DATASCI 510 - Lesson 7

Feature Engineering:

Natural Language Processing

PROFESSIONAL & CONTINUING EDUCATION
UNIVERSITY of WASHINGTON

Lesson 7 Agenda

- A reminder of the Wordle interview question
  - Bootstrap compared to the analytical comparison
- Text processing: TF-IDF
- Lesson_07_a_TF-IDF.ipynb
- Break
- Lesson_07_b_student.ipynb
- Break
- Classification Accuracy (Confusion Matrix, and RoC)
- Lesson_07_c_RFM.ipynb (Time Permitting)
- Interview question

Term Frequency Inverse Document Frequency

How to calculate the TF-IDF from a *corpus*

Term Frequency-Inverse Document Frequency Matrix (TF-IDF)

The **T**erm **F**requency - **I**nverse **D**ocument **F**requency matrix (TF-IDF) is a matrix where each row represents one document and each feature is a term (word) that may appear in a document.

TF-IDF can be used in machine learning to create a model. These models might predict something about a document or describe a collection of documents (corpus).

TF-IDF matrix

	voice	actor	traffic	bribe	study	home
Doc 1	0	0	0.22	0.15	0	0
Doc 2	0.03	0.07	0	0	0	0.11
Doc 3	0	0	0.32	0.07	0	0
Doc 4	0.05	0.07	0	0	0	0.05
Doc 5	0.08	0	0	0	0	0
Doc 6	0.02	0.11	0	0.07	0	0
Doc 7	0.12	0	0	0	0	0

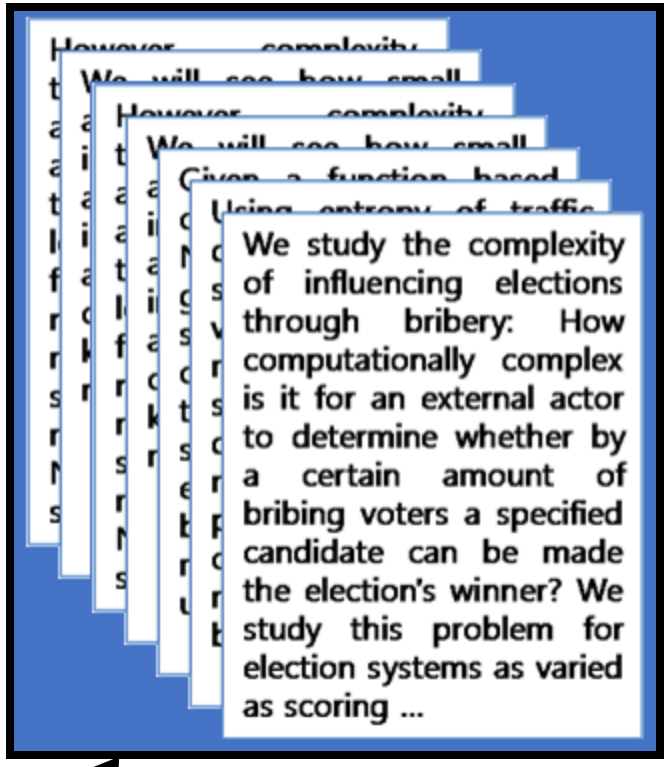
Glossary

- Document: Written text like a book or a tweet
- Corpus: A collection of documents
- DTM: **D**ocument **T**erm **M**atrix
- TF: **T**erm **F**requency matrix
- IDF: **I**nverse **D**ocument **F**requency matrix
- TF-IDF: **T**erm **F**requency - **I**nverse **D**ocument **F**requency matrix

Steps in Creating the Term-Frequency-Inverse Document Frequency (TF-IDF) Matrix

1. Get a collection of documents (corpus) like books or tweets
2. Calculate **D**ocument-**T**erm **M**atrix (DTM) from the corpus
3. Calculate **T**erm **F**requencies (TF) from the DTM
4. Calculate **I**nverse **D**ocument **F**requencies (IDF) from the DTM
5. Calculate the **T**erm **F**requency-**I**nverse **D**ocument **F**requency (TF-IDF) Matrix from the TF and IDF

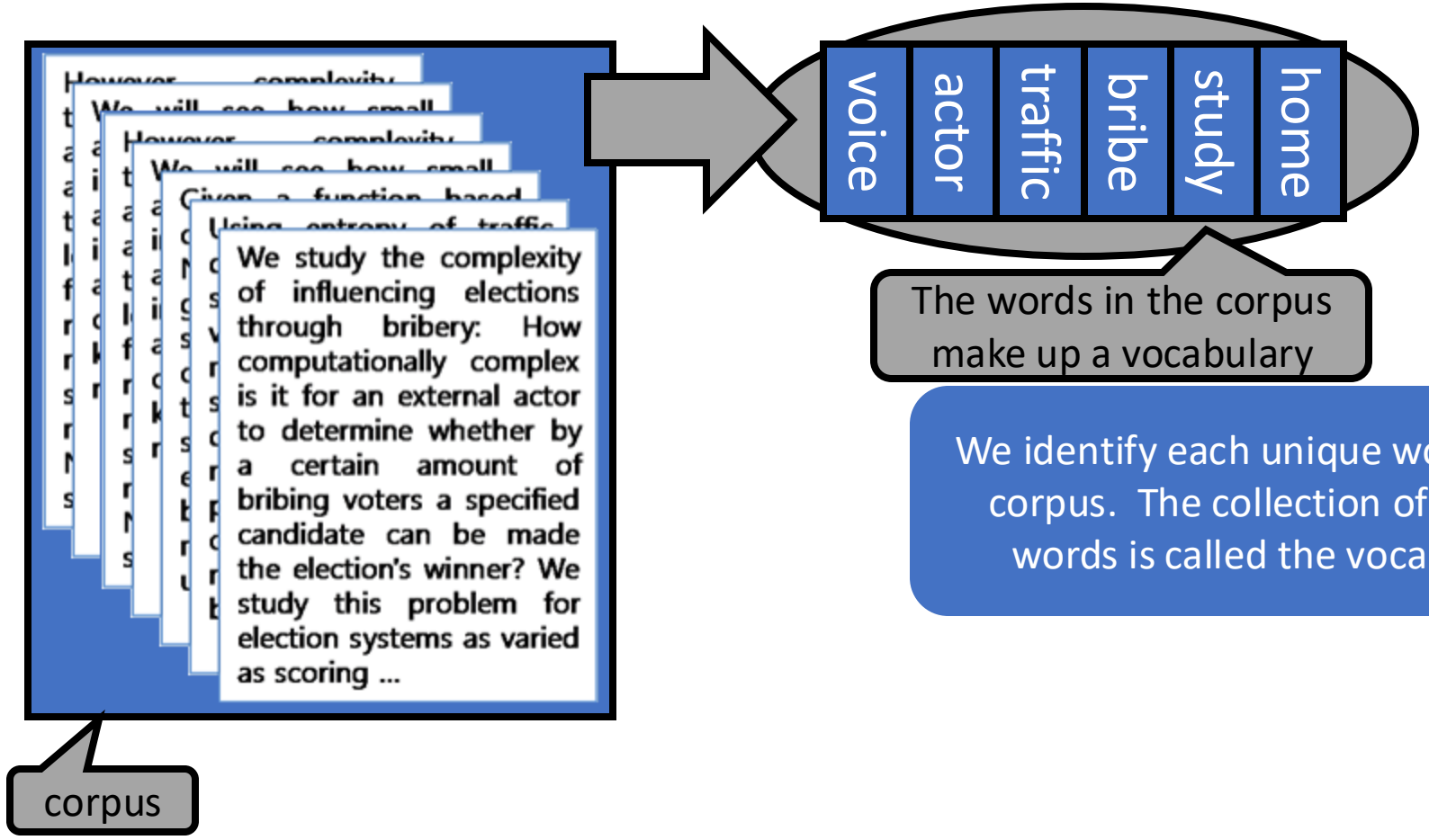
Document-Term Matrix (DTM)



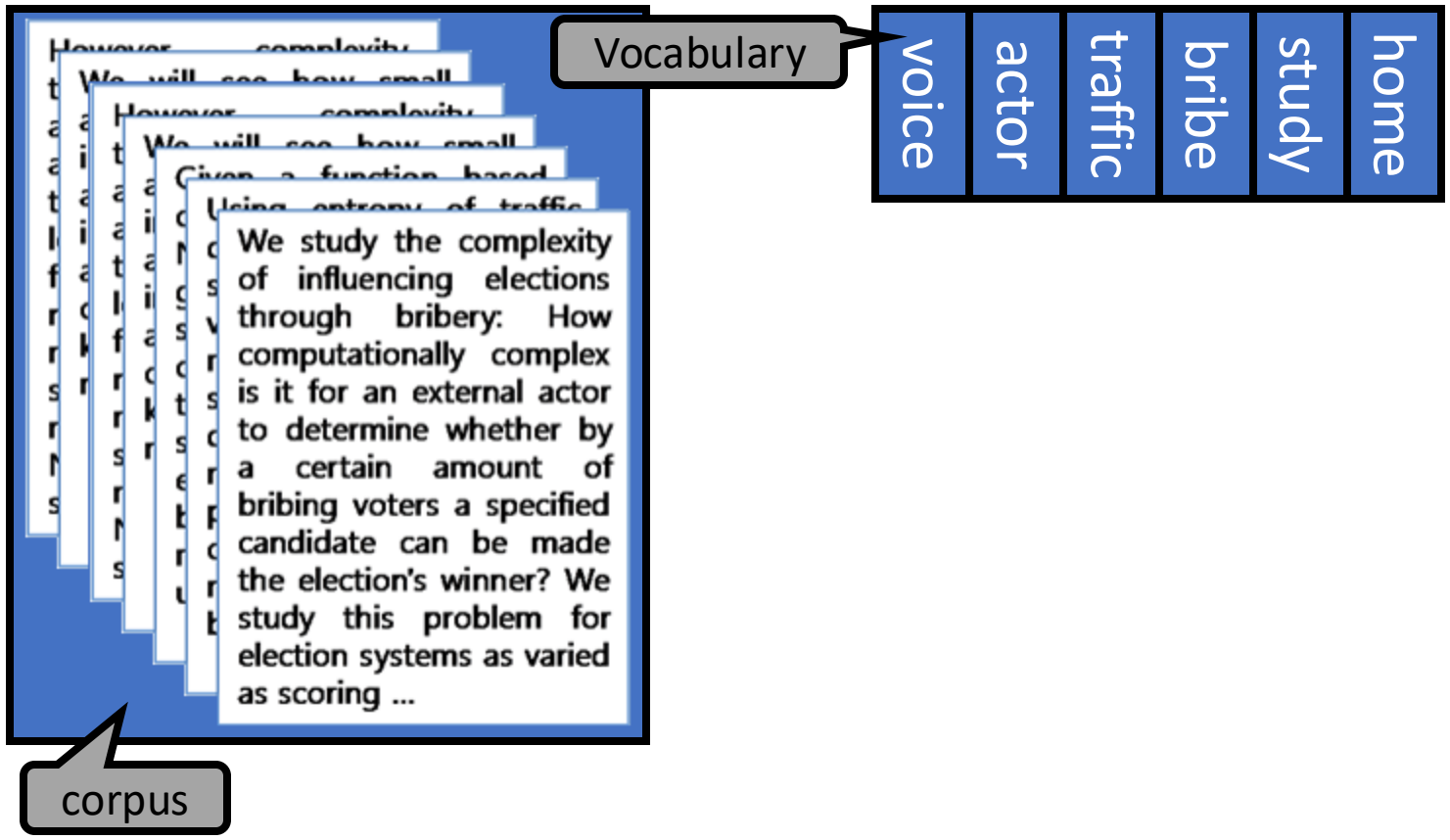
A collection of documents is called a corpus

We start with a collection of documents, also called a *corpus*

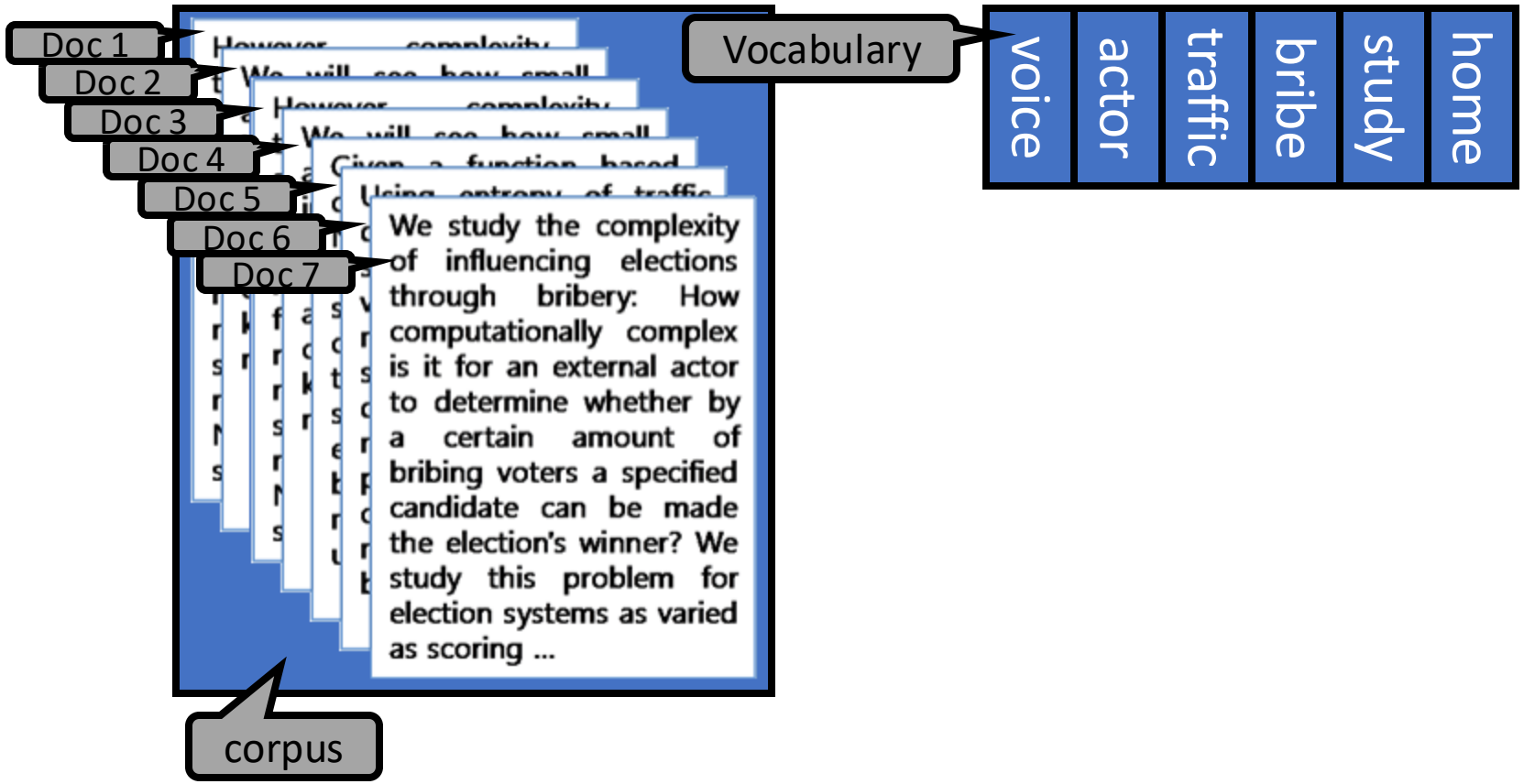
Document-Term Matrix (DTM)



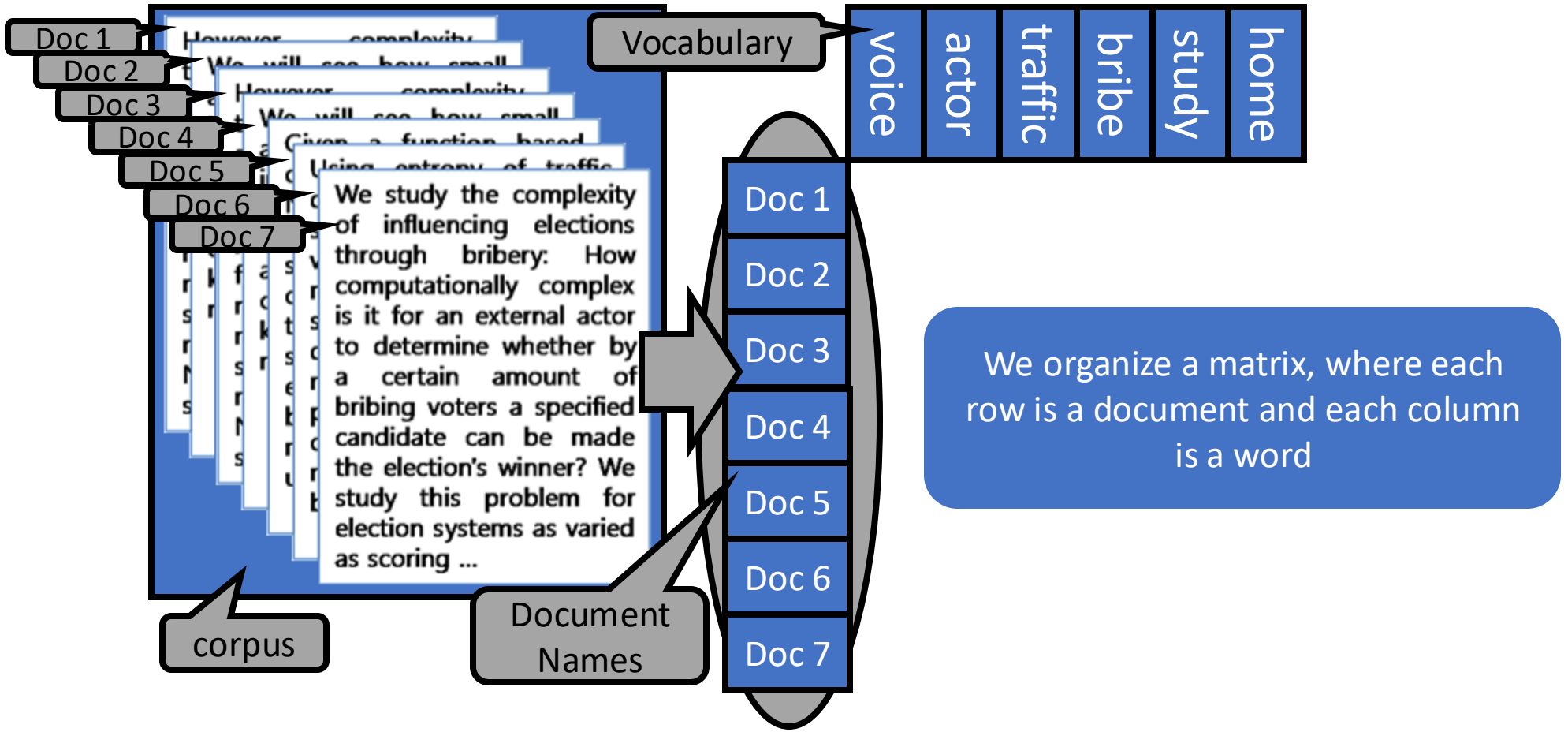
Document-Term Matrix (DTM)



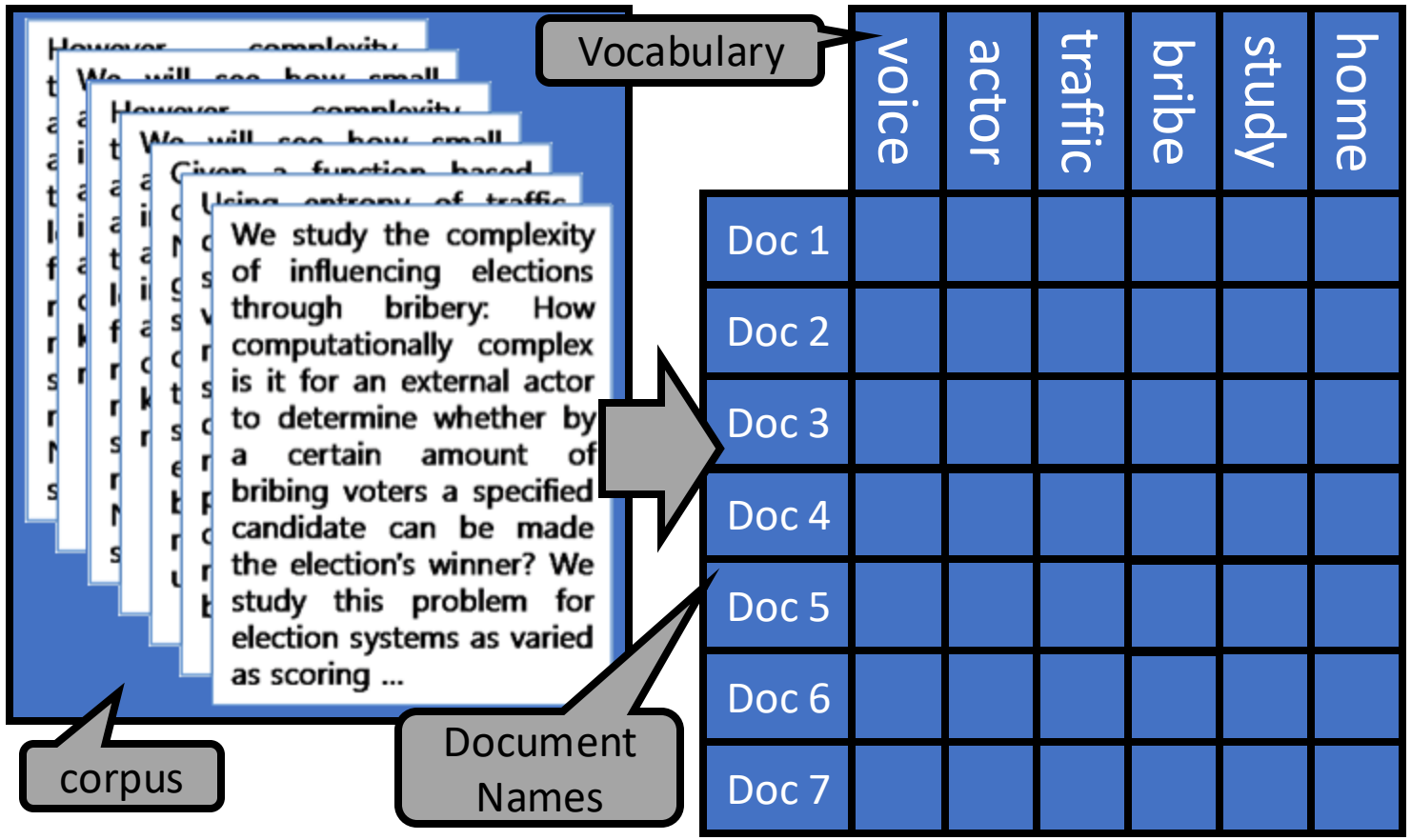
Document-Term Matrix (DTM)



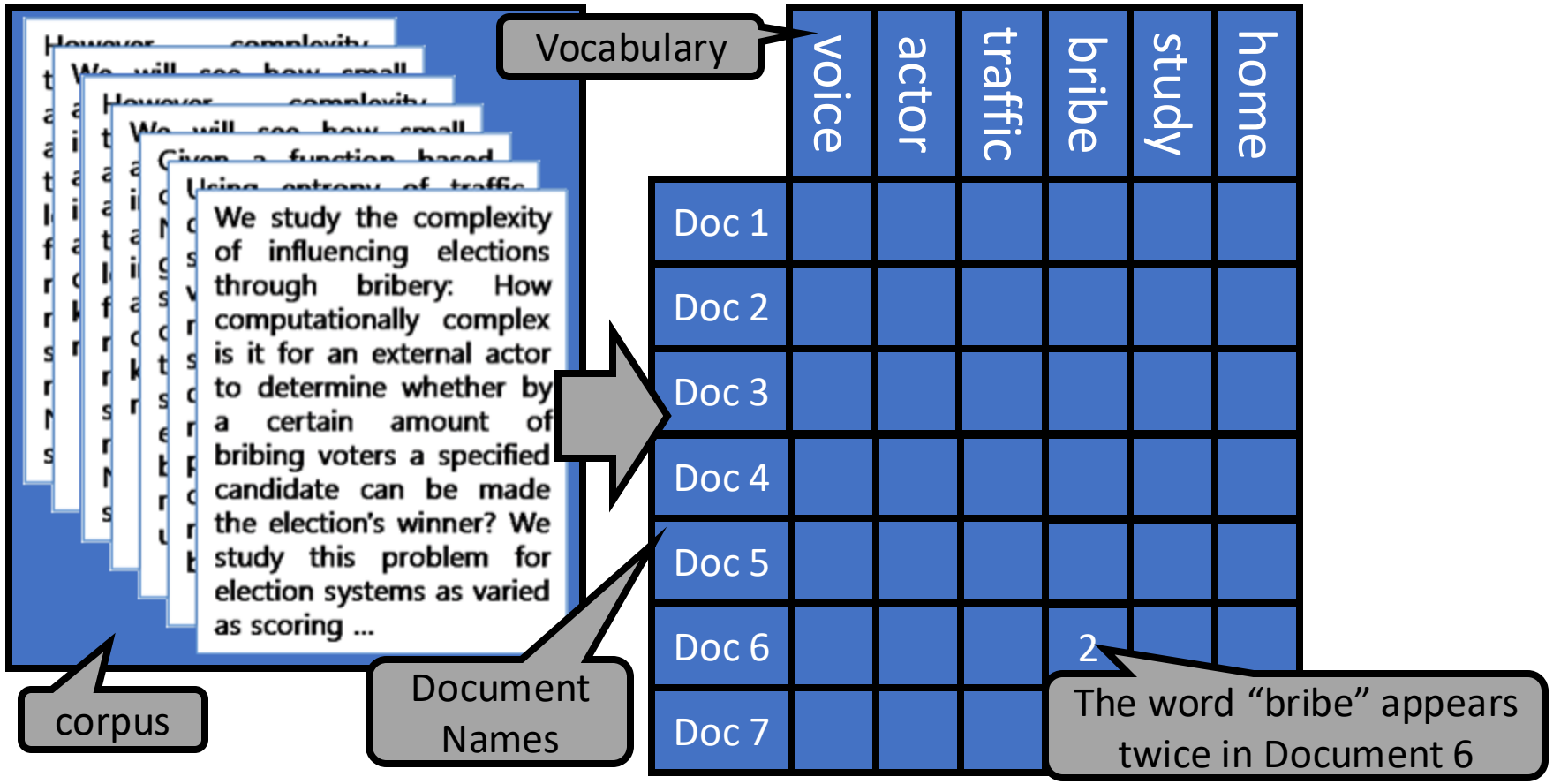
Document-Term Matrix (DTM)



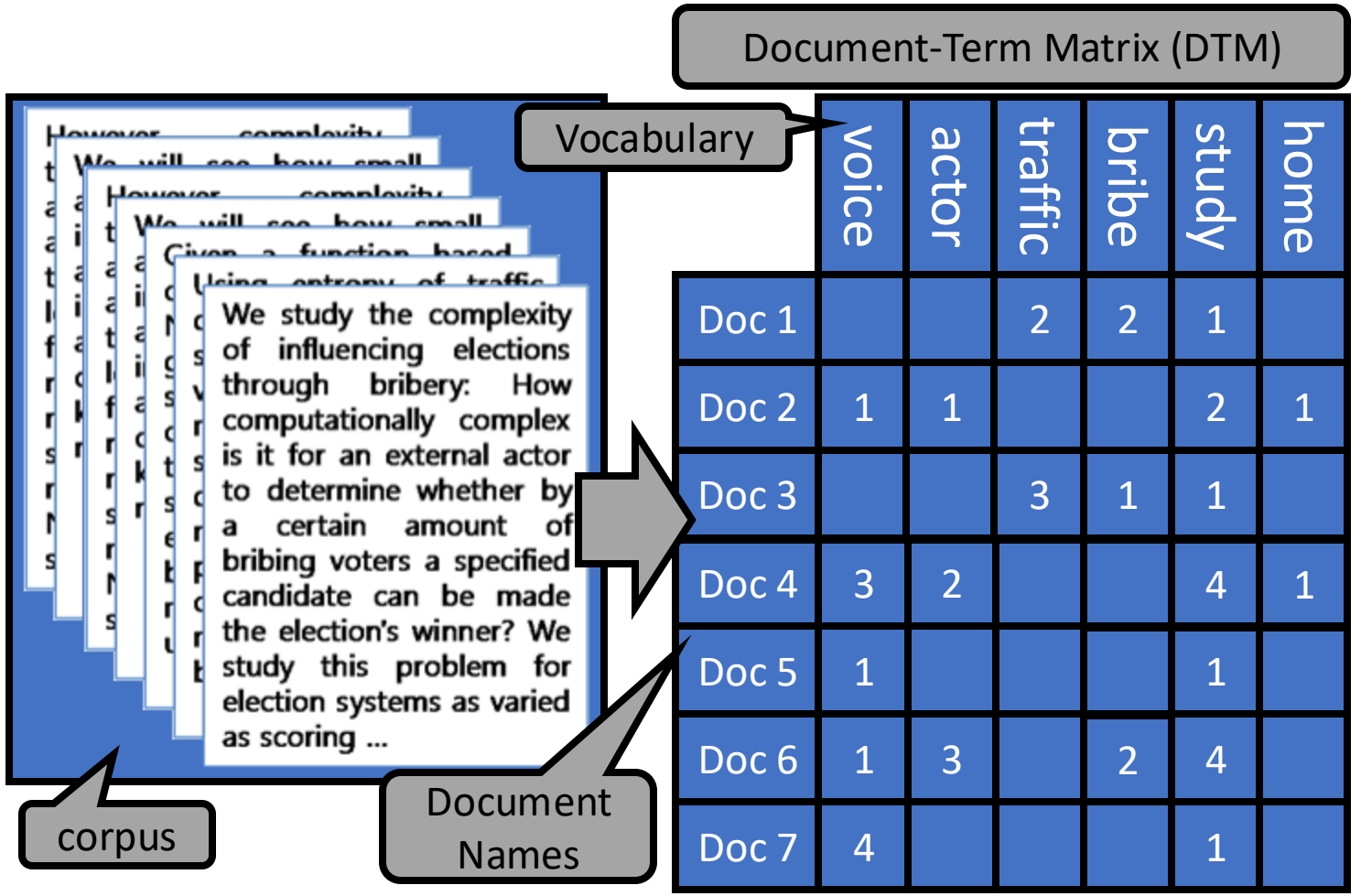
Document-Term Matrix (DTM)



Document-Term Matrix (DTM)



Document-Term Matrix (DTM)



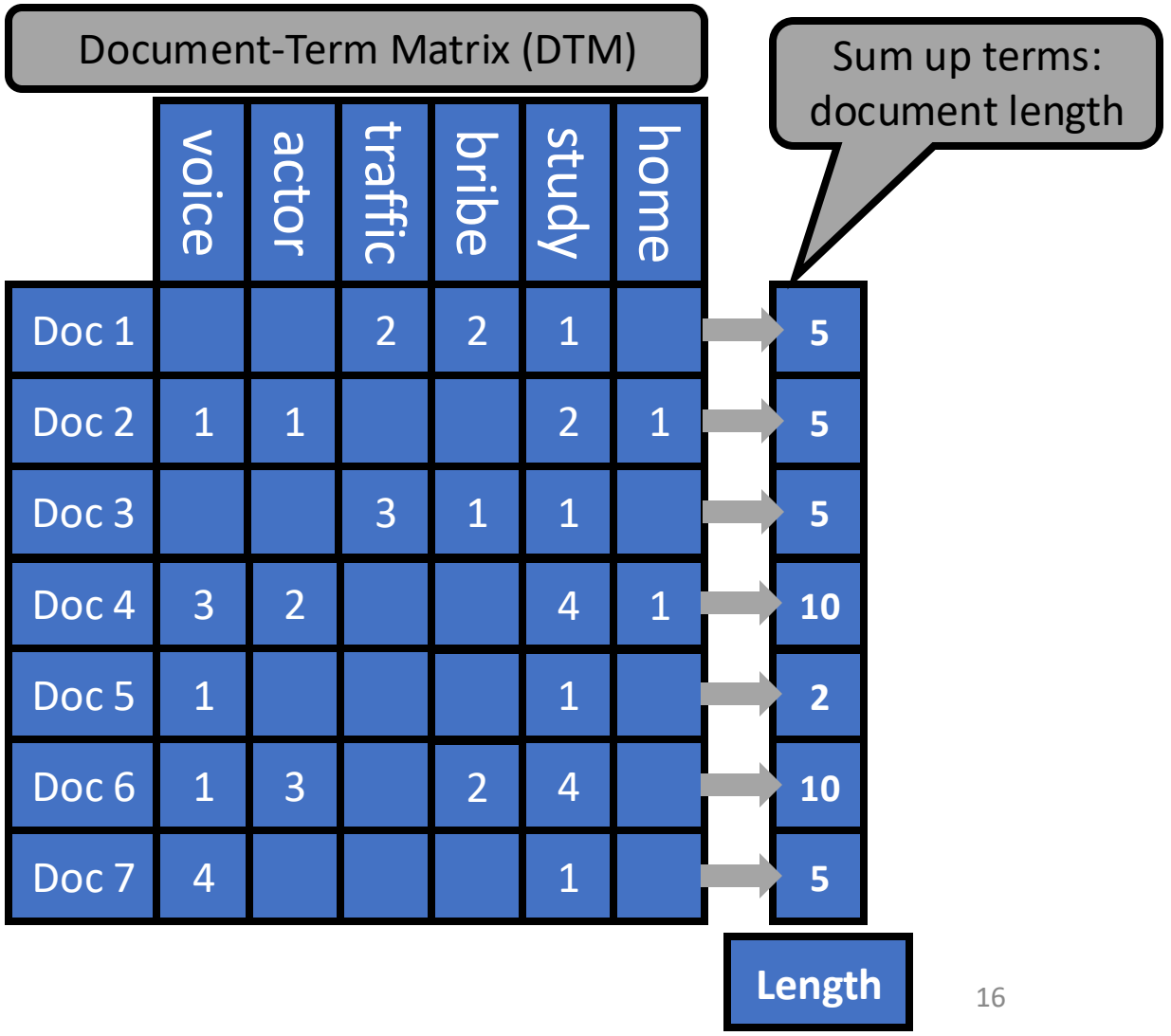
Document-Term Matrix (DTM)

The Document-Term Matrix is the data set that is used to create the Term-Frequencies (TF) and the Inverse Document Frequencies (IDF) and finally the Term-Frequency-Inverse-Document Frequency (TF-IDF) matrix

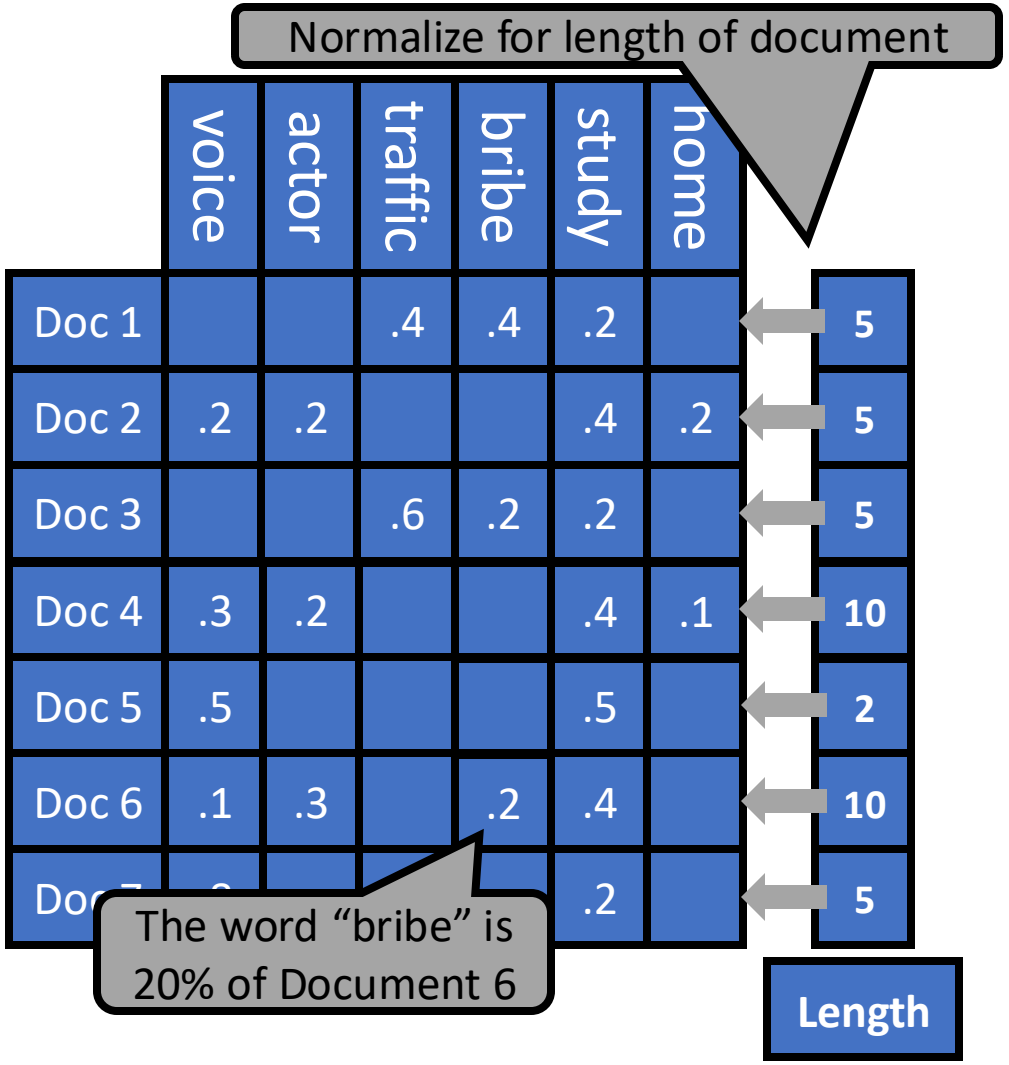
Document-Term Matrix (DTM)

	voice	actor	traffic	bribe	study	home
Doc 1			2	2	1	
Doc 2	1	1			2	1
Doc 3			3	1	1	
Doc 4	3	2			4	1
Doc 5	1				1	
Doc 6	1	3		2	4	
Doc 7	4				1	

Term Frequencies (TF)



Term Frequencies (TF)



Term Frequencies (TF)

The Term-Frequencies (TF) will be used later to calculate the Term-Frequency-Inverse-Document Frequency (TF-IDF) matrix. We will remember it now

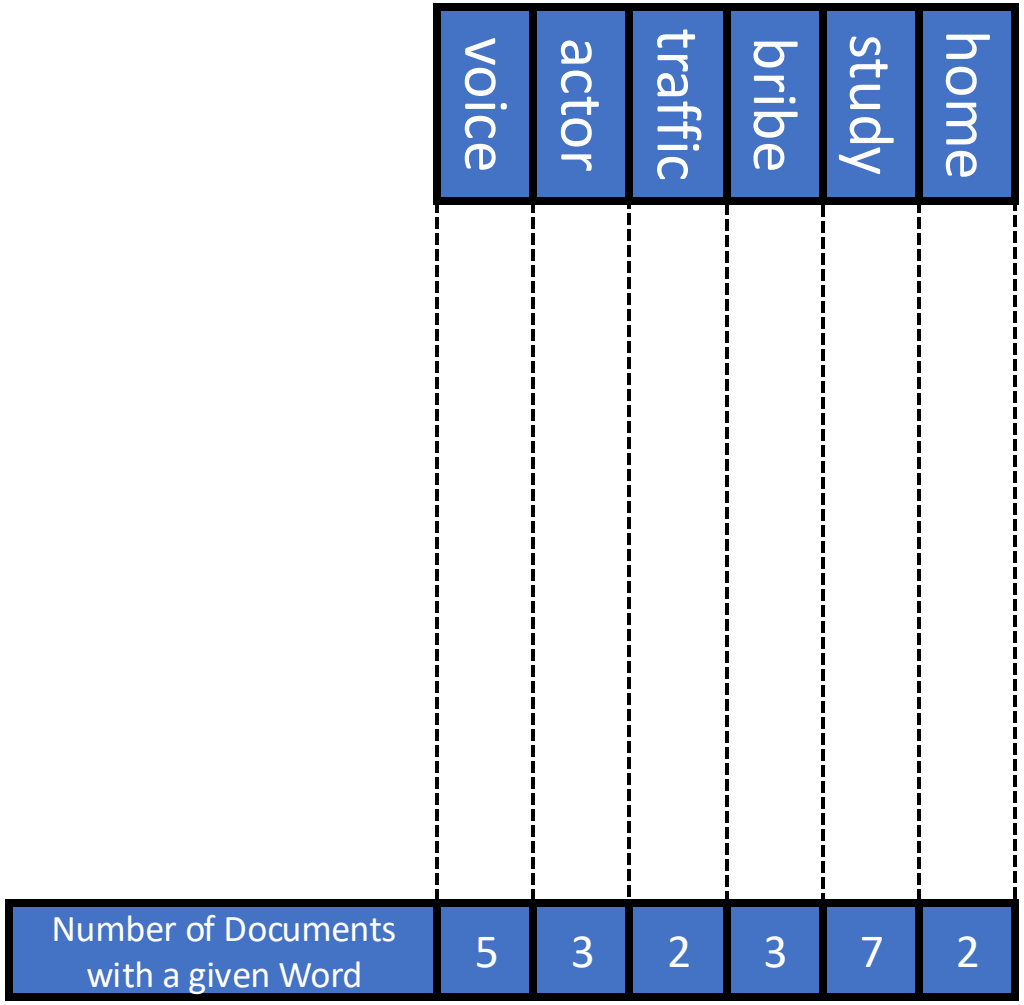
Term Frequencies (TF)	voice	actor	traffic	bribe	study	home
Doc 1			.4	.4	.2	
Doc 2	.2	.2			.4	.2
Doc 3			.6	.2	.2	
Doc 4	.3	.2			.4	.1
Doc 5	.5				.5	
Doc 6	.1	.3		.2	.4	
Doc 7	.8				.2	

Inverse Document Frequency (IDF)

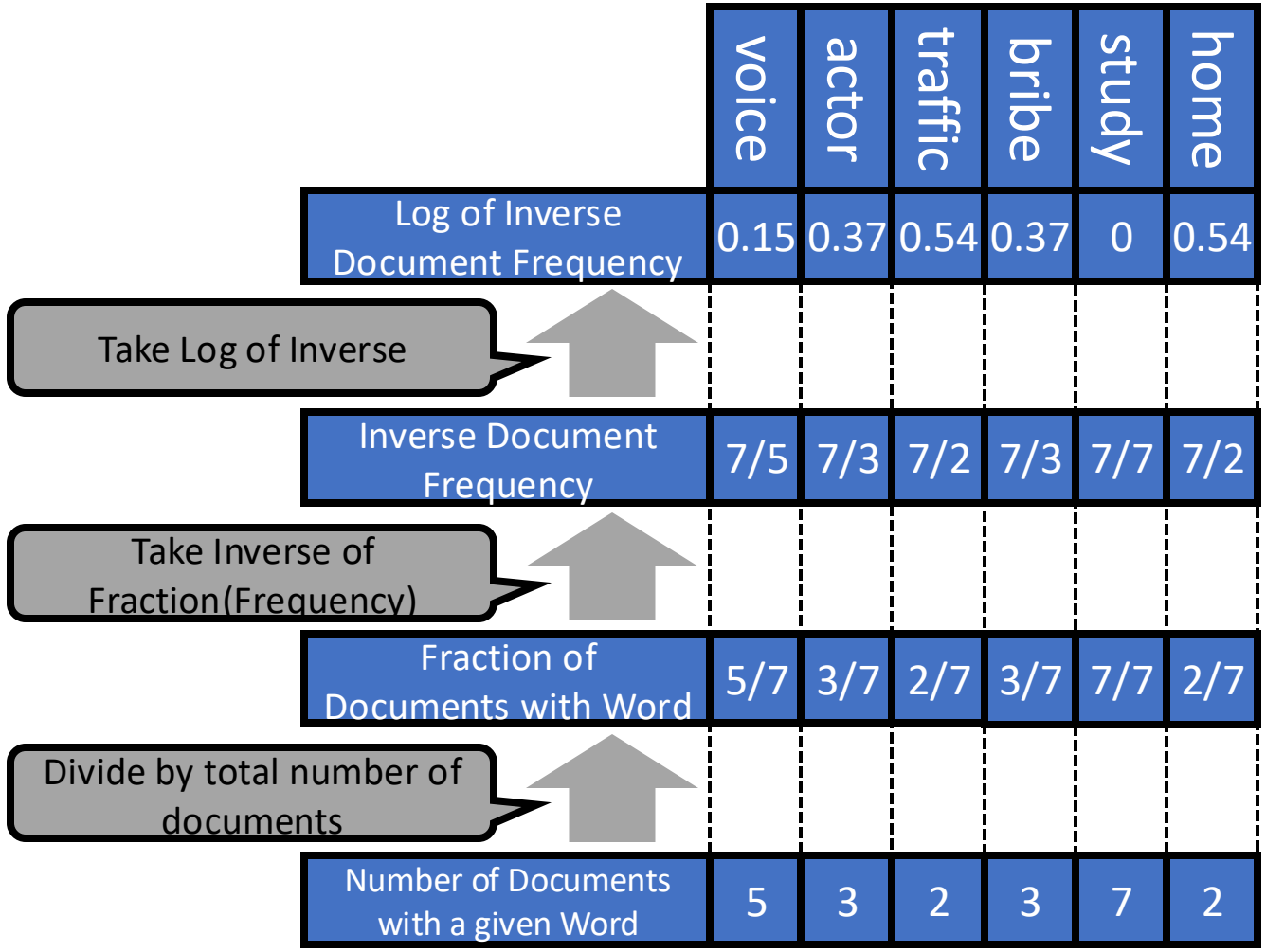
Document-Term Matrix (DTM)

	voice	actor	traffic	bribe	study	home
Doc 1			2	2	1	
Doc 2	1	1			2	1
Doc 3			3	1	1	
Doc 4	3	2			4	1
Doc 5	1				1	
Doc 6	1	3		2	4	
Doc 7	4				1	
Number of Documents with a given Word	5	3	2	3	7	2

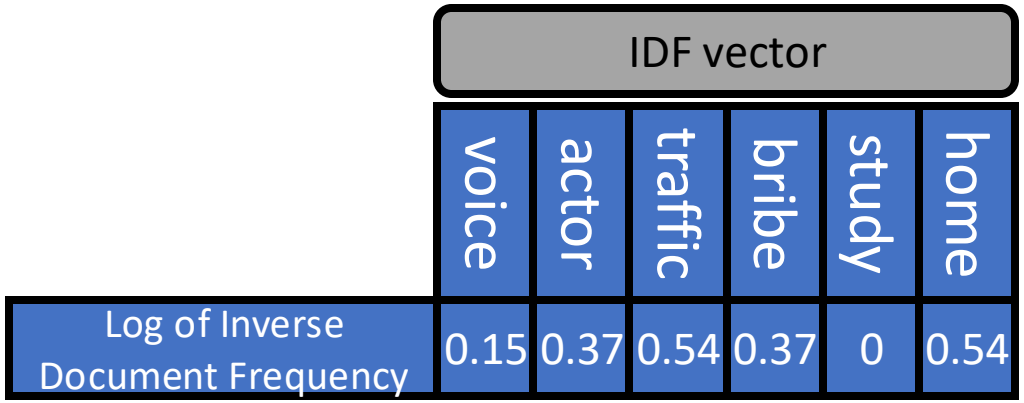
Inverse Document Frequency (IDF)



Inverse Document Frequency (IDF)



Inverse Document Frequency (IDF)



Term-Frequency Inverse Document Frequency (TF-IDF)

TF from earlier

Term Frequencies (TF)

	voice	actor	traffic	bribe	study	home
Doc 1			.4	.4	.2	
Doc 2	.2	.2			.4	.2
Doc 3			.6	.2	.2	
Doc 4	.3	.2			.4	.1
Doc 5	.5				.5	
Doc 6	.1	.3		.2	.4	
Doc 7	.8				.2	

⊙

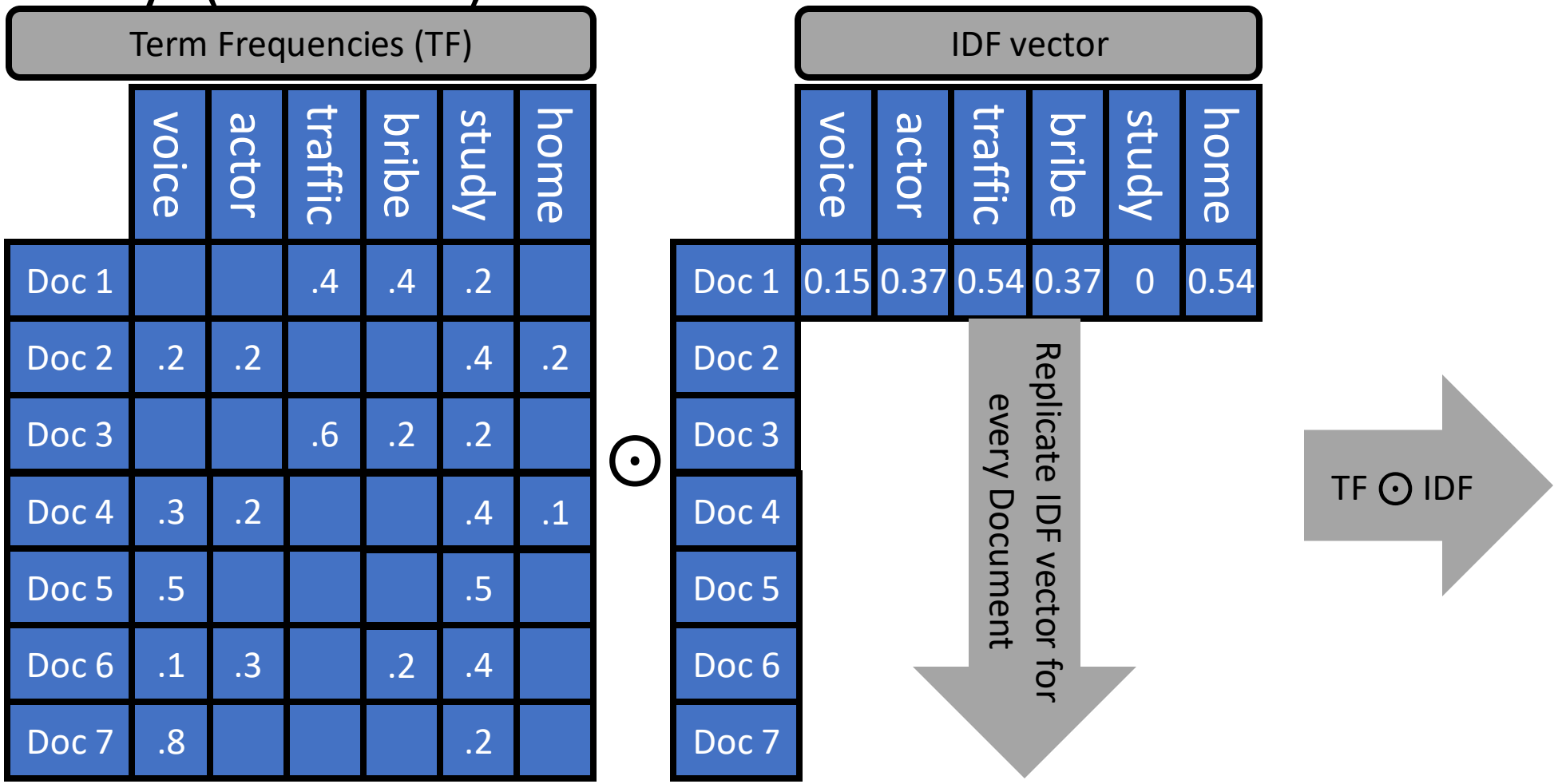
IDF vector

voice	actor	traffic	bribe	study	home
0.15	0.37	0.54	0.37	0	0.54

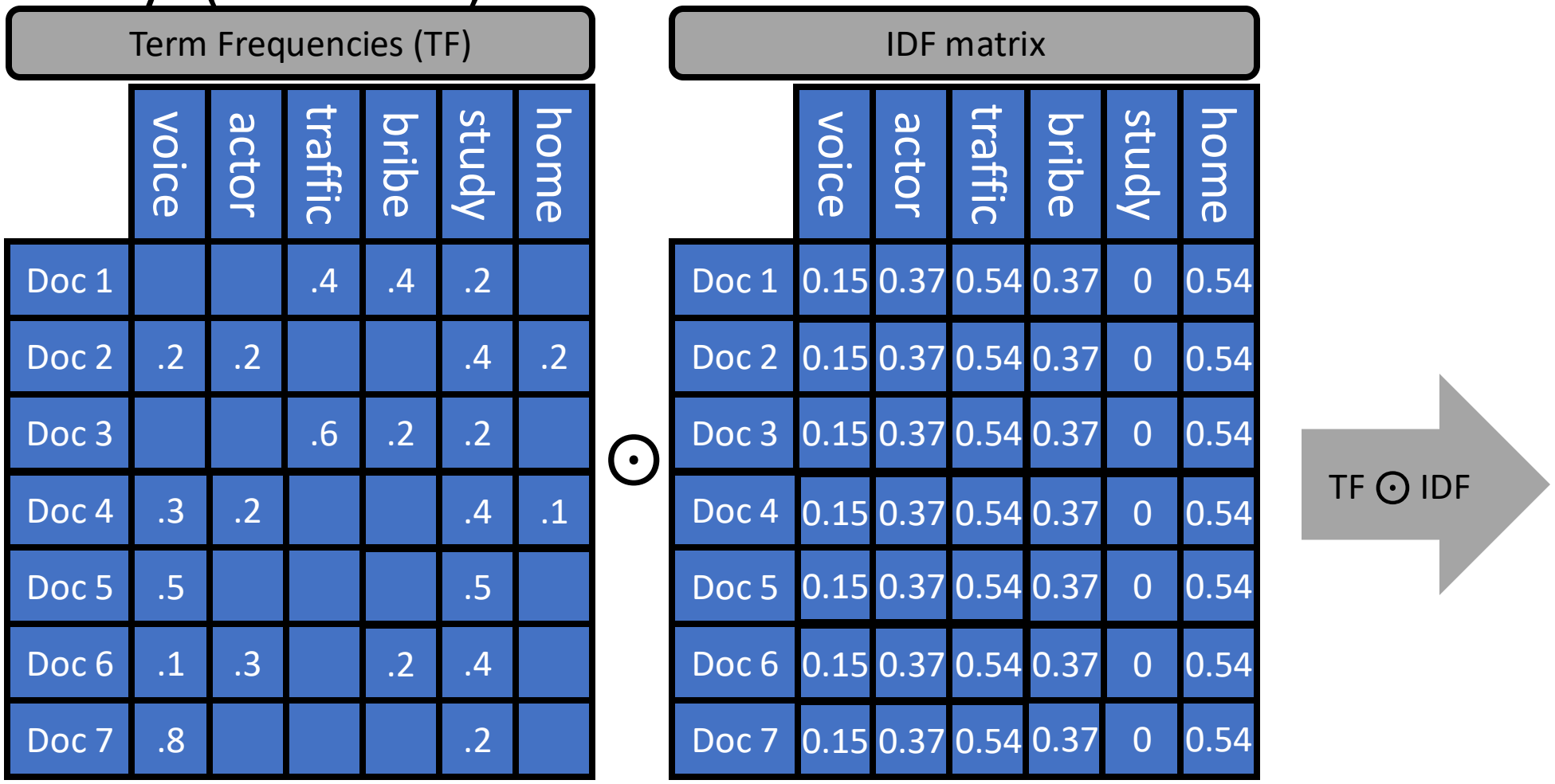
TF-IDF = TF ⊙ IDF

Element-by-element multiplication of TF with IDF vector creates the TF-IDF matrix

Term-Frequency Inverse Document Frequency (TF-IDF)



Term-Frequency Inverse Document Frequency (TF-IDF)

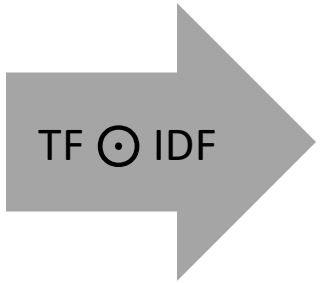


	voice	actor	traffic	bribe	study	home
Doc 1			.4	.4	.2	
Doc 2	.2	.2			.4	.2
Doc 3			.6	.2	.2	
Doc 4	.3	.2			.4	.1
Doc 5	.5				.5	
Doc 6	.1	.3		.2	.4	
Doc 7	.8				.2	

⊙

	voice	actor	traffic	bribe	study	home
Doc 1	0.15	0.37	0.54	0.37	0	0.54
Doc 2	0.15	0.37	0.54	0.37	0	0.54
Doc 3	0.15	0.37	0.54	0.37	0	0.54
Doc 4	0.15	0.37	0.54	0.37	0	0.54
Doc 5	0.15	0.37	0.54	0.37	0	0.54
Doc 6	0.15	0.37	0.54	0.37	0	0.54
Doc 7	0.15	0.37	0.54	0.37	0	0.54

Term-Frequency Inverse Document Frequency (TF-IDF)



TF-IDF matrix

	voice	actor	traffic	bribe	study	home
Doc 1	0	0	0.22	0.15	0	0
Doc 2	0.03	0.07	0	0	0	0.11
Doc 3	0	0	0.32	0.07	0	0
Doc 4	0.05	0.07	0	0	0	0.05
Doc 5	0.08	0	0	0	0	0
Doc 6	0.02	0.11	0	0.07	0	0
Doc 7	0.12	0	0	0	0	0

Term-Frequency Inverse Document Frequency (TF-IDF)

The TF-IDF can be used for Unsupervised Learning. For Example: We could cluster documents into topics. (Useful for curation of legal documents)

TF-IDF matrix

	voice	actor	traffic	bribe	study	home
Doc 1	0	0	0.22	0.15	0	0
Doc 2	0.03	0.07	0	0	0	0.11
Doc 3	0	0	0.32	0.07	0	0
Doc 4	0.05	0.07	0	0	0	0.05
Doc 5	0.08	0	0	0	0	0
Doc 6	0.02	0.11	0	0.07	0	0
Doc 7	0.12	0	0	0	0	0

Cluster Label
Topic 1
Topic 2
Topic 1
Topic 2
Topic 3
Topic 2
Topic 3

Term-Frequency Inverse Document Frequency (TF-IDF)

The TF-IDF can be used for Supervised Learning. For Example: We could add labels to classify documents for sentiment analysis or fake news detection

TF-IDF matrix

	voice	actor	traffic	bribe	study	home	FAKE? (Class)
Doc 1	0	0	0.22	0.15	0	0	No
Doc 2	0.03	0.07	0	0	0	0.11	Yes
Doc 3	0	0	0.32	0.07	0	0	Yes
Doc 4	0.05	0.07	0	0	0	0.05	No
Doc 5	0.08	0	0	0	0	0	No
Doc 6	0.02	0.11	0	0.07	0	0	No
Doc 7	0.12	0	0	0	0	0	Yes

Model to Classify Fake News

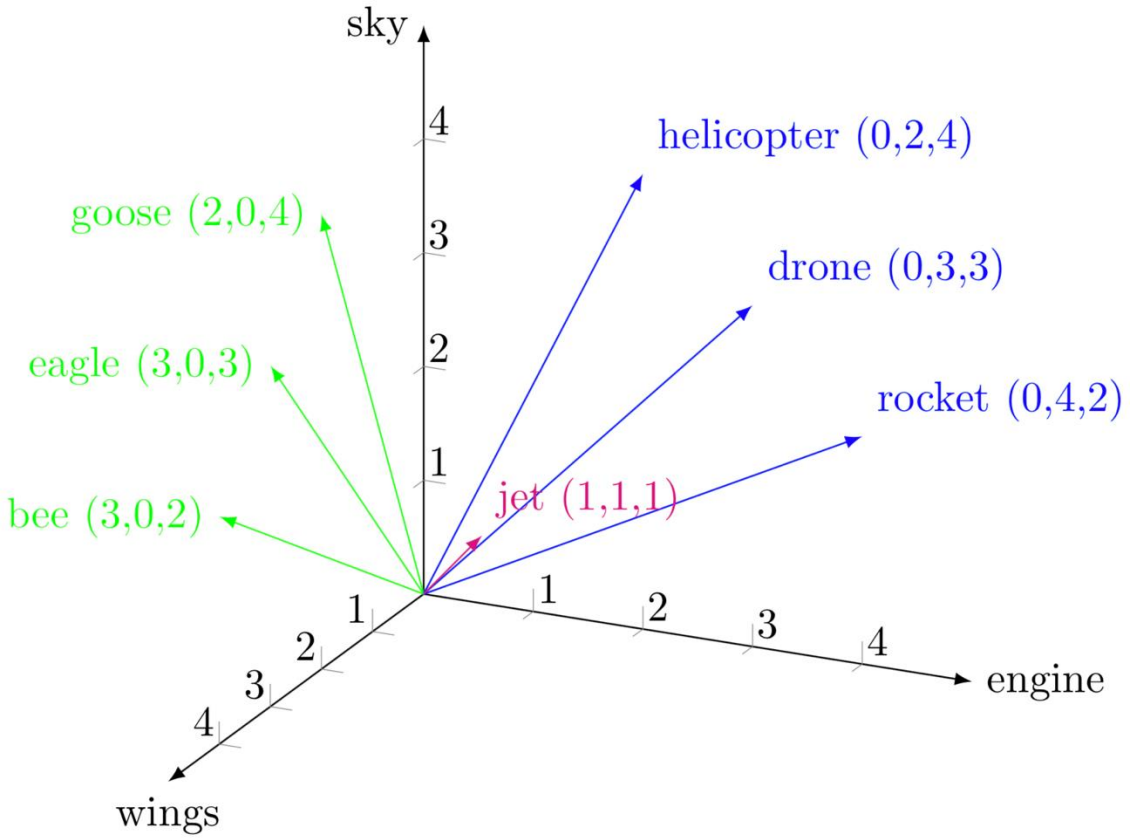
TF-IDF Summary

1. Corpus
   - A document is a unit of written natural language like a book or a tweet
   - A corpus is a collection of documents
2. document-term matrix (DTM)
   - The DTM is derived from a corpus
   - The DTM is sparse
3. term frequencies (TF)
   - The TF are derived from the DTM
   - The TF is sparse
4. inverse document frequencies (IDF)
   - IDF are derived from the DTM
   - IDF for a given term is the same for all documents in the corpus
5. term frequency – inverse document frequency (TF-IDF)
   - TF-IDF is calculated by multiplying TF with IDF
   - The TF-IDF can be used as training data for machine learning

Break

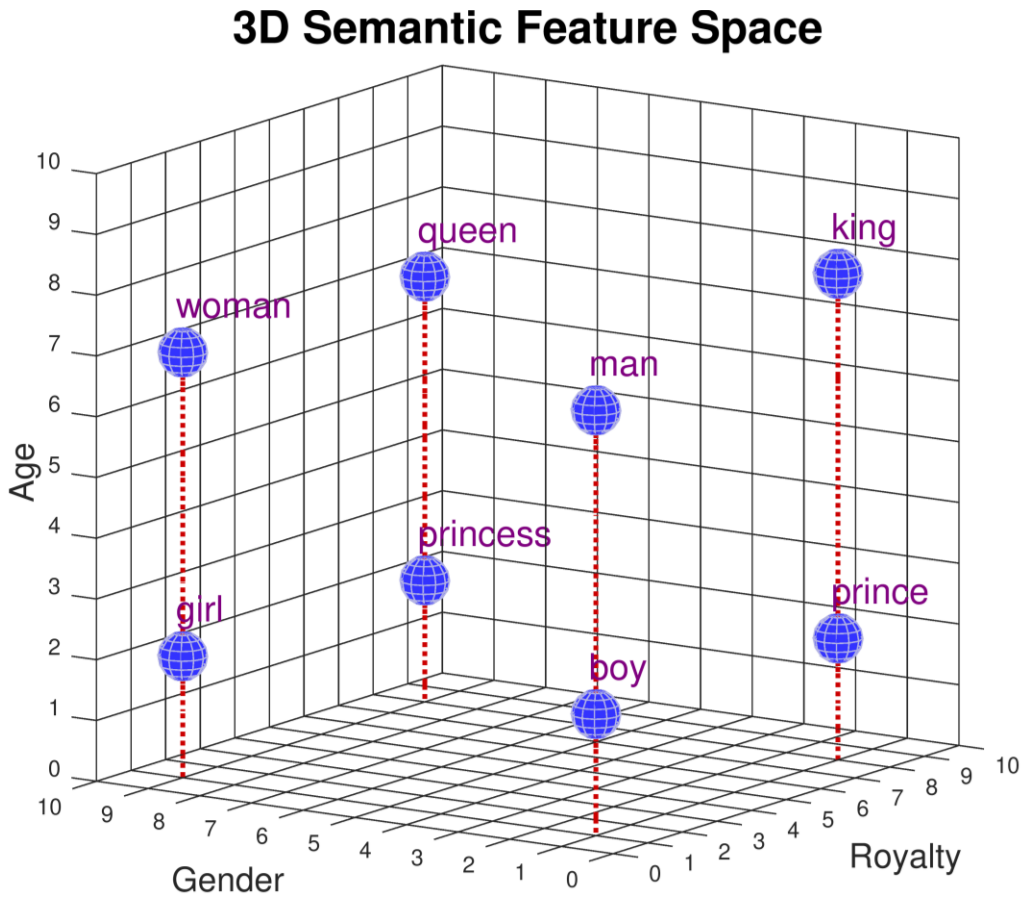
Word Embedding: basic element of LLMs

- Numerical representation of a word in a high-dimensional space
- This representation captures the semantic and syntactic meaning of the word, allowing machines to understand and process language in a more sophisticated way



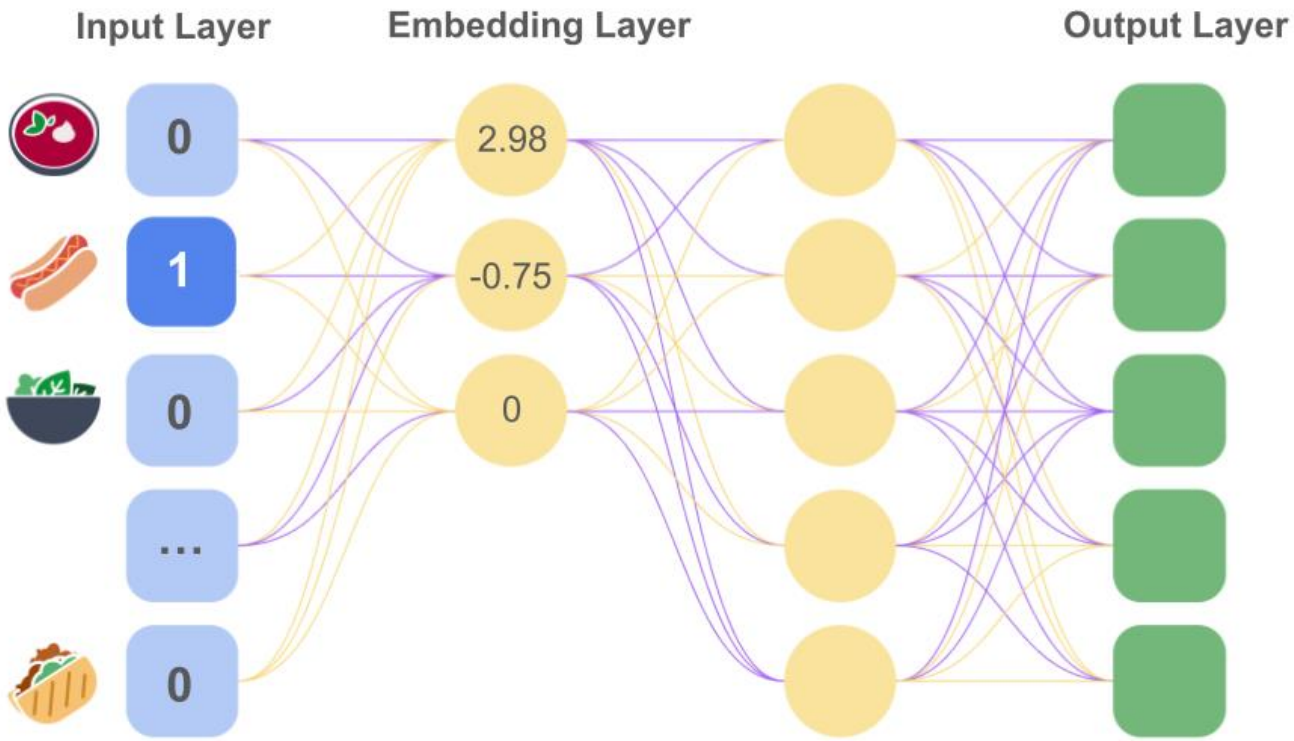
Word Embedding: basic element of LLMs

- In the examples here we have very low dimensionality
- In practice, most word embedding have quite high dimensionalities (100+)
- Often, the dimensionalities *a priori* do not mean anything
- Some famous methods:
  - Word2Vec
  - GloVe
  - BERT



Word Embedding: basic element of LLMs

- We will talk about some of these methods next week(s)
- Here we just want to mention that Word Embedding, is another feature engineering tool, that maps a word from a vocabulary into a vector of some (usually many) real numbers



Curse of Dimensionality

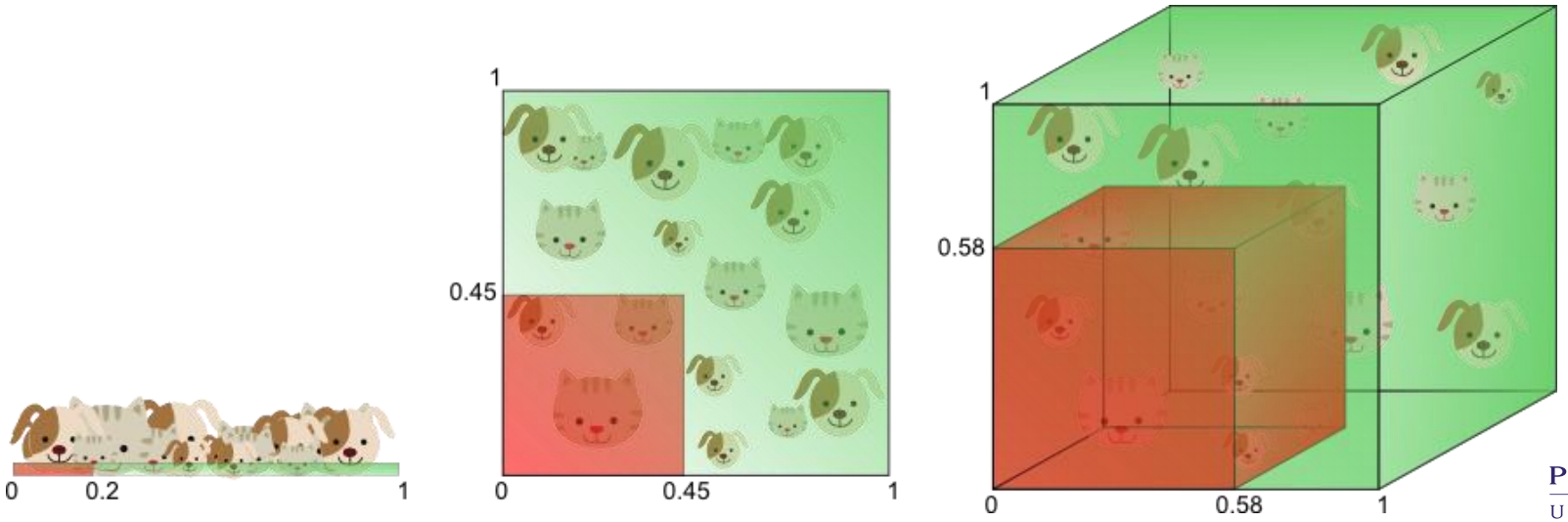
Increasing the number of dimensions can improve a machine learning model.

The curse of dimensionality means that too many dimensions cause problems with:

(1) overfitting
(2) computational effort
(3) interpretation

The question "How many dimensions is too many?" is usually answered by trial and error.

https://en.wikipedia.org/wiki/Curse_of_dimensionality



Interview question

A. Write a Python code to measure the median distance of two random points in a *d* dimensional space, for *d*=1, 2, …, 20 with simulation
B. Write a Python code to measure the median distance of a random point in a *d* dimensional space to the origin, for *d*=1, 2, …, 20 with simulation
C. Write a Python code to measure the probability for a random point in a *d* dimensional space to fit in the *unit hypersphere*, for *d*=1, 2, …, 20 with simulation