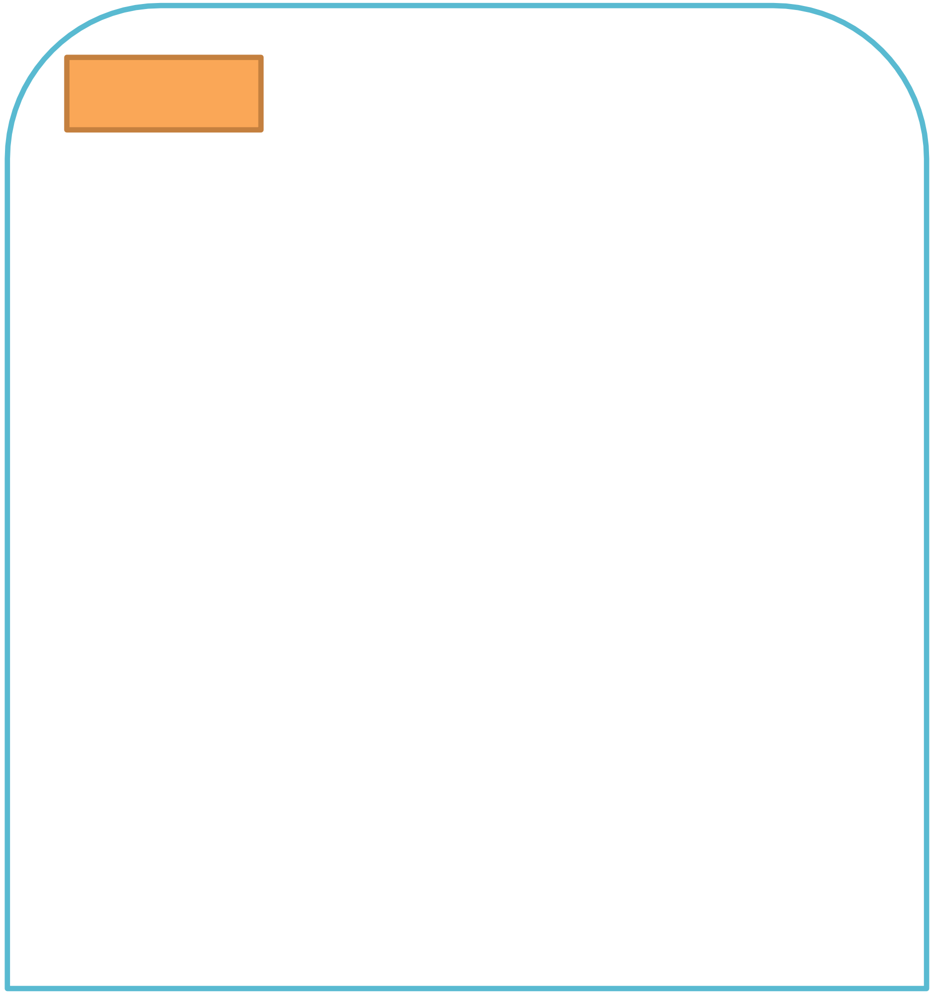
Ok. I'll write the document word by word (bag of words). First word!

Choose which topic this word will be about. Roll the dice, pick randomly from the topic distribution for the doc.

Topic Modeling: LDA

Let’s use an algorithm specifically developed to find topics.

Model the process of writing



Empty page: I’ll write a document.

First, I’ll decide what topics to write on.
Choose the topic distribution.
Sex: 2%, Drugs: 33%, Rock’n Roll: 65%

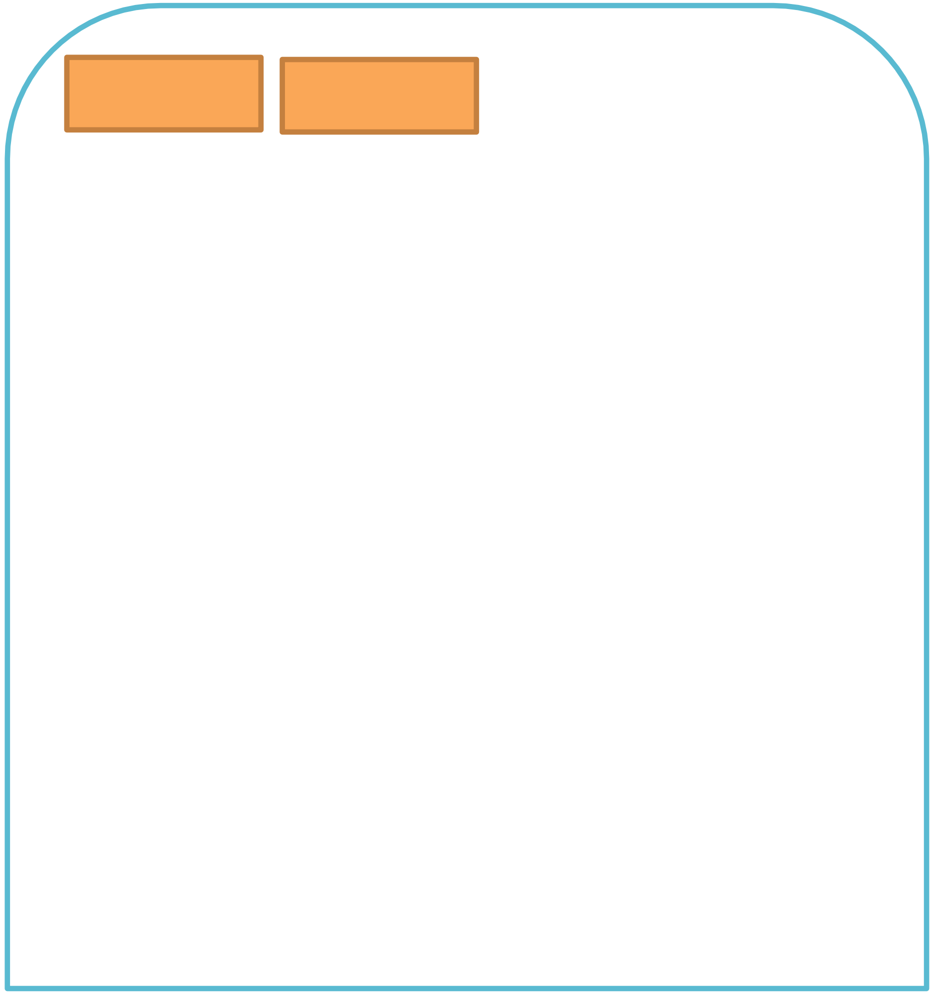
Ok. I’ll write the document word by word (bag of words). First word!

A Rock’n Roll word. Randomly pick a word according to the probability distribution of the Rock’n Roll topic.

Topic Modeling: LDA

Let’s use an algorithm specifically developed to find topics.

Model the process of writing



Empty page: I’ll write a document.

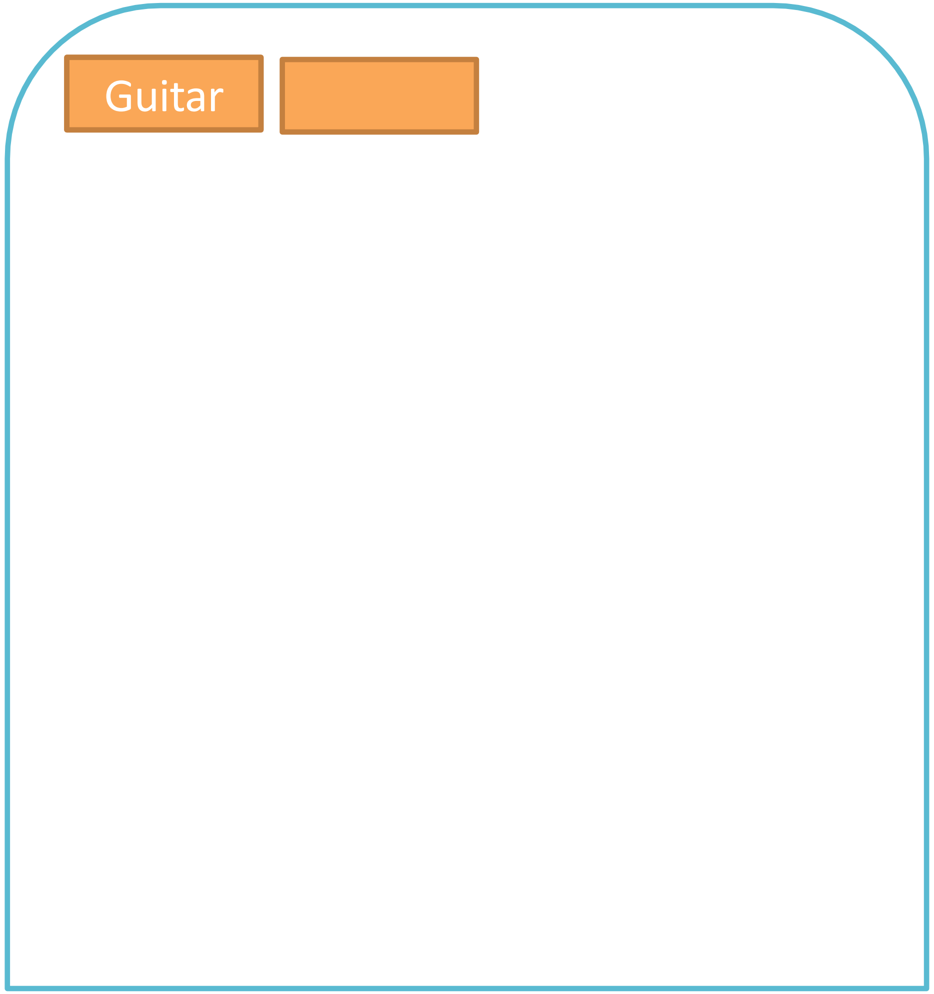
First, I’ll decide what topics to write on.
Choose the topic distribution.
Sex: 2%, Drugs: 33%, Rock’n Roll: 65%

Choose next word’s topic.
Roll the dice.

Topic Modeling: LDA

Let’s use an algorithm specifically developed to find topics.

Model the process of writing



Empty page: I’ll write a document.

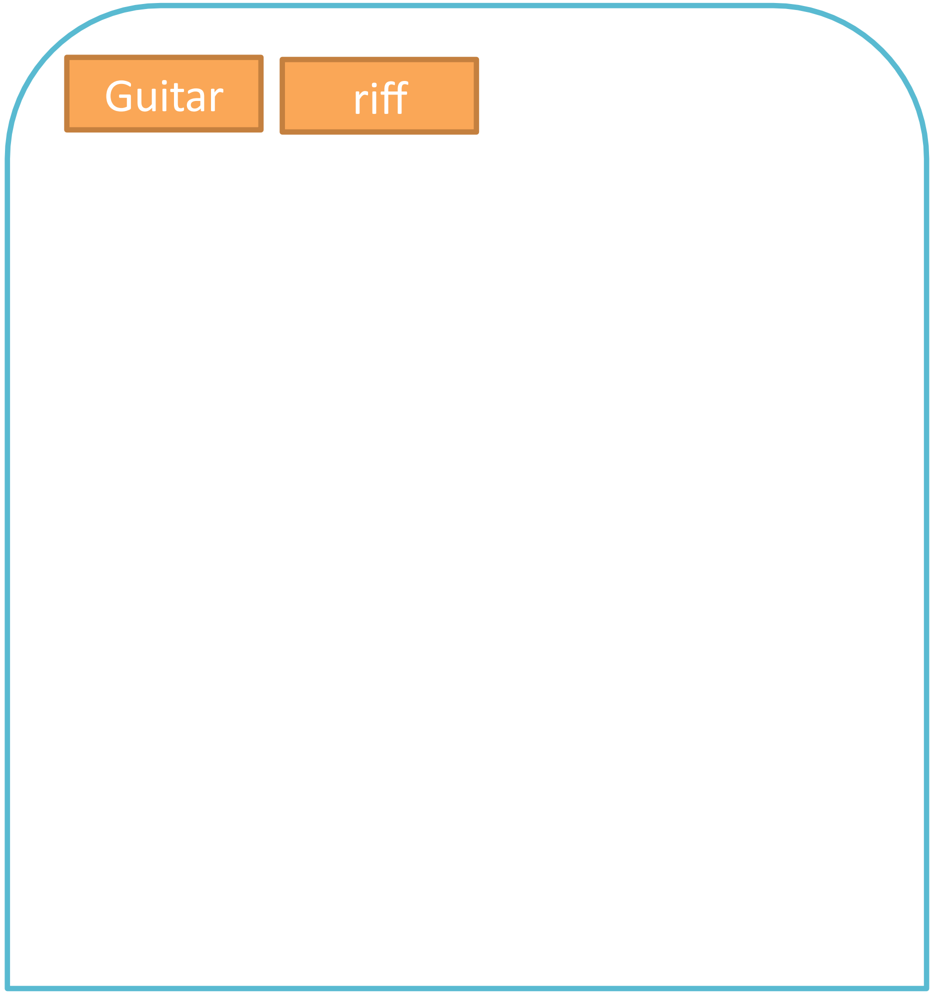
First, I’ll decide what topics to write on.
Choose the topic distribution.
Sex: 2%, Drugs: 33%, Rock’n Roll: 65%

Choose next word’s topic.
Roll the dice.

Topic Modeling: LDA

Let’s use an algorithm specifically developed to find topics.

Model the process of writing



Empty page: I’ll write a document.

First, I’ll decide what topics to write on.
Choose the topic distribution.
Sex: 2%, Drugs: 33%, Rock’n Roll: 65%

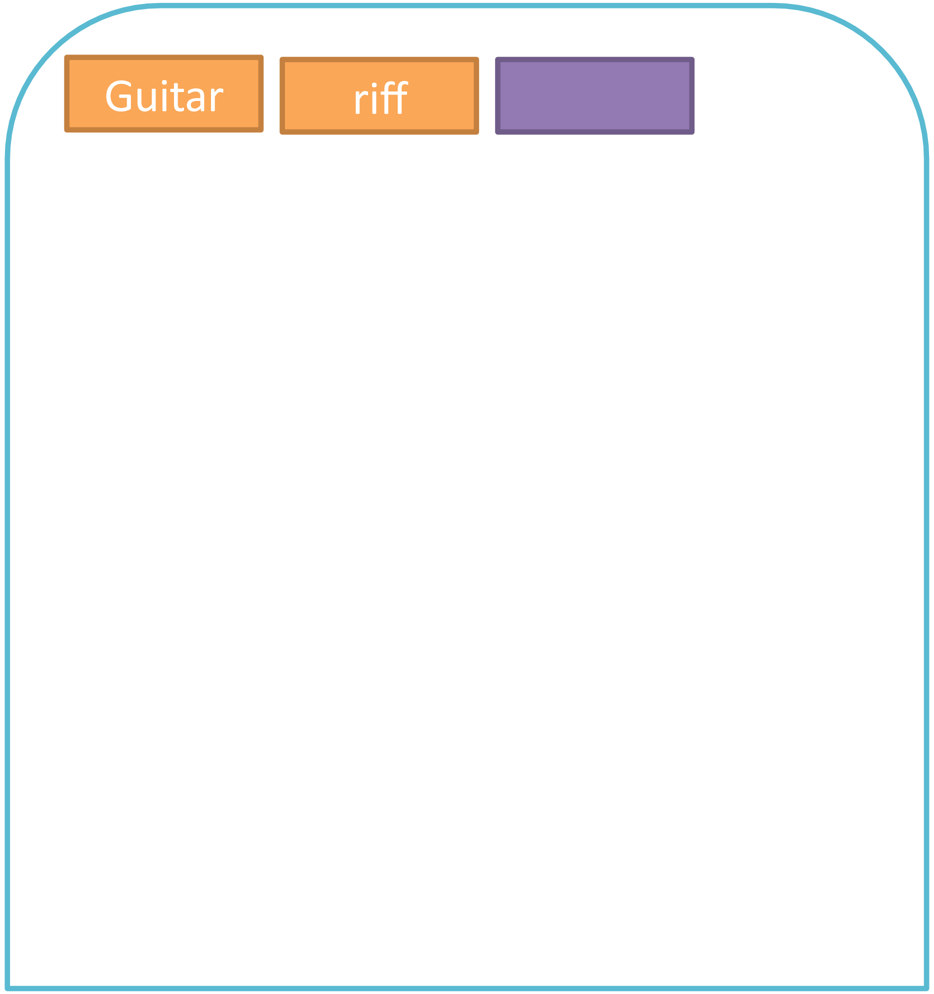
Choose next word’s topic.
Roll the dice.

Choose the word according to this topic.
Roll the dice.

Topic Modeling: LDA

Let's use an algorithm specifically developed to find topics.

Model the process of writing



Empty page: I'll write a document.

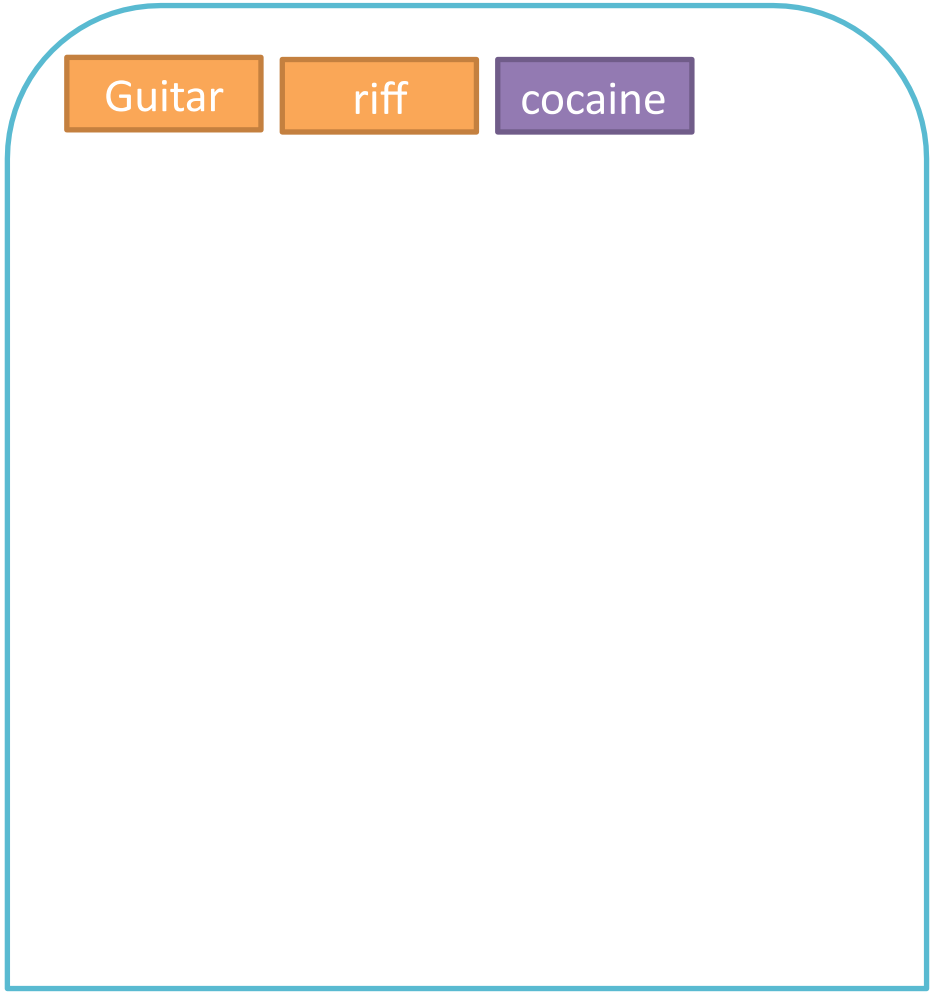
First, I'll decide what topics to write on.
Choose the topic distribution.
Sex: 2%, Drugs: 33%, Rock'n Roll: 65%

Choose next word's topic.
Roll the dice.

Topic Modeling: LDA

Let's use an algorithm specifically developed to find topics.

Model the process of writing



Empty page: I'll write a document.

First, I'll decide what topics to write on.
Choose the topic distribution.
Sex: 2%, Drugs: 33%, Rock'n Roll: 65%

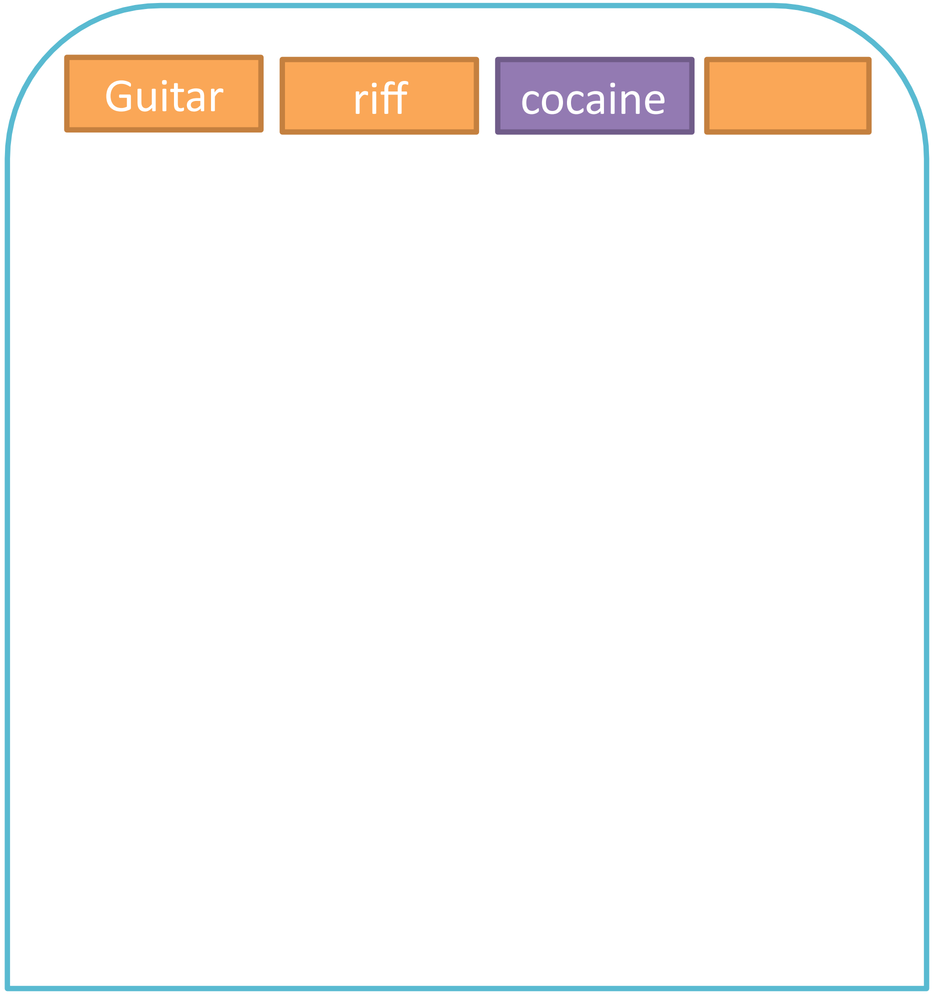
Choose next word's topic.
Roll the dice.

Choose the word according to this topic.
Roll the dice.

Topic Modeling: LDA

Let's use an algorithm specifically developed to find topics.

Model the process of writing



Empty page: I'll write a document.

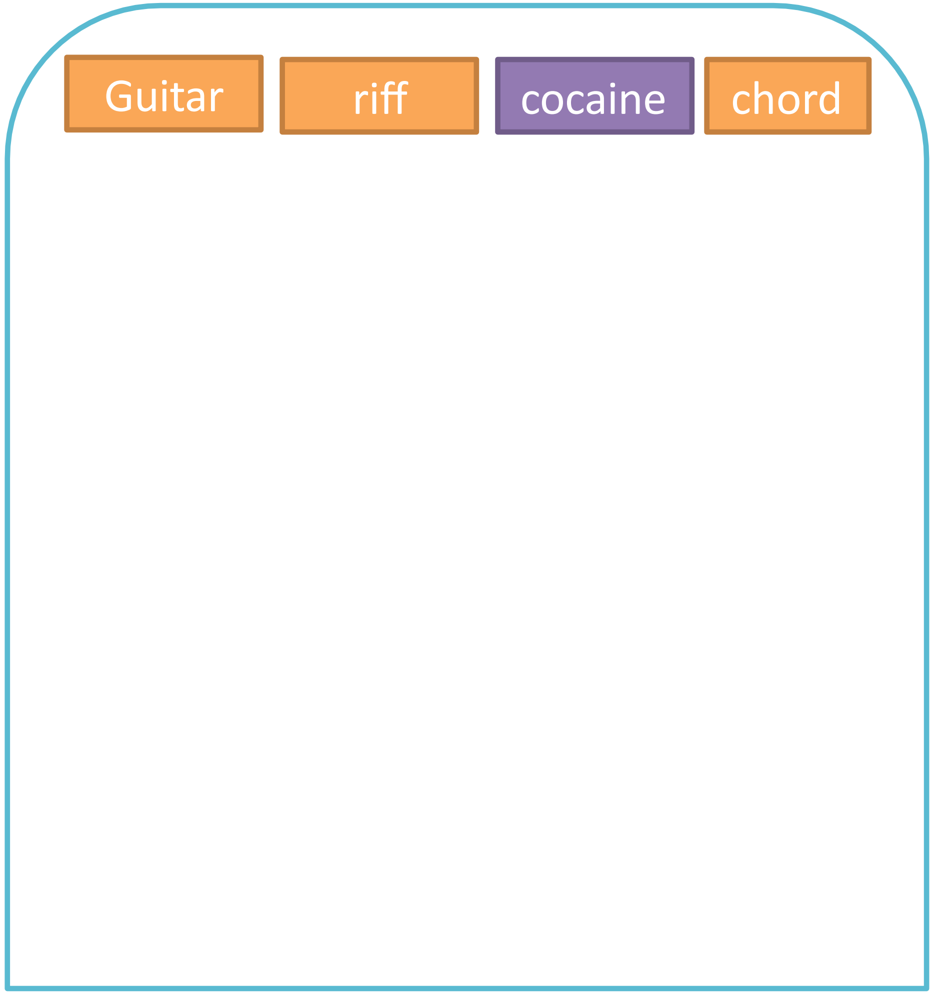
First, I'll decide what topics to write on.
Choose the topic distribution.
Sex: 2%, Drugs: 33%, Rock'n Roll: 65%

Choose next word's topic.
Roll the dice.

Topic Modeling: LDA

Let’s use an algorithm specifically developed to find topics.

Model the process of writing



Empty page: I’ll write a document.

First, I’ll decide what topics to write on.
Choose the topic distribution.
Sex: 2%, Drugs: 33%, Rock’n Roll: 65%

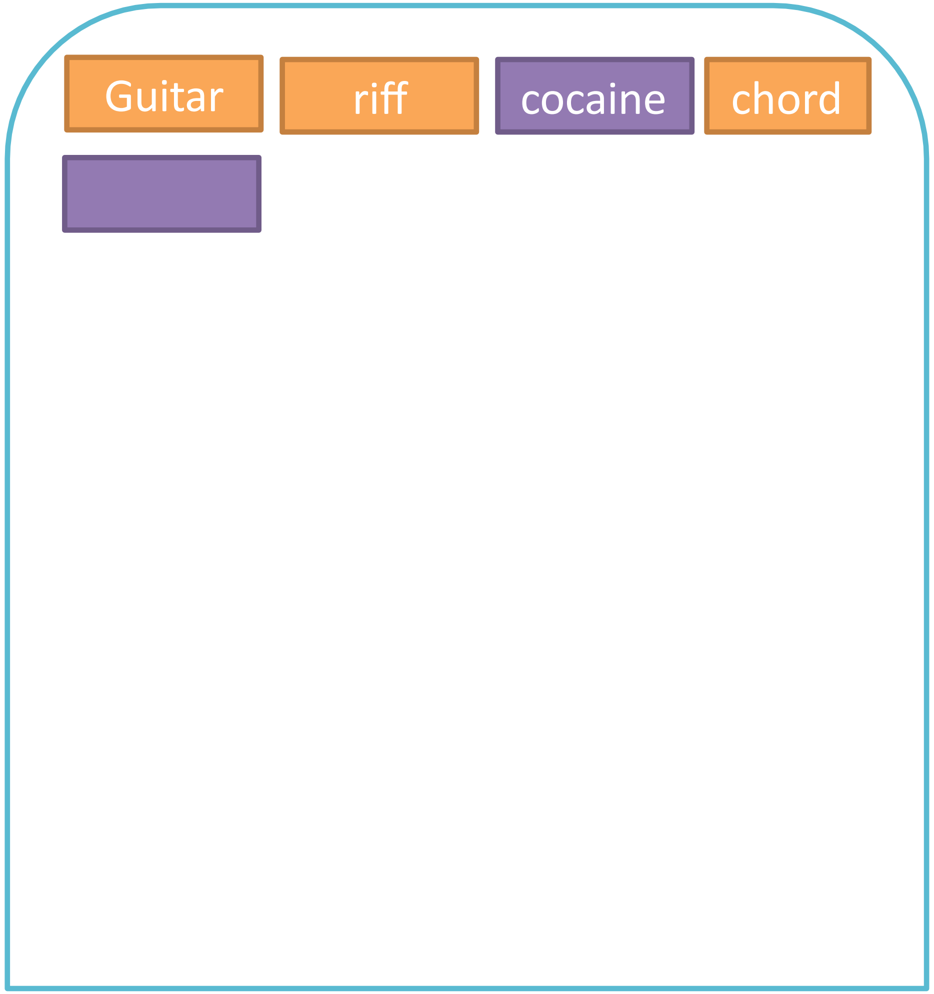
Choose next word’s topic.
Roll the dice.

Choose the word according to this topic.
Roll the dice.

Topic Modeling: LDA

Let's use an algorithm specifically developed to find topics.

Model the process of writing



Empty page: I'll write a document.

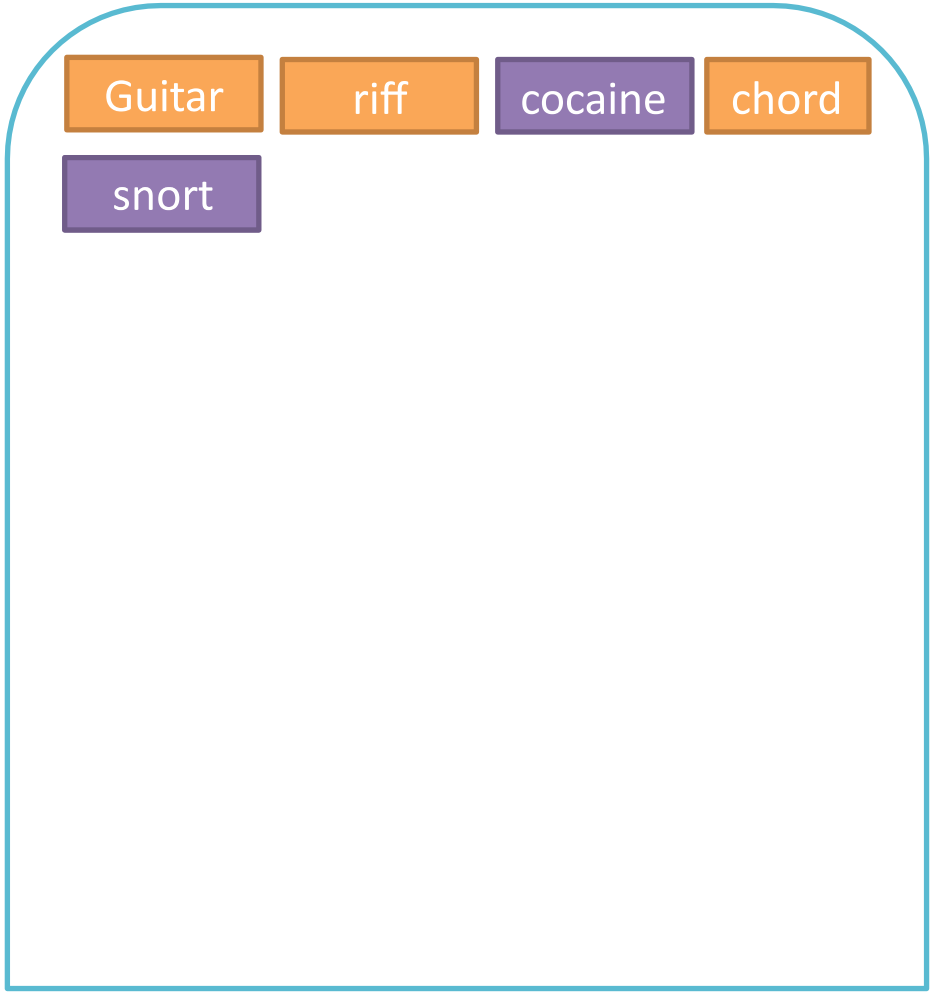
First, I'll decide what topics to write on.
Choose the topic distribution.
Sex: 2%, Drugs: 33%, Rock'n Roll: 65%

Choose next word's topic.
Roll the dice.

Topic Modeling: LDA

Let's use an algorithm specifically developed to find topics.

Model the process of writing



Empty page: I'll write a document.

First, I'll decide what topics to write on.
Choose the topic distribution.
Sex: 2%, Drugs: 33%, Rock'n Roll: 65%

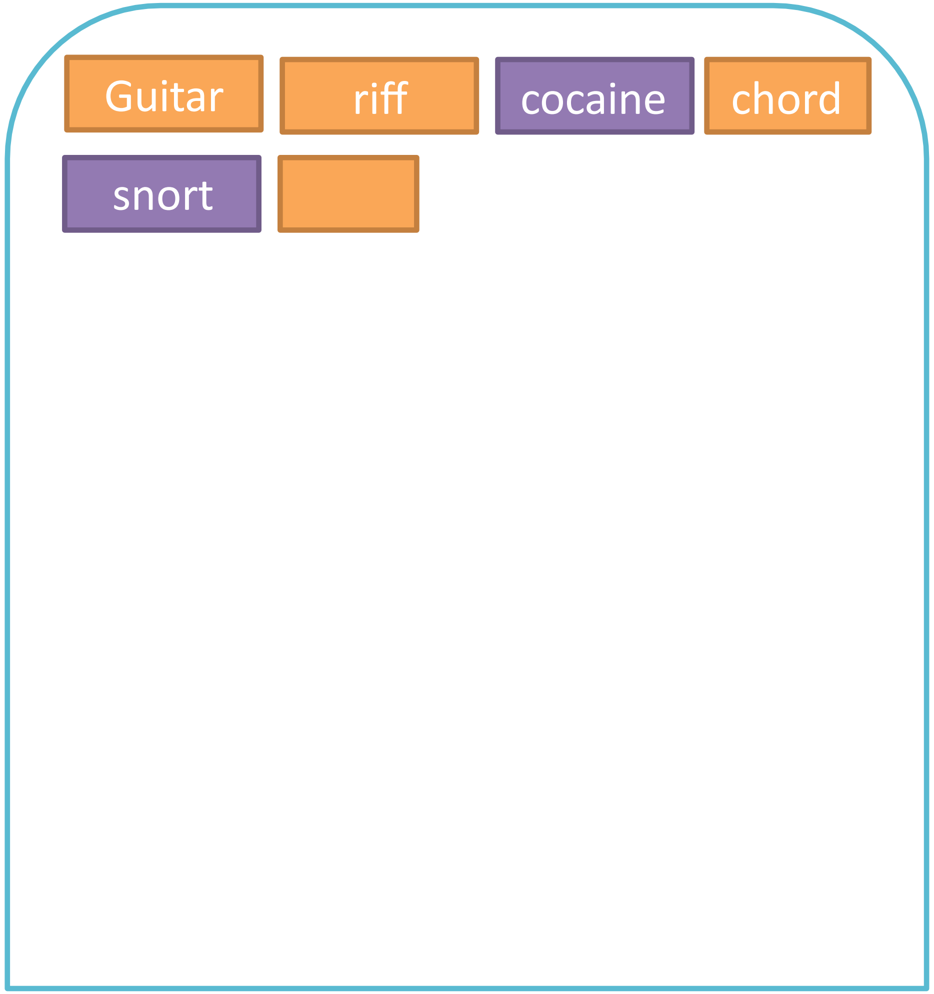
Choose next word's topic.
Roll the dice.

Choose the word according to this topic.
Roll the dice.

Topic Modeling: LDA

Let's use an algorithm specifically developed to find topics.

Model the process of writing



Empty page: I'll write a document.

First, I'll decide what topics to write on.

Choose the topic distribution.

Sex: 2%, Drugs: 33%, Rock'n Roll: 65%

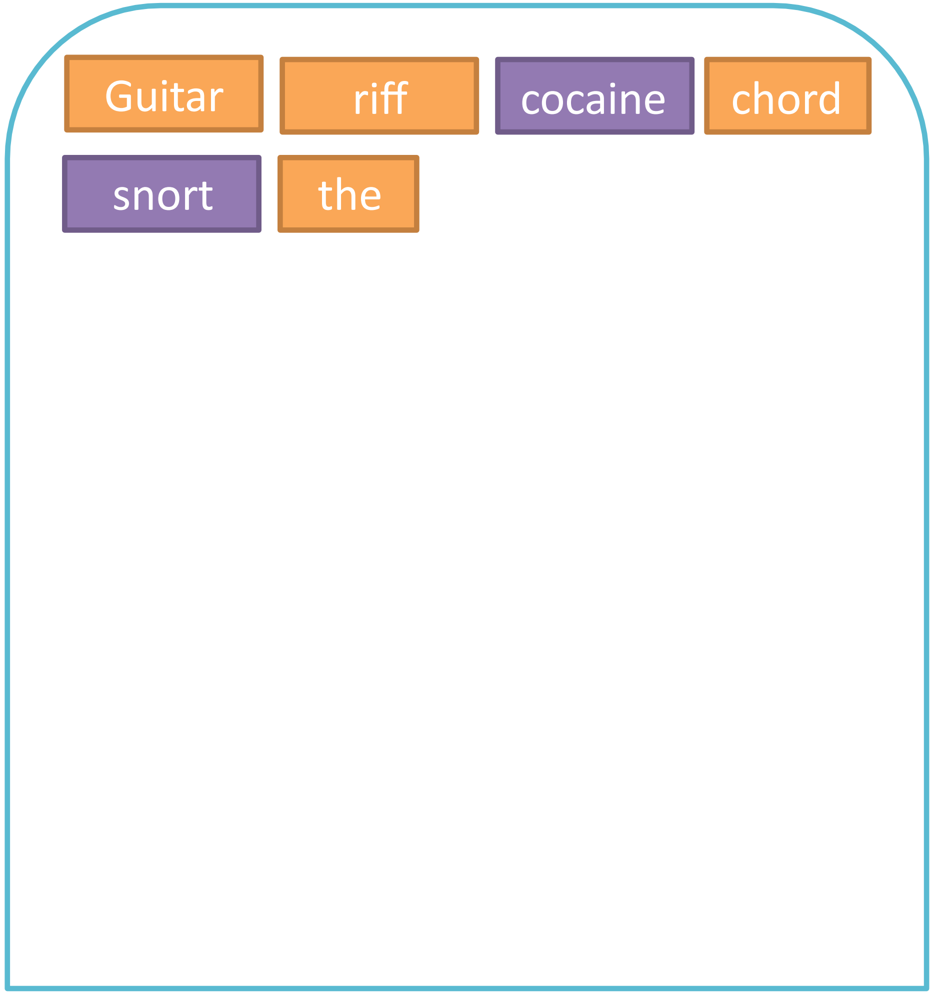
Choose next word's topic.

Roll the dice.

Topic Modeling: LDA

Let's use an algorithm specifically developed to find topics.

Model the process of writing



Empty page: I'll write a document.

First, I'll decide what topics to write on.
Choose the topic distribution.
Sex: 2%, Drugs: 33%, Rock'n Roll: 65%

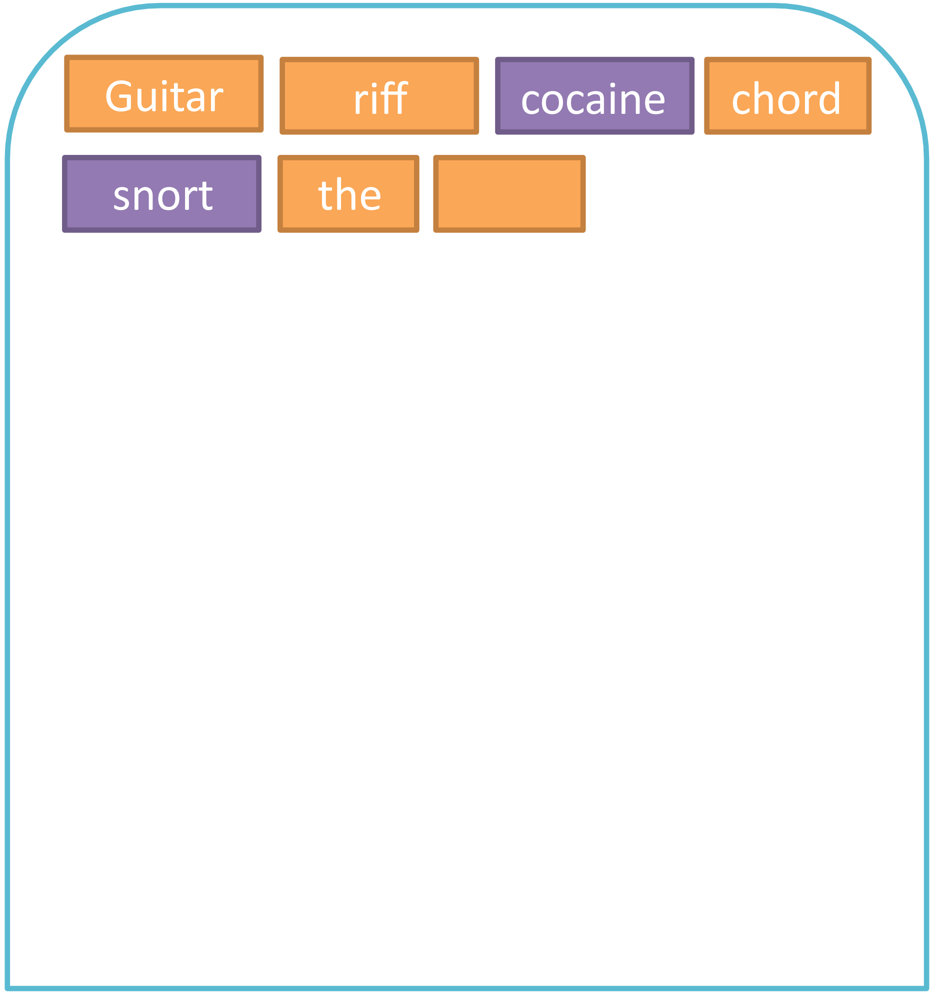
Choose next word's topic.
Roll the dice.

Choose the word according to this topic.
Roll the dice.

Topic Modeling: LDA

Let's use an algorithm specifically developed to find topics.

Model the process of writing



Empty page: I'll write a document.

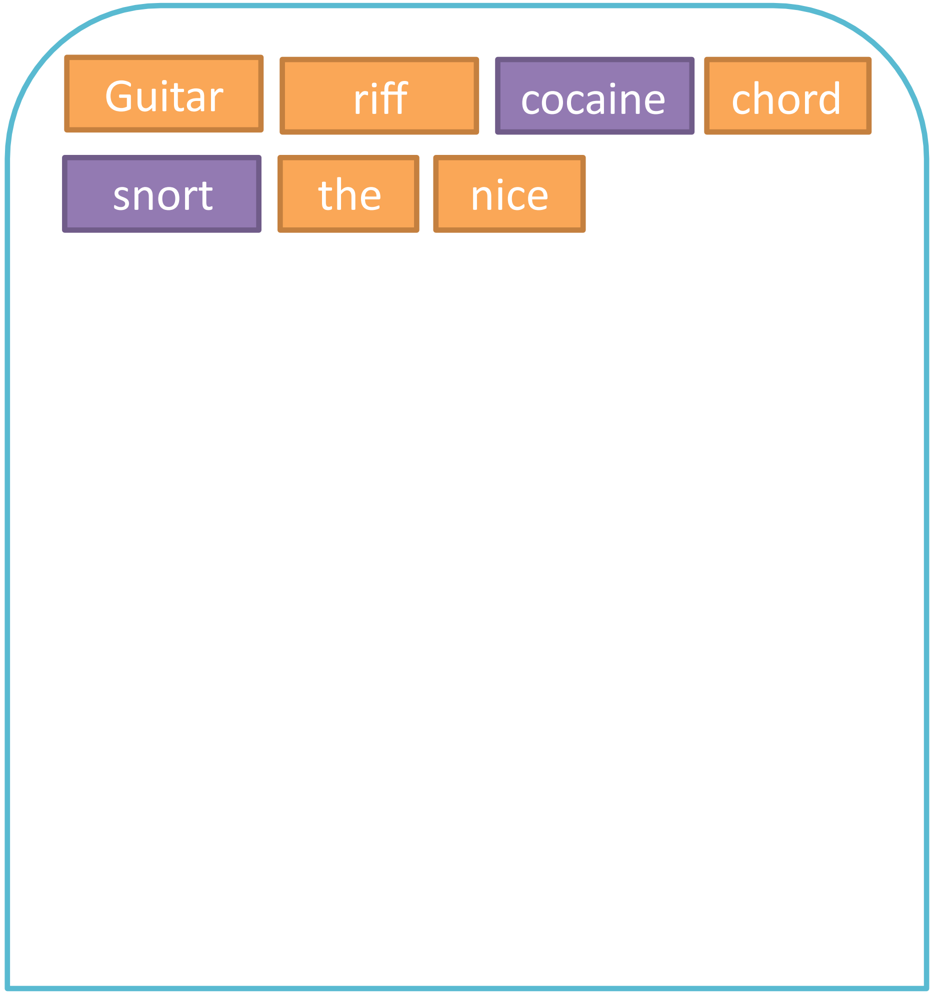
First, I'll decide what topics to write on.
Choose the topic distribution.
Sex: 2%, Drugs: 33%, Rock'n Roll: 65%

Choose next word's topic.
Roll the dice.

Topic Modeling: LDA

Let's use an algorithm specifically developed to find topics.

Model the process of writing



Empty page: I'll write a document.

First, I'll decide what topics to write on.
Choose the topic distribution.
Sex: 2%, Drugs: 33%, Rock'n Roll: 65%

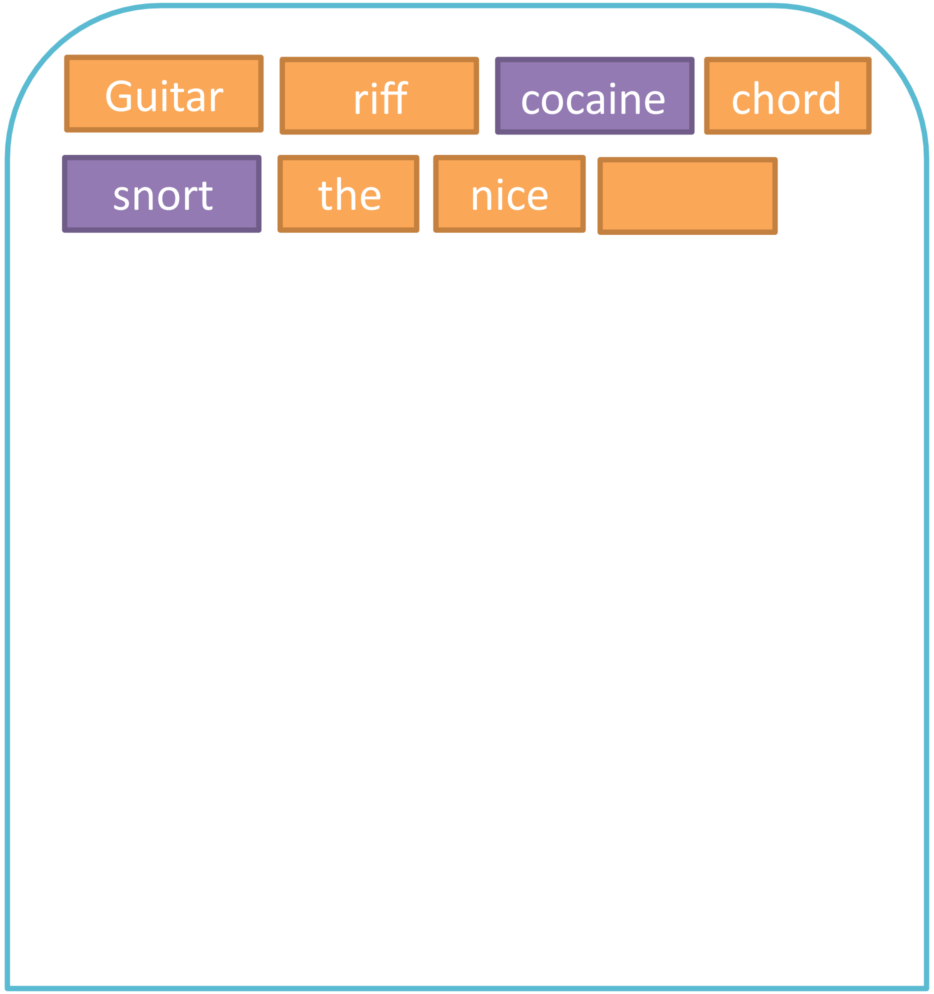
Choose next word's topic.
Roll the dice.

Choose the word according to this topic.
Roll the dice.

Topic Modeling: LDA

Let's use an algorithm specifically developed to find topics.

Model the process of writing



Empty page: I'll write a document.

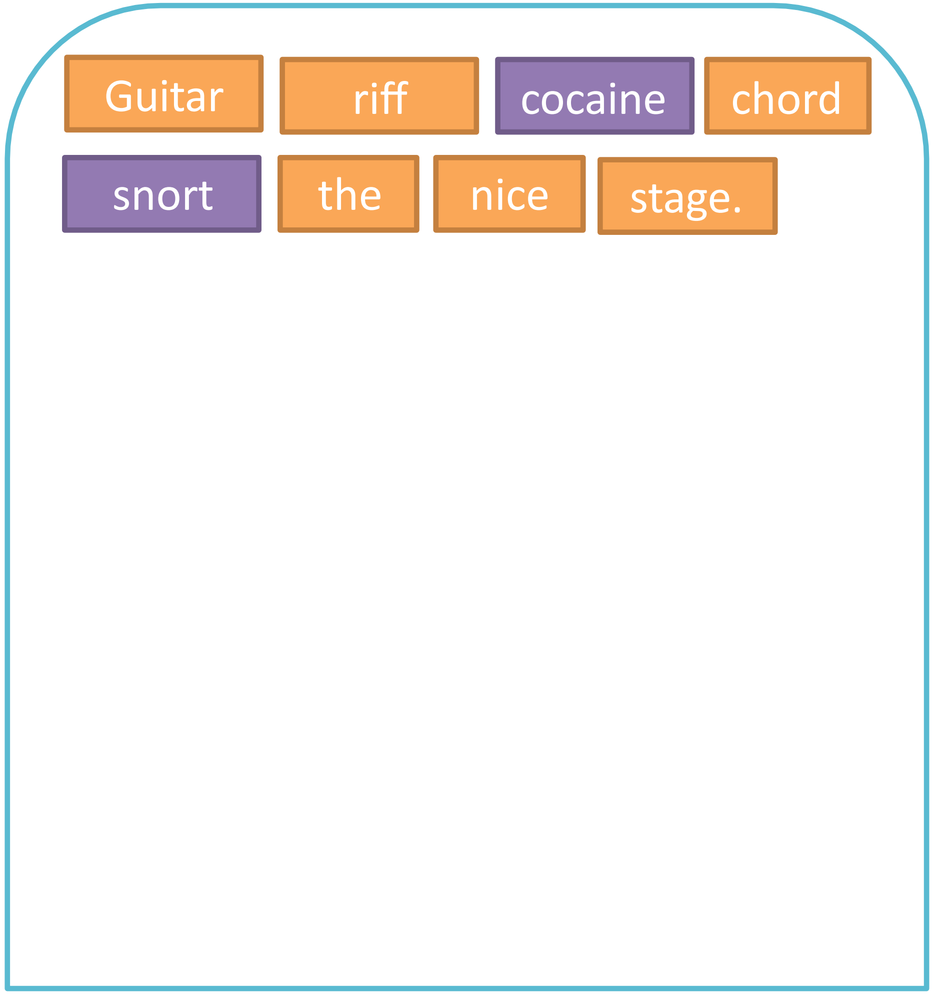
First, I'll decide what topics to write on.
Choose the topic distribution.
Sex: 2%, Drugs: 33%, Rock'n Roll: 65%

Choose next word's topic.
Roll the dice.

Topic Modeling: LDA

Let's use an algorithm specifically developed to find topics.

Model the process of writing



Empty page: I'll write a document.

First, I'll decide what topics to write on.
Choose the topic distribution.
Sex: 2%, Drugs: 33%, Rock'n Roll: 65%

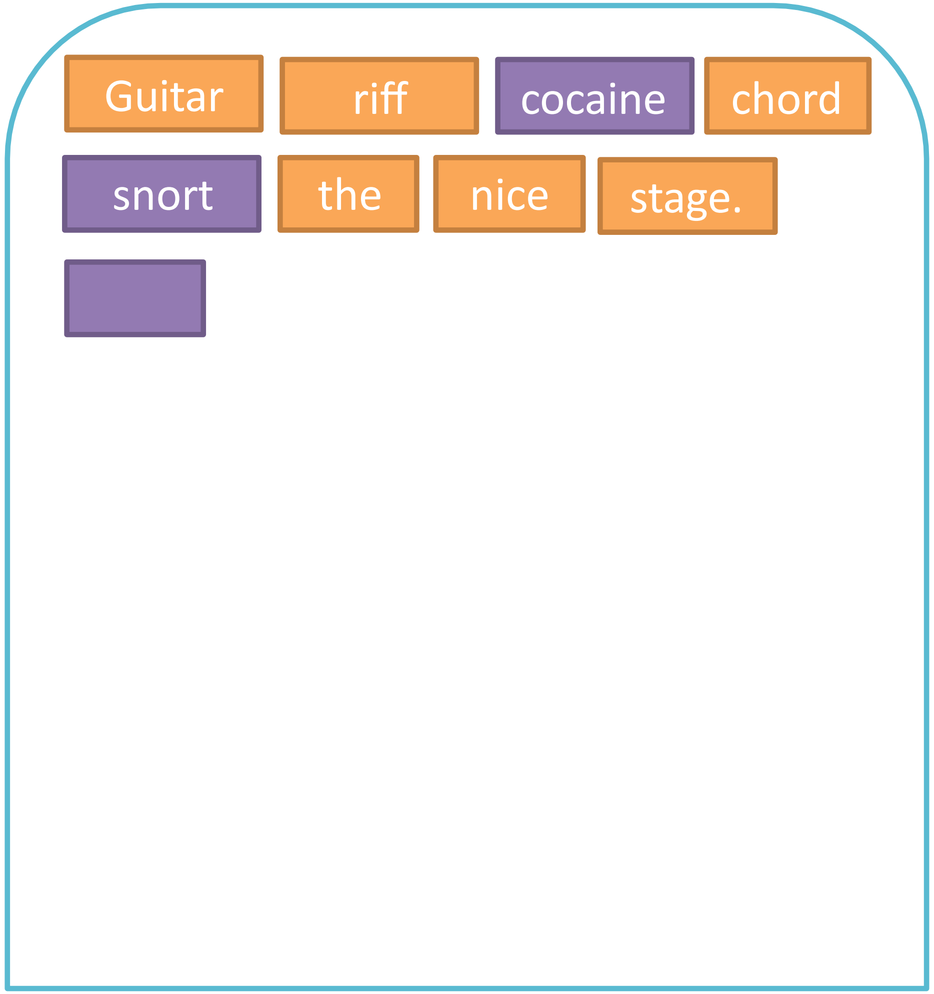
Choose next word's topic.
Roll the dice.

Choose the word according to this topic.
Roll the dice.

Topic Modeling: LDA

Let's use an algorithm specifically developed to find topics.

Model the process of writing



Empty page: I'll write a document.

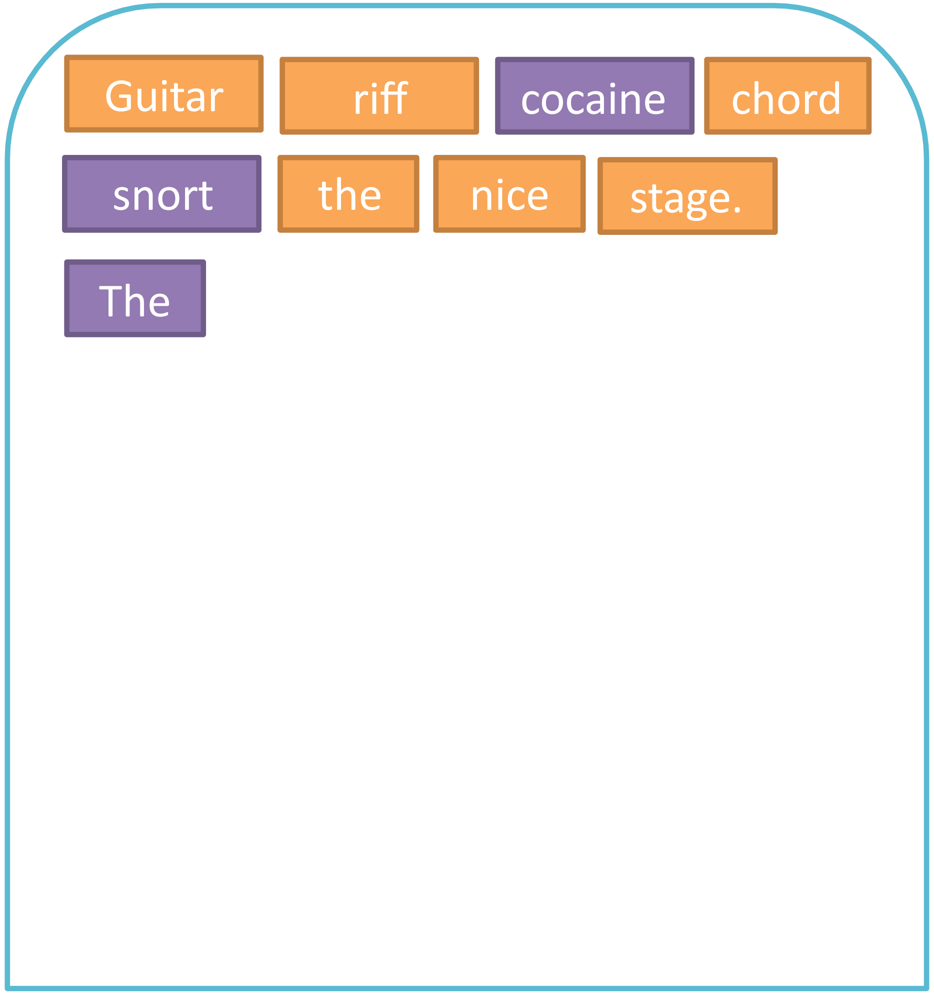
First, I'll decide what topics to write on.
Choose the topic distribution.
Sex: 2%, Drugs: 33%, Rock'n Roll: 65%

Choose next word's topic.
Roll the dice.

Topic Modeling: LDA

Let's use an algorithm specifically developed to find topics.

Model the process of writing



Empty page: I'll write a document.

First, I'll decide what topics to write on.
Choose the topic distribution.
Sex: 2%, Drugs: 33%, Rock'n Roll: 65%

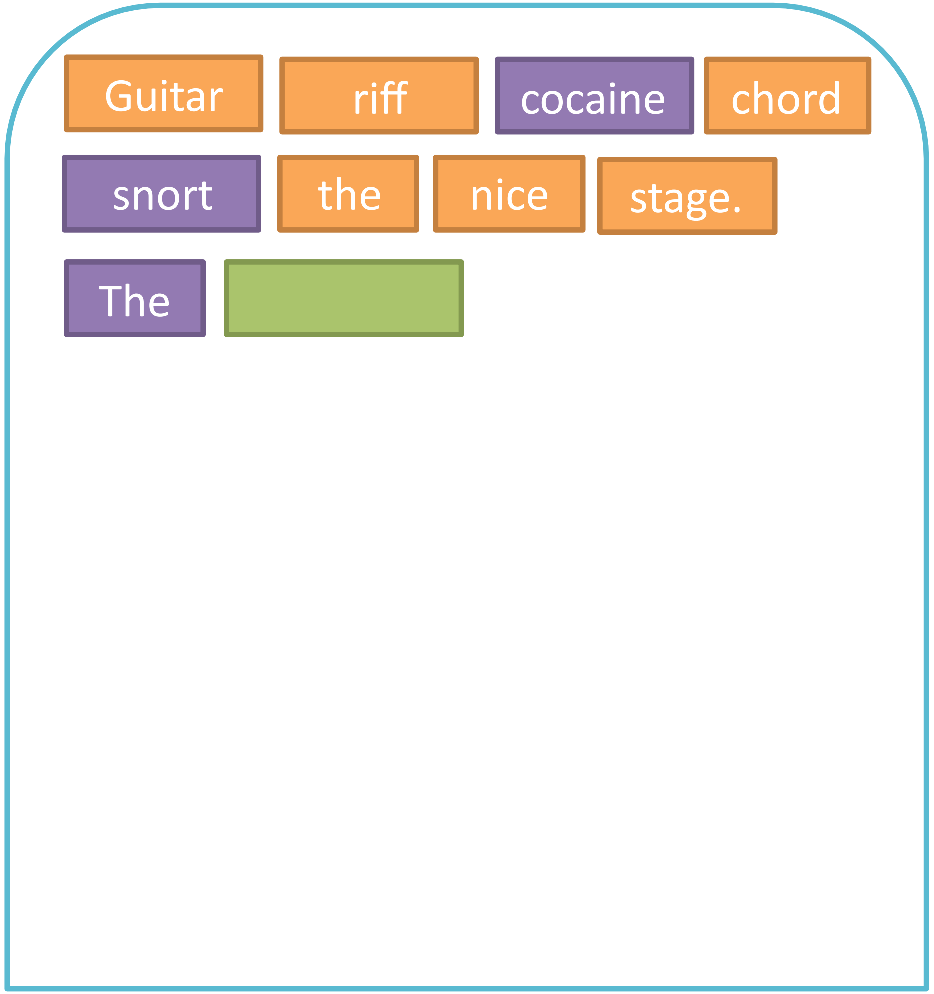
Choose next word's topic.
Roll the dice.

Choose the word according to this topic.
Roll the dice.

Topic Modeling: LDA

Let's use an algorithm specifically developed to find topics.

Model the process of writing



Empty page: I'll write a document.

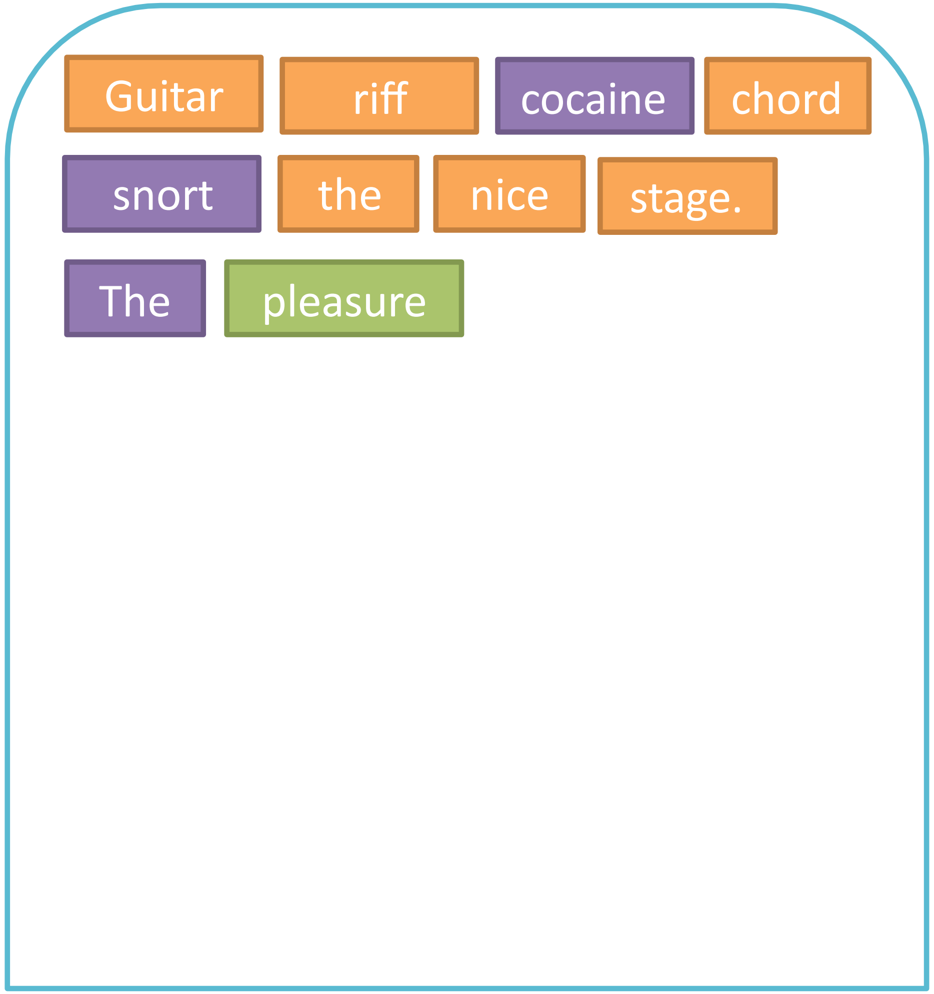
First, I'll decide what topics to write on.
Choose the topic distribution.
Sex: 2%, Drugs: 33%, Rock'n Roll: 65%

Choose next word's topic.
Roll the dice.

Topic Modeling: LDA

Let's use an algorithm specifically developed to find topics.

Model the process of writing



Empty page: I'll write a document.

First, I'll decide what topics to write on.
Choose the topic distribution.
Sex: 2%, Drugs: 33%, Rock'n Roll: 65%

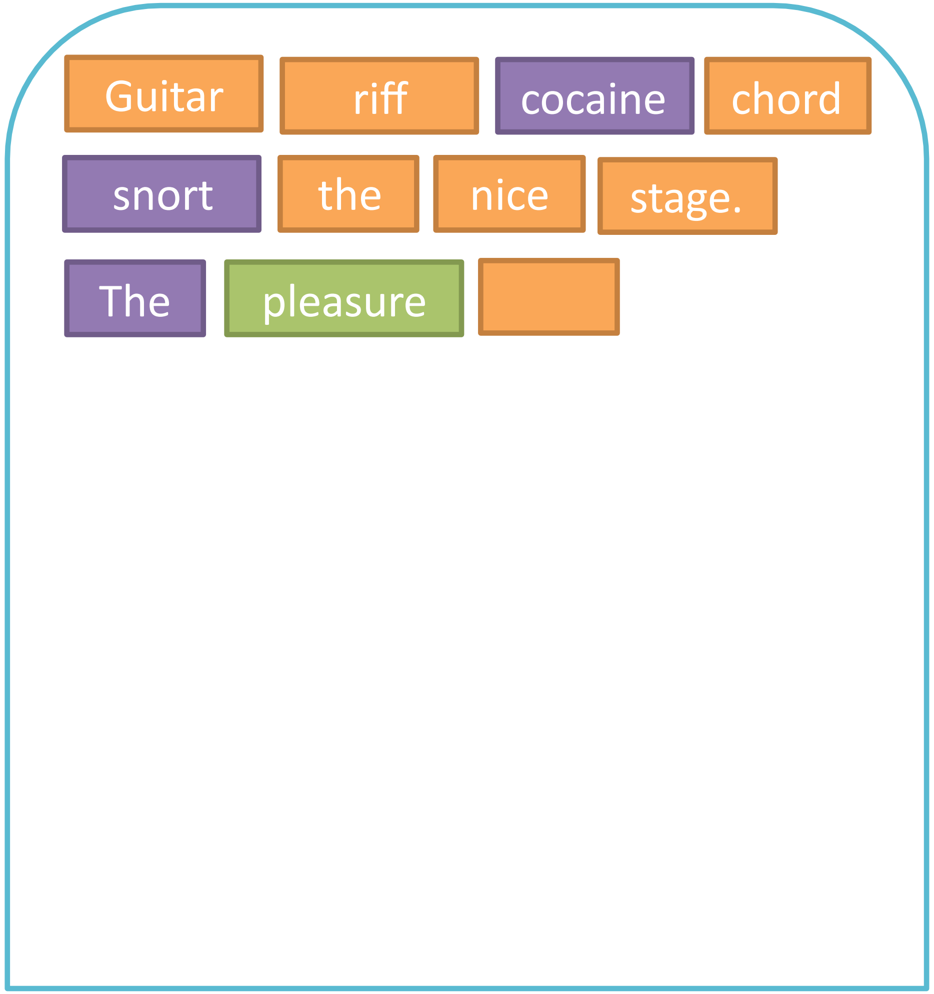
Choose next word's topic.
Roll the dice.

Choose the word according to this topic.
Roll the dice.

Topic Modeling: LDA

Let's use an algorithm specifically developed to find topics.

Model the process of writing



Empty page: I'll write a document.

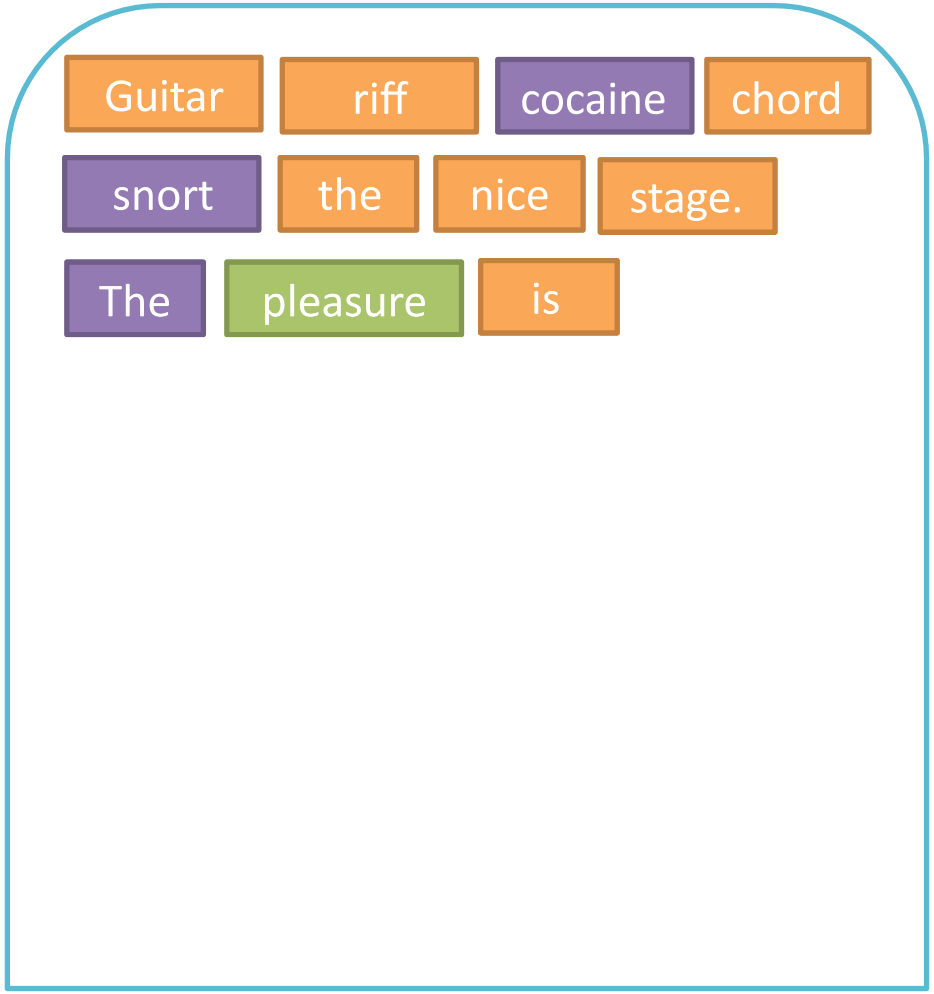
First, I'll decide what topics to write on.
Choose the topic distribution.
Sex: 2%, Drugs: 33%, Rock'n Roll: 65%

Choose next word's topic.
Roll the dice.

Topic Modeling: LDA

Let's use an algorithm specifically developed to find topics.

Model the process of writing



Empty page: I'll write a document.

First, I'll decide what topics to write on.
Choose the topic distribution.
Sex: 2%, Drugs: 33%, Rock'n Roll: 65%

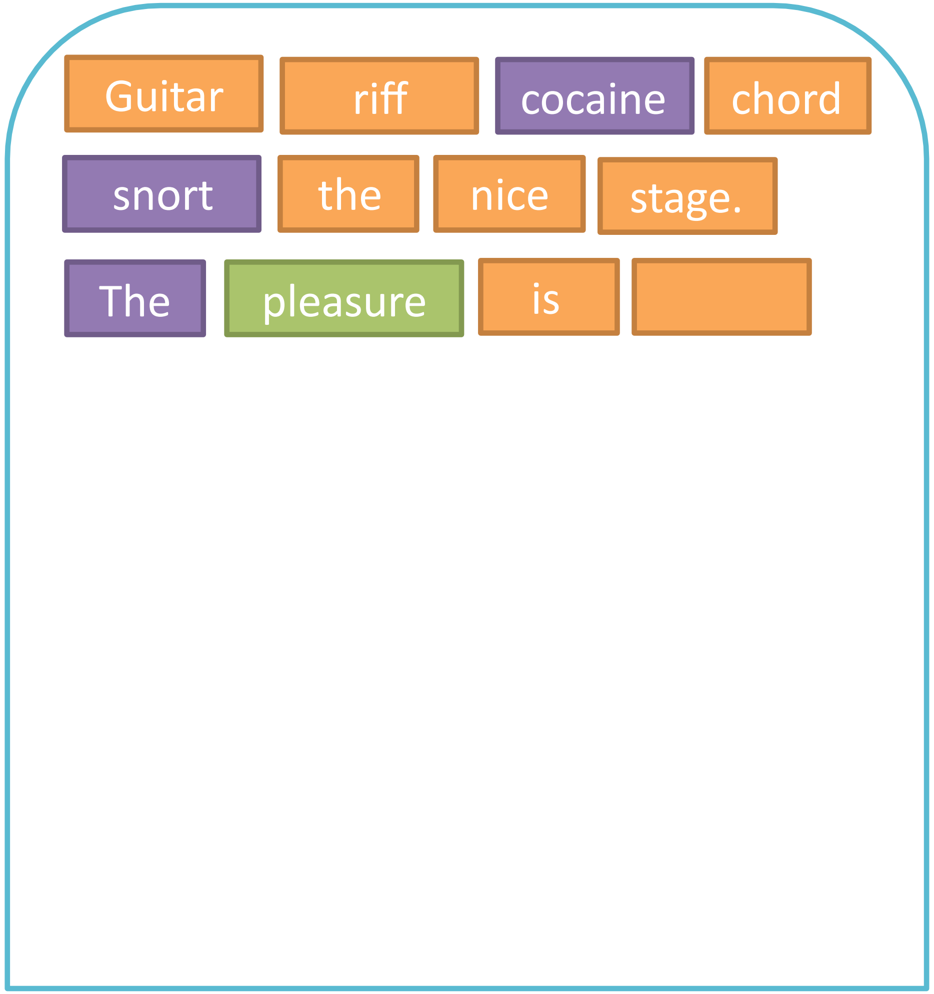
Choose next word's topic.
Roll the dice.

Choose the word according to this topic.
Roll the dice.

Topic Modeling: LDA

Let’s use an algorithm specifically developed to find topics.

Model the process of writing



Empty page: I’ll write a document.

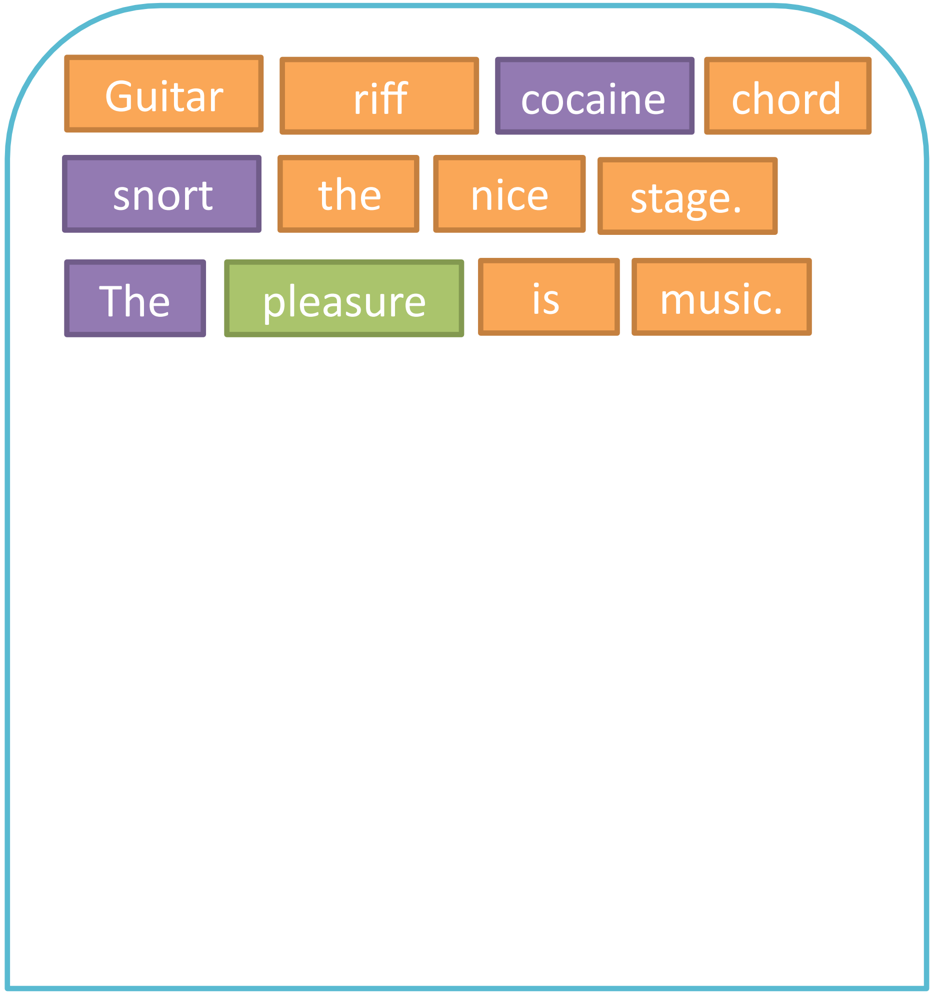
First, I’ll decide what topics to write on.
Choose the topic distribution.
Sex: 2%, Drugs: 33%, Rock’n Roll: 65%

Choose next word’s topic.
Roll the dice.

Topic Modeling: LDA

Let’s use an algorithm specifically developed to find topics.

Model the process of writing



Empty page: I’ll write a document.

First, I’ll decide what topics to write on.
Choose the topic distribution.
Sex: 2%, Drugs: 33%, Rock’n Roll: 65%

Choose next word’s topic.
Roll the dice.

Choose the word according to this topic.
Roll the dice.

LDA DOES THIS IN REVERSE

- Start with a corpus of documents

- Assume some topic distribution in each document
- Assume some word distribution in each topic

- Look at the corpus and try to find what topic and word distributions would be most likely to generate that data

How LDA Works

Perspective #2

LDA STEP BY STEP

- **Goal:** You want LDA to learn the topic mix in each document, and the word mix in each topic.

LDA STEP BY STEP

- **Goal:** You want LDA to learn the topic mix in each document, and the word mix in each topic.
- Choose the number of topics you think there are in your corpus
  Example: K = 2

LDA STEP BY STEP

Document #1

Topic A: Food

Topic B: Animals

- **Goal:** You want LDA to learn the topic mix in each document, and the word mix
- Choose the number of topics you think there are in your corpus
  Example: K = 2
- Randomly assign each word in each document to one of 2 topics
  Example: The word ‘banana’ in Document #1 is randomly assigned to Topic B (animal-like topic)

LDA STEP BY STEP

Document #1

Topic A: **Food**

Topic B: **Animals**

- **Goal:** You want LDA to learn the topic mix in each document, and the word mix
- Choose the number of topics you think there are in your corpus
  Example: K = 2
- Randomly assign each word in each document to one of 2 topics
  Example: The word ‘banana’ in Document #1 is randomly assigned to Topic B (animal-like topic)
- Go through every word and its topic assignment in each document. Look at (1) how often the topic occurs in the document and (2) how often the word occurs in the topic overall. Based on this info, assign the word a new topic.
  Example: It looks like (1) animals don’t occur often in Document #1 and (2) ‘banana’ doesn’t occur much in Topic B, so the word ‘banana’ should be assigned to Topic A instead

LDA STEP BY STEP

Document #1

Topic A: Food

Topic B: Animals

- **Goal:** You want LDA to learn the topic mix in each document, and the word mix
- Choose the number of topics you think there are in your corpus
  Example: K = 2
- Randomly assign each word in each document to one of 2 topics
  Example: The word ‘banana’ in Document #1 is randomly assigned to Topic B (animal-like topic)
- Go through every word and its topic assignment in each document. Look at (1) how often the topic occurs in the document and (2) how often the word occurs in the topic overall. Based on this info, assign the word a new topic.
  Example: It looks like (1) animals don’t occur often in Document #1 and (2) ‘banana’ doesn’t occur much in Topic B, so the word ‘banana’ should be assigned to Topic A instead
- Go through multiple iterations of this. Eventually, the topics will start making sense
  Interpret them.

LDA STEP BY STEP

Document #1

Topic A: Food

Topic B: Animals

- **Goal:** You want LDA to learn the topic mix in each document, and the word mix
- Choose the number of topics you think there are in your corpus
  Example: K = 2

You can use a Python library like gensim to do this part for you.

- Go through multiple iterations of this. Eventually, the topics will start making sense
  Interpret them.

LDA IN PYTHON

- **Inputs**
  - Term-Document Matrix
  - Dictionary of Words

- **Parameters**
  - LDA Specific(# Topics, # Passes…)
  - Text Preprocessing (stop words, min / max doc freq, parts of speech, bi-grams…)

- **Outputs**
  - Word Distribution in Each Topic
  - Topic Distribution in Each Document

TOPIC MODELING SUMMARY

WAYS TO USE TOPIC MODELING

- **Exploratory Data Analysis**
  - Example: Look at how the topic distribution for documents change over time
  - Example: If reduced to 2 or 3 dimensions, can visualize documents on a plot
- **Supervised Learning**
  - Use to reduce dimensions before applying a regression or classification technique
  - Example: Instead of inputting 100 words (100 features) into a spam classification model, input in 5 topics (5 features) into the model instead
- **Unsupervised Learning**
  - Use to reduce dimensions before determining how similar documents are
  - Example: Two articles may look very different in terms of words, but when represented as topic distributions, can look more similar (doc 1: 90% sports vs doc 2: 95% sports)

TOPIC MODELING WORKFLOW

1. Choose an algorithm (LSA, NMF, LDA)
2. Transform your data from the word space to the latent topic space
3. Each axis in the latent space represents a topic - it is your job as a human to interpret them
4. Tune the parameters of the algorithms until the topics make sense
5. Once your topics make sense, you are done

LSA vs NMF vs LDA

- LSA and NMF tend to work better on smaller documents (tweets)
- LDA tends to work better on larger documents (books)
- Try multiple techniques on your data set

- This is an iterative process - your topics likely won't make sense on the first try
- Apply a technique, look at the results, and continuously tweak your parameters until the topics make sense

QUESTIONS?

LET'S GO TO AN LDA EXERCISE!