Artificial Intelligence for Text Analytics

Text Similarity and Clustering
Text Summarization and Topic Models

1121AITA07
MBA, IM, NTPU (M5265) (Fall 2023)
Tue 2, 3, 4 (9:10-12:00) (B3F17)

Min-Yuh Day, Ph.D,
Associate Professor

Institute of Information Management, National Taipei University

https://web.ntpu.edu.tw/~myday

https://meet.google.com/miy-fbif-max

2023-11-08; 2023-11-15

Syllabus

Week	Date	Subject/Topics
1	2023/09/13	Introduction to Artificial Intelligence for Text Analytics
2	2023/09/20	Foundations of Text Analytics: Natural Language Processing (NLP)
3	2023/09/27	Python for Natural Language Processing
4	2023/10/04	Natural Language Processing with Transformers
5	2023/10/11	Case Study on Artificial Intelligence for Text Analytics I
6	2023/10/18	Text Classification and Sentiment Analysis

Syllabus

Week	Date	Subject/Topics
7	2023/10/25	Multilingual Named Entity Recognition (NER)
8	2023/11/01	Midterm Project Report
9	2023/11/08	Text Similarity and Clustering
10	2023/11/15	Text Summarization and Topic Models
11	2023/11/22	Text Generation with Large Language Models (LLMs)
12	2023/11/29	Case Study on Artificial Intelligence for Text Analytics II

Syllabus

Week	Date	Subject/Topics
13	2023/12/06	Question Answering and Dialogue Systems
14	2023/12/13	Deep Learning, Generative AI, Transfer Learning, Zero-Shot, and Few-Shot Learning for Text Analytics
15	2023/12/20	Final Project Report I
16	2023/12/27	Final Project Report II
17	2024/01/03	Self-learning
18	2024/01/10	Self-learning

Text Similarity
Text Clustering
Text Summarization
Topic Models

Outline

- **Text Similarity**
  - **Analyzing and quantifying the likeness between text documents.**
- **Text Clustering**
  - **Grouping similar text documents using various algorithms.**
- **Text Summarization**
  - **Condensing text data into a shorter, coherent form.**
- **Topic Models**
  - **Identifying underlying themes or topics within text collections.**

Source: Dipanjan Sarkar (2019), Text Analytics with Python: A Practitioner's Guide to Natural Language Processing, Second Edition. APress.

Text Similarity and Clustering

Text Similarity and Clustering

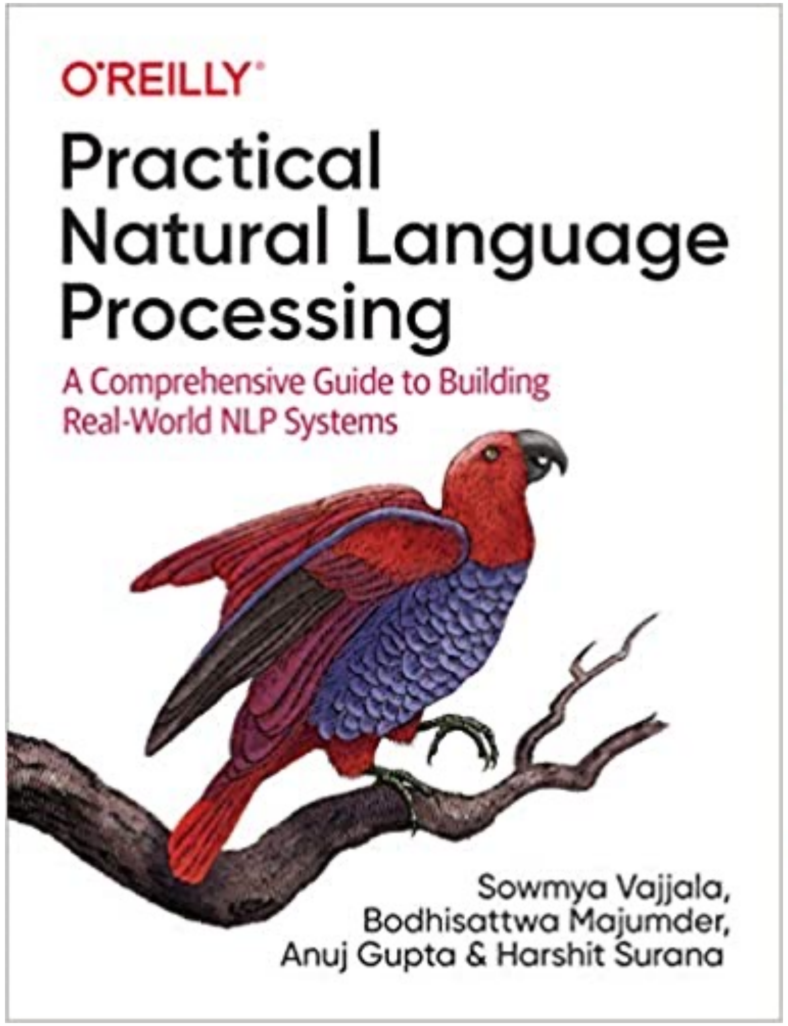
Text Dataset (Unsupervised)

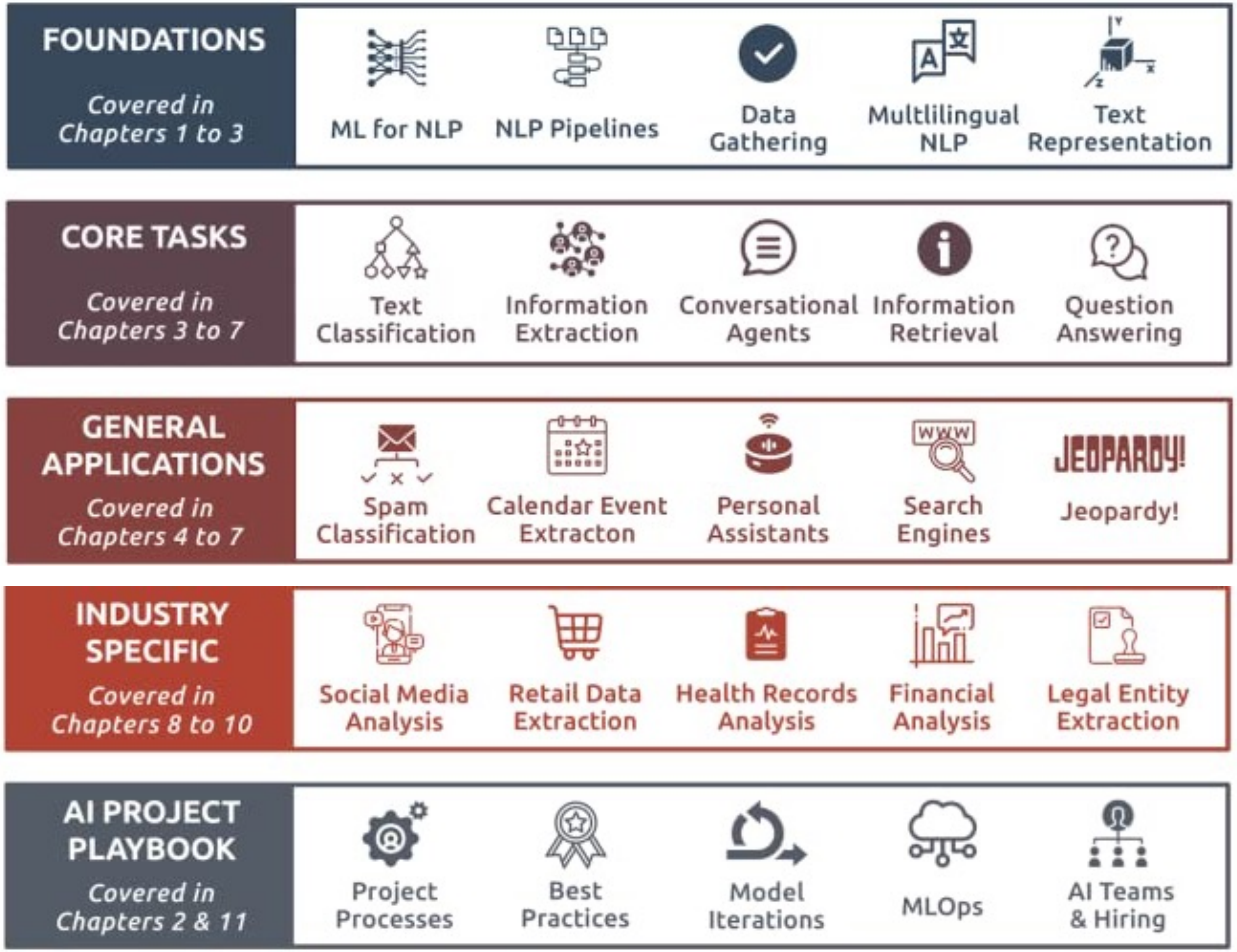
Text Pre-Processing

Feature Extraction (Vectorization) (TF-IDF)(Embedding)

Text Similarity

Text Clustering





Source: Sowmya Vajjala, Bodhisattwa Majumder, Anuj Gupta (2020), Practical Natural Language Processing: A Comprehensive Guide to Building Real-World NLP Systems, O'Reilly Media.

Source: https://www.amazon.com/Practical-Natural-Language-Processing-Pragmatic/dp/1492054054

Text Similarity and Clustering

- How do we measure similarity between terms and documents?
- How can we use distance measures to find the most relevant documents?
- How can we build a recommender system from text similarity?
- How do we group similar documents (document clustering)?

Text Similarity and Clustering

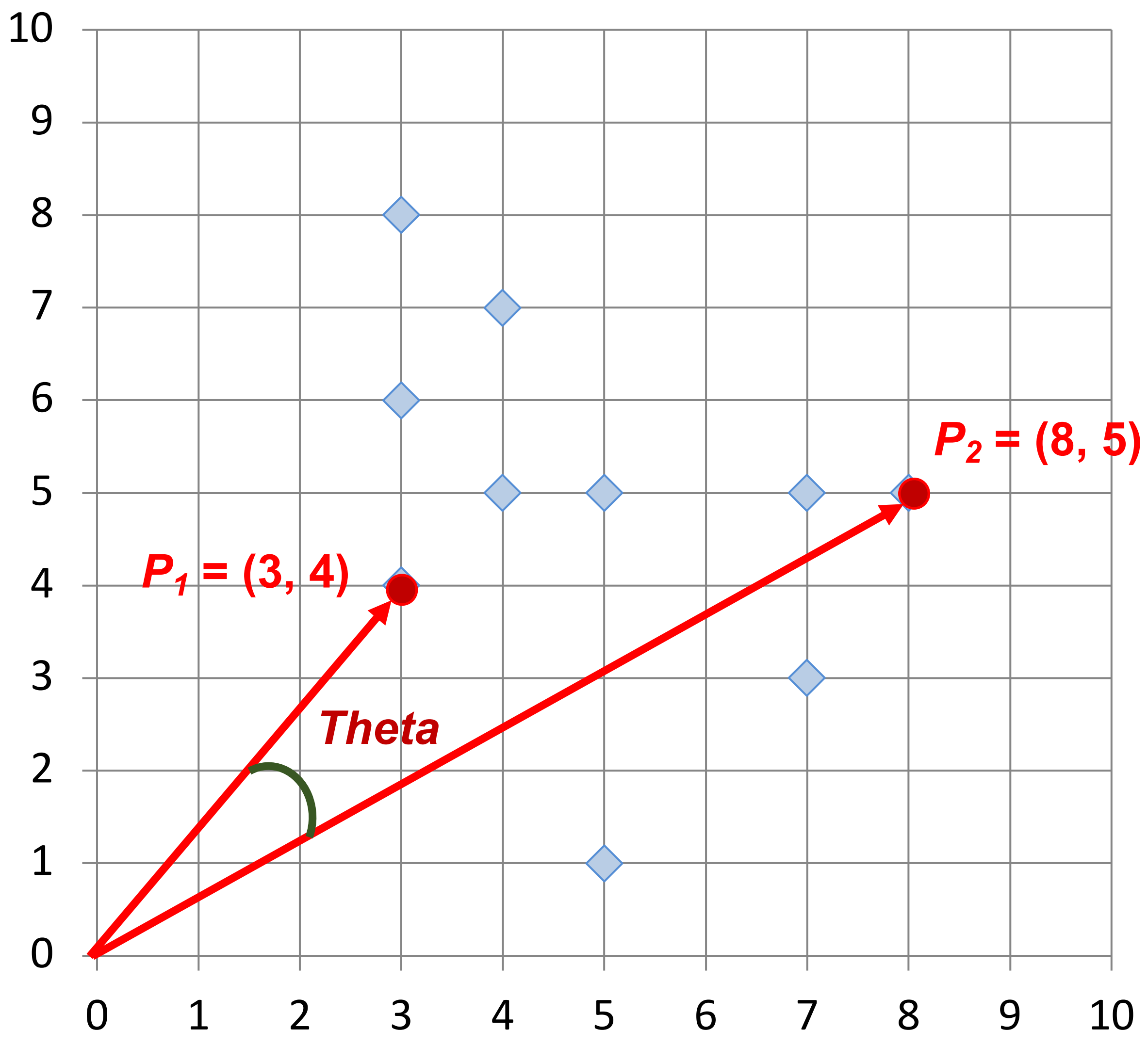
- **Information Retrieval (IR)**
- **Feature Engineering**
- **Similarity Measures**
- **Unsupervised Machine Learning Algorithms**

Text Similarity

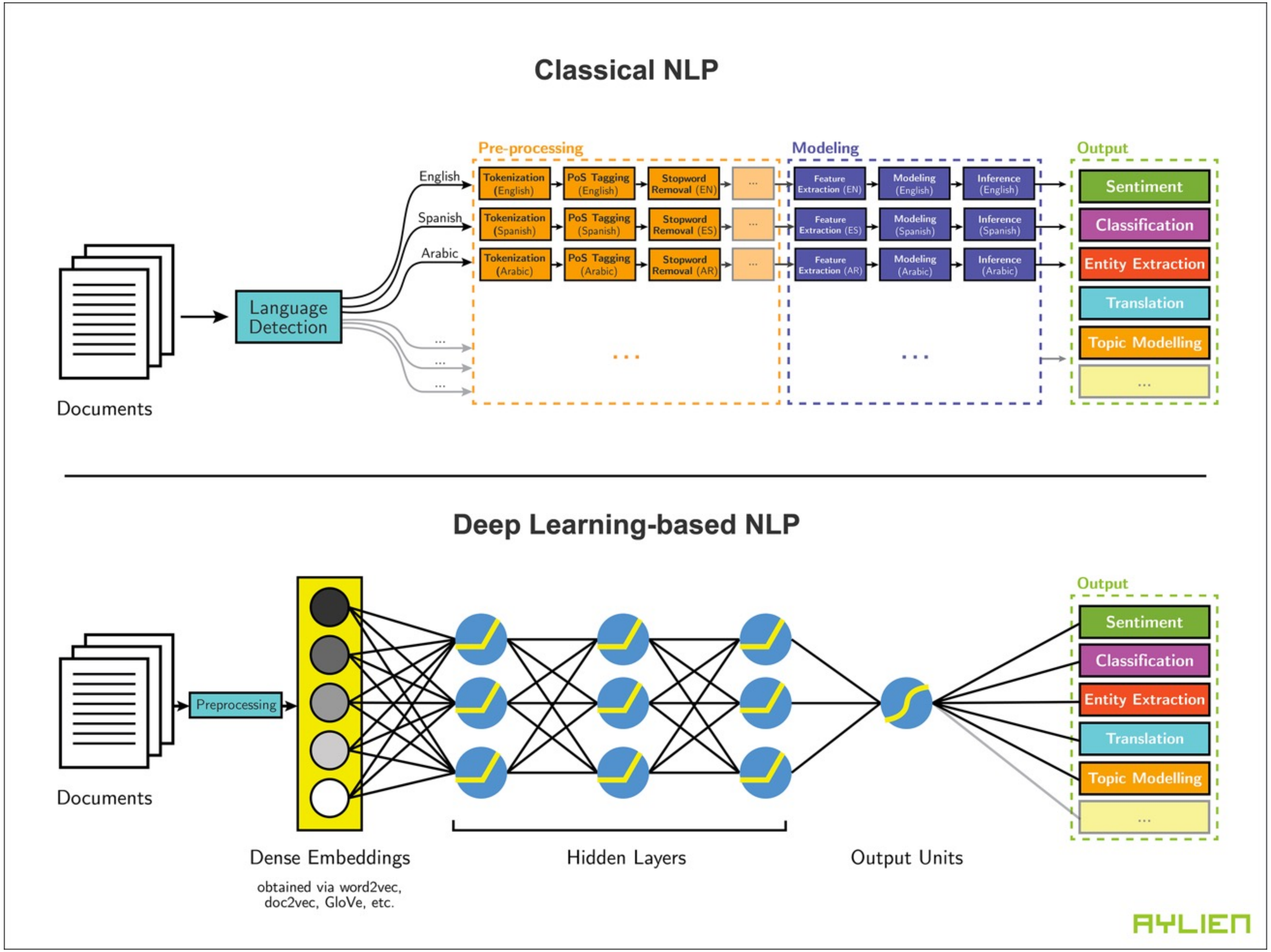
- **Lexical similarity**
  - **Syntax, structure, and content of the documents**
- **Semantic similarity**
  - **Semantics, meaning, and context of the documents**

Source: Dipanjan Sarkar (2019), Text Analytics with Python: A Practitioner's Guide to Natural Language Processing, Second Edition. APress.

Cosine Similarity

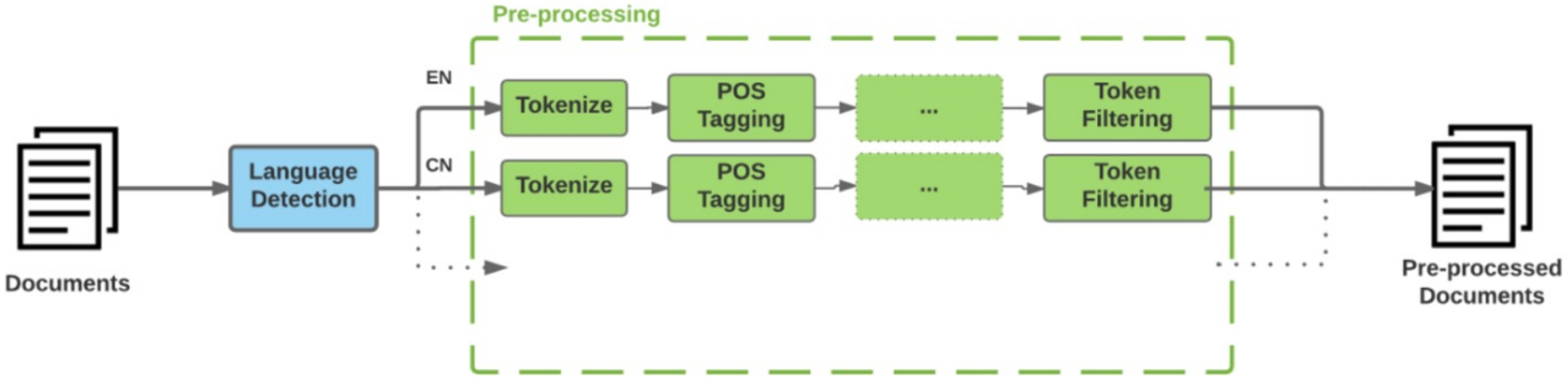


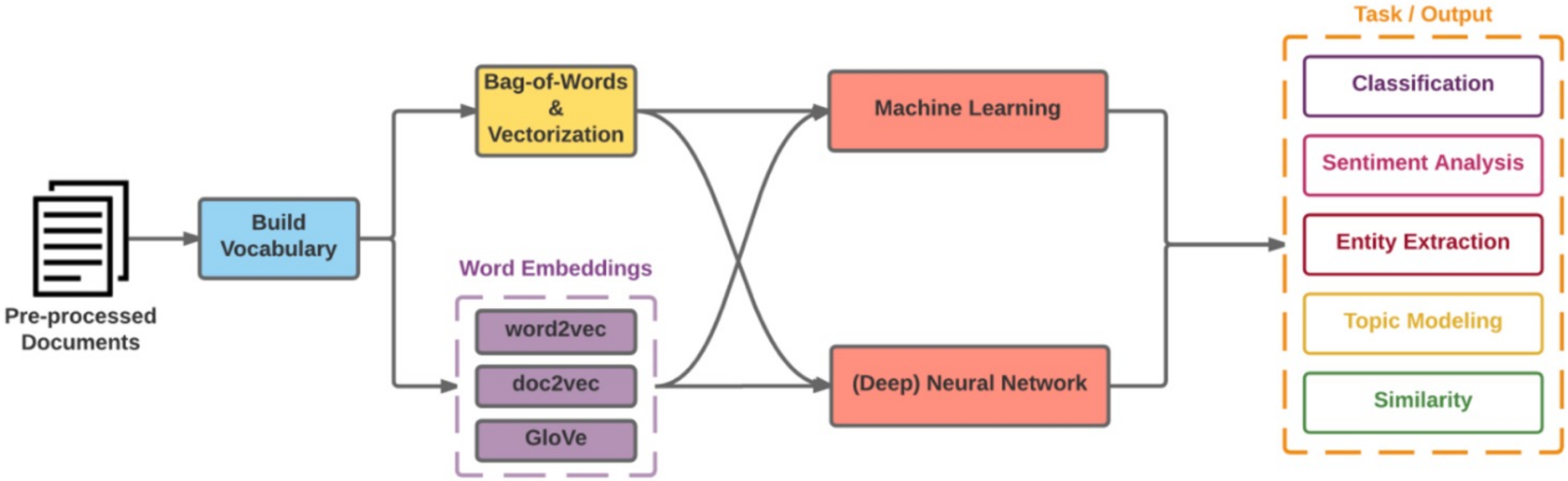
NLP



Source: http://blog.aylien.com/leveraging-deep-learning-for-multilingual/

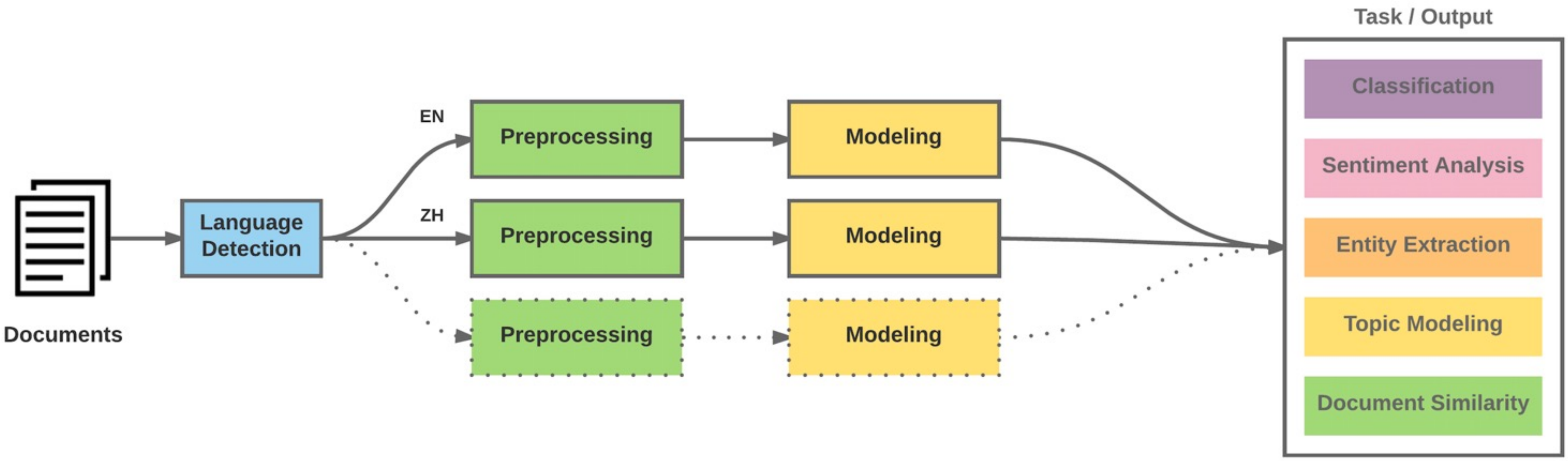
Modern NLP Pipeline





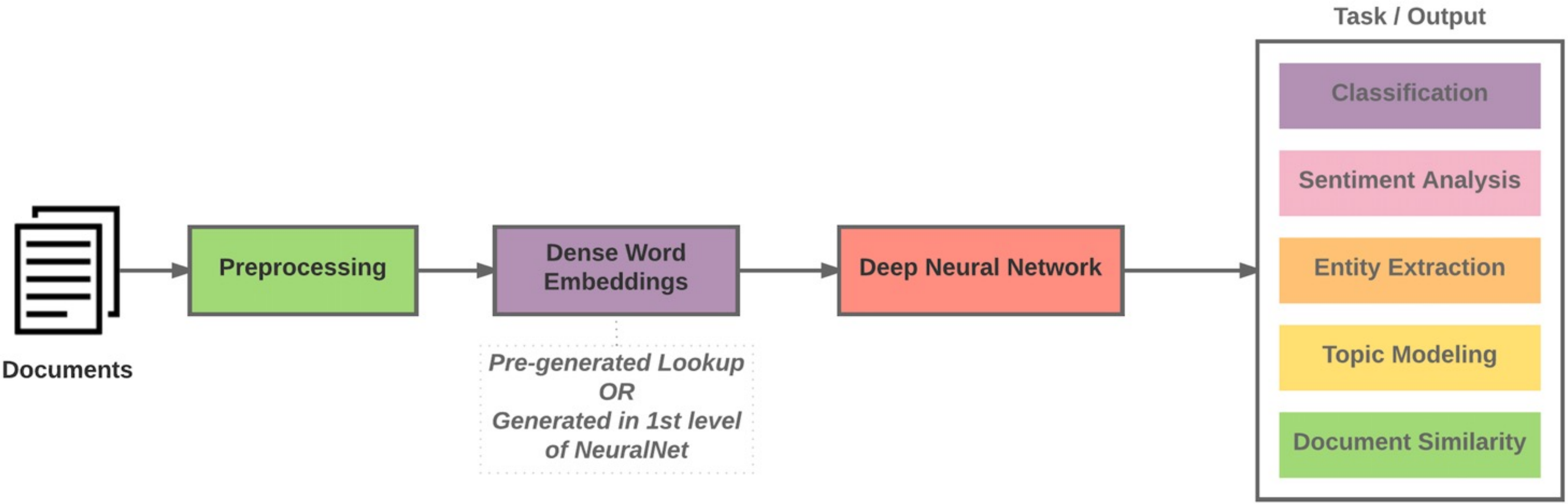
Source: https://github.com/fortiema/talks/blob/master/opendata2016sh/pragmatic-nlp-opendata2016sh.pdf

Modern NLP Pipeline



Source: http://mattfortier.me/2017/01/31/nlp-intro-pt-1-overview/

Deep Learning NLP

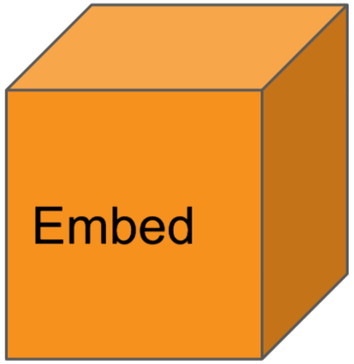


Source: http://mattfortier.me/2017/01/31/nlp-intro-pt-1-overview/

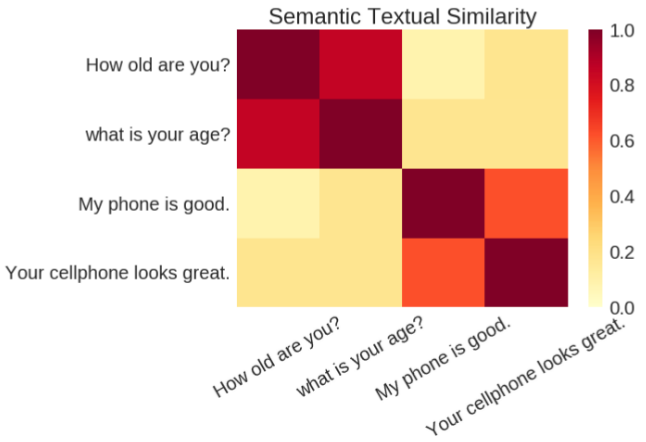
Text Similarity

Semantic Similarity

```
"How old are you?"
"What is your age?"
"My phone is good."
...
```




```
[0.3, 0.2, …]
[0.2, 0.1, …]
[0.9, 0.6, …]
...
```



Source: https://tfhub.dev/google/universal-sentence-encoder/4

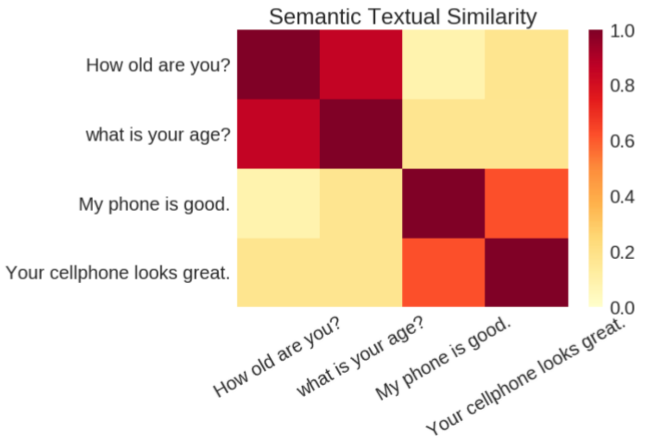
Text Similarity
Text Classification

Semantic Similarity

"How old are you?"
"What is your age?"
"My phone is good."
...

Embed

[0.3, 0.2, …]
[0.2, 0.1, …]
[0.9, 0.6, …]
...



Classification

"How old are you?"
"What is your age?"
"My phone is good."
...

Embed

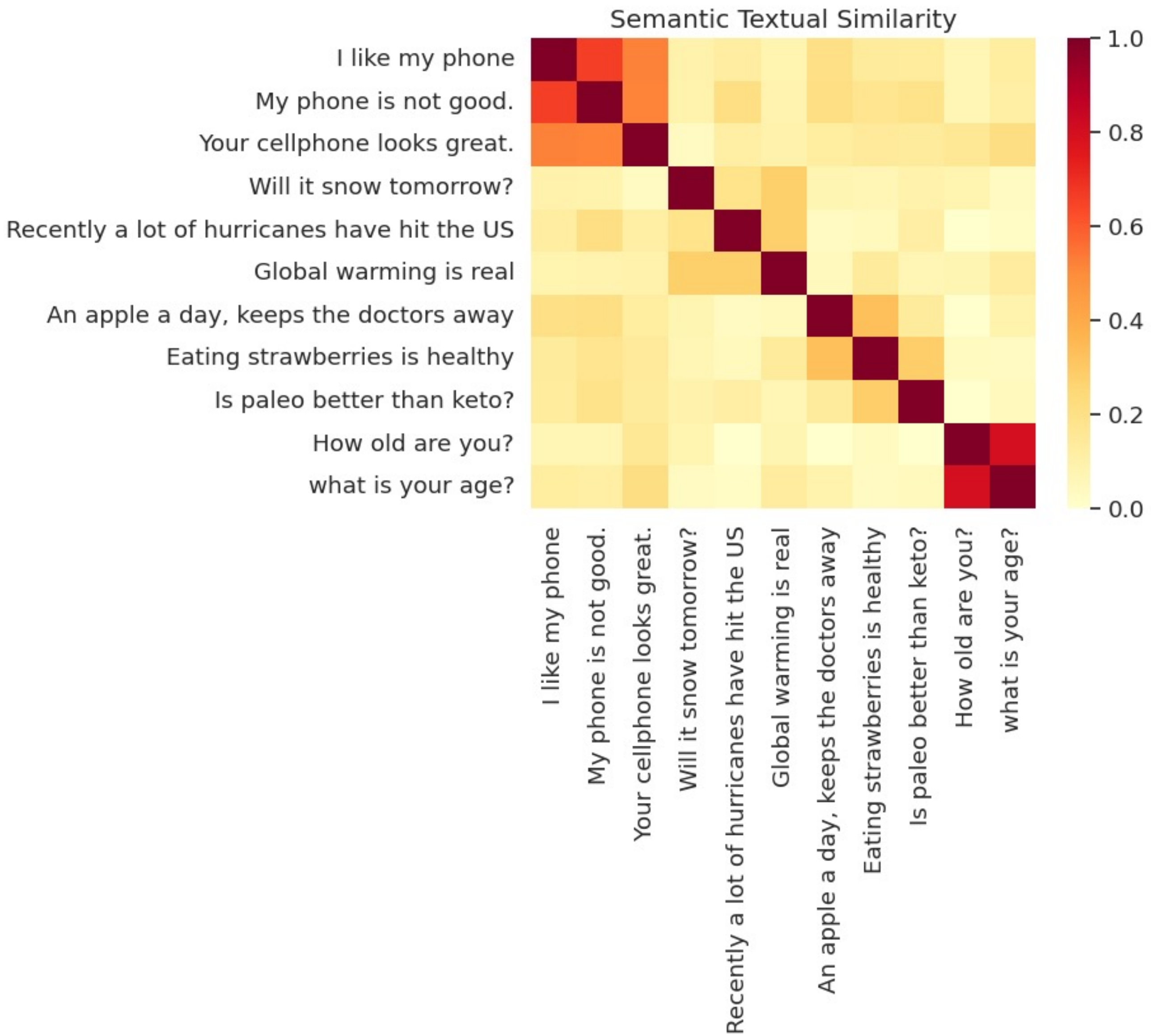
[0.3, 0.2, …]
[0.2, 0.1, …]
[0.9, 0.6, …]
...

Question Classifier

Confidence is a question
(96%) "How old are you?"
(98%) "What is your age?"
(7%) "My phone is good."
...

Source: https://tfhub.dev/google/universal-sentence-encoder/4

Semantic Textual Similarity



Source: https://tfhub.dev/google/universal-sentence-encoder/4

Natural Language Processing (NLP) and Text Mining

- Raw text
- Sentence Segmentation
- Tokenization
- Part-of-Speech (POS)
- Stop word removal
- Stemming / Lemmatization
- Dependency Parser
- String Metrics & Matching

word's stem	word's lemma
am → am	am → be
having → hav	having → have

Source: Nitin Hardeniya (2015), NLTK Essentials, Packt Publishing; Florian Leitner (2015), Text mining - from Bayes rule to dependency parsing

Data Mining Tasks & Methods

Data Mining Tasks & Methods		Data Mining Algorithms	Learning Type
Prediction			
	Classification	Decision Trees, Neural Networks, Support Vector Machines, kNN, Naïve Bayes, GA	Supervised
	Regression	Linear/Nonlinear Regression, ANN, Regression Trees, SVM, kNN, GA	Supervised
	Time series	Autoregressive Methods, Averaging Methods, Exponential Smoothing, ARIMA	Supervised
Association			
	Market-basket	Apriori, OneR, ZeroR, Eclat, GA	Unsupervised
	Link analysis	Expectation Maximization, Apriori Algorithm, Graph-Based Matching	Unsupervised
	Sequence analysis	Apriori Algorithm, FP-Growth, Graph-Based Matching	Unsupervised
Segmentation			
	Clustering	*k*-means, Expectation Maximization (EM)	Unsupervised
	Outlier analysis	*k*-means, Expectation Maximization (EM)	Unsupervised

Source: Ramesh Sharda, Dursun Delen, and Efraim Turban (2017), Business Intelligence, Analytics, and Data Science: A Managerial Perspective, 4th Edition, Pearson

Example of Cluster Analysis

Point	P	P(x,y)
p01	a	(3, 4)
p02	b	(3, 6)
p03	c	(3, 8)
p04	d	(4, 5)
p05	e	(4, 7)
p06	f	(5, 1)
p07	g	(5, 5)
p08	h	(7, 3)
p09	i	(7, 5)
p10	j	(8, 5)

K-Means Clustering

Point	P	P(x,y)	m1 distance	m2 distance	Cluster
p01	a	(3, 4)	1.95	3.78	Cluster1
p02	b	(3, 6)	0.69	4.51	Cluster1
p03	c	(3, 8)	2.27	5.86	Cluster1
p04	d	(4, 5)	0.89	3.13	Cluster1
p05	e	(4, 7)	1.22	4.45	Cluster1
p06	f	(5, 1)	5.01	3.05	Cluster2
p07	g	(5, 5)	1.57	2.30	Cluster1
p08	h	(7, 3)	4.37	0.56	Cluster2
p09	i	(7, 5)	3.43	1.52	Cluster2
p10	j	(8, 5)	4.41	1.95	Cluster2

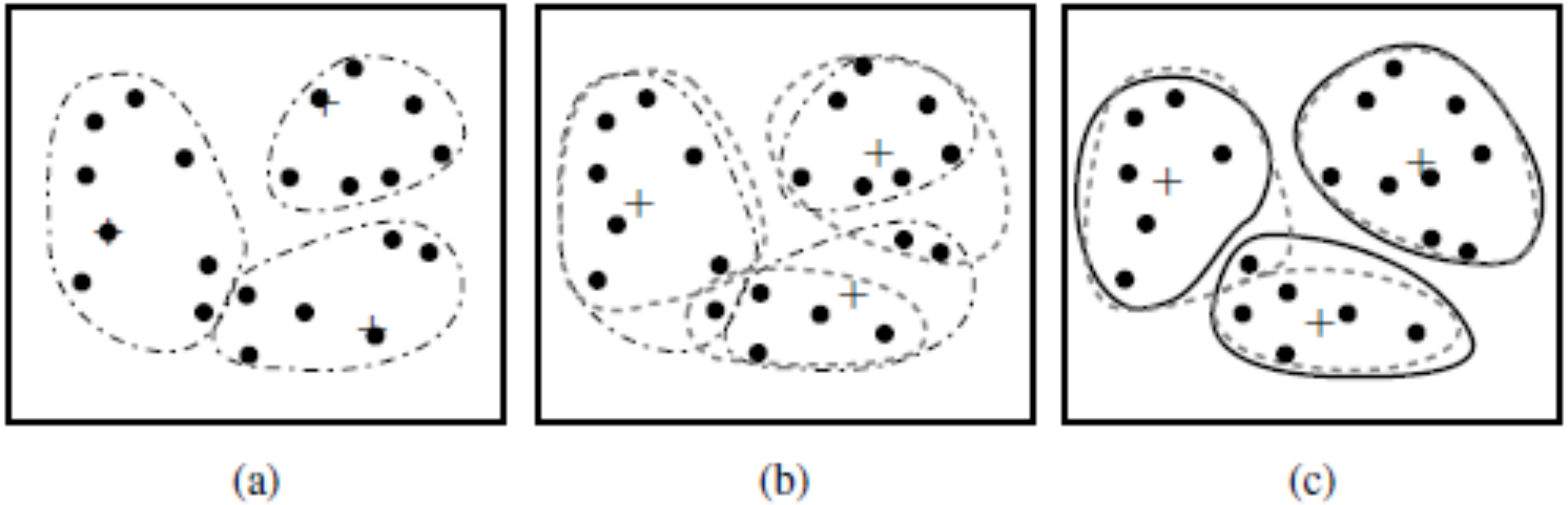
m1	(3.67, 5.83)
m2	(6.75, 3.50)

Cluster Analysis

Cluster Analysis

- Used for automatic identification of natural groupings of things
- Part of the machine-learning family
- Employ unsupervised learning
- Learns the clusters of things from past data, then assigns new instances
- There is not an output variable
- Also known as segmentation

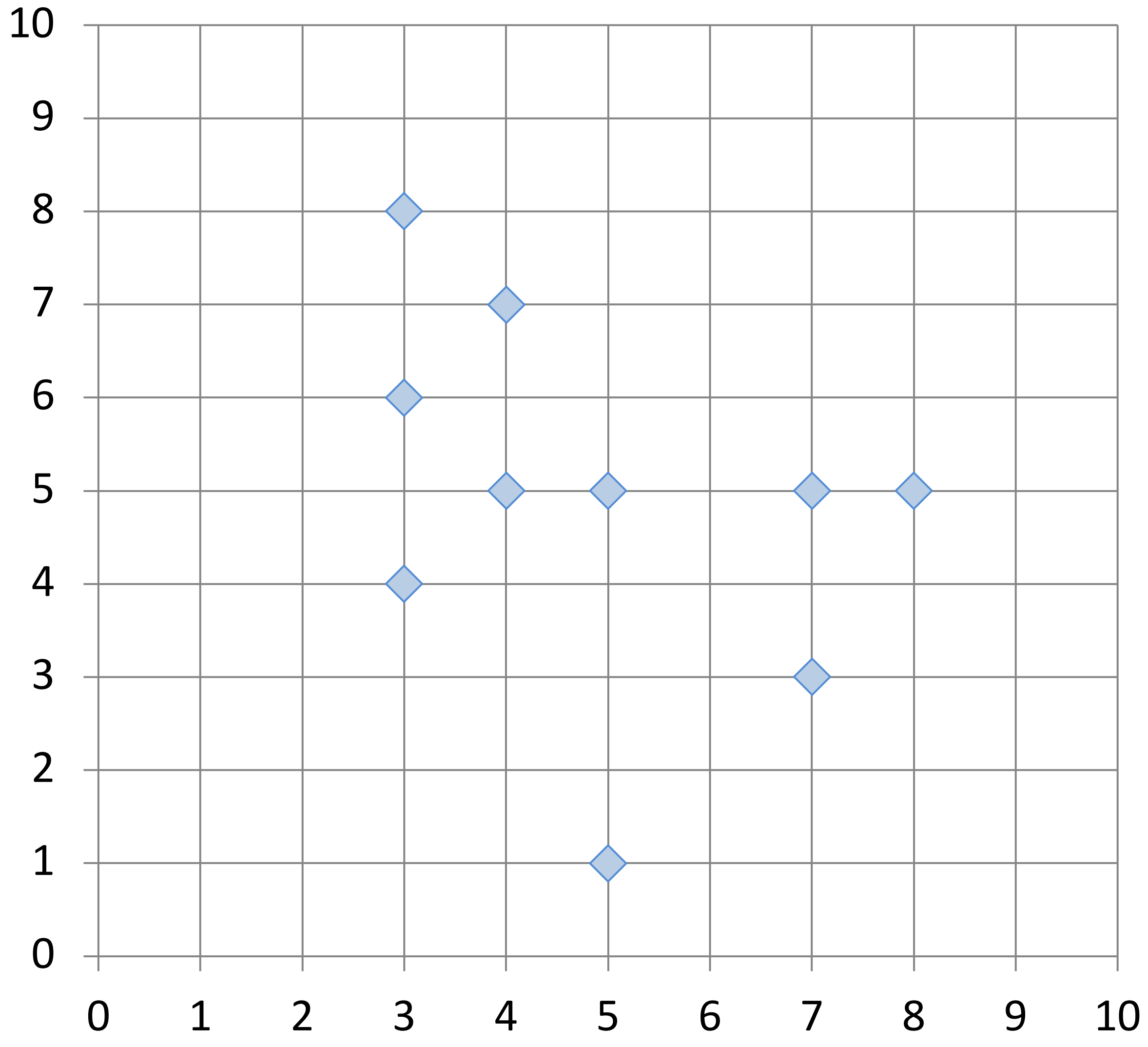
Cluster Analysis



Clustering of a set of objects based on the *k-means method.*
(The mean of each cluster is marked by a “+”.)

Source: Han & Kamber (2006)

Example of Cluster Analysis



Point	P	P(x,y)
p01	a	(3, 4)
p02	b	(3, 6)
p03	c	(3, 8)
p04	d	(4, 5)
p05	e	(4, 7)
p06	f	(5, 1)
p07	g	(5, 5)
p08	h	(7, 3)
p09	i	(7, 5)
p10	j	(8, 5)

Cluster Analysis for Data Mining

- **How many clusters?**
  - **There is not a “truly optimal” way to calculate it**
  - **Heuristics are often used**
    1. Look at the sparseness of clusters
    2. Number of clusters = $(n/2)^{1/2}$ (n: no of data points)
    3. Use Akaike information criterion (AIC)
    4. Use Bayesian information criterion (BIC)
- **Most cluster analysis methods involve the use of a distance measure to calculate the closeness between pairs of items**
  - **Euclidian versus Manhattan (rectilinear) distance**

Source: Turban et al. (2011), Decision Support and Business Intelligence Systems

k-Means Clustering Algorithm

- *k* : pre-determined number of clusters
- Algorithm (**Step 0:** determine value of *k*)

Step 1: Randomly generate *k* random points as initial cluster centers

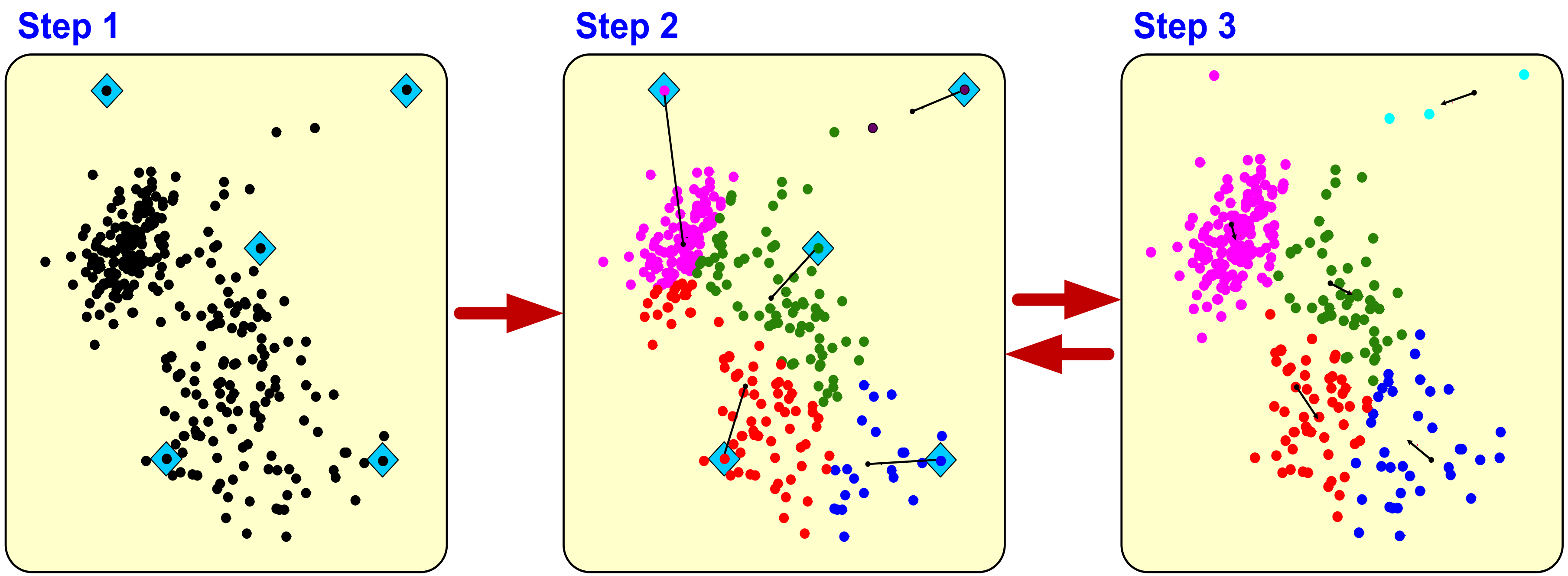
Step 2: Assign each point to the nearest cluster center

Step 3: Re-compute the new cluster centers

Repetition step: Repeat steps 2 and 3 until some convergence criterion is met (usually that the assignment of points to clusters becomes stable)

Source: Turban et al. (2011), Decision Support and Business Intelligence Systems

Cluster Analysis for Data Mining - *k*-Means Clustering Algorithm



Source: Turban et al. (2011), Decision Support and Business Intelligence Systems

Similarity

Distance

Similarity and Dissimilarity Between Objects

- Distances are normally used to measure the similarity or dissimilarity between two data objects
- Some popular ones include: ***Minkowski distance***:

$$d(i,j)=\sqrt[q]{(|x_{i_1}-x_{j_1}|^q+|x_{i_2}-x_{j_2}|^q+...+|x_{i_p}-x_{j_p}|^q)}$$

where $i = (x_{i1}, x_{i2}, ..., x_{ip})$ and $j = (x_{j1}, x_{j2}, ..., x_{jp})$ are two p-dimensional data objects, and q is a positive integer

- If $q = 1$, d is Manhattan distance

$$d(i,j)=|x_{i_1}-x_{j_1}|+|x_{i_2}-x_{j_2}|+...+|x_{i_p}-x_{j_p}|$$

Similarity and Dissimilarity Between Objects (Cont.)

- ***If q = 2, d*** **is Euclidean distance:**

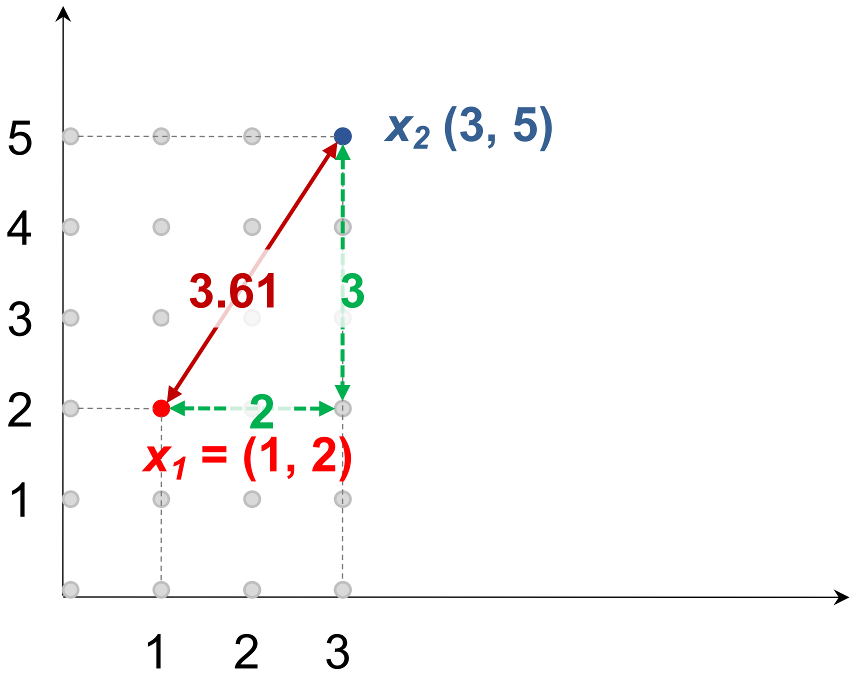
$$d(i,j) = \sqrt{(|x_{i_1} - x_{j_1}|^2 + |x_{i_2} - x_{j_2}|^2 + ... + |x_{i_p} - x_{j_p}|^2)}$$

  - **Properties**
    - $d(i,j) \geq 0$
    - $d(i,i) = 0$
    - $d(i,j) = d(j,i)$
    - $d(i,j) \leq d(i,k) + d(k,j)$
- **Also, one can use weighted distance, parametric Pearson product moment correlation, or other disimilarity measures**

Source: Han & Kamber (2006)

Euclidean distance vs Manhattan distance

- **Distance of two point $x_1 = (1, 2)$ and $x_2\ (3, 5)$**

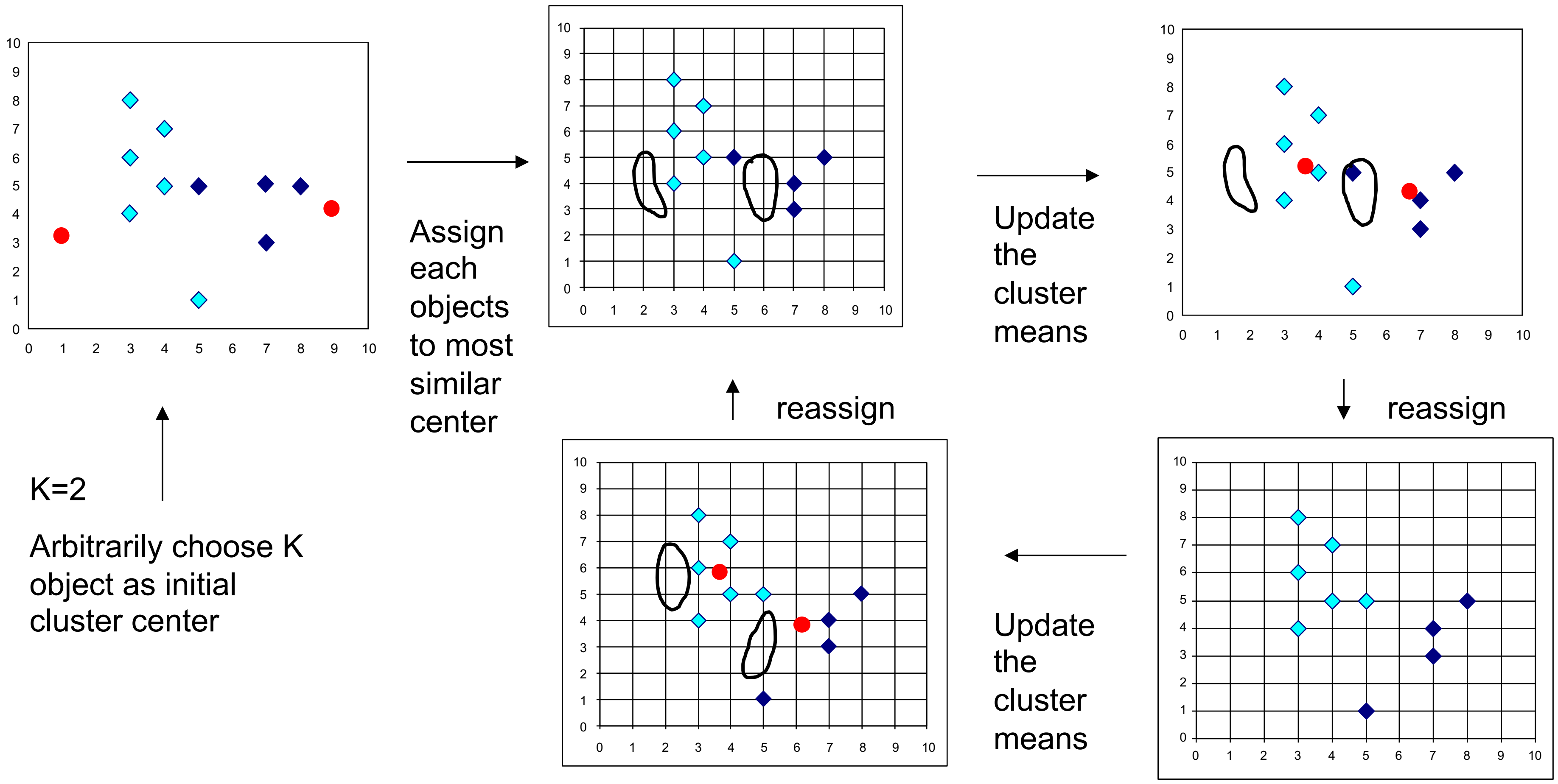


Euclidean distance:
$= ((3-1)^2 + (5-2)^2\ )^{1/2}$
$= (2^2 + 3^2)^{1/2}$
$= (4 + 9)^{1/2}$
$= (13)^{1/2}$
$= 3.61$

Manhattan distance:
$= (3-1) + (5-2)$
$= 2 + 3$
$= 5$

The *K-Means* Clustering Method

- **Example**



Source: Han & Kamber (2006)

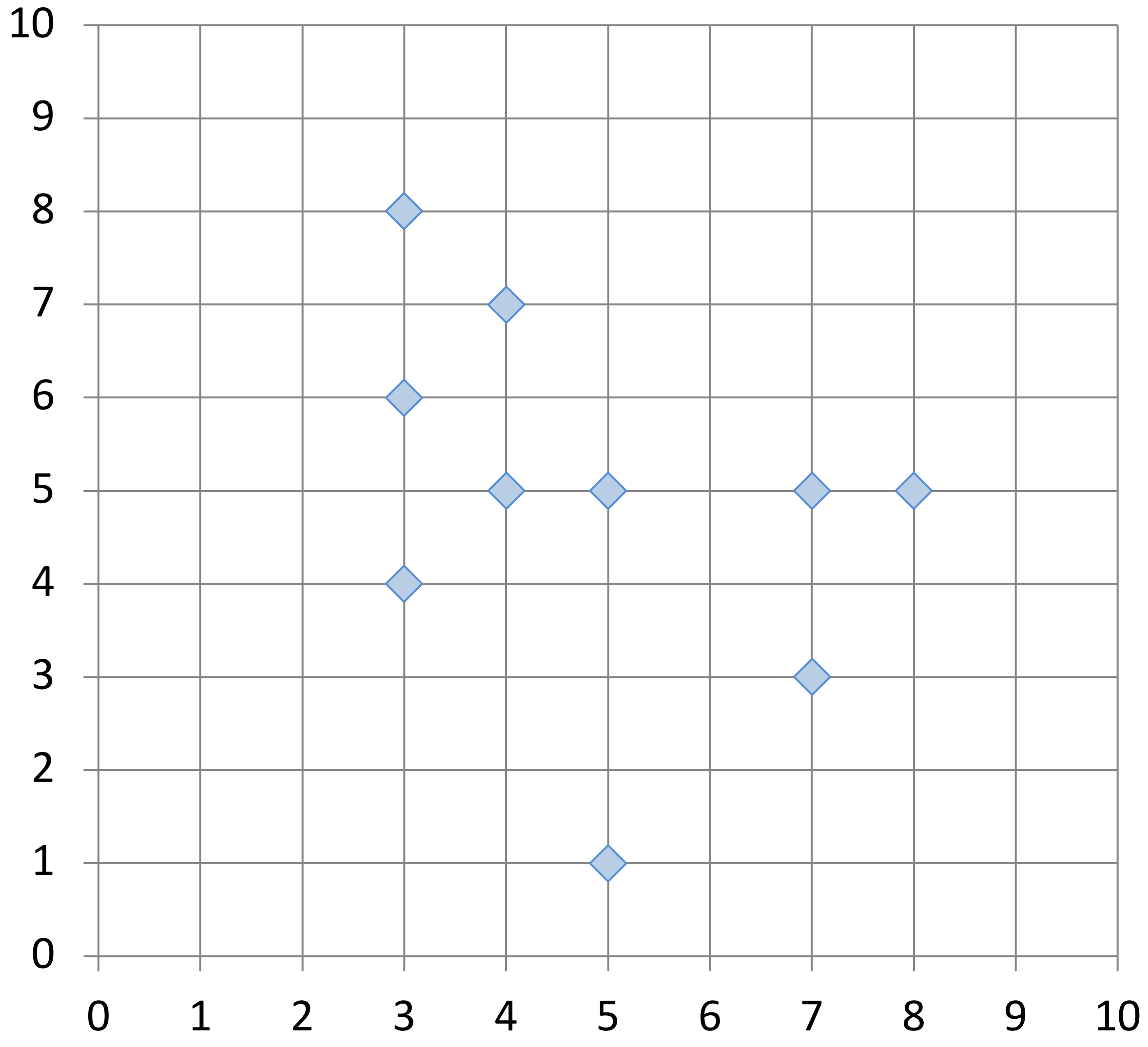
K-Means Clustering

Example of Cluster Analysis

Point	P	P(x,y)
p01	a	(3, 4)
p02	b	(3, 6)
p03	c	(3, 8)
p04	d	(4, 5)
p05	e	(4, 7)
p06	f	(5, 1)
p07	g	(5, 5)
p08	h	(7, 3)
p09	i	(7, 5)
p10	j	(8, 5)

K-Means Clustering

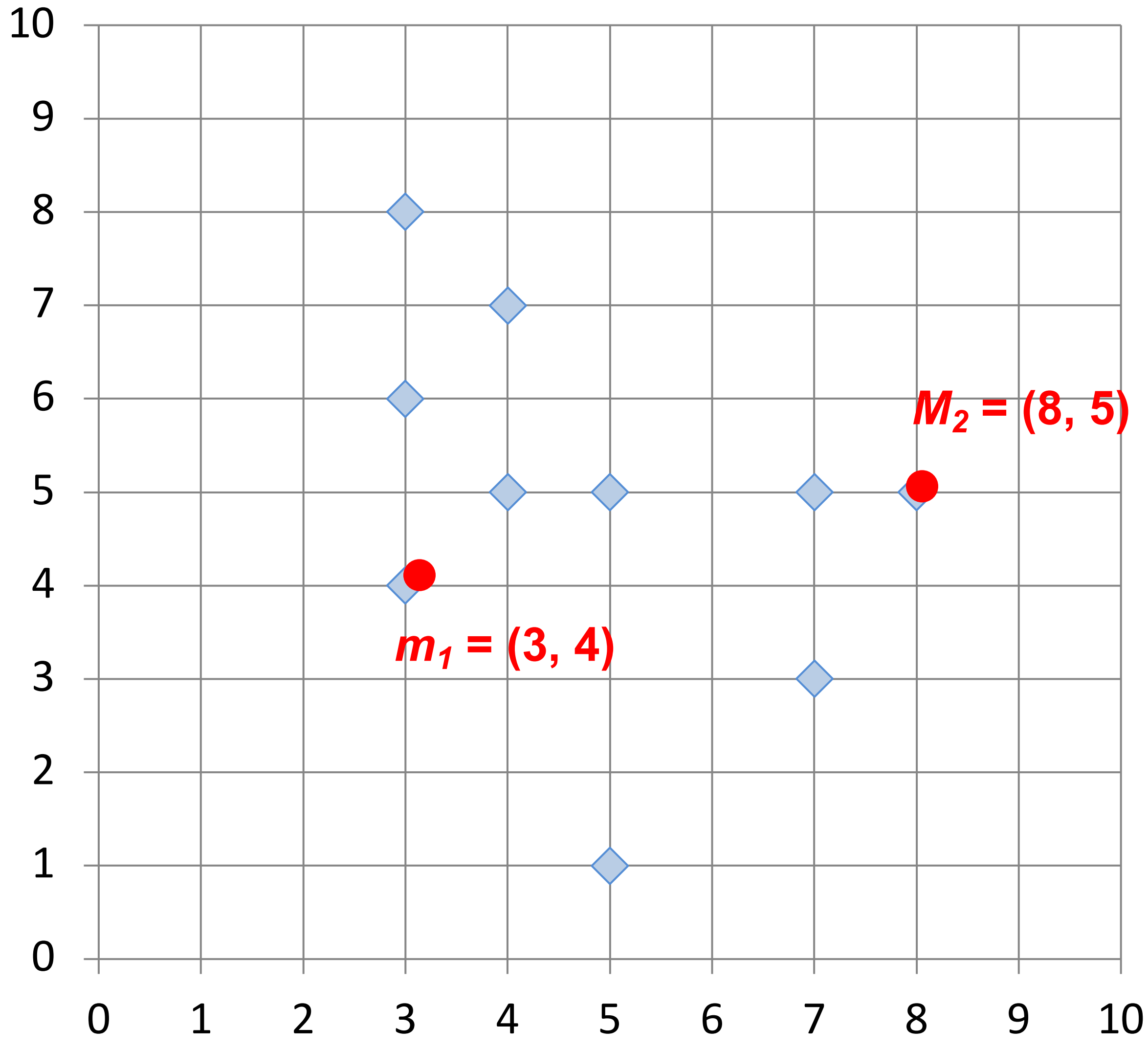
Step by Step



Point	P	P(x,y)
p01	a	(3, 4)
p02	b	(3, 6)
p03	c	(3, 8)
p04	d	(4, 5)
p05	e	(4, 7)
p06	f	(5, 1)
p07	g	(5, 5)
p08	h	(7, 3)
p09	i	(7, 5)
p10	j	(8, 5)

K-Means Clustering

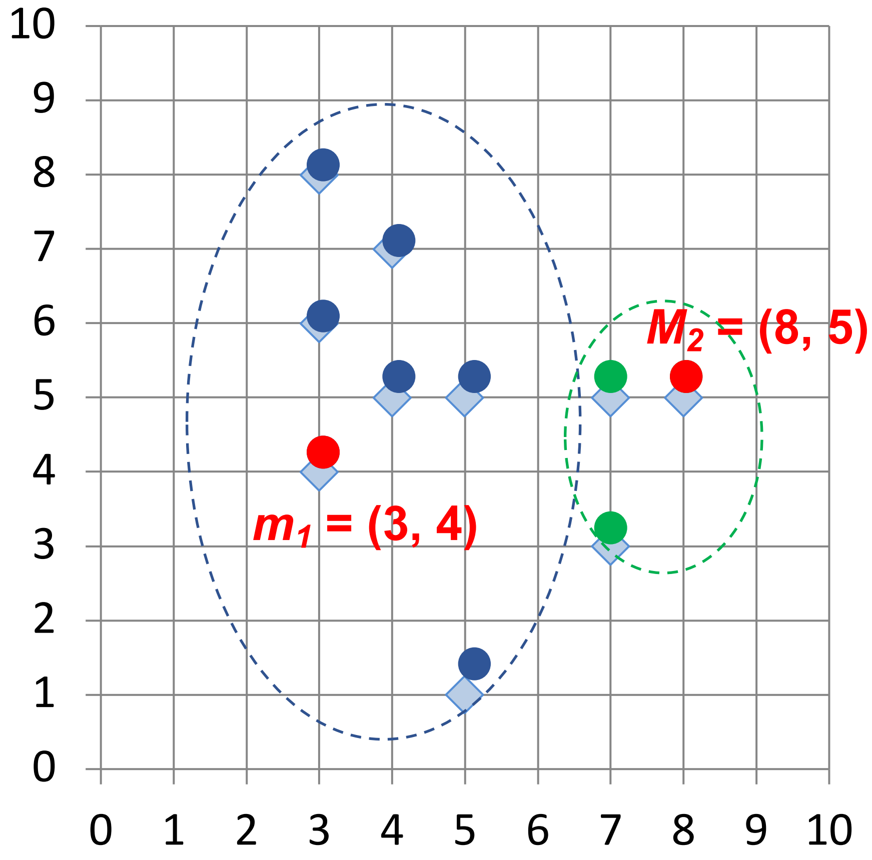
Step 1: K=2, Arbitrarily choose K object as initial cluster center



Point	P	P(x,y)
p01	a	(3, 4)
p02	b	(3, 6)
p03	c	(3, 8)
p04	d	(4, 5)
p05	e	(4, 7)
p06	f	(5, 1)
p07	g	(5, 5)
p08	h	(7, 3)
p09	i	(7, 5)
p10	j	(8, 5)
Initial	m1	(3, 4)
Initial	m2	(8, 5)

Step 2: Compute seed points as the centroids of the clusters of the current partition

Step 3: Assign each objects to most similar center



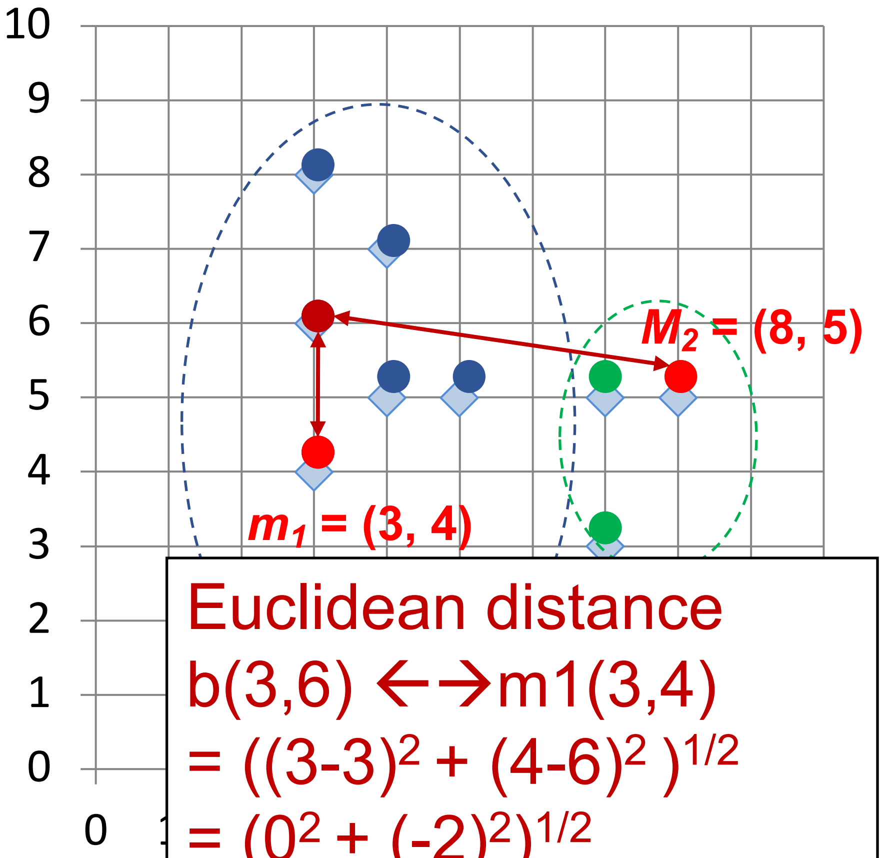
Point	P	P(x,y)	m1 distance	m2 distance	Cluster
p01	a	(3, 4)	0.00	5.10	Cluster1
p02	b	(3, 6)	2.00	5.10	Cluster1
p03	c	(3, 8)	4.00	5.83	Cluster1
p04	d	(4, 5)	1.41	4.00	Cluster1
p05	e	(4, 7)	3.16	4.47	Cluster1
p06	f	(5, 1)	3.61	5.00	Cluster1
p07	g	(5, 5)	2.24	3.00	Cluster1
p08	h	(7, 3)	4.12	2.24	Cluster2
p09	i	(7, 5)	4.12	1.00	Cluster2
p10	j	(8, 5)	5.10	0.00	Cluster2

Initial	m1	(3, 4)
Initial	m2	(8, 5)

K-Means Clustering

Step 2: Compute seed points as the centroids of the clusters of the current partition

Step 3: Assign each objects to most similar center



Euclidean distance

b(3,6) ←→m1(3,4)

$= ((3-3)^2 + (4-6)^2)^{1/2}$

$= (0^2 + (-2)^2)^{1/2}$

$= (0 + 4)^{1/2}$

$= (4)^{1/2}$

= 2.00

K-

Point	P	P(x,y)	m1 distance	m2 distance	Cluster
p01	a	(3, 4)	0.00	5.10	Cluster1
p02	b	(3, 6)	2.00	5.10	Cluster1
p03	c	(3, 8)	4.00	5.83	Cluster1
p04	d	(4, 5)	1.41	4.00	Cluster1
p05					ster1
p06					ster1
p07					ster1
p08					ster2
p09					ster2
p10					ster2

Euclidean distance

b(3,6) ←→m2(8,5)

$= ((8-3)^2 + (5-6)^2)^{1/2}$

$= (5^2 + (-1)^2)^{1/2}$

$= (25 + 1)^{1/2}$

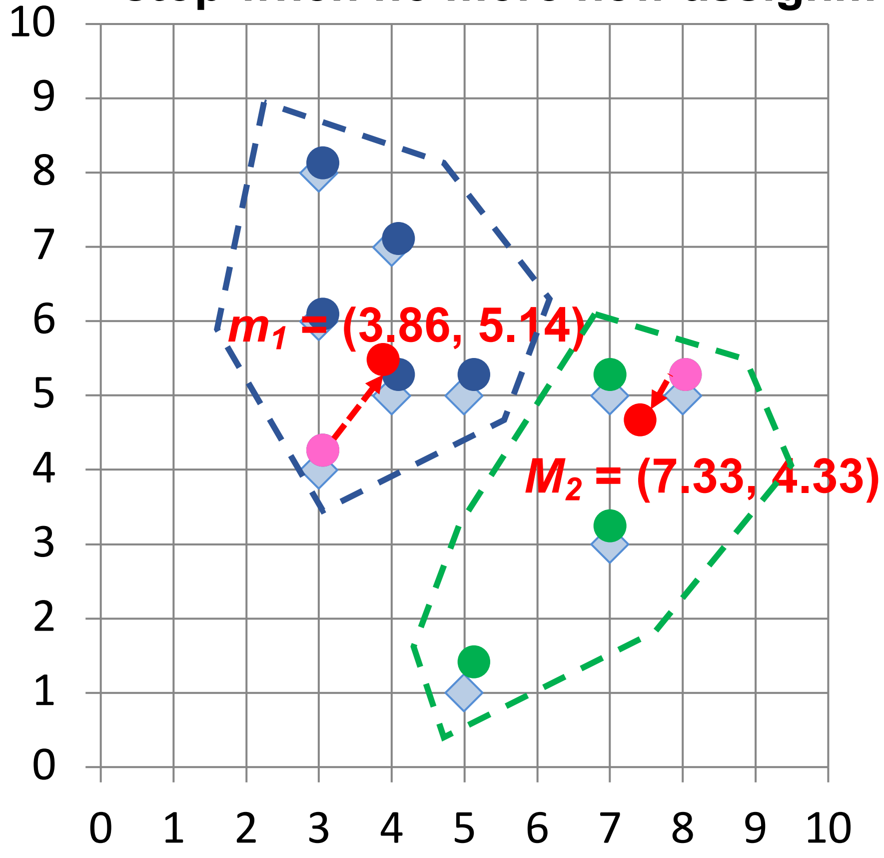
$= (26)^{1/2}$

= 5.10

Initial m1 (3, 4)

Initial m2 (8, 5)

Step 4: Update the cluster means,
Repeat Step 2, 3,
stop when no more new assignment



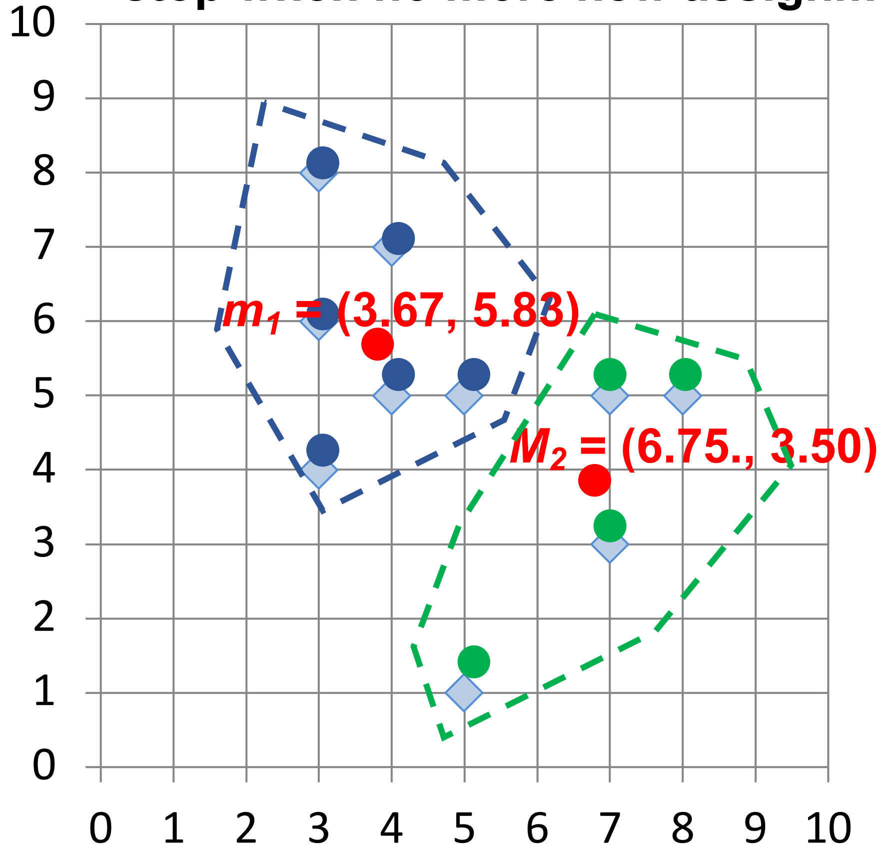
Point	P	P(x,y)	m1 distance	m2 distance	Cluster
p01	a	(3, 4)	1.43	4.34	Cluster1
p02	b	(3, 6)	1.22	4.64	Cluster1
p03	c	(3, 8)	2.99	5.68	Cluster1
p04	d	(4, 5)	0.20	3.40	Cluster1
p05	e	(4, 7)	1.87	4.27	Cluster1
p06	f	(5, 1)	4.29	4.06	Cluster2
p07	g	(5, 5)	1.15	2.42	Cluster1
p08	h	(7, 3)	3.80	1.37	Cluster2
p09	i	(7, 5)	3.14	0.75	Cluster2
p10	j	(8, 5)	4.14	0.95	Cluster2

m1 (3.86, 5.14)

m2 (7.33, 4.33)

K-Means Clustering

Step 4: Update the cluster means,
Repeat Step 2, 3,
stop when no more new assignment

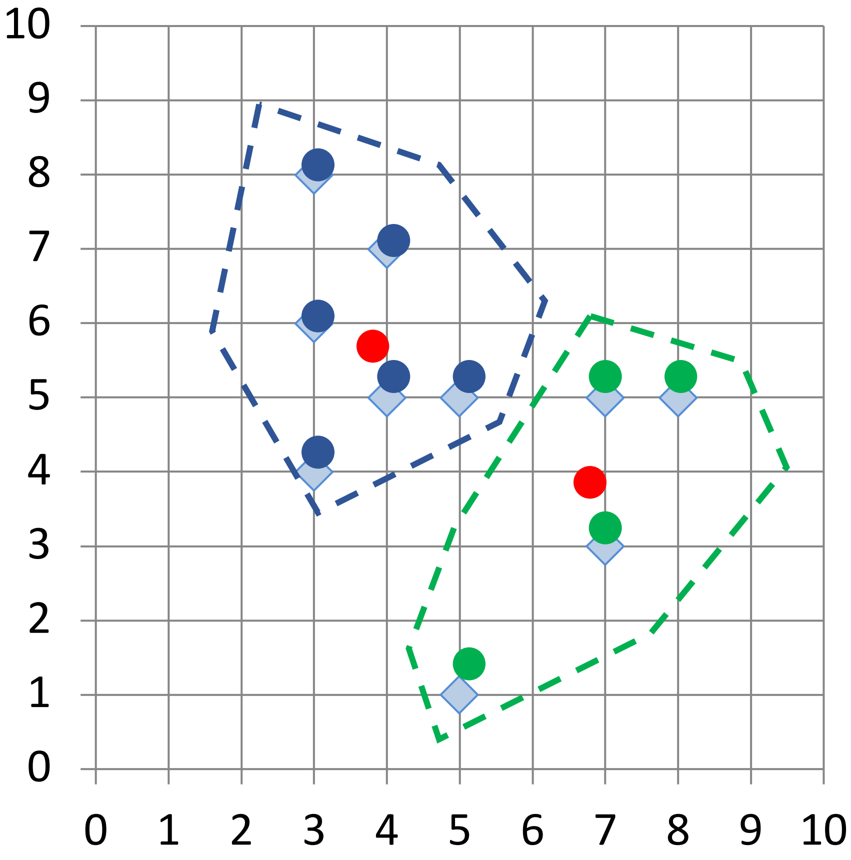


Point	P	P(x,y)	m1 distance	m2 distance	Cluster
p01	a	(3, 4)	1.95	3.78	Cluster1
p02	b	(3, 6)	0.69	4.51	Cluster1
p03	c	(3, 8)	2.27	5.86	Cluster1
p04	d	(4, 5)	0.89	3.13	Cluster1
p05	e	(4, 7)	1.22	4.45	Cluster1
p06	f	(5, 1)	5.01	3.05	Cluster2
p07	g	(5, 5)	1.57	2.30	Cluster1
p08	h	(7, 3)	4.37	0.56	Cluster2
p09	i	(7, 5)	3.43	1.52	Cluster2
p10	j	(8, 5)	4.41	1.95	Cluster2

m1 (3.67, 5.83)
m2 (6.75, 3.50)

K-Means Clustering

stop when no more new assignment



Point	P	P(x,y)	m1 distance	m2 distance	Cluster
p01	a	(3, 4)	1.95	3.78	Cluster1
p02	b	(3, 6)	0.69	4.51	Cluster1
p03	c	(3, 8)	2.27	5.86	Cluster1
p04	d	(4, 5)	0.89	3.13	Cluster1
p05	e	(4, 7)	1.22	4.45	Cluster1
p06	f	(5, 1)	5.01	3.05	Cluster2
p07	g	(5, 5)	1.57	2.30	Cluster1
p08	h	(7, 3)	4.37	0.56	Cluster2
p09	i	(7, 5)	3.43	1.52	Cluster2
p10	j	(8, 5)	4.41	1.95	Cluster2

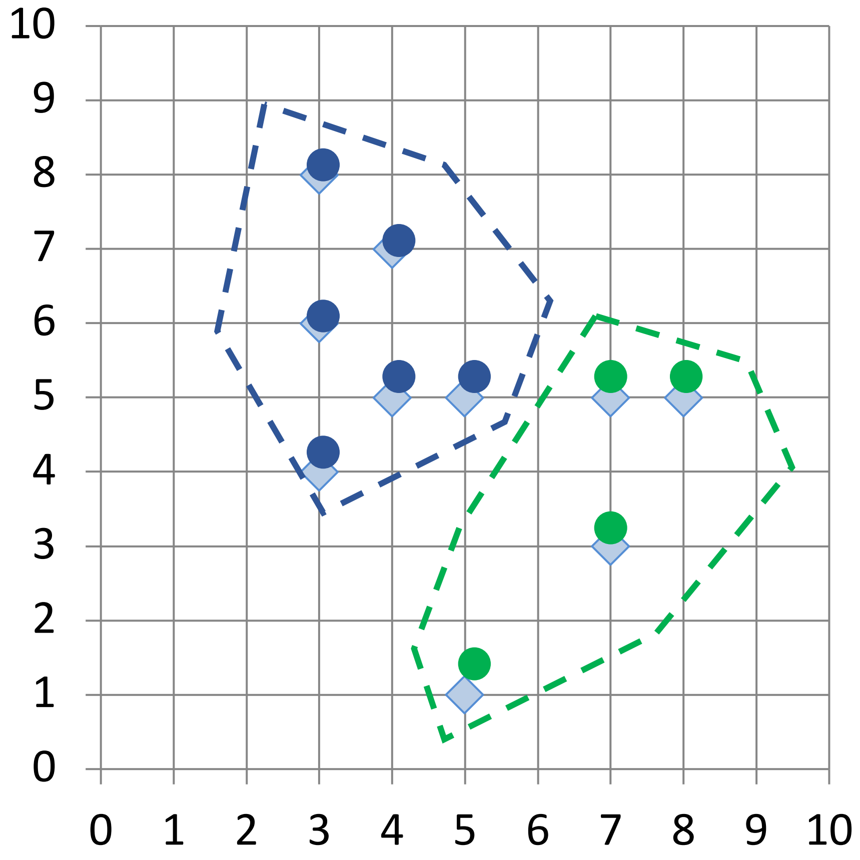
m1 (3.67, 5.83)

m2 (6.75, 3.50)

K-Means Clustering

K-Means Clustering (*K*=2, two clusters)

stop when no more new assignment



Point	P	P(x,y)	m1 distance	m2 distance	Cluster
p01	a	(3, 4)	1.95	3.78	Cluster1
p02	b	(3, 6)	0.69	4.51	Cluster1
p03	c	(3, 8)	2.27	5.86	Cluster1
p04	d	(4, 5)	0.89	3.13	Cluster1
p05	e	(4, 7)	1.22	4.45	Cluster1
p06	f	(5, 1)	5.01	3.05	Cluster2
p07	g	(5, 5)	1.57	2.30	Cluster1
p08	h	(7, 3)	4.37	0.56	Cluster2
p09	i	(7, 5)	3.43	1.52	Cluster2
p10	j	(8, 5)	4.41	1.95	Cluster2

m1 (3.67, 5.83)

m2 (6.75, 3.50)

***K-Means* Clustering**

K-Means Clustering

Point	P	P(x,y)	m1 distance	m2 distance	Cluster
p01	a	(3, 4)	1.95	3.78	Cluster1
p02	b	(3, 6)	0.69	4.51	Cluster1
p03	c	(3, 8)	2.27	5.86	Cluster1
p04	d	(4, 5)	0.89	3.13	Cluster1
p05	e	(4, 7)	1.22	4.45	Cluster1
p06	f	(5, 1)	5.01	3.05	Cluster2
p07	g	(5, 5)	1.57	2.30	Cluster1
p08	h	(7, 3)	4.37	0.56	Cluster2
p09	i	(7, 5)	3.43	1.52	Cluster2
p10	j	(8, 5)	4.41	1.95	Cluster2

m1	(3.67, 5.83)
m2	(6.75, 3.50)

gensim

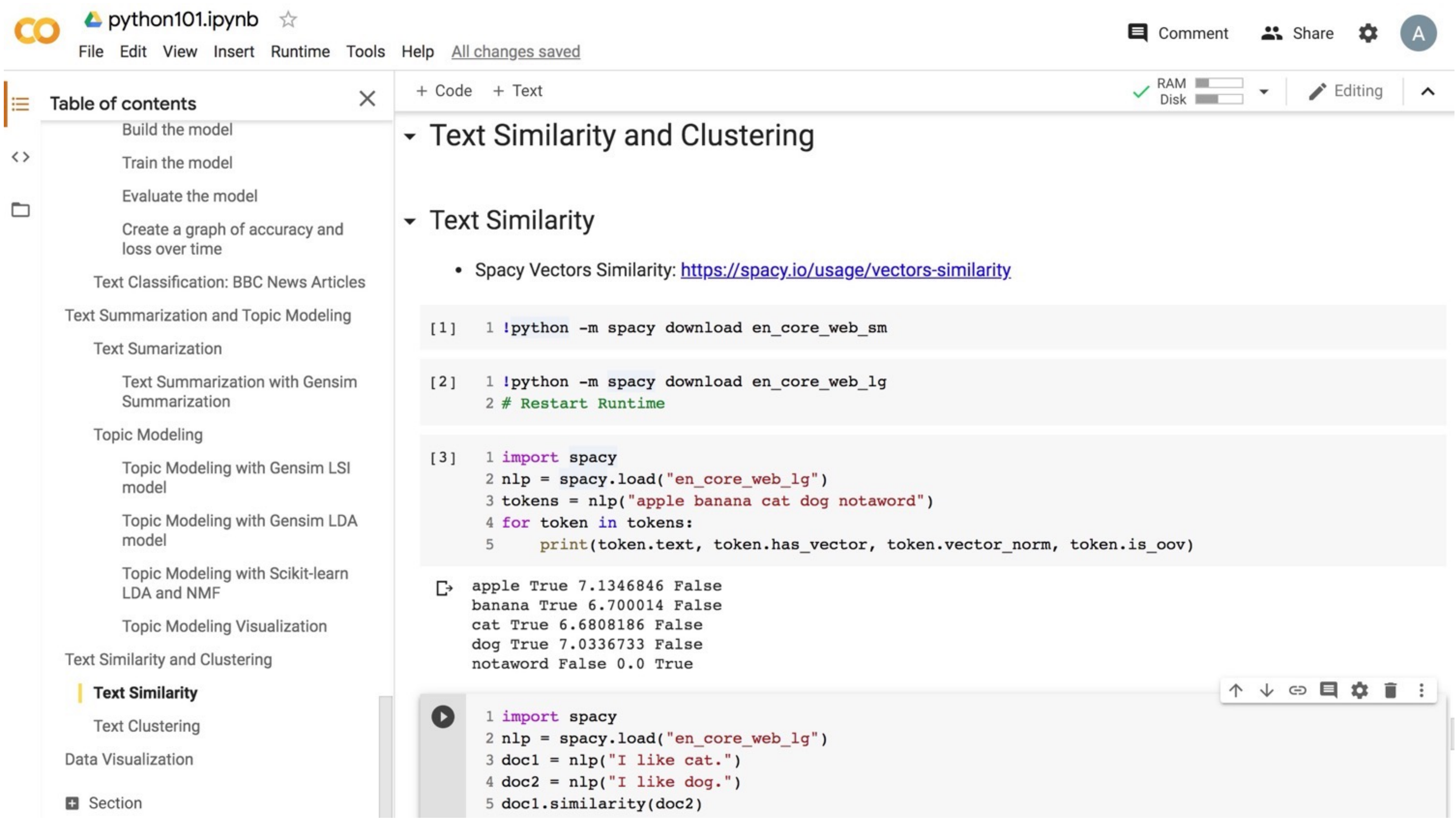
https://radimrehurek.com/gensim/

spaCy

https://spacy.io/

Python in Google Colab (Python101)

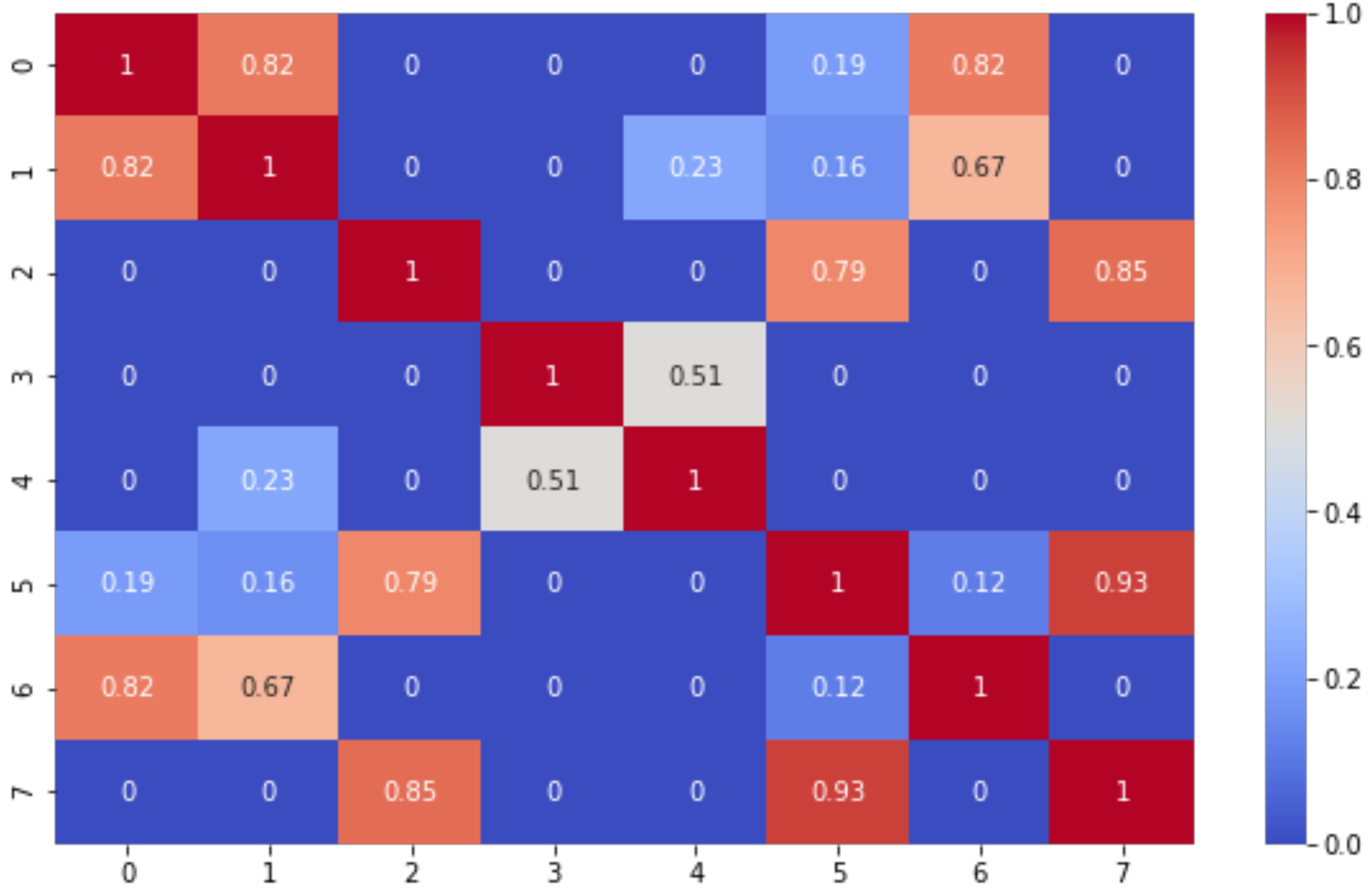
https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT



https://tinyurl.com/aintpupython101

Python in Google Colab (Python101)

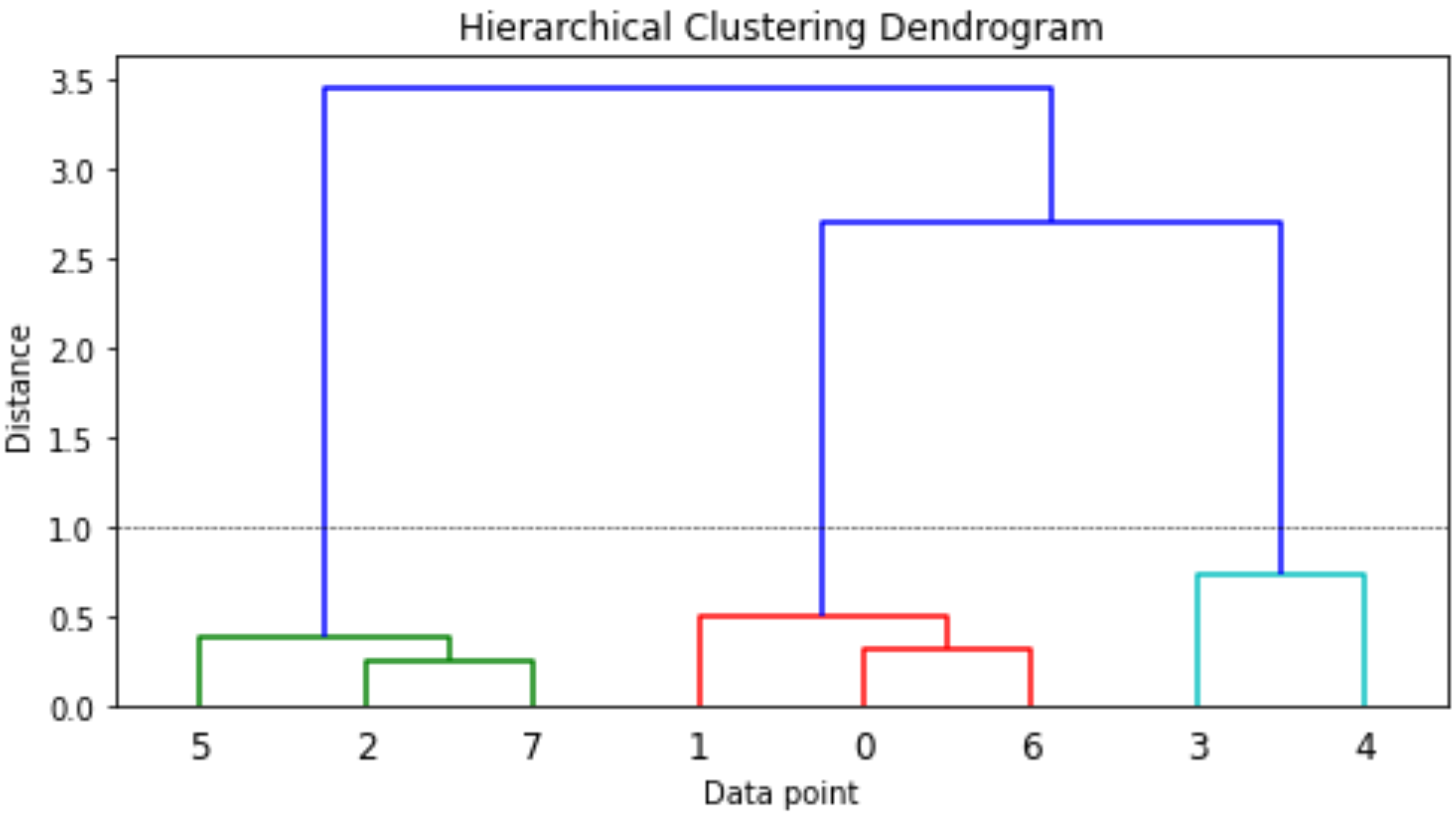
https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT



https://tinyurl.com/aintpupython101

Python in Google Colab (Python101)

https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT



https://tinyurl.com/aintpupython101

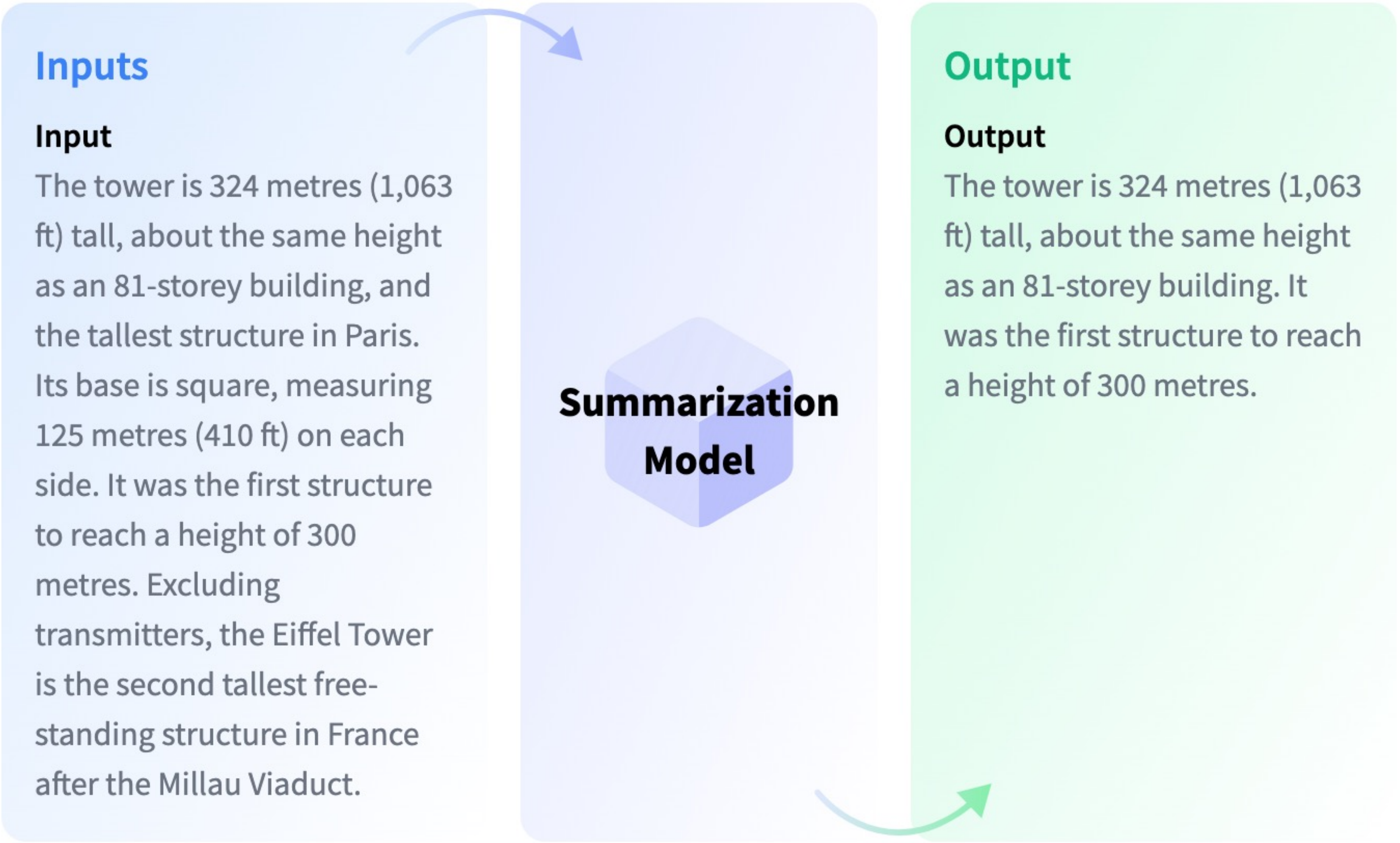
Text Summarization
and
Topic Models

Outline

- Text Summarization
  - Extractive Text Summarization
  - Abstractive Text Summarization
    - PEGASUS: Abstractive Summarization
- Topic Models
  - Topic Modeling
  - Latent Dirichlet Allocation (LDA)
  - BERTopic

Text Summarization

Text Summarization



https://huggingface.co/tasks/summarization

Text Summarization

Summarization

Examples

The tower is 324 metres (1,063 ft) tall, about the same height as an 81-storey building, and the tallest structure in Paris. Its base is square, measuring 125 metres (410 ft) on each side. During its construction, the Eiffel Tower surpassed the Washington Monument to become the tallest man-made structure in the world, a title it held for 41 years until the Chrysler Building in New York City was finished in 1930. It was the first structure to reach a height of 300 metres. Due to the addition of a broadcasting aerial at the top of the tower in 1957, it is now taller than the Chrysler Building by 5.2 metres (17 ft). Excluding transmitters, the Eiffel Tower is the second tallest free-standing structure in France after the Millau Viaduct.

Compute

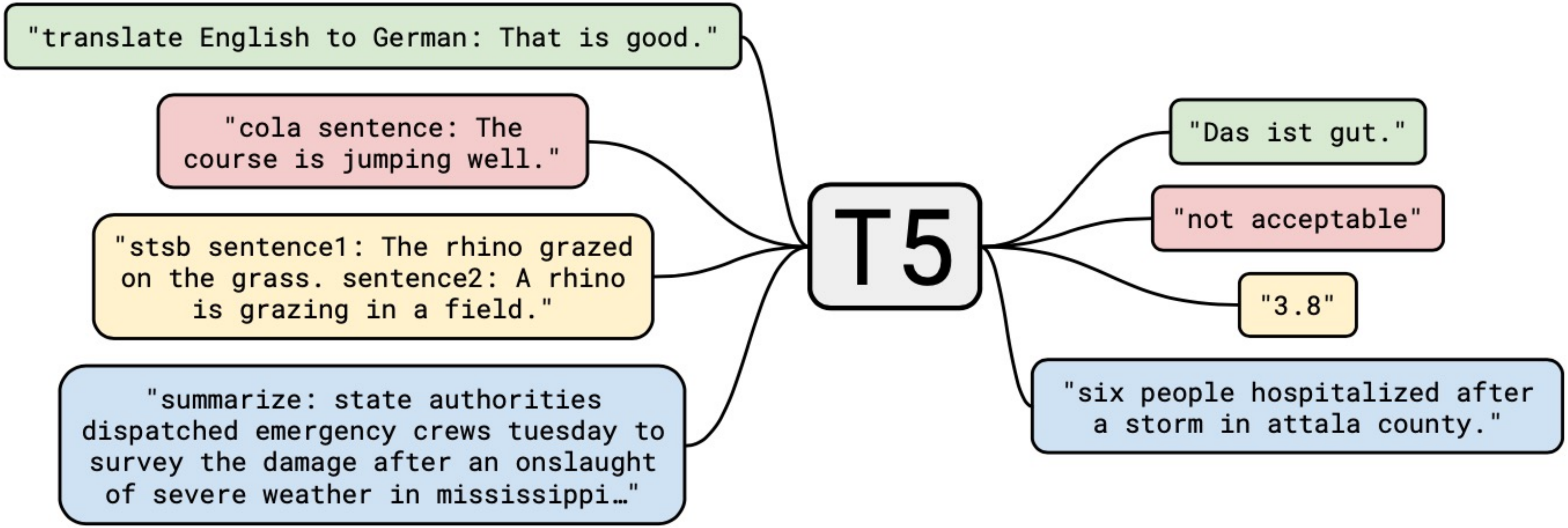
Computation time on cpu: cached

The tower is 324 metres (1,063 ft) tall, about the same height as an 81-storey building . It was the first structure to reach a height of 300 metres . It is now taller than the Chrysler Building in New York City by 5.2 metres (17 ft) Excluding transmitters, the Eiffel Tower is the second tallest free-standing structure in France .

https://huggingface.co/tasks/summarization

T5

Text-to-Text Transfer Transformer



Source: JColin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu (2019). "Exploring the limits of transfer learning with a unified text-to-text transformer." arXiv preprint arXiv:1910.10683 (2019).

Text Summarization and Information Extraction

- Key-phrase extraction
  - **extracting key influential phrases from the documents.**
- Topic modeling
  - **Extract various diverse concepts or topics present in the documents, retaining the major themes.**
- Document summarization
  - **Summarize entire text documents to provide a gist that retains the important parts of the whole corpus.**

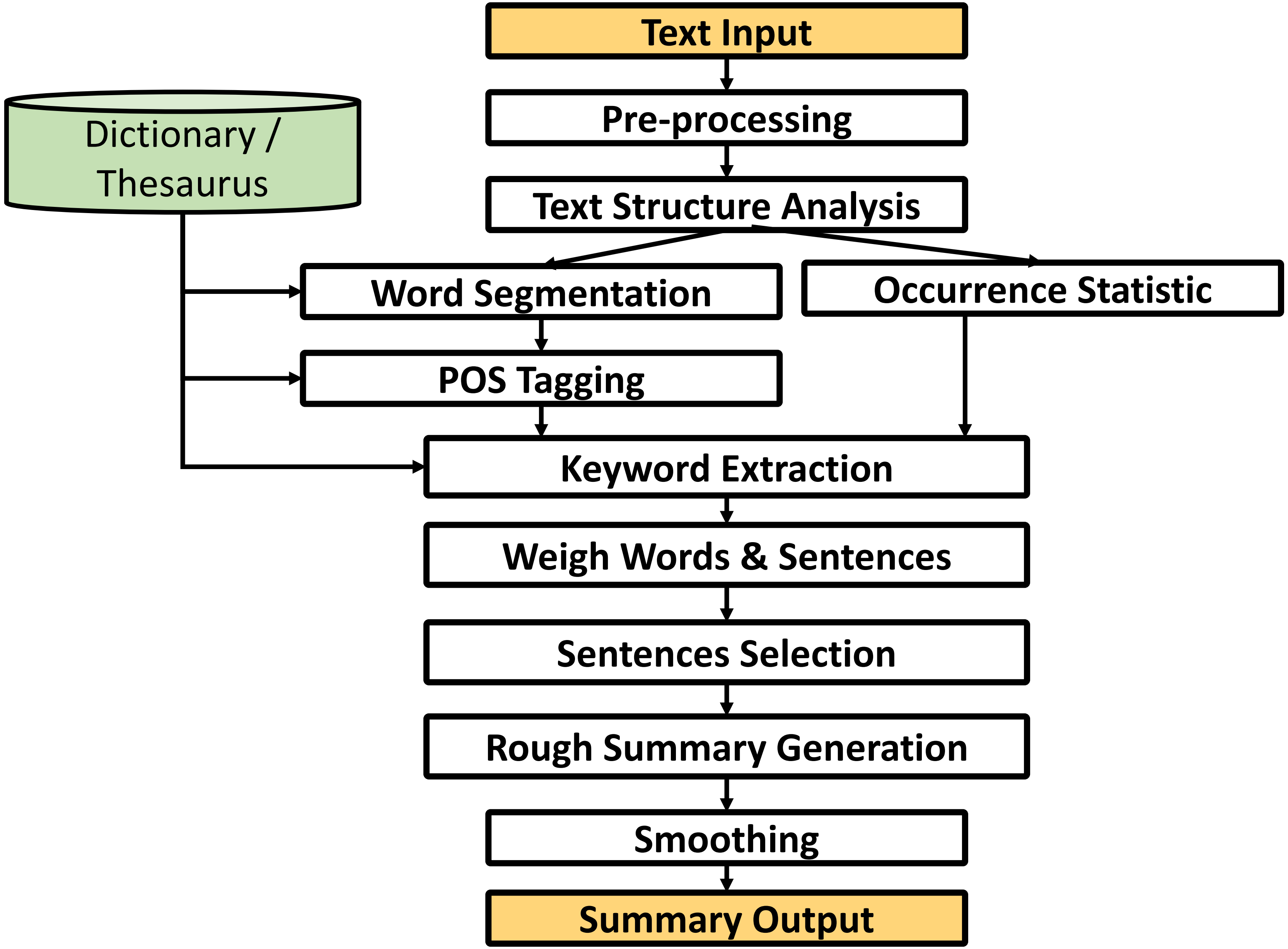
Natural Language Processing (NLP) and Text Mining

- Raw text
- Sentence Segmentation
- Tokenization
- Part-of-Speech (POS)
- Stop word removal
- Stemming / Lemmatization
- Dependency Parser
- String Metrics & Matching

word's stem	word's lemma
am → am	am → be
having → hav	having → have

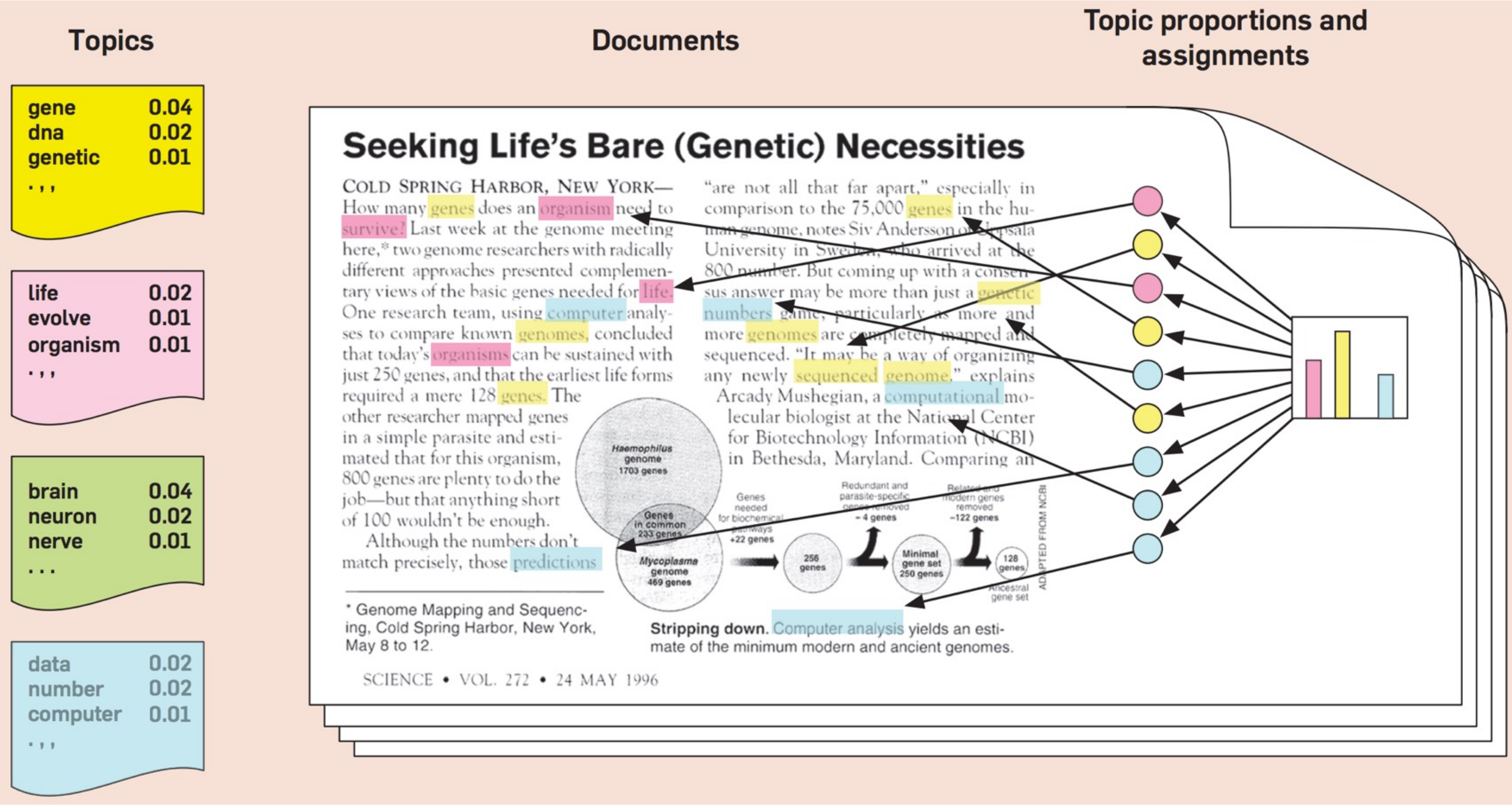
Source: Nitin Hardeniya (2015), NLTK Essentials, Packt Publishing; Florian Leitner (2015), Text mining - from Bayes rule to dependency parsing

Text Summarization



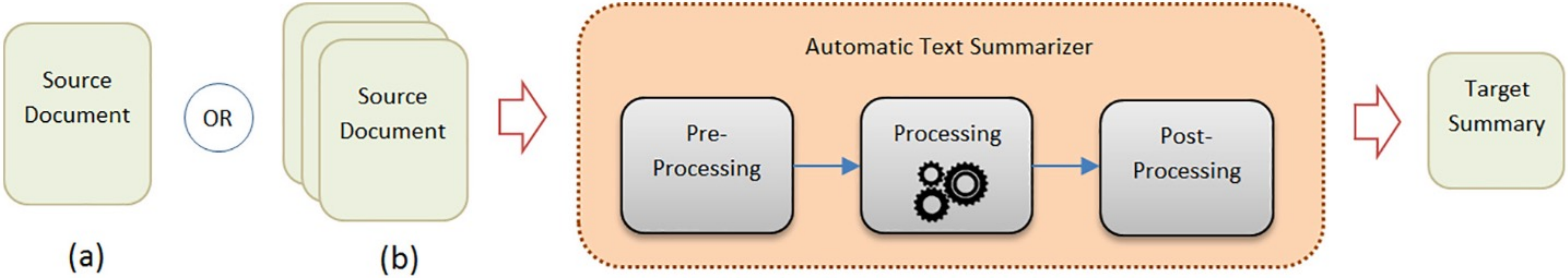
Source: Vishal Gupta and Gurpreet S. Lehal (2009), "A survey of text mining techniques and applications," Journal of emerging technologies in web intelligence, vol. 1, no. 1, pp. 60-76.

Topic Modeling



Source: Blei, David M. "Probabilistic topic models." *Communications of the ACM* 55, no. 4 (2012): 77-84.

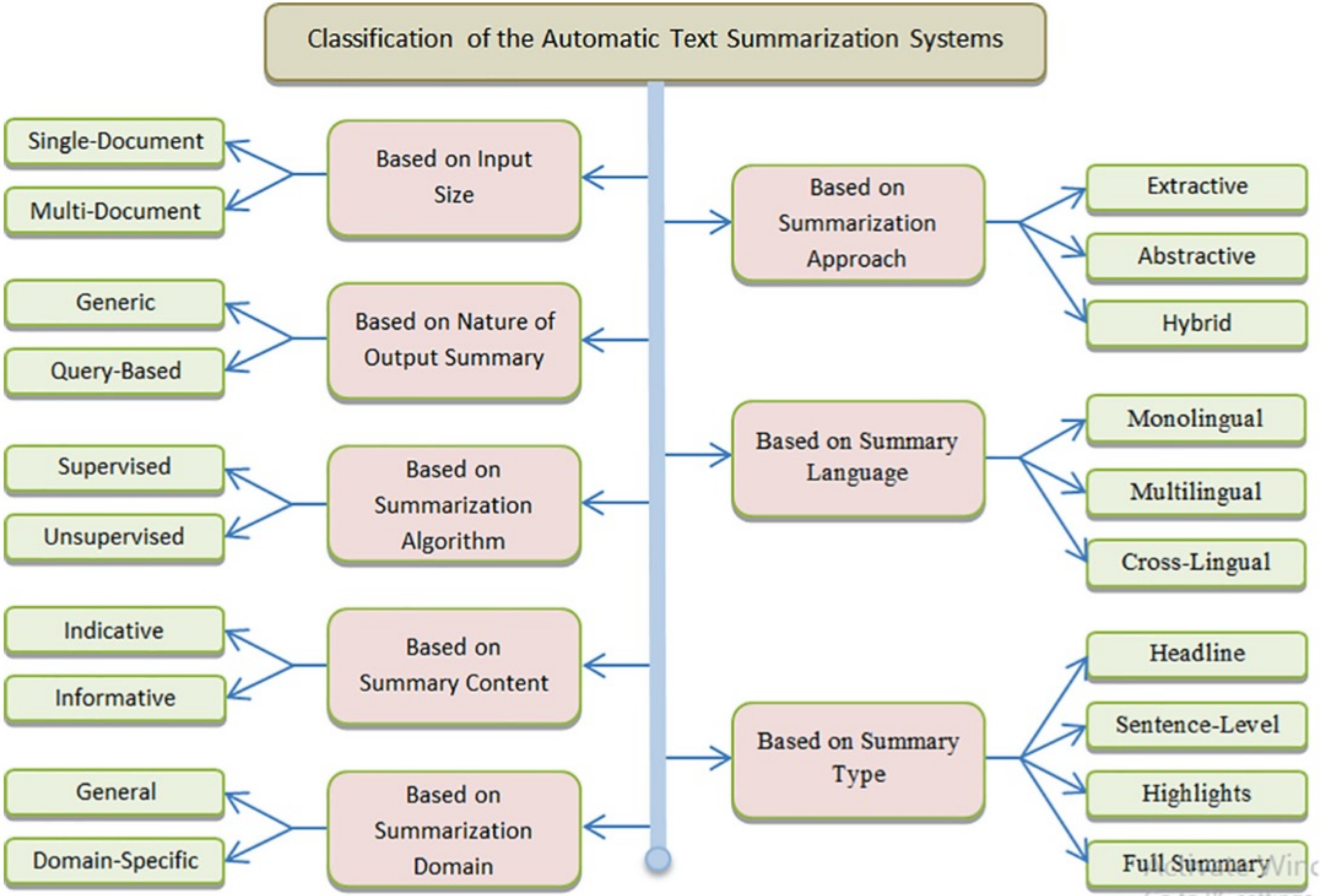
Automatic Text Summarization



(a) Single-document or (b) Multi-document, automatic text summarizer

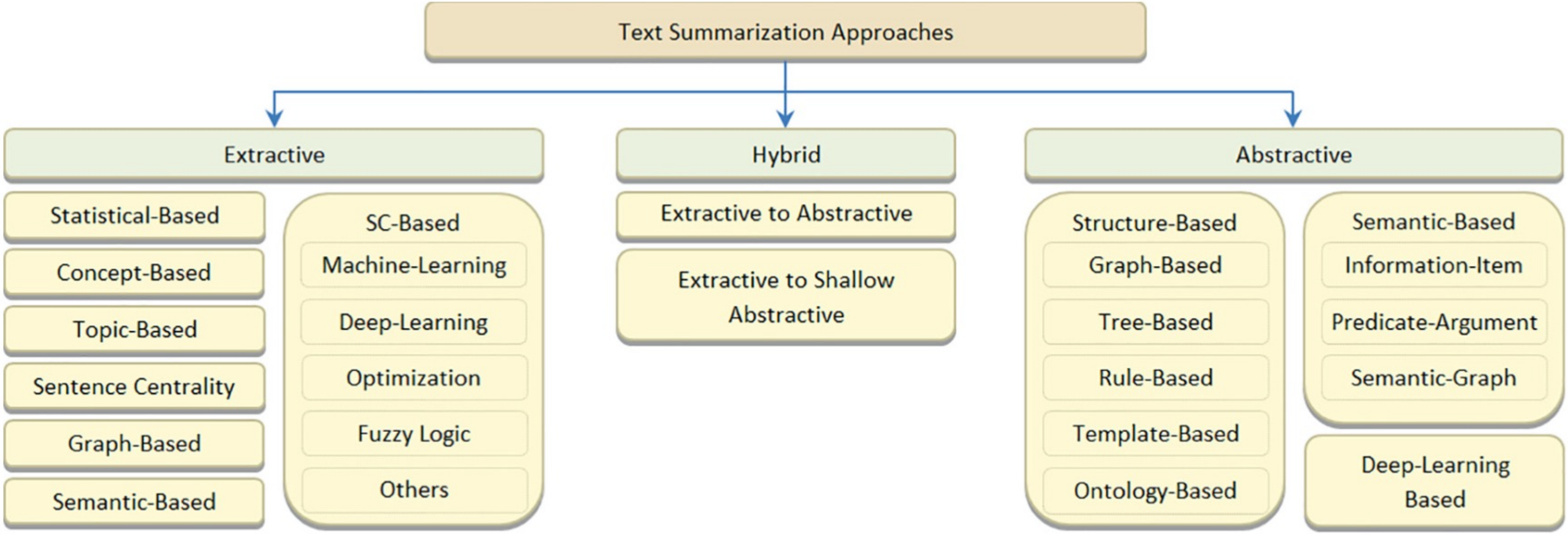
Source: Wafaa S. El-Kassas, Cherif R. Salama, Ahmed A. Rafea, and Hoda K. Mohamed (2021). "Automatic text summarization: A comprehensive survey." Expert Systems with Applications 165 (2021): 113679.

Classification of Automatic Text Summarization Systems



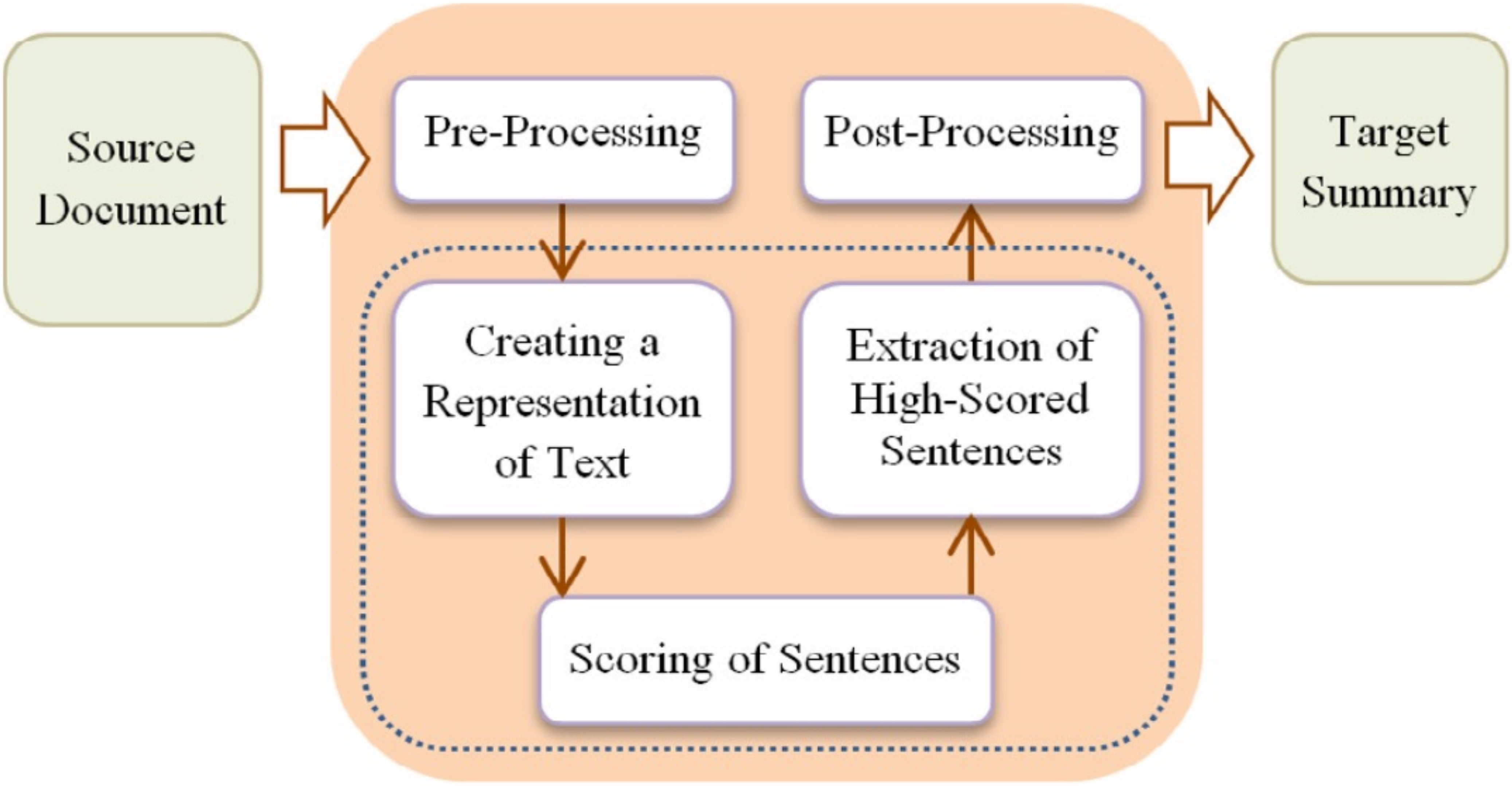
Source: Wafaa S. El-Kassas, Cherif R. Salama, Ahmed A. Rafea, and Hoda K. Mohamed (2021). "Automatic text summarization: A comprehensive survey." Expert Systems with Applications 165 (2021): 113679.

Automatic Text Summarization Approaches



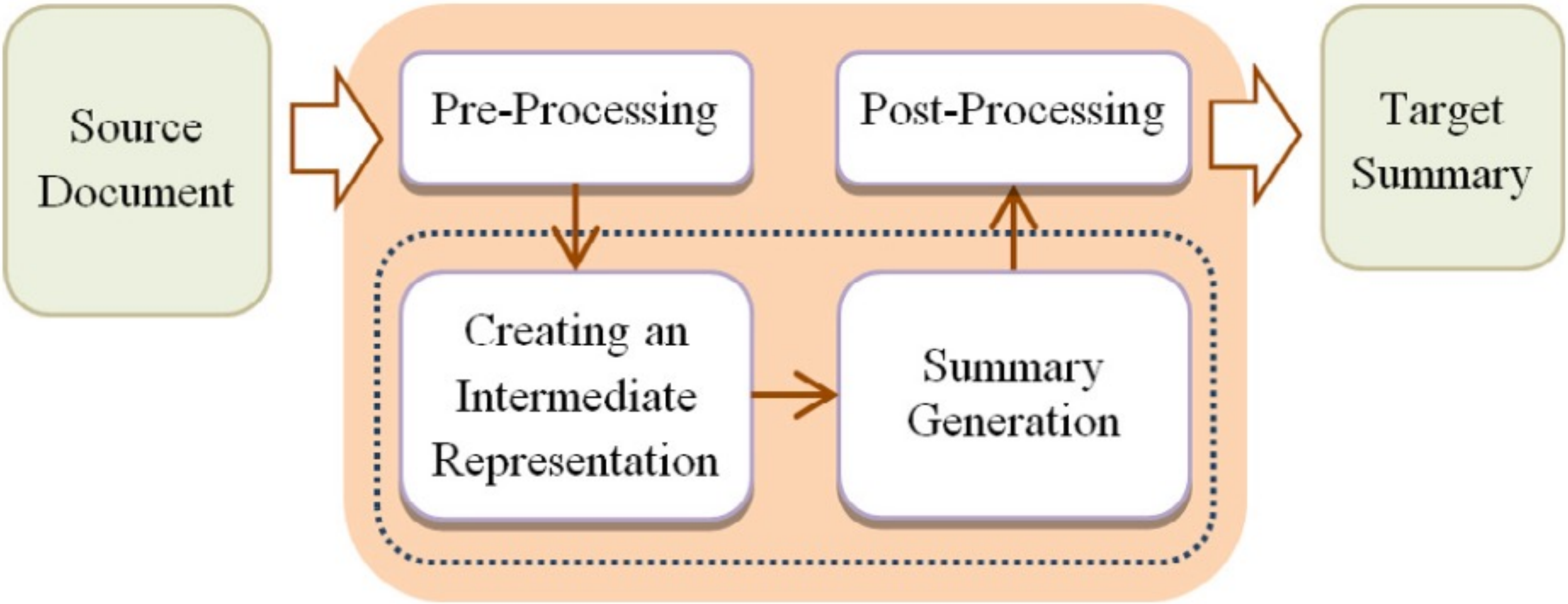
Source: Wafaa S. El-Kassas, Cherif R. Salama, Ahmed A. Rafea, and Hoda K. Mohamed (2021). "Automatic text summarization: A comprehensive survey." Expert Systems with Applications 165 (2021): 113679.

Extractive Text Summarization System



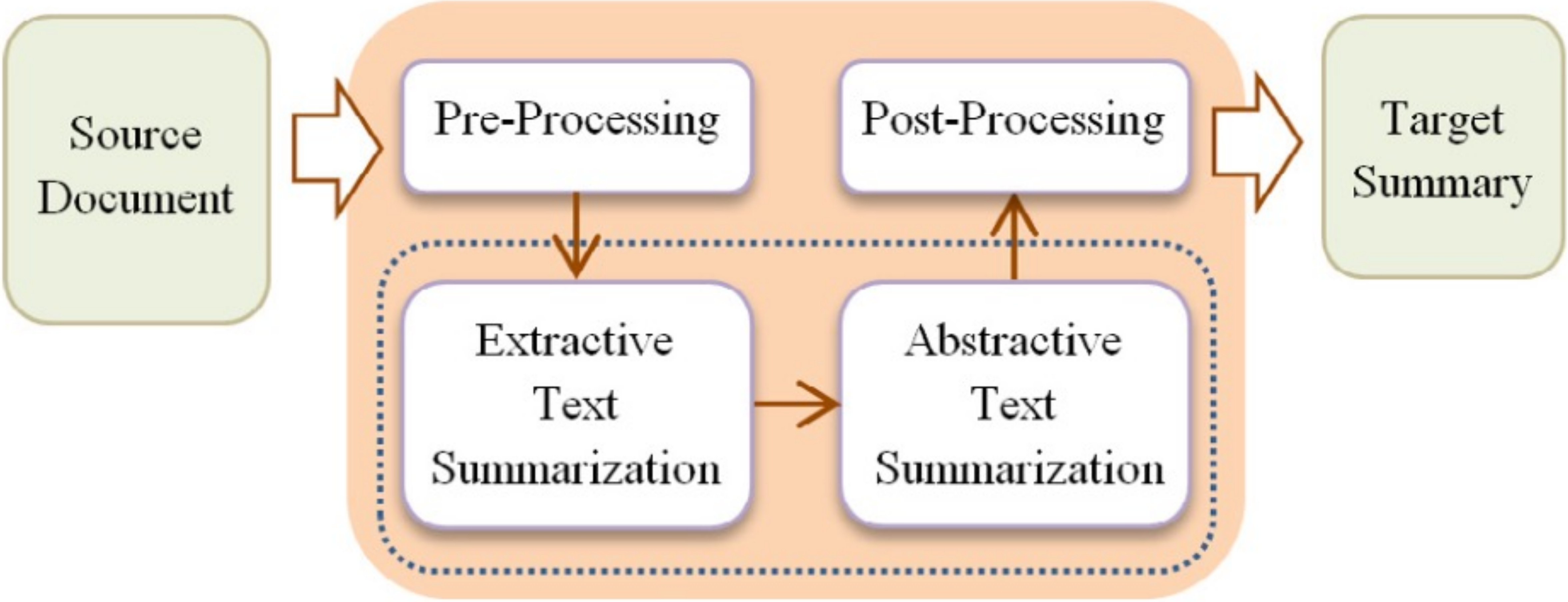
Source: Wafaa S. El-Kassas, Cherif R. Salama, Ahmed A. Rafea, and Hoda K. Mohamed (2021). "Automatic text summarization: A comprehensive survey." Expert Systems with Applications 165 (2021): 113679.

Abstractive Text Summarization System



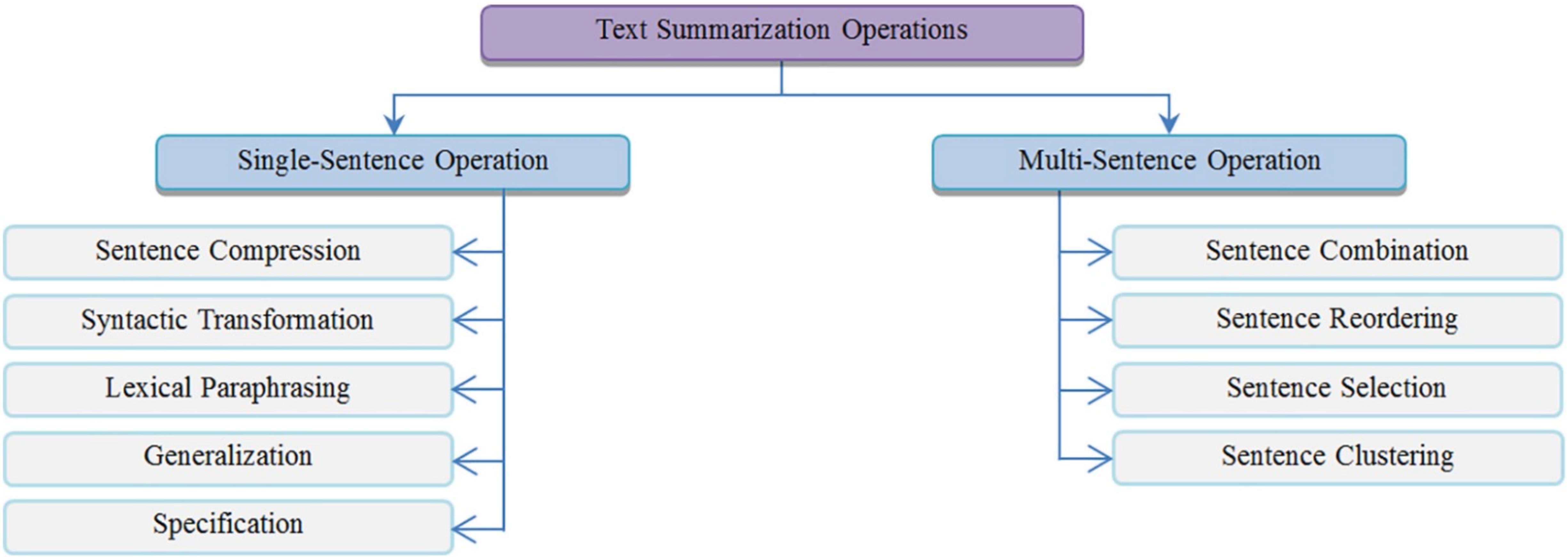
Source: Wafaa S. El-Kassas, Cherif R. Salama, Ahmed A. Rafea, and Hoda K. Mohamed (2021). "Automatic text summarization: A comprehensive survey." Expert Systems with Applications 165 (2021): 113679.

Hybrid Text Summarization System



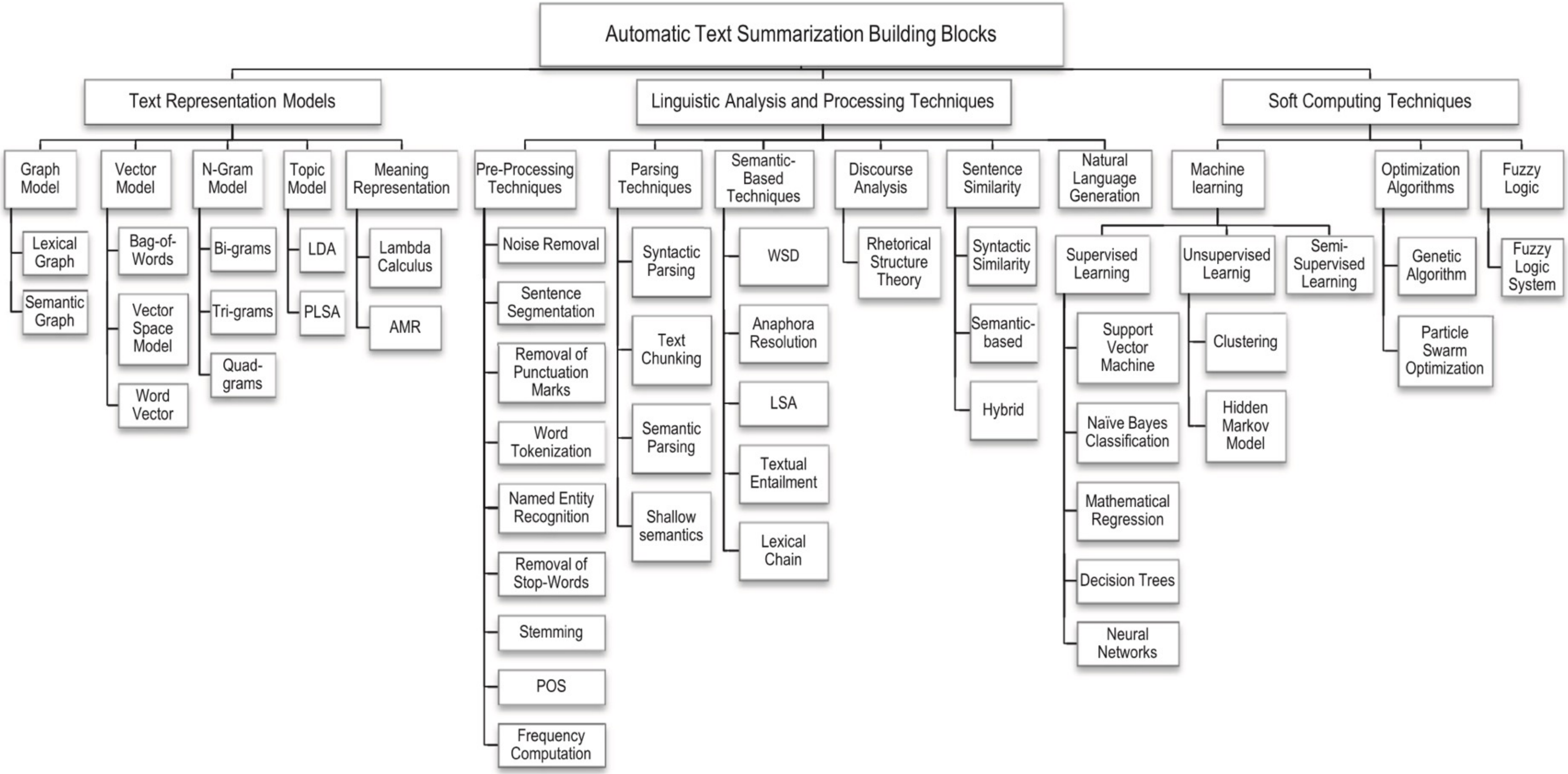
Source: Wafaa S. El-Kassas, Cherif R. Salama, Ahmed A. Rafea, and Hoda K. Mohamed (2021). "Automatic text summarization: A comprehensive survey." Expert Systems with Applications 165 (2021): 113679.

Single-sentence and Multi-sentence Text Summarization Operations



Source: Wafaa S. El-Kassas, Cherif R. Salama, Ahmed A. Rafea, and Hoda K. Mohamed (2021). "Automatic text summarization: A comprehensive survey." Expert Systems with Applications 165 (2021): 113679.

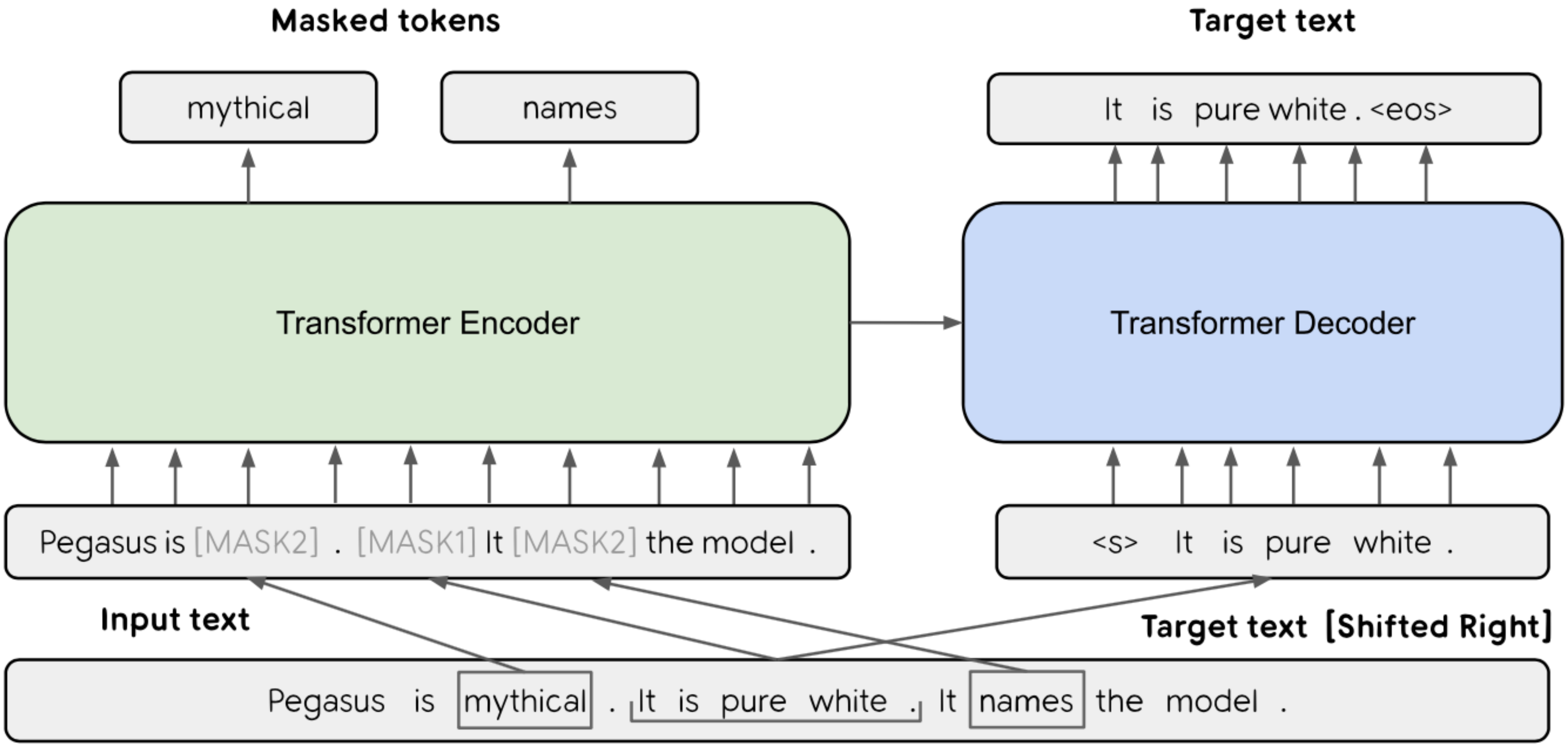
Automatic Text Summarization Building Blocks



Source: Wafaa S. El-Kassas, Cherif R. Salama, Ahmed A. Rafea, and Hoda K. Mohamed (2021). "Automatic text summarization: A comprehensive survey." Expert Systems with Applications 165 (2021): 113679.

PEGASUS:

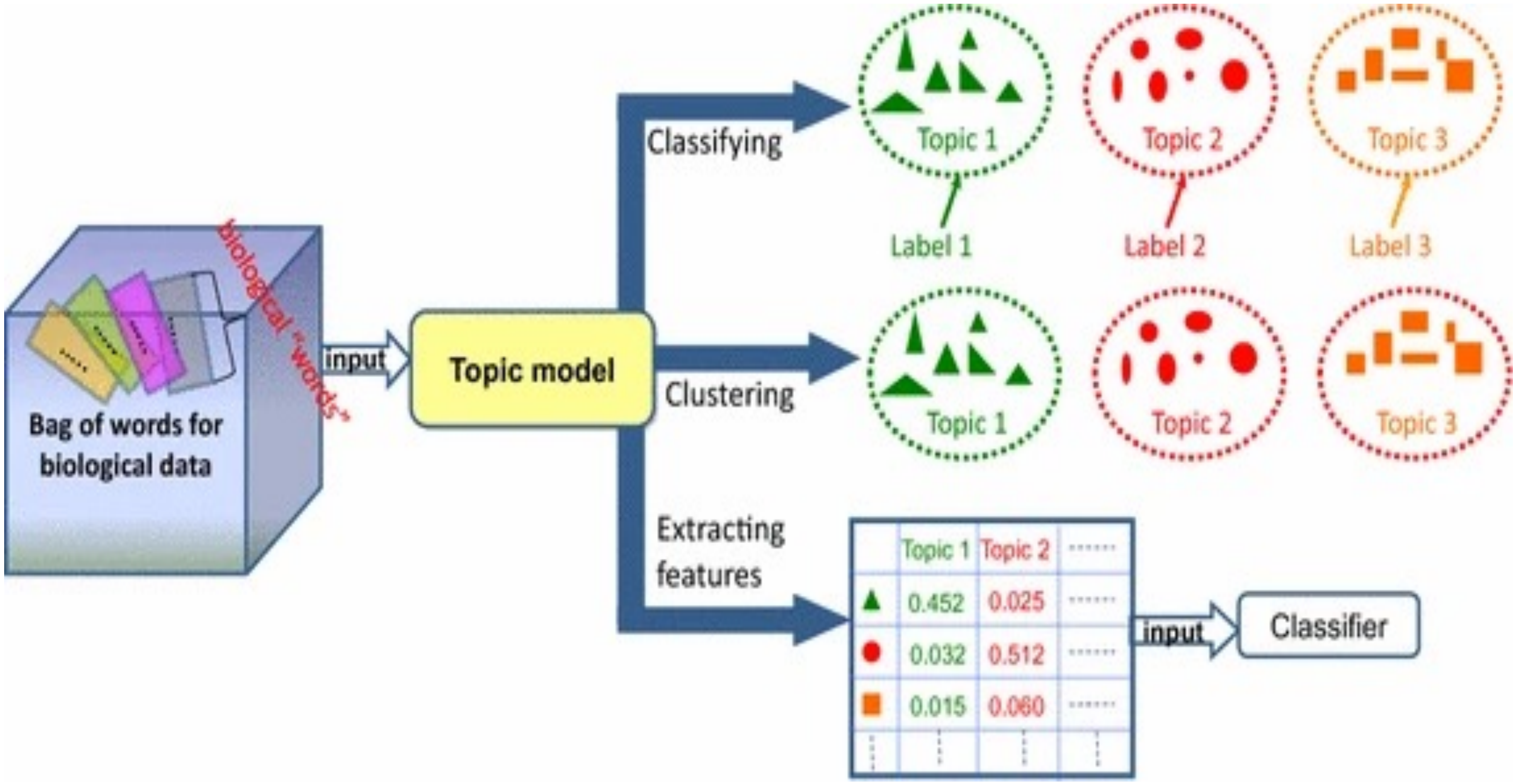
Pre-training with Extracted Gap-sentences for Abstractive Summarization



Source: Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu (2020). "Pegasus: Pre-training with extracted gap-sentences for abstractive summarization." In International Conference on Machine Learning, pp. 11328-11339. PMLR, 2020.

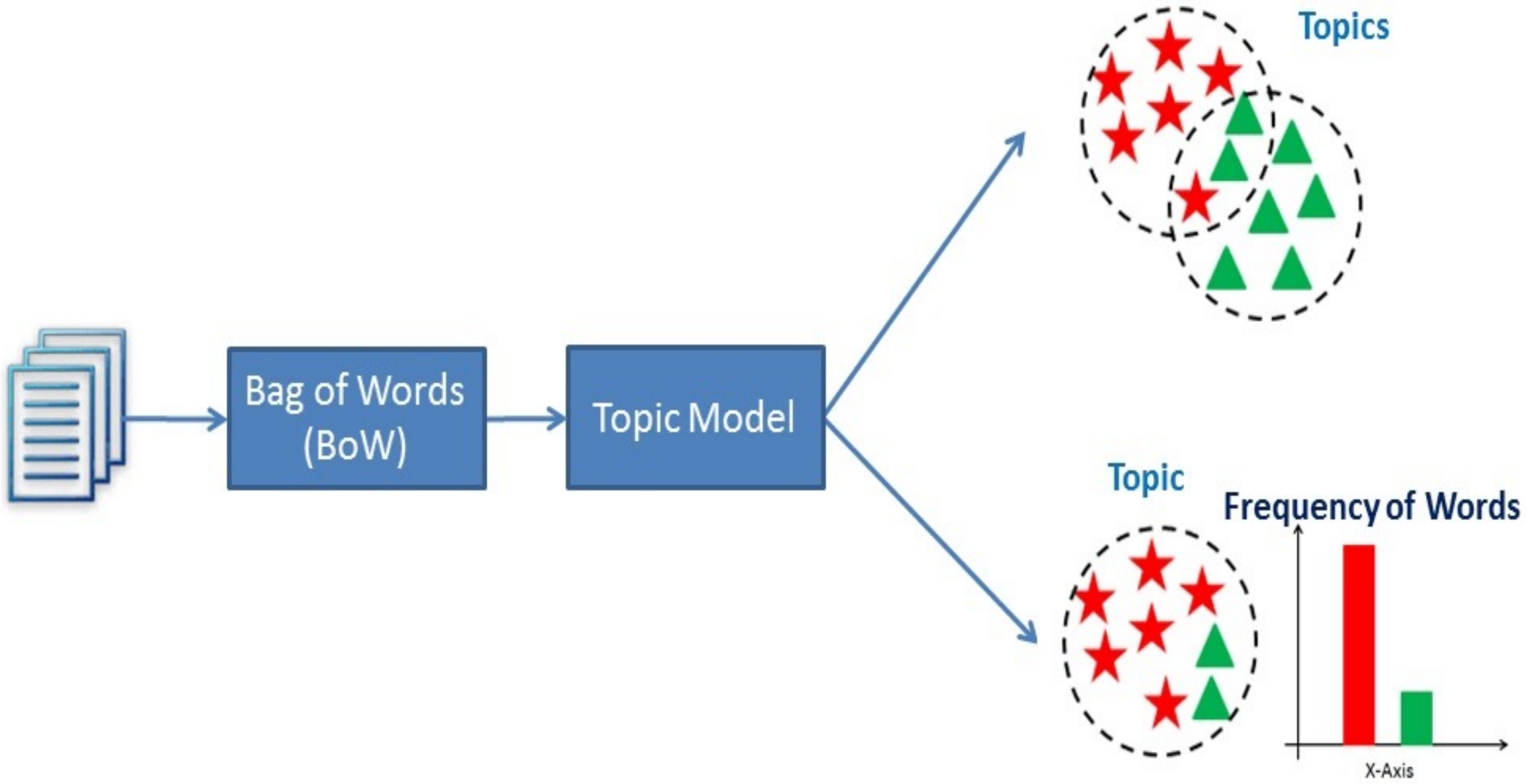
Topic Modeling

Topic Model in Bioinformatics



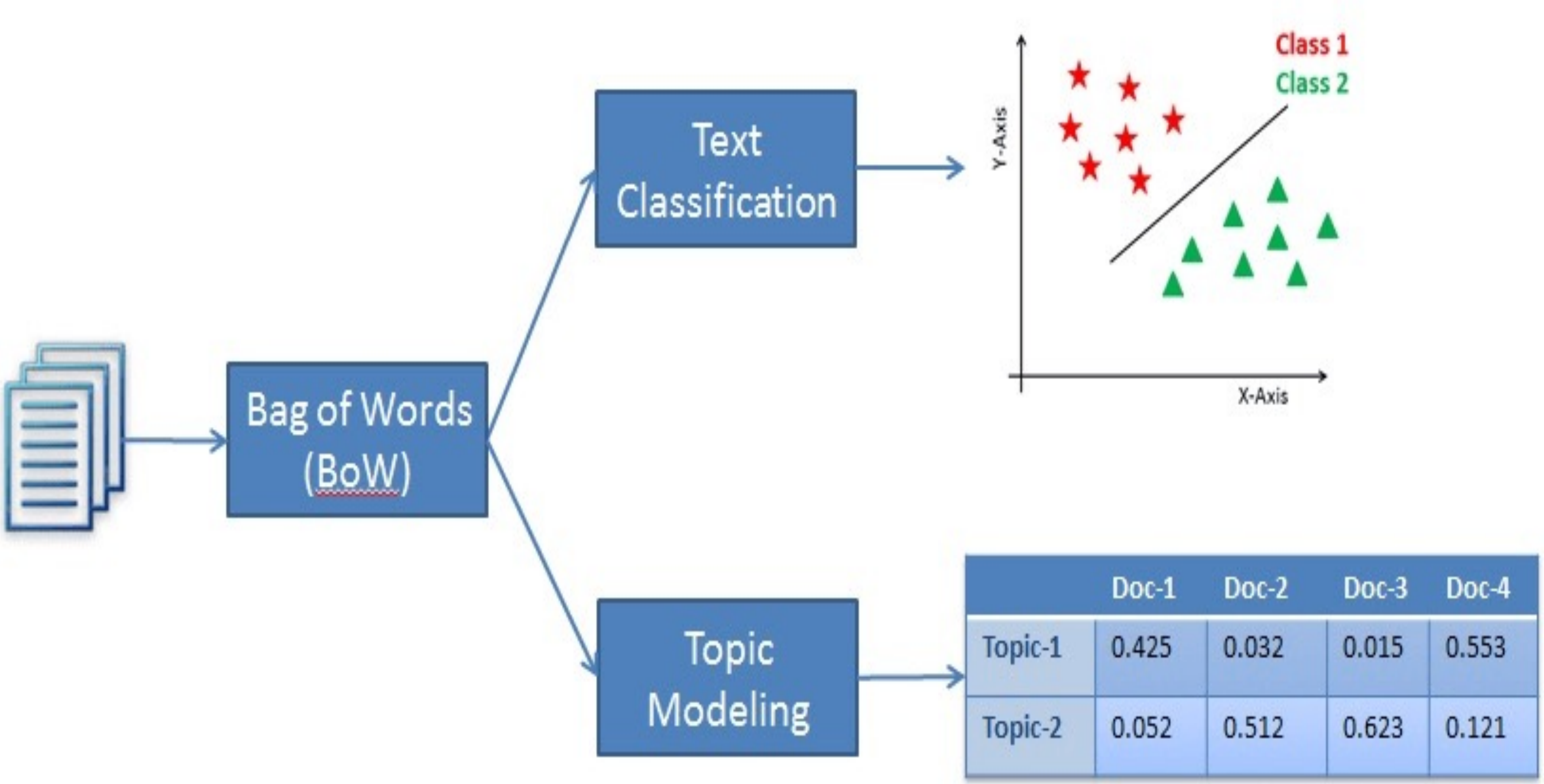
Source: Liu, Lin, Lin Tang, Wen Dong, Shaowen Yao, and Wei Zhou. "An overview of topic modeling and its current applications in bioinformatics." SpringerPlus 5, no. 1 (2016): 1608.

Topic Modeling



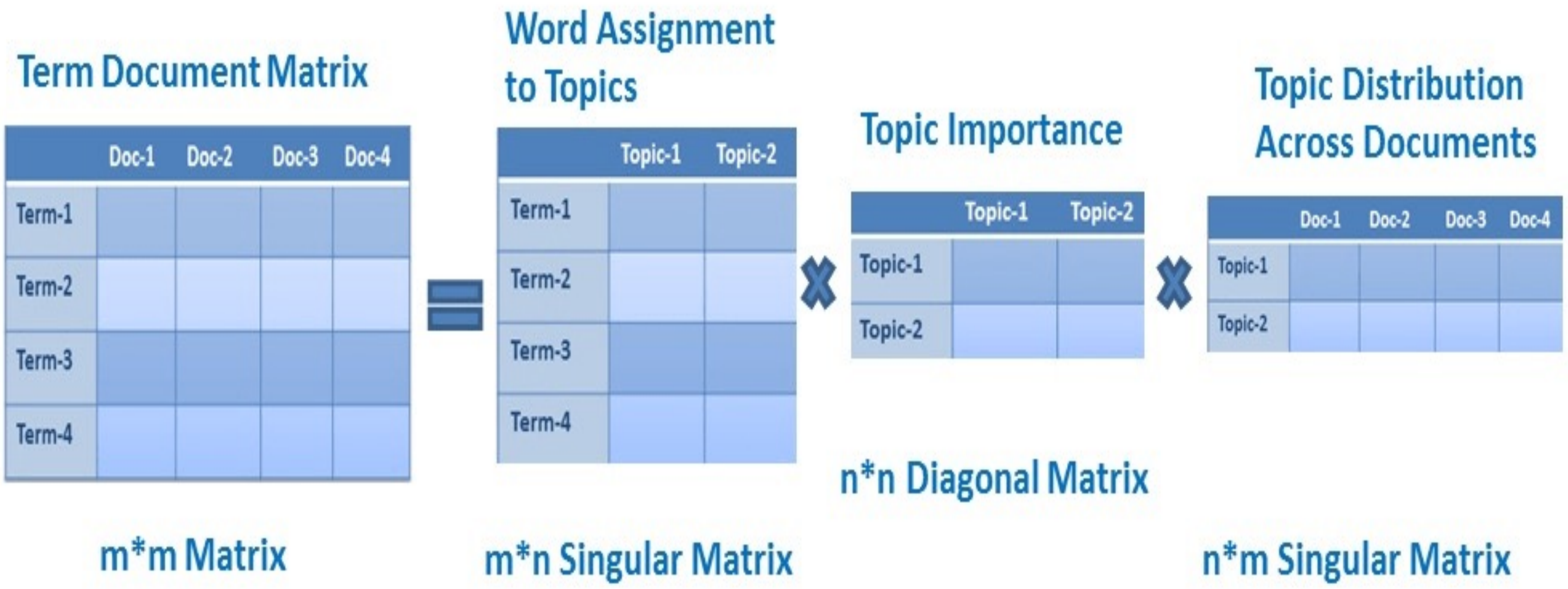
Source: Avinash Navlani (2018), Latent Semantic Analysis using Python,
https://www.datacamp.com/community/tutorials/discovering-hidden-topics-python

Topic Modeling (Unsupervised Learning)
vs.
Text Classification (Supervised Learning)



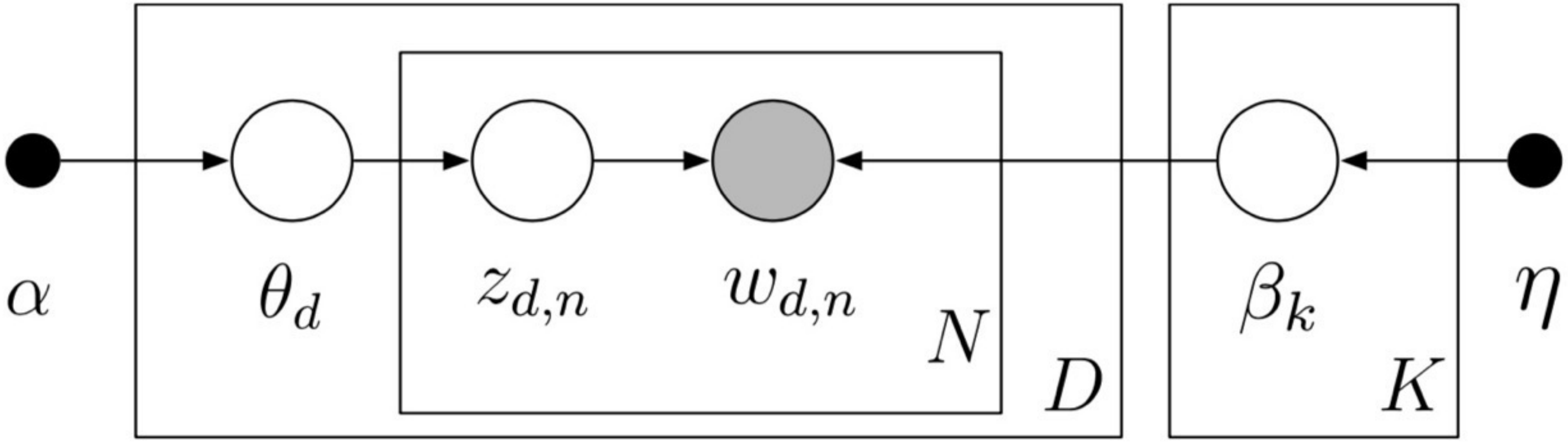
	Doc-1	Doc-2	Doc-3	Doc-4
Topic-1	0.425	0.032	0.015	0.553
Topic-2	0.052	0.512	0.623	0.121

Topic Modeling
Term Document Matrix to Topic Distribution



Source: Avinash Navlani (2018), Latent Semantic Analysis using Python, https://www.datacamp.com/community/tutorials/discovering-hidden-topics-python

Topic Modeling
Latent Dirichlet Allocation (LDA)



D **documents**
N **words**
K **topics**

Source: Hoffman, Matthew D., David M. Blei, Chong Wang, and John Paisley. "Stochastic variational inference." The Journal of Machine Learning Research 14, no. 1 (2013): 1303-1347.

Latent Dirichlet Allocation (Blei et al., 2003)

Journal of Machine Learning Research 3 (2003) 993-1022 Submitted 2/02; Published 1/03

Latent Dirichlet Allocation

David M. Blei BLEI@CS.BERKELEY.EDU
Computer Science Division
University of California
Berkeley, CA 94720, USA

Andrew Y. Ng ANG@CS.STANFORD.EDU
Computer Science Department
Stanford University
Stanford, CA 94305, USA

Michael I. Jordan JORDAN@CS.BERKELEY.EDU
Computer Science Division and Department of Statistics
University of California
Berkeley, CA 94720, USA

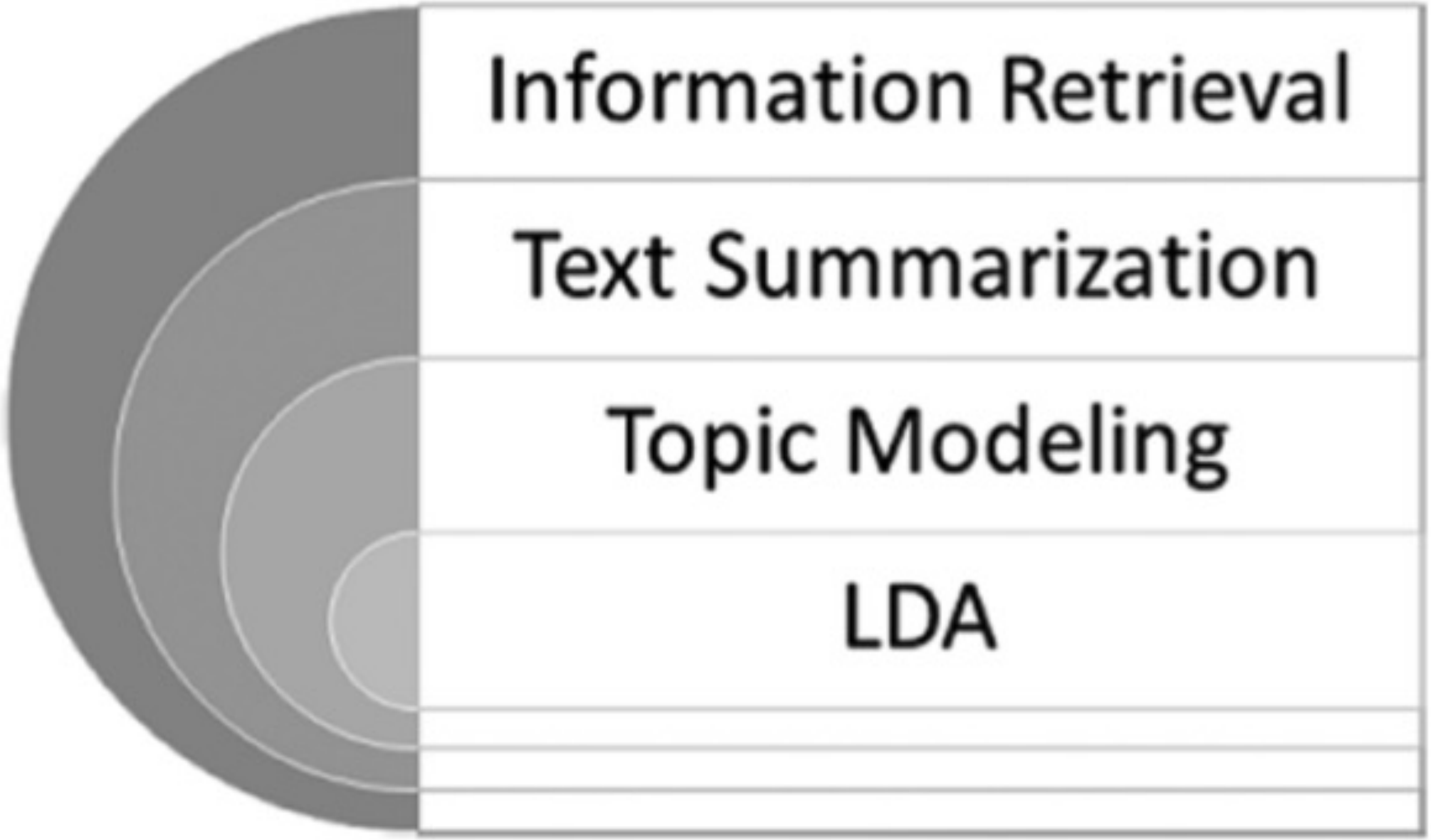
Editor: John Lafferty

Abstract

We describe *latent Dirichlet allocation* (LDA), a generative probabilistic model for collections of discrete data such as text corpora. LDA is a three-level hierarchical Bayesian model, in which each item of a collection is modeled as a finite mixture over an underlying set of topics. Each topic is, in turn, modeled as an infinite mixture over an underlying set of topic probabilities. In the context of text modeling, the topic probabilities provide an explicit representation of a document. We present efficient approximate inference techniques based on variational methods and an EM algorithm for empirical Bayes parameter estimation. We report results in document modeling, text classification, and collaborative filtering, comparing to a mixture of unigrams model and the probabilistic LSI model.

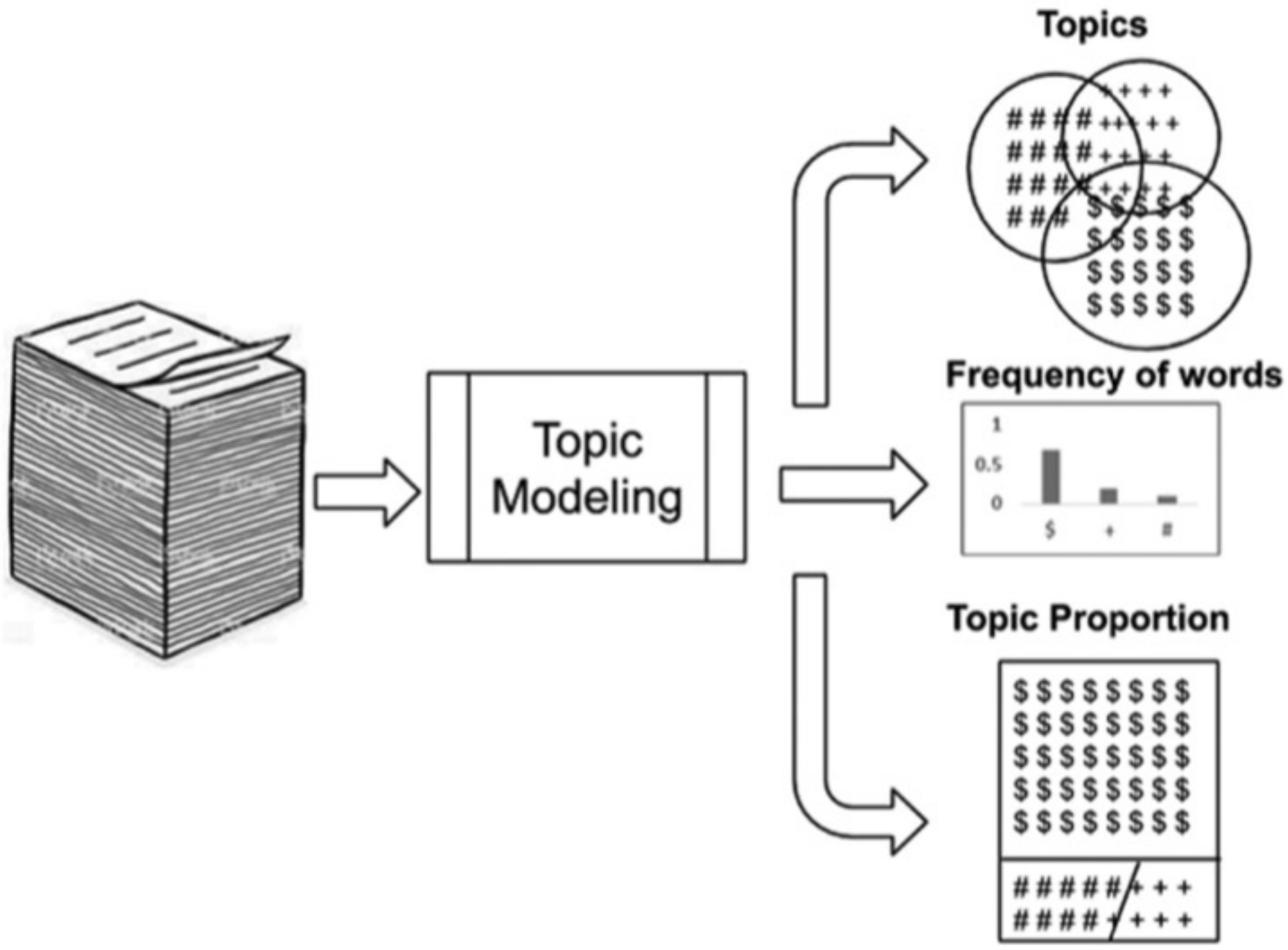
Source: Blei, David M., Andrew Y. Ng, and Michael I. Jordan. "Latent dirichlet allocation." Journal of machine Learning research 3, no. Jan (2003): 993-1022.

Topic Modeling Using Latent Dirichlet allocation (LDA)



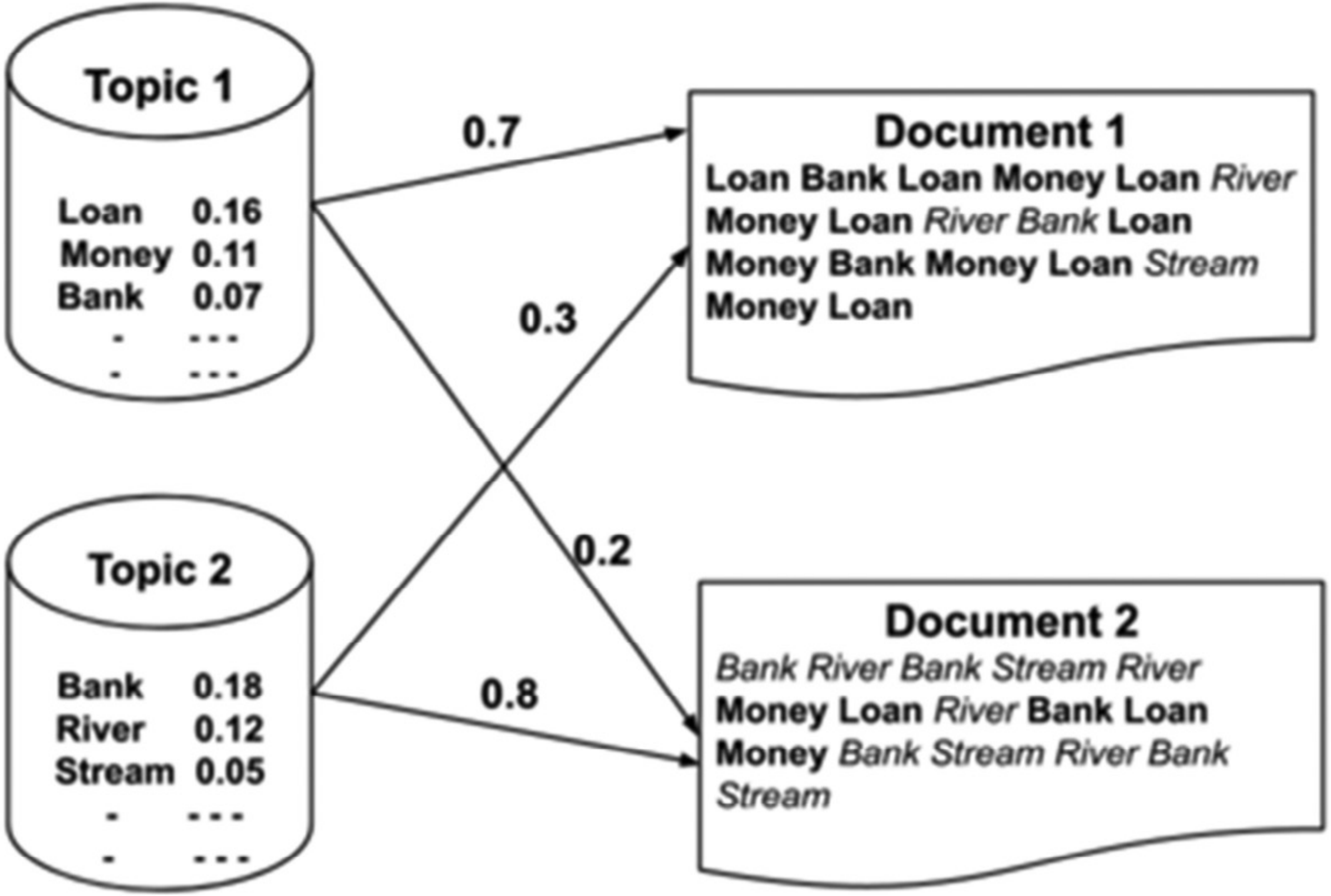
Source: Uttam Chauhan, and Apurva Shah (2021). "Topic modeling using latent Dirichlet allocation: A survey." ACM Computing Surveys (CSUR) 54, no. 7 (2021): 1-35.

Topic Modeling Technique



Source: Uttam Chauhan, and Apurva Shah (2021). "Topic modeling using latent Dirichlet allocation: A survey." ACM Computing Surveys (CSUR) 54, no. 7 (2021): 1-35.

The Generative Process of Latent Dirichlet Allocation (LDA)

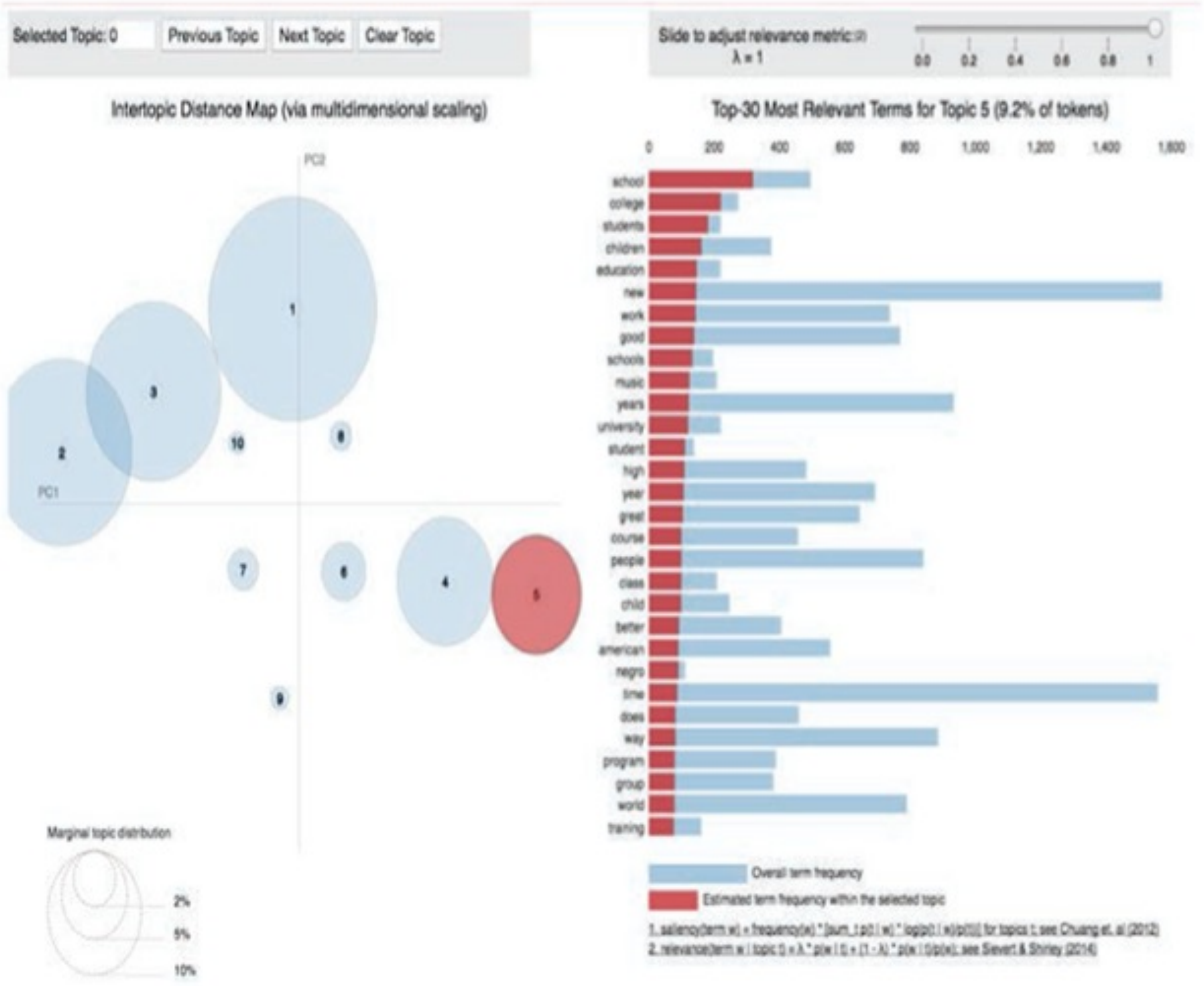


Source: Uttam Chauhan, and Apurva Shah (2021). "Topic modeling using latent Dirichlet allocation: A survey." ACM Computing Surveys (CSUR) 54, no. 7 (2021): 1-35.

Topic Visualization as Word Clouds

Source: Uttam Chauhan, and Apurva Shah (2021). "Topic modeling using latent Dirichlet allocation: A survey." ACM Computing Surveys (CSUR) 54, no. 7 (2021): 1-35.

LDAvis: Gensim Topic Model Visualization



Source: Uttam Chauhan, and Apurva Shah (2021). "Topic modeling using latent Dirichlet allocation: A survey." ACM Computing Surveys (CSUR) 54, no. 7 (2021): 1-35.

BERTopic

Neural topic modeling with a class-based TF-IDF procedure

Maarten Grootendorst (2022). "BERTopic: Neural topic modeling with a class-based TF-IDF procedure." arXiv preprint arXiv:2203.05794 (2022).

https://github.com/MaartenGr/BERTopic

Topic Modeling

- **Latent Dirichlet Allocation (LDA)**
  - **Versatile for large datasets.**
- **BERTopic**
  - **Advanced, contextual language understanding.**
- **Non-Negative Matrix Factorization (NMF)**
  - **Fast, effective clustering.**
- **Latent Semantic Analysis (LSA)**
  - **Latent Semantic Indexing (LSI), Efficient, initial data exploration.**
- **Hierarchical Dirichlet Process (HDP)**
  - **Adaptable, nonparametric approach.**

gensim

https://radimrehurek.com/gensim/

spaCy

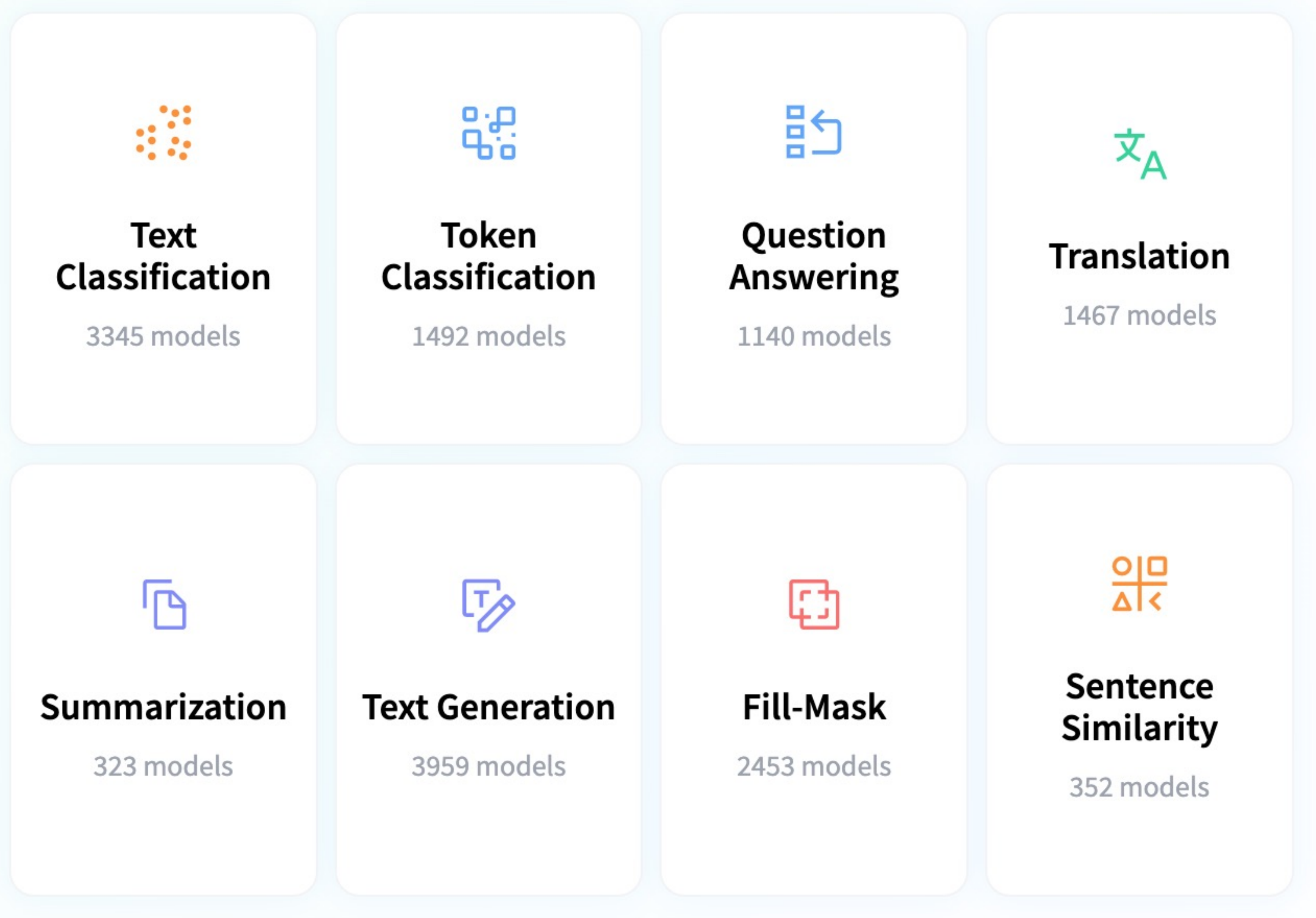
https://spacy.io/

NLP Benchmark Datasets

Task	Dataset	Link
Machine Translation	WMT 2014 EN-DE WMT 2014 EN-FR	http://www-lium.univ-lemans.fr/~schwenk/cslm_joint_paper/
Text Summarization	CNN/DM Newsroom DUC Gigaword	https://cs.nyu.edu/~kcho/DMQA/ https://summari.es/ https://www-nlpir.nist.gov/projects/duc/data.html https://catalog.ldc.upenn.edu/LDC2012T21
Reading Comprehension Question Answering Question Generation	ARC CliCR CNN/DM NewsQA RACE SQuAD Story Cloze Test NarativeQA Quasar SearchQA	http://data.allenai.org/arc/ http://aclweb.org/anthology/N18-1140 https://cs.nyu.edu/~kcho/DMQA/ https://datasets.maluuba.com/NewsQA http://www.qizhexie.com/data/RACE_leaderboard https://rajpurkar.github.io/SQuAD-explorer/ http://aclweb.org/anthology/W17-0906.pdf https://github.com/deepmind/narrativeqa https://github.com/bdhingra/quasar https://github.com/nyu-dl/SearchQA
Semantic Parsing	AMR parsing ATIS (SQL Parsing) WikiSQL (SQL Parsing)	https://amr.isi.edu/index.html https://github.com/jkkummerfeld/text2sql-data/tree/master/data https://github.com/salesforce/WikiSQL
Sentiment Analysis	IMDB Reviews SST Yelp Reviews Subjectivity Dataset	http://ai.stanford.edu/~amaas/data/sentiment/ https://nlp.stanford.edu/sentiment/index.html https://www.yelp.com/dataset/challenge http://www.cs.cornell.edu/people/pabo/movie-review-data/
Text Classification	AG News DBpedia TREC 20 NewsGroup	http://www.di.unipi.it/~gulli/AG_corpus_of_news_articles.html https://wiki.dbpedia.org/Datasets https://trec.nist.gov/data.html http://qwone.com/~jason/20Newsgroups/
Natural Language Inference	SNLI Corpus MultiNLI SciTail	https://nlp.stanford.edu/projects/snli/ https://www.nyu.edu/projects/bowman/multinli/ http://data.allenai.org/scitail/
Semantic Role Labeling	Proposition Bank OneNotes	http://propbank.github.io/ https://catalog.ldc.upenn.edu/LDC2013T19

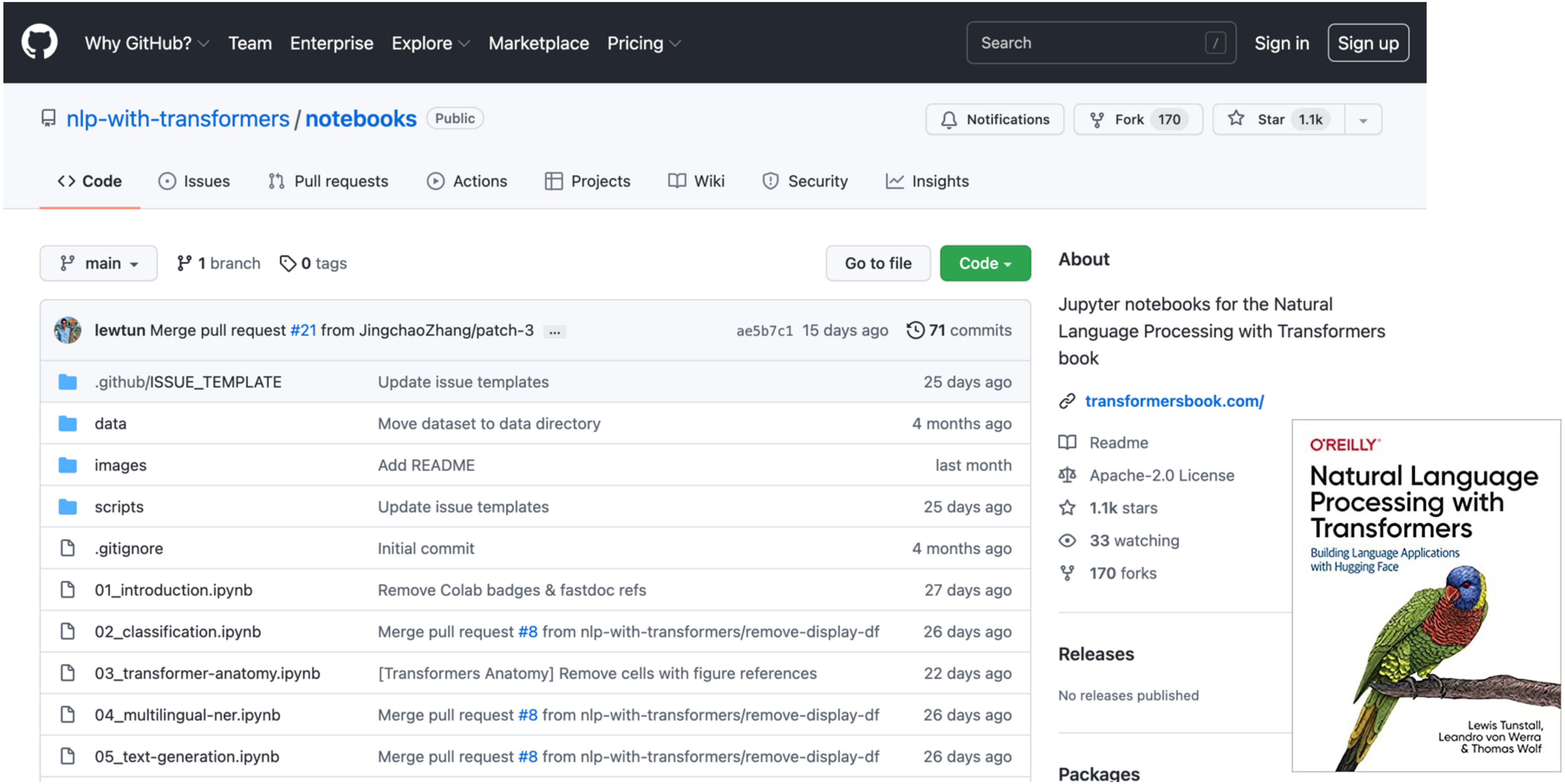
Source: Amirsina Torfi, Rouzbeh A. Shirvani, Yaser Keneshloo, Nader Tavvaf, and Edward A. Fox (2020). "Natural Language Processing Advancements By Deep Learning: A Survey." arXiv preprint arXiv:2003.01200.

Hugging Face Tasks
Natural Language Processing



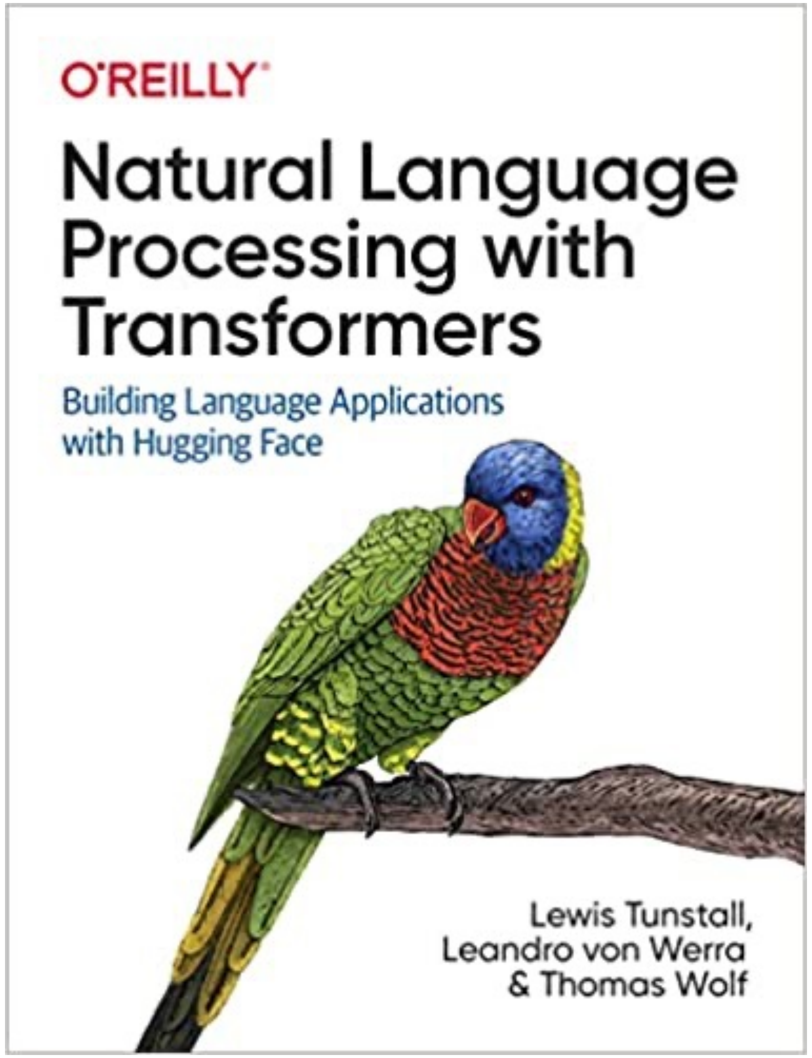
https://huggingface.co/tasks

NLP with Transformers Github



https://github.com/nlp-with-transformers/notebooks

NLP with Transformers Github Notebooks



Running on a cloud platform

To run these notebooks on a cloud platform, just click on one of the badges in the table below:

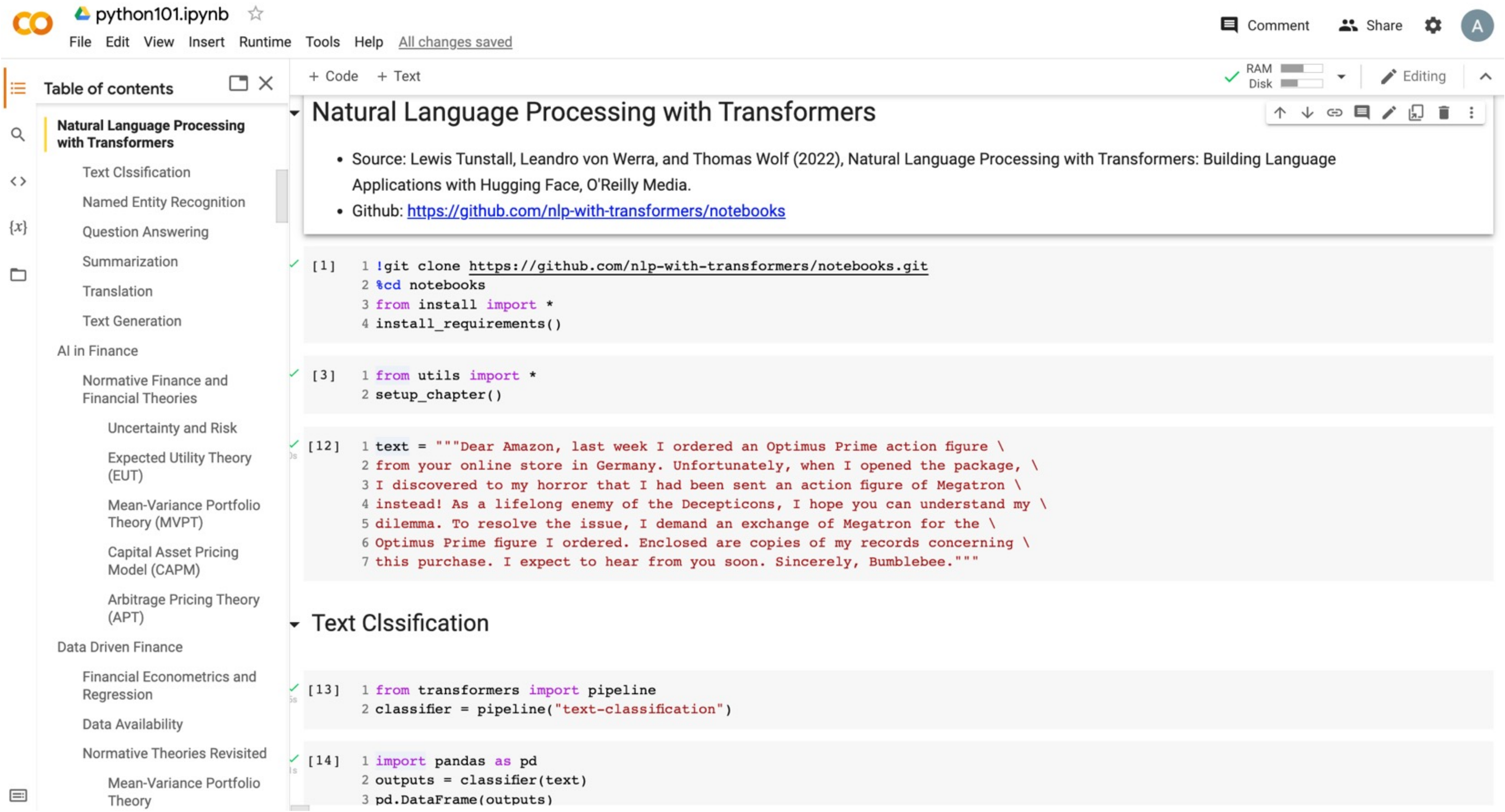
Chapter	Colab	Kaggle	Gradient	Studio Lab
Introduction	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Text Classification	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Transformer Anatomy	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Multilingual Named Entity Recognition	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Text Generation	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Summarization	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Question Answering	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Making Transformers Efficient in Production	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Dealing with Few to No Labels	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Training Transformers from Scratch	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Future Directions	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab

Nowadays, the GPUs on Colab tend to be K80s (which have limited memory), so we recommend using Kaggle, Gradient, or SageMaker Studio Lab. These platforms tend to provide more performant GPUs like P100s, all for free!

https://github.com/nlp-with-transformers/notebooks

Python in Google Colab (Python101)

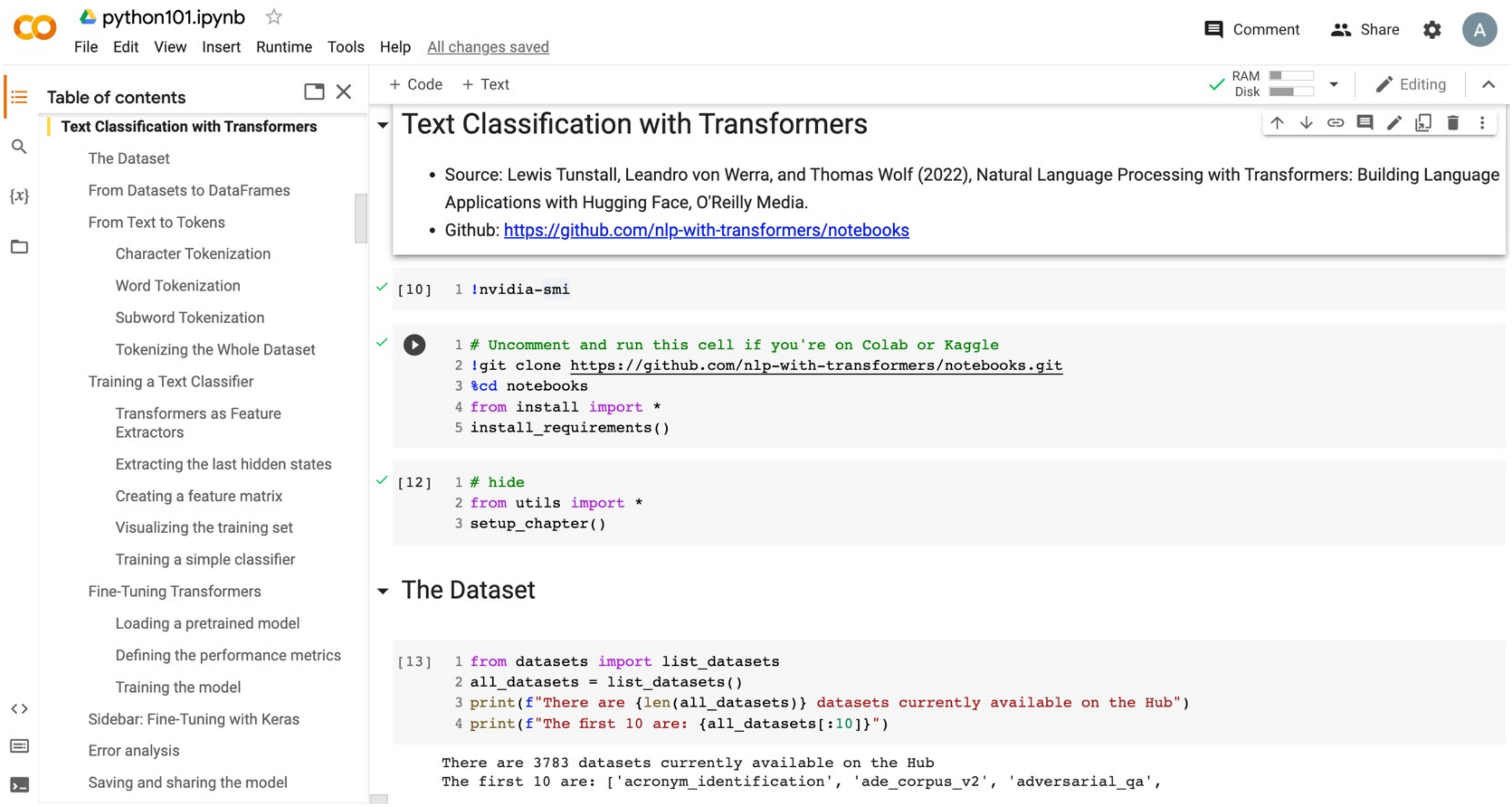
https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT



https://tinyurl.com/aintpupython101

Python in Google Colab (Python101)

https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT



https://tinyurl.com/aintpupython101

Python in Google Colab (Python101)

https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT

python101.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

Multilingual Named Entity Recognition (NER)

- Source: Lewis Tunstall, Leandro von Werra, and Thomas Wolf (2022), Natural Language Processing with Transformers: Building Language Applications with Hugging Face, O'Reilly Media.
- Github: https://github.com/nlp-with-transformers/notebooks

```
#NER: https://huggingface.co/tasks/token-classification
!pip install transformers
from transformers import pipeline
classifier = pipeline("ner")
classifier("Hello I'm Omar and I live in Zürich.")
```

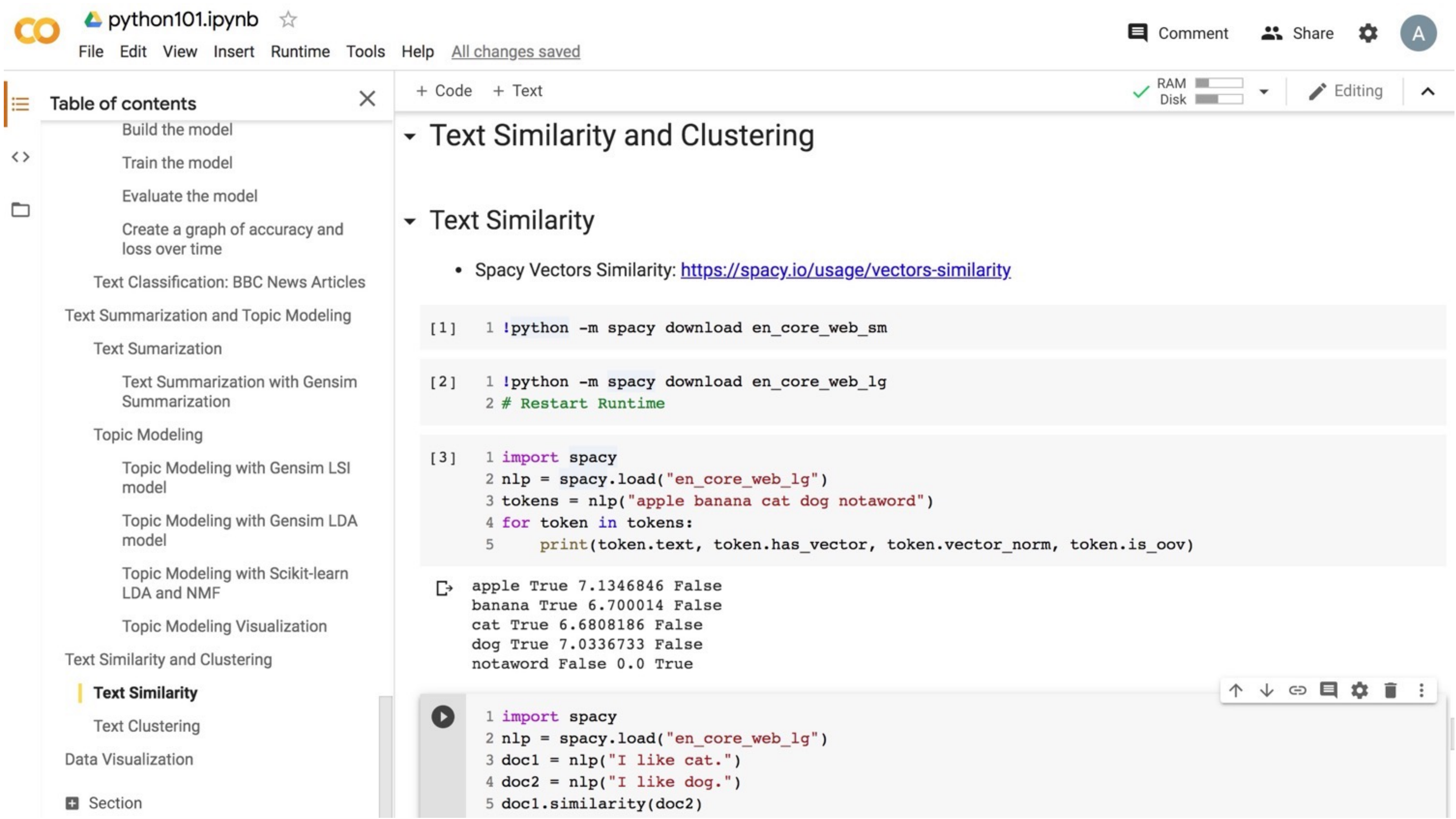
```
from transformers import pipeline
classifier = pipeline("ner")
classifier("Hello I'm Omar and I live in Zürich.")
```

```
No model was supplied, defaulted to dbmdz/bert-large-cased-finetuned-conll03-english (https://huggingface.co/dbmdz/bert-large-cased-finetuned-conll03-eng
[{'end': 14,
  'entity': 'I-PER',
  'index': 5,
  'score': 0.99770516,
  'start': 10,
  'word': 'Omar'},
 {'end': 35,
  'entity': 'I-LOC',
  'index': 10,
  'score': 0.9968976,
  'start': 29,
  'word': 'Zürich'}]
```

https://tinyurl.com/aintpupython101

Python in Google Colab (Python101)

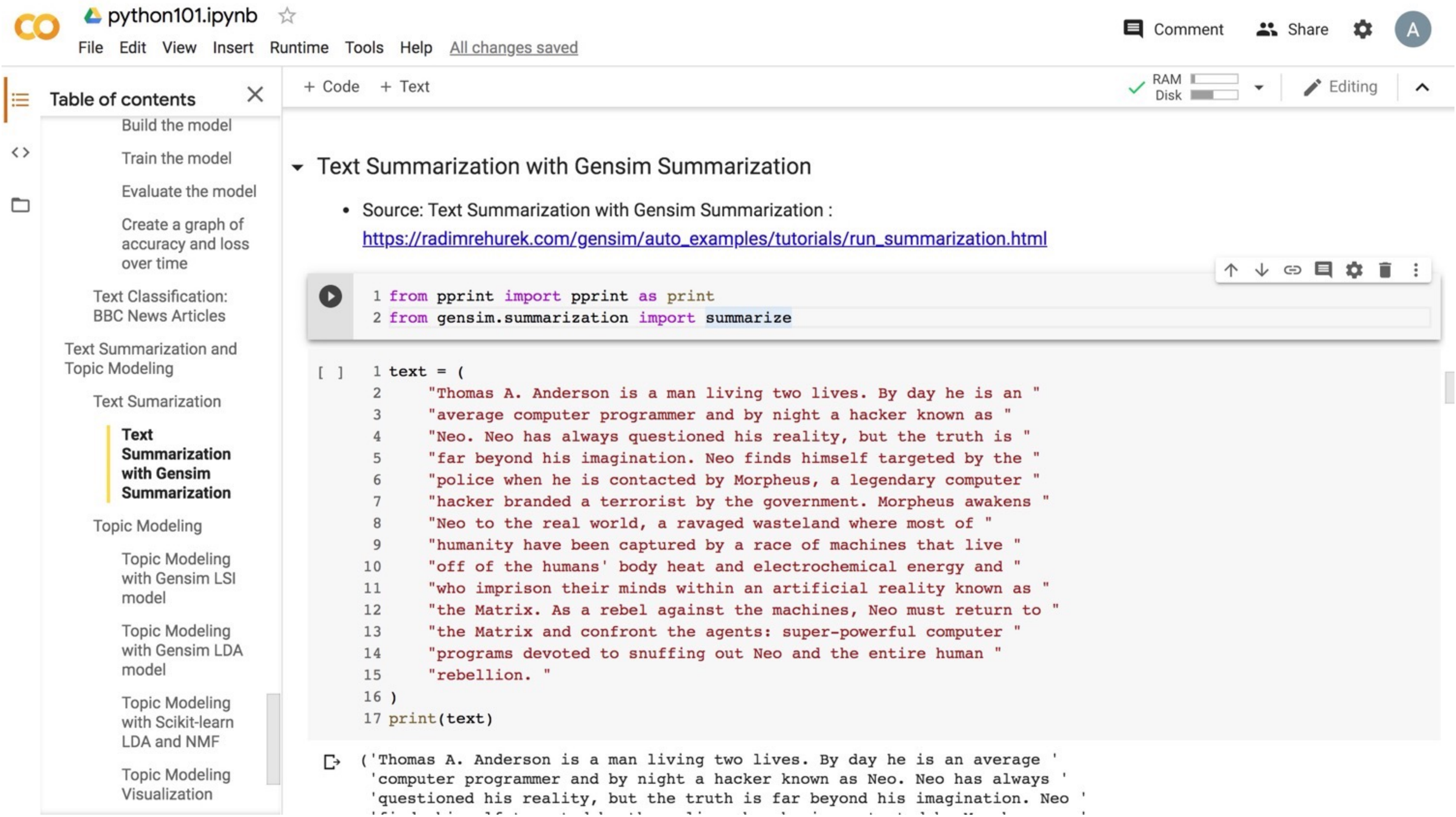
https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT



https://tinyurl.com/aintpupython101

Python in Google Colab (Python101)

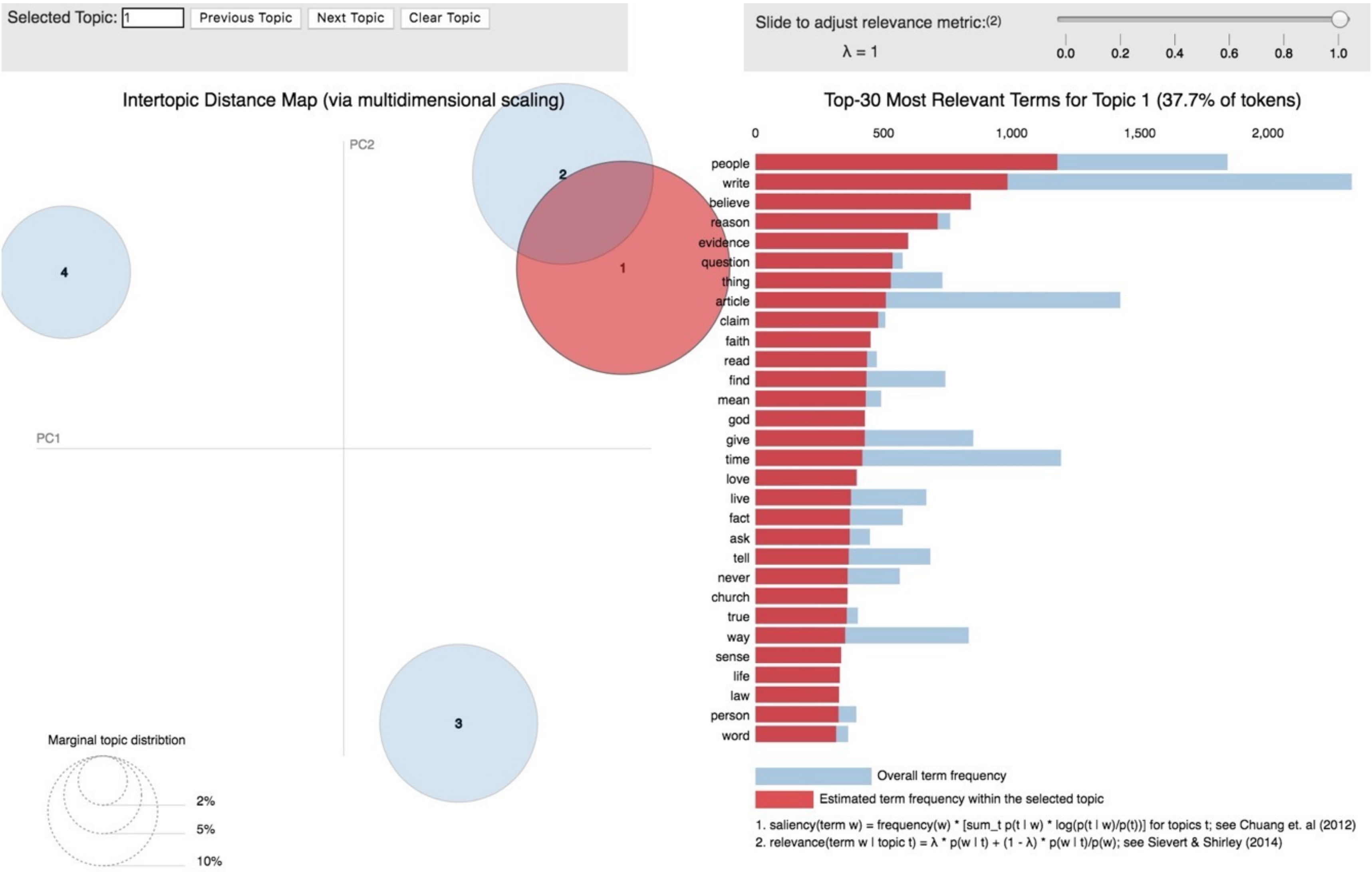
https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT



https://tinyurl.com/aintpupython101

Python in Google Colab (Python101)

https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT



https://tinyurl.com/aintpupython101

Python in Google Colab (Python101)

https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT

python101.ipynb

File Edit View Insert Runtime Tools Help Saving...

Comment Share

+ Code + Text

RAM Disk Editing

Text Summarization

- Source: Lewis Tunstall, Leandro von Werra, and Thomas Wolf (2022), Natural Language Processing with Transformers: Building Language Applications with Hugging Face, O'Reilly Media.
- Github: https://github.com/nlp-with-transformers/notebooks

```
#Source: https://huggingface.co/tasks/summarization
!pip install transformers
from transformers import pipeline
classifier = pipeline("summarization")
text = "Paris is the capital and most populous city of France, with an estimated population of 2,175,601 residents as of 2018, in an area of more thar
classifier(text, max_length=30)
```

```
No model was supplied, defaulted to sshleifer/distilbart-cnn-12-6 (https://huggingface.co/sshleifer/distilbart-cnn-12-6)
Your min_length=56 must be inferior than your max_length=30.
[{'summary_text': ' Paris is the capital and most populous city of France, with an estimated population of 2,175,601 residents . The City of Paris'}]
```

```
#!pip install transformers
text = """Dear Amazon, last week I ordered an Optimus Prime action figure \
from your online store in Germany. Unfortunately, when I opened the package, \
I discovered to my horror that I had been sent an action figure of Megatron \
instead! As a lifelong enemy of the Decepticons, I hope you can understand my \
dilemma. To resolve the issue, I demand an exchange of Megatron for the \
Optimus Prime figure I ordered. Enclosed are copies of my records concerning \
this purchase. I expect to hear from you soon. Sincerely, Bumblebee."""
from transformers import pipeline
summarizer = pipeline("summarization")
outputs = summarizer(text, max_length=45, clean_up_tokenization_spaces=True)
print(outputs[0]['summary_text'])
```

https://tinyurl.com/aintpupython101

Text Summarization

```
text = """Dear Amazon, last week I ordered an Optimus Prime action figure \
from your online store in Germany. Unfortunately, when I opened the package, \
I discovered to my horror that I had been sent an action figure of Megatron \
instead! As a lifelong enemy of the Decepticons, I hope you can understand my \
dilemma. To resolve the issue, I demand an exchange of Megatron for the \
Optimus Prime figure I ordered. Enclosed are copies of my records concerning \
this purchase. I expect to hear from you soon. Sincerely, Bumblebee."""
```

```
from transformers import pipeline
summarizer = pipeline("summarization")
outputs = summarizer(text, max_length=45, clean_up_tokenization_spaces=True)
print(outputs[0]['summary_text'])
```

```
Bumblebee ordered an Optimus Prime action figure from your
online store in Germany. Unfortunately, when I opened the
package, I discovered to my horror that I had been sent an
action figure of Megatron instead.
```

https://tinyurl.com/aintpupython101

Summary

- **Text Similarity**
  - **Analyzing and quantifying the likeness between text documents.**
- **Text Clustering**
  - **Grouping similar text documents using various algorithms.**
- **Text Summarization**
  - **Condensing text data into a shorter, coherent form.**
- **Topic Models**
  - **Identifying underlying themes or topics within text collections.**

Source: Dipanjan Sarkar (2019), Text Analytics with Python: A Practitioner's Guide to Natural Language Processing, Second Edition. APress.

References

- Lewis Tunstall, Leandro von Werra, and Thomas Wolf (2022), Natural Language Processing with Transformers: Building Language Applications with Hugging Face, O'Reilly Media.
- Denis Rothman (2021), Transformers for Natural Language Processing: Build innovative deep neural network architectures for NLP with Python, PyTorch, TensorFlow, BERT, RoBERTa, and more, Packt Publishing.
- Savaş Yıldırım and Meysam Asgari-Chenaghlu (2021), Mastering Transformers: Build state-of-the-art models from scratch with advanced natural language processing techniques, Packt Publishing.
- Sowmya Vajjala, Bodhisattwa Majumder, Anuj Gupta (2020), Practical Natural Language Processing: A Comprehensive Guide to Building Real-World NLP Systems, O'Reilly Media.
- Wafaa S. El-Kassas, Cherif R. Salama, Ahmed A. Rafea, and Hoda K. Mohamed (2021). "Automatic text summarization: A comprehensive survey." Expert Systems with Applications 165 (2021): 113679.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu (2019). "Exploring the limits of transfer learning with a unified text-to-text transformer." arXiv preprint arXiv:1910.10683 (2019).
- Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu (2020). "Pegasus: Pre-training with extracted gap-sentences for abstractive summarization." In International Conference on Machine Learning, pp. 11328-11339. PMLR, 2020.
- Uttam Chauhan, and Apurva Shah (2021). "Topic modeling using latent Dirichlet allocation: A survey." ACM Computing Surveys (CSUR) 54, no. 7 (2021): 1-35.
- Maarten Grootendorst (2022). "BERTopic: Neural topic modeling with a class-based TF-IDF procedure." arXiv preprint arXiv:2203.05794 (2022).
- Benjamin Bengfort, Rebecca Bilbro, and Tony Ojeda (2018), Applied Text Analysis with Python: Enabling Language-Aware Data Products with Machine Learning, O'Reilly.
- Selva Prabhakaran (2020), Topic modeling visualization – How to present the results of LDA models?, https://www.machinelearningplus.com/nlp/topic-modeling-visualization-how-to-present-results-lda-models/
- Rajesh Arumugam (2018), Hands-On Natural Language Processing with Python: A practical guide to applying deep learning architectures to your NLP applications, Packt.
- Jake VanderPlas (2016), Python Data Science Handbook: Essential Tools for Working with Data, O'Reilly Media.
- Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." arXiv preprint arXiv:1810.04805.
- The Super Duper NLP Repo, https://notebooks.quantumstat.com/
- Jay Alammar (2018), The Illustrated Transformer, http://jalammar.github.io/illustrated-transformer/
- Jay Alammar (2019), A Visual Guide to Using BERT for the First Time, http://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/
- NLP with Transformer, https://github.com/nlp-with-transformers/notebooks
- Min-Yuh Day (2023), Python 101, https://tinyurl.com/aintpupython101