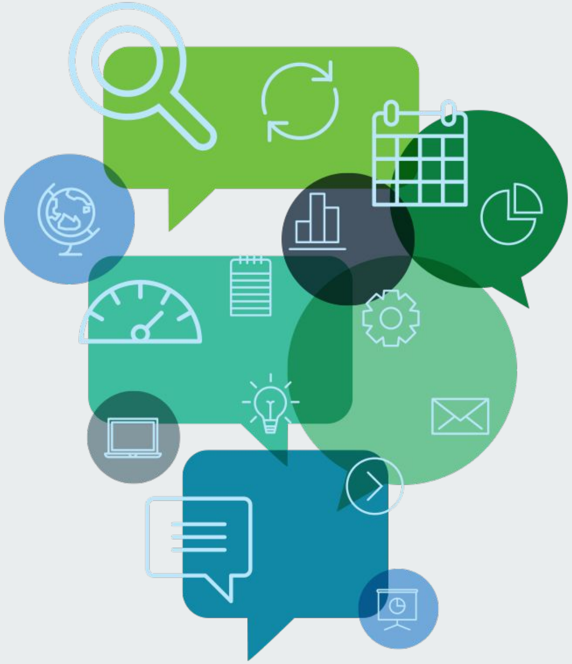
Deep Dive 2



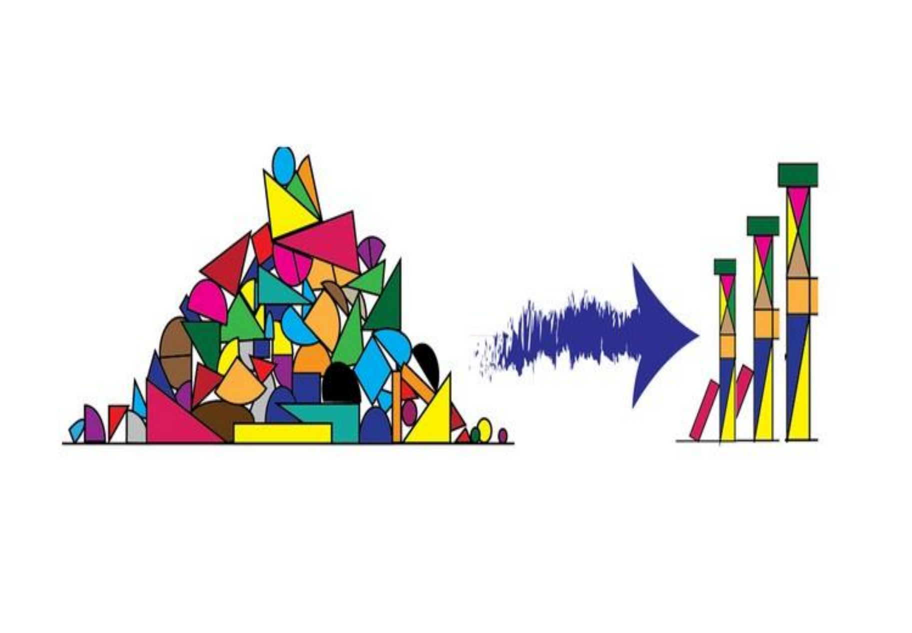
Feature Engineering - TF-IDF, Density Functions, Implement with NLTK, Spacy

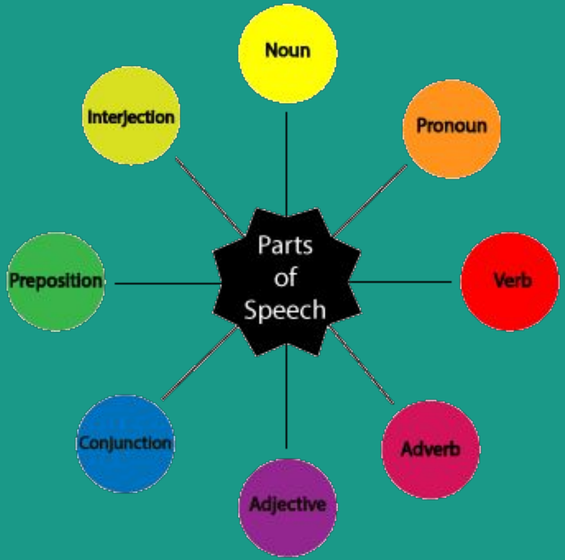
WOMEN WHO CODE DATA SCIENCE

Agenda

1. Recap Week 2
2. TF-IDF
3. Bag-of-Words
4. Document Similarity - Use Cases
5. Implement with Google Colab
6. Questions?
7. Wrap-up and Next Steps

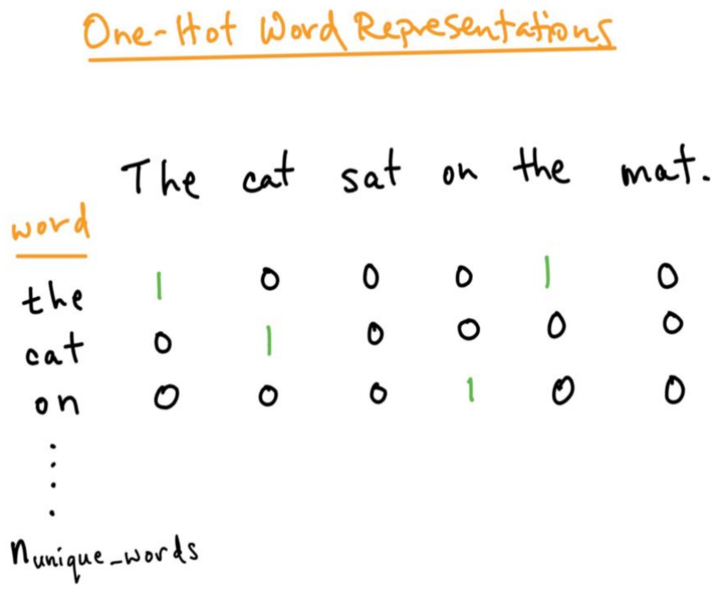
Recap - Week 2




Noun
Interjection
Pronoun
Parts of Speech
Preposition
Verb
Conjunction
Adverb
Adjective

Feature Engineering Overview

- ML algorithms cannot work on the raw text directly
- Algorithms can only process numeric representation of an actual text
- FE techniques used to convert text into a matrix (or vector)
- Popular methods of feature extraction are : Bag-of-Words, TF-IDF



Source: https://www.kdnuggets.com/2019/10/introduction-natural-language-processing.html

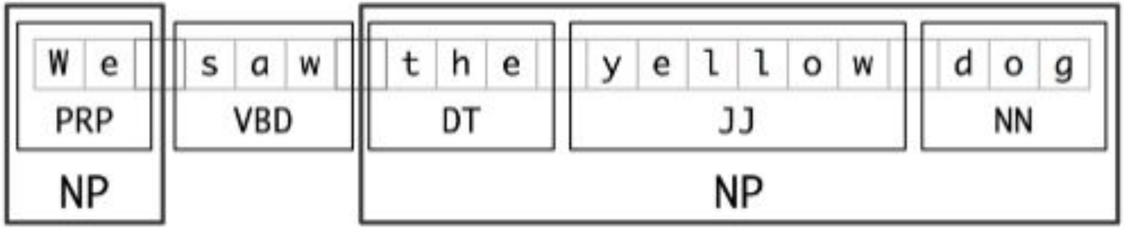
Techniques to Understand Text

Shallow Parsing or Chunking

Named Entity Recognition (NER)

N-Grams

Shallow Parsing or Chunking



WOMEN WHO CODE DATA SCIENCE

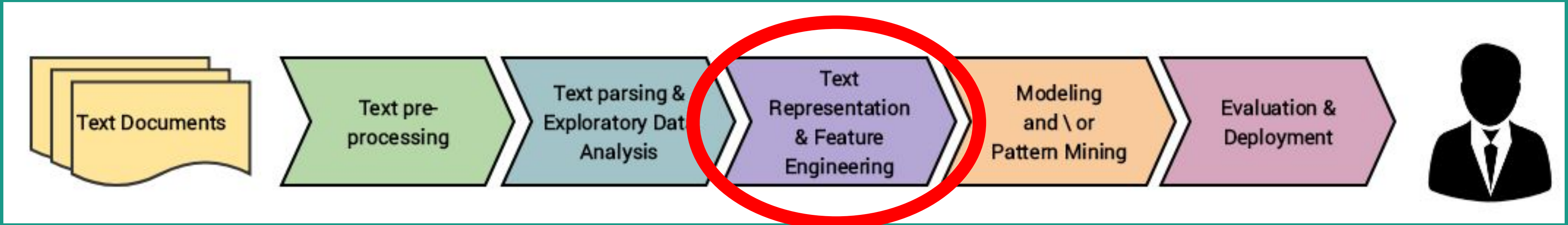
Named Entity Recognition

F.B.I. Agent Peter Strzok PERSON , Who Criticized Trump PERSON in Texts, Is Fired GPE - The New York Times ORG SectionsSEARCHSkip to contentSkip to site indexPoliticsSubscribeLog InSubscribeLog InToday's PaperAdvertisementSupported ORG byF.B.I. Agent Peter Strzok PERSON , Who Criticized Trump PERSON in Texts, Is FiredImagePeter Strzok, a top F.B.I. GPE counterintelligence agent who was taken off the special counsel investigation after his disparaging texts about President Trump PERSON were uncovered, was fired. CreditT.J. Kirkpatrick PERSON for The New York TimesBy Adam Goldman ORG and Michael S. SchmidtAug PERSON . 13 CARDINAL , 2018WASHINGTON CARDINAL — Peter Strzok PERSON , the F.B.I. GPE senior counterintelligence agent who disparaged President Trump PERSON in inflammatory text messages and helped oversee the Hillary Clinton PERSON email and Russia GPE investigations, has been fired for violating bureau policies, Mr. Strzok PERSON 's lawyer said Monday DATE .Mr. Trump and his allies seized on the texts — exchanged during the 2016 DATE campaign with a former F.B.I. GPE lawyer, Lisa Page — in PERSON assailing the Russia GPE investigation as an illegitimate “witch hunt.” Mr. Strzok PERSON , who rose over 20 years DATE at the F.B.I. GPE to become one of its most experienced counterintelligence agents, was a key figure in the early months DATE of the inquiry.Along with writing the texts, Mr. Strzok PERSON was accused of sending a highly sensitive search warrant to his personal email account.The F.B.I. GPE had been under immense political pressure by Mr. Trump PERSON to dismiss Mr. Strzok PERSON , who was removed last summer DATE from the staff of the special counsel, Robert S. Mueller III PERSON . The president has repeatedly denounced Mr. Strzok PERSON in posts on Twitter EVENT , and on Monday DATE expressed satisfaction that he had been sacked.Mr. Trump's ORG victory traces back to June DATE , when Mr. Strzok PERSON 's conduct was laid out in a wide-ranging inspector general's report on how the F.B.I. GPE handled the investigation of Hillary Clinton's PERSON emails in the run-up to the 2016 DATE election. The report was critical of Mr. Strzok PERSON 's conduct in sending the

WOMEN WHO CODE DATA SCIENCE

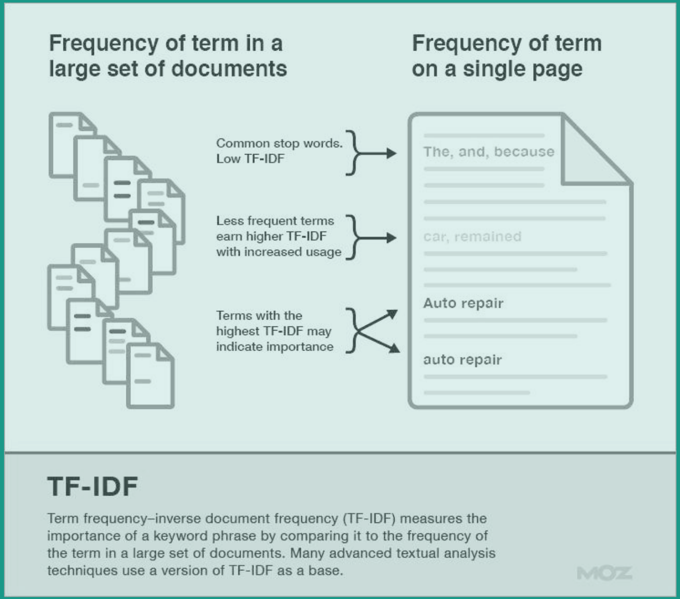
N-Grams

NLP Workflow



Source: https://towardsdatascience.com/a-practitioners-guide-to-natural-language-processing-part-i-processing-understanding-text-9f4abfd13e72

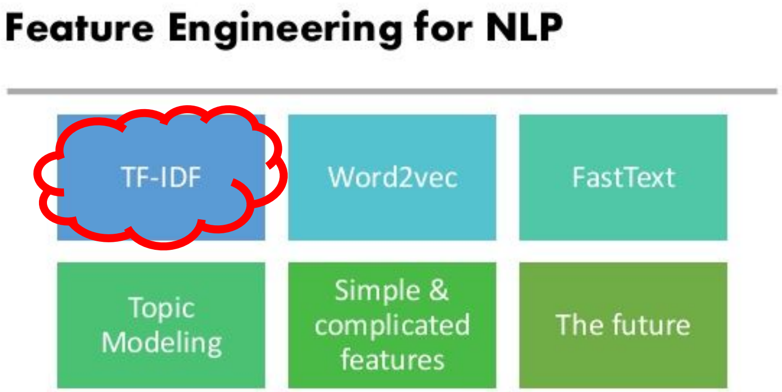
TF-IDF



Frequency of term in a large set of documents
Frequency of term on a single page
Common stop words. Low TF-IDF
Less frequent terms earn higher TF-IDF with increased usage
Terms with the highest TF-IDF may indicate importance
The, and, because
car, remained
Auto repair
auto repair
TF-IDF
Term frequency–inverse document frequency (TF-IDF) measures the importance of a keyword phrase by comparing it to the frequency of the term in a large set of documents. Many advanced textual analysis techniques use a version of TF-IDF as a base.
MOZ

TF-IDF

- Term frequency-inverse document frequency
- Combination of two metrics: **term frequency** and **inverse document frequency**
- Commonly used in information retrieval



Source: https://www.slideshare.net/BillLiu31/feature-engineering-for-nlp

Goal of TF-IDF

- Text documents → vector models, based on word occurrence (without considering the exact ordering)
- Given: dataset of N text documents, TF and IDF defined as…
  - TF: count of a term “t” in a document “d”
  - IDF: logarithm of ratio of total documents (d) in the entire corpus and number of documents (d) containing the term “t”
- Together: relative importance of a term in a corpus

TF-IDF: Formula (1)

$$w_{x,y} = tf_{x,y} \times \log\left(\frac{N}{df_x}\right)$$

TF-IDF

Term x within document y

$tf_{x,y}$ = frequency of x in y

df_x = number of documents containing x

N = total number of documents

Source: http://filotechnologia.blogspot.com/2014/01/a-simple-java-class-for-tfidf-scoring.html

TF-IDF: Formula (2)

$$TF_{ij} = \frac{f_{ij}}{n_j}$$ Equation (1)

Where f_{ij} is the frequency of term i in document j. n_j is the total number of words in document j.

$$IDF_i = 1 + log(\frac{N}{c_i})$$ Equation (2)

Where N is the total number of documents in the corpus. c_i is the number of documents that contain word i.

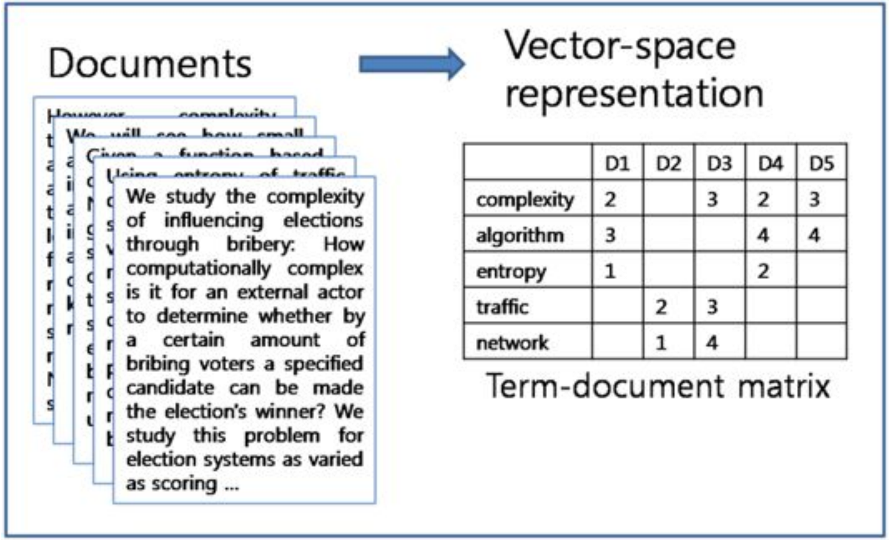
$$w_{ij} = TF_{ij} \times IDF_i$$ Equation (3)

Where w_{ij} is the TF-IDF score of term i in document j.

Source:

TF-IDF: Properties

- Compute for every word in every document
- Result: matrix with shape = # words * # documents
- A single value for 1 word → matrix of values when considering all documents



Source: https://www.quora.com/What-is-a-tf-idf-vector

TF-IDF: Simple Example

3 Documents

- Document 1: “Machine learning teaches machine how to learn”
- Document 2: “Machine translation is my favorite subject”
- Document 3: “Term frequency and inverse document frequency is important”

Source:

TF-IDF: Step 1

- Compute f_{ij}: frequency of term i in document j, Equation 1
- Pure counting: each term i in documents 1-3
  - f{i1}

	machine	learning	teaches	how	to	learn
f_{i1}	2	1	1	1	1	1

  - f{i2}

	machine	translation	is	my	favorite	subject
f_{i2}	1	1	1	1	1	1

  - f{i3}

	term	frequency	and	inverse	document	is	important
f_{i3}	1	2	1	1	1	1	1

Source:

TF-IDF: Step 2

- Compute normalized term frequency TF_{ij}, Equation 1
- Each word in single document should divide total # words in that document

	machine	learning	teaches	how	to	learn
TF_{i1}	0.29	0.14	0.14	0.14	0.14	0.14

	machine	translation	is	my	favorite	subject
TF_{i2}	0.17	0.17	0.17	0.17	0.17	0.17

	term	frequency	and	inverse	document	is	important
TF_{i3}	0.125	0.25	0.125	0.125	0.125	0.125	0.125

Source:

TF-IDF: Step 3

- Compute IDF of each term i, Equation 2

Terms	IDF
Machine	1.4
learning	2.1
teaches	2.1
how	2.1
to	2.1
learn	2.1
translation	2.1
is	1.4
my	2.1
favorite	2.1
subject	2.1
Term	2.1
frequency	2.1
and	2.1
inverse	2.1
document	2.1
important	2.1

Source:

WOMEN WHO CODE DATA SCIENCE

TF-IDF: Step 4

- Compute TF-IDF for each word i in document j, Equation 3

	machine	learning	teaches	how	to	learn
TFIDF_{i1}	0.40	0.30	0.30	0.30	0.30	0.30

	machine	translation	is	my	favorite	subject
TFIDF_{i2}	0.23	0.35	0.23	0.35	0.35	0.35

	term	frequency	and	inverse	document	is	important
TFIDF_{i3}	0.26	0.52	0.26	0.26	0.26	0.18	0.26

Source:

TF-IDF: Give a Query

- Given a query: “machine learning” → compute TF-IDF

	TF	IDF	TF*IDF
machine	0.5	1.4	0.7
learning	0.5	2.1	1.05

	Document1	Document2	Document3
machine	0.4	0.23	0.0
learning	0.3	0	0.0

Source:

TF-IDF ✔️


PHEW!!

WOMEN WHO CODE DATA SCIENCE

Bag-of-Words

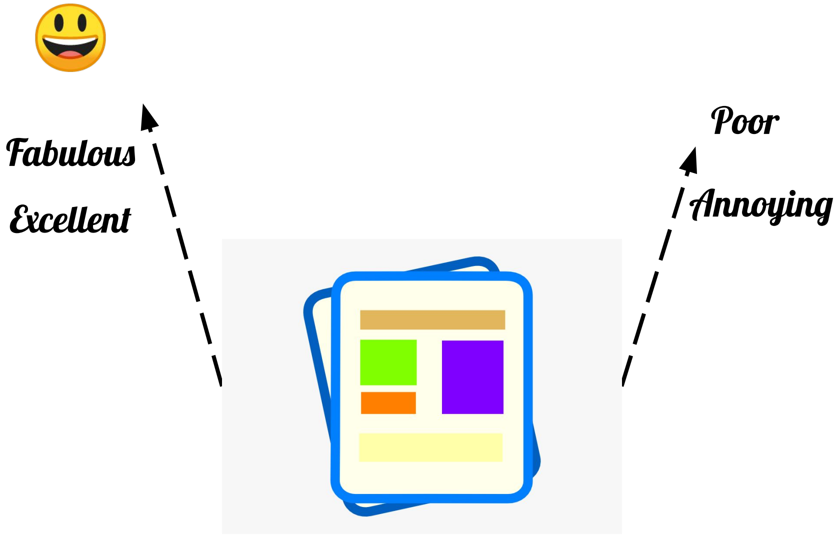
Why a "bag"?

- Any information about the order or structure of words is discarded
- Only concerned with whether known words occur in the document, *not where*
- General intuition: similar documents have similar content

Source: https://medium.com/greyatom/an-introduction-to-bag-of-words-in-nlp-ac967d43b428

Bag-of-Words Model

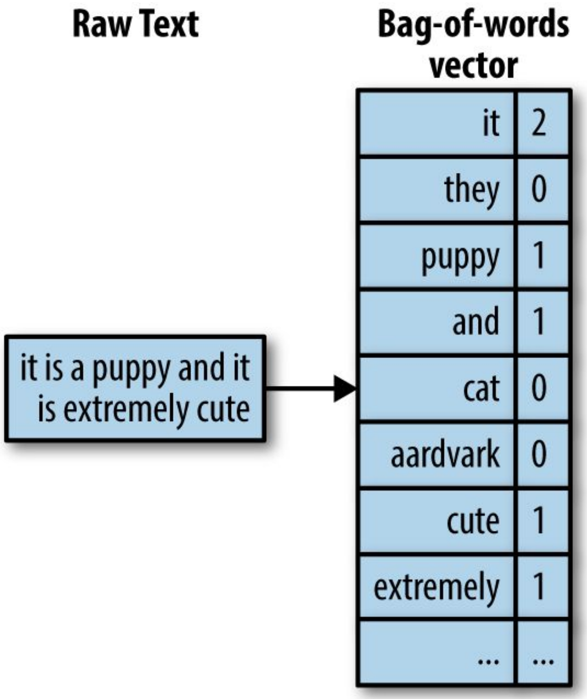
- Transform tokens into a set of features
- Used in document classification, each word used as a feature for training the classifier
- Ex: review-based sentiment analysis



Source: https://towardsdatascience.com/understanding-feature-engineering-part-3-traditional-methods-for-text-data-f6f7d70acd41

Bag-of-Words Model

- Each text document = numeric vector
- Each dimension = specific word (from the corpus)
- Value = frequency in document, occurrence (1/0), or weighted value



Source: Source: http://uc-r.github.io/creating-text-features

Bag-of-Words Model: Simple Example

"A Tale of Two Cities"

by Charles Dickens

> *It was the best of times,*
> *it was the worst of times,*
> *it was the age of wisdom,*
> *it was the age of foolishness,*

Source: https://machinelearningmastery.com/gentle-introduction-bag-words-model/

Step 1: Collect Data

- Data: first few lines of text from book
- Each line treated as a separate “document”
- Four lines = entire corpus of documents

Source: https://machinelearningmastery.com/gentle-introduction-bag-words-model/

Step 2: Design the Vocabulary

- Construct list of all words in the mode's vocabulary
- *Only* the *unique* words (note: we ignore case and punctuation)
- Our list: vocab of 10 words, from corpus of 24 words

- "it"
- "was"
- "the"
- "best"
- "of"
- "times"
- "worst"
- "age"
- "wisdom"
- "foolishness"

Source: https://machinelearningmastery.com/gentle-introduction-bag-words-model/

Step 3: Create Document Vectors

- Score the words in each document
- Document of free text → vector
- Vector = input *or* output of model
- Vocab of 10 words → fixed-length vector
- Use 0/1 binary scoring

- "it" = 1
- "was" = 1
- "the" = 1
- "best" = 1
- "of" = 1
- "times" = 1
- "worst" = 0
- "age" = 0
- "wisdom" = 0
- "foolishness" = 0

Source: https://machinelearningmastery.com/gentle-introduction-bag-words-model/

Managing Vocabulary

- Vocab size ⬆ Vector representation of documents ⬆
- Issue of sparse vector/representation
  - Memory, computational resources
- Decrease vocab size → text cleansing!
  - Ignore case
  - Ignore punctuation
  - Ignore stop words
  - Fix misspelled words
  - Stemming

Source: https://machinelearningmastery.com/gentle-introduction-bag-words-model/

WOMEN WHO CODE DATA SCIENCE

Bag-of-Words ✔️

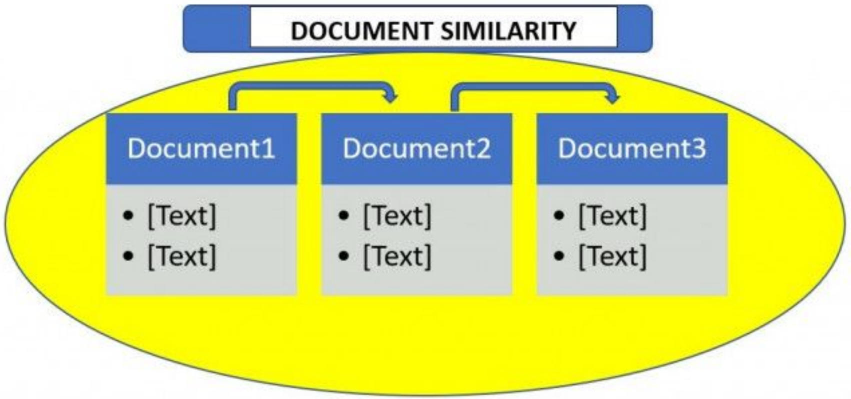

PHEW!!

WOMEN WHO CODE DATA SCIENCE

Document Similarity - Use Cases

Document Similarity

- Use a distance (or similarity-based metric) to identify how similar two text documents are
- Based on features extracted from documents (i.e. bag-of-words, TF-IDF)



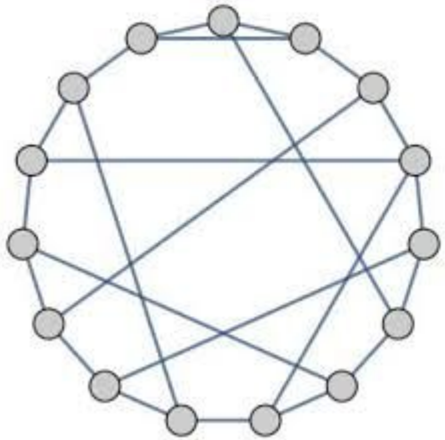
Source: https://www.geeksforgeeks.org/measuring-the-document-similarity-in-python/

Pairwise Similarity

Compute for each pair of documents in a corpus

Given: **D** documents in corpus **C**

D * D matrix: each row & column represents similarity score for a pair of documents, which represent indices at the row and column



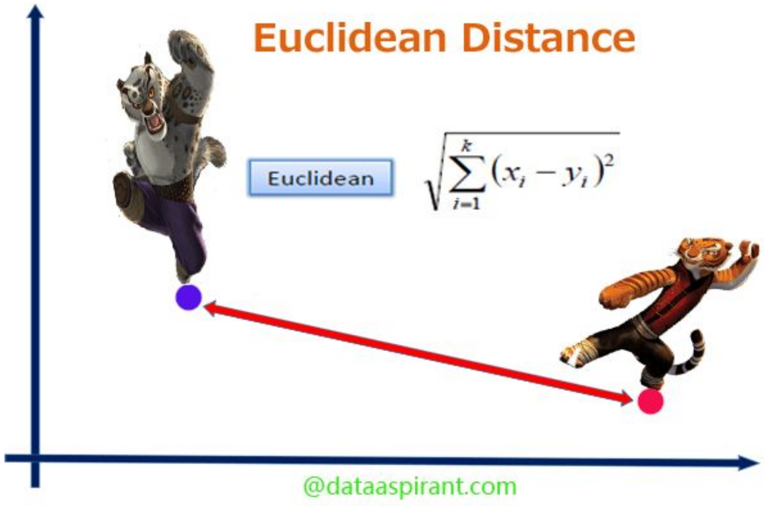
Source: https://arxiv.org/pdf/2006.01127.pdf

WOMEN WHO CODE DATA SCIENCE

Use Cases

1. Euclidean Distance
2. Cosine Similarity
3. Jaccard Similarity
4. Document Clustering with Similarity Features
5. Topic Models

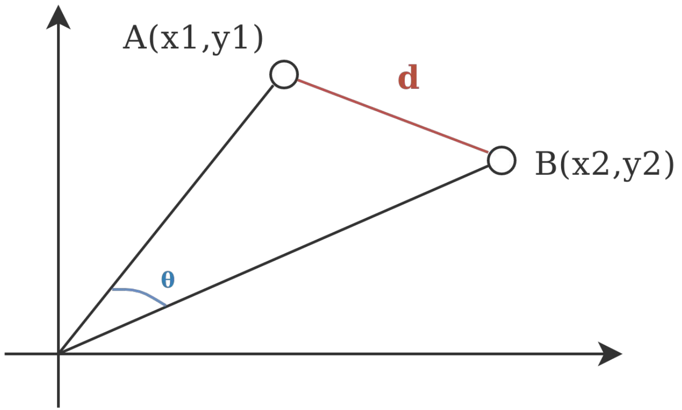
Euclidean Distance (1)



- Compare shortest distance among two objects
- Use Pythagorean Theorem
- Output: score
  - Zero: both objects are identical

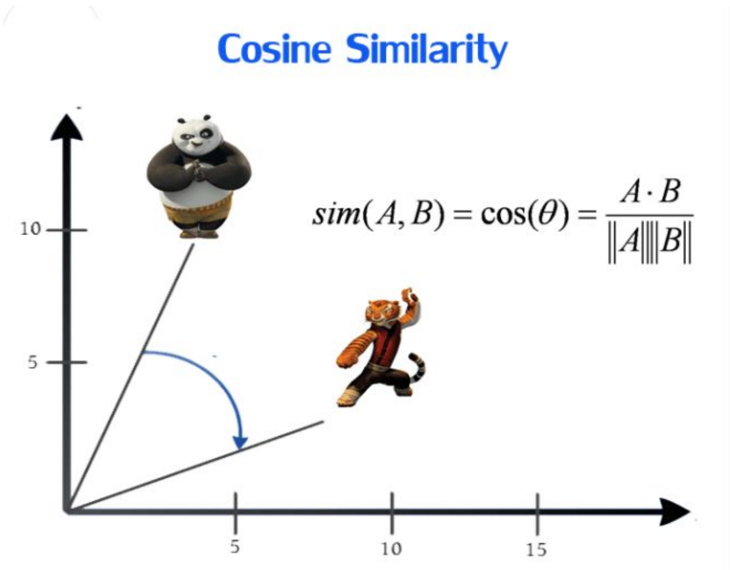
Euclidean Distance (2)

- Limitations:
  - Overcome the “count-the-common-words” approach
- Resolution → Cosine Similarity!



Source: https://towardsdatascience.com/3-basic-distance-measurement-in-text-mining-5852becff1d7

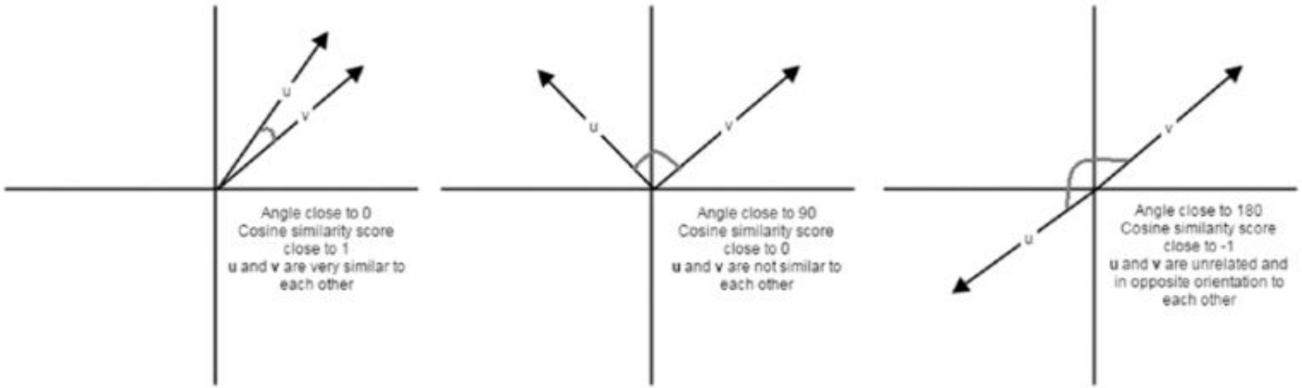
Cosine Similarity (1)



- Determine angle between two objects to find similarity
- Range of score: 0 to 1

Source: https://towardsdatascience.com/3-basic-distance-measurement-in-text-mining-5852becff1d7

Cosine Similarity (2)



- Most popular, widely-used metric
- Represent cosine of angle between feature vector representations of two text documents
    - Lower angle = closer + more similar

Source: https://towardsdatascience.com/understanding-feature-engineering-part-3-traditional-methods-for-text-data-f6f7d70acd41

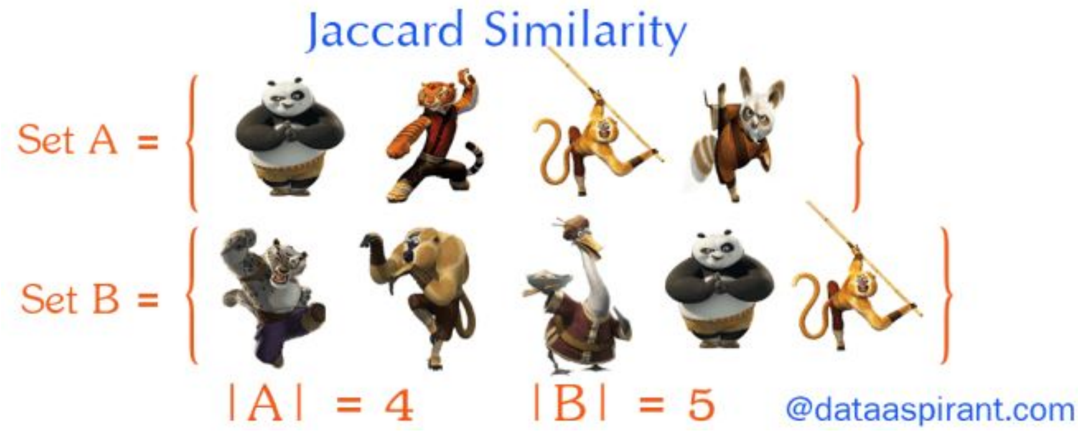
Cosine Similarity (3)

- Compare pairwise document similarity based on TF-IDF feature vectors
- sklearn: cosine_similarity() → call this function on matrix of vectors

	0	1	2	3	4	5	6	7
0	1.000000	0.820599	0.000000	0.000000	0.000000	0.192353	0.817246	0.000000
1	0.820599	1.000000	0.000000	0.000000	0.225489	0.157845	0.670631	0.000000
2	0.000000	0.000000	1.000000	0.000000	0.000000	0.791821	0.000000	0.850516
3	0.000000	0.000000	0.000000	1.000000	0.506866	0.000000	0.000000	0.000000
4	0.000000	0.225489	0.000000	0.506866	1.000000	0.000000	0.000000	0.000000
5	0.192353	0.157845	0.791821	0.000000	0.000000	1.000000	0.115488	0.930989
6	0.817246	0.670631	0.000000	0.000000	0.000000	0.115488	1.000000	0.000000
7	0.000000	0.000000	0.850516	0.000000	0.000000	0.930989	0.000000	1.000000

Source: https://towardsdatascience.com/understanding-feature-engineering-part-3-traditional-methods-for-text-data-f6f7d70acd41

Jaccard Similarity



- Measure number of common words over all words
- More common = both objects should be similar

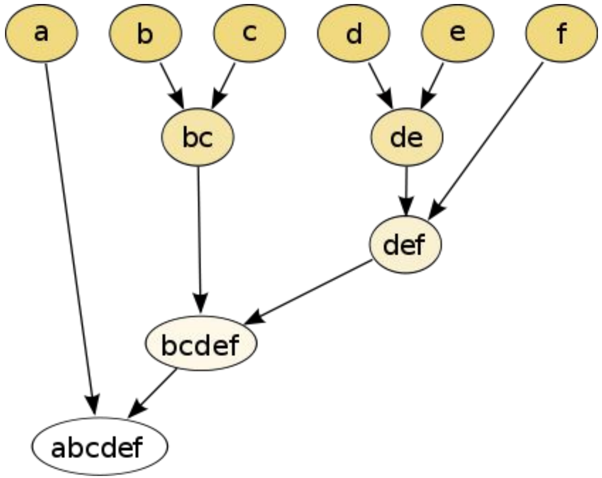
$$(A \cap B) / (A \cup B)$$

- Range of score: 0 to 1
  - Score = 1: identical

Source: https://towardsdatascience.com/3-basic-distance-measurement-in-text-mining-5852becff1d7

Document Clustering (1)

- Leverage unsupervised learning to group data points (i.e. documents) into groups/clusters
- Use hierarchical clustering algorithm to group similar documents, leverage document similarity features
- Two types of algorithms:
  - **Agglomerative**
  - Divisive



Source: https://towardsdatascience.com/understanding-feature-engineering-part-3-traditional-methods-for-text-data-f6f7d70acd41

Document Clustering (2)

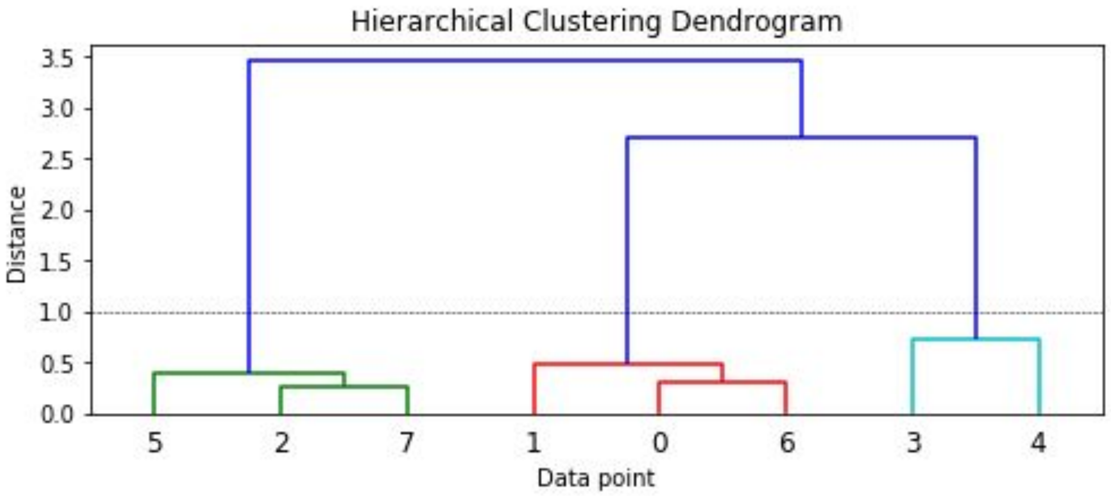
- Strategy: selection of linkage criterion - choosing cluster pairs to merge at each step, based on optimal value of an objective function
- Ward's minimum variance method: find pair leading to minimum increase in total within-cluster variance after merging

	Document\Cluster 1	Document\Cluster 2	Distance	Cluster Size
0	2	7	0.253098	2
1	0	6	0.308539	2
2	5	8	0.386952	3
3	1	9	0.489845	3
4	3	4	0.732945	2
5	11	12	2.69565	5
6	10	13	3.45108	8

Source: https://towardsdatascience.com/understanding-feature-engineering-part-3-traditional-methods-for-text-data-f6f7d70acd41

Document Clustering (3)

Visualization: dendrogram → better understanding!



Source: https://towardsdatascience.com/understanding-feature-engineering-part-3-traditional-methods-for-text-data-f6f7d70acd41

Document Clustering (4)

Distance metric ≥ 1 → obtain our cluster labels

	Document	Category	ClusterLabel
0	The sky is blue and beautiful.	weather	2
1	Love this blue and beautiful sky!	weather	2
2	The quick brown fox jumps over the lazy dog.	animals	1
3	A king's breakfast has sausages, ham, bacon, eggs, toast and beans	food	0
4	I love green eggs, ham, sausages and bacon!	food	0
5	The brown fox is quick and the blue dog is lazy!	animals	1
6	The sky is very blue and the sky is very beautiful today	weather	2
7	The dog is lazy but the brown fox is quick!	animals	1

Source: https://towardsdatascience.com/understanding-feature-engineering-part-3-traditional-methods-for-text-data-f6f7d70acd41

Topic Models (1)

- Extract topic-based features from text documents
- Each topic represented as bag/collection of words/terms from the document corpus
- Various techniques:
  - Latent Semantic Indexing
  - **Latent Dirichlet Allocation**

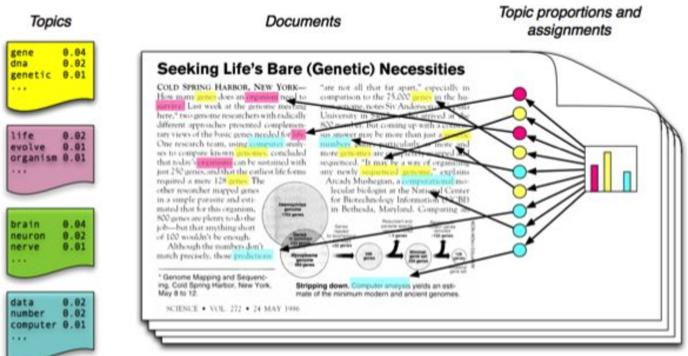
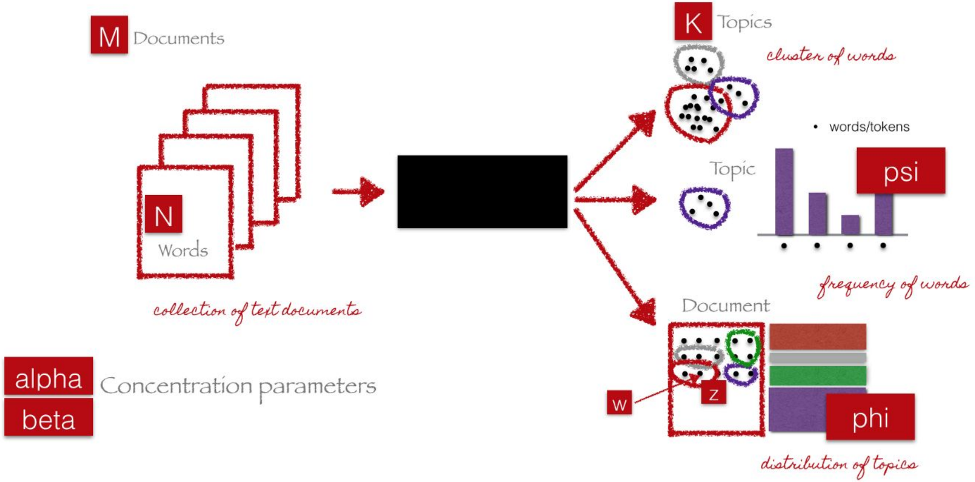


Figure source: Blei, D. M. (2012). Probabilistic topic models. *Communications of the ACM*, *55*(4), 77-84.

Source: https://towardsdatascience.com/understanding-feature-engineering-part-3-traditional-methods-for-text-data-f6f7d70acd41

Topic Models (2)

- Extract **K** topics, from **M** documents
- Black-box algorithm
- Output: topic mixtures for each document → generate constituents of each topic from terms pointing to that topic



Source: https://towardsdatascience.com/understanding-feature-engineering-part-3-traditional-methods-for-text-data-f6f7d70acd41

Topic Models (3)

- LDA decomposed into two components:
    - Document-topic matrix
    - Topic-term matrix
- Leverage scikit-learn for document-topic matrix

	T1	T2	T3
0	0.832191	0.083480	0.084329
1	0.863554	0.069100	0.067346
2	0.047794	0.047776	0.904430
3	0.037243	0.925559	0.037198
4	0.049121	0.903076	0.047802
5	0.054901	0.047778	0.897321
6	0.888287	0.055697	0.056016
7	0.055704	0.055689	0.888607

Source: https://towardsdatascience.com/understanding-feature-engineering-part-3-traditional-methods-for-text-data-f6f7d70acd41

Topic Models (4)

- Use B-o-W model-based features to build topic model-based features, using LDA
- Group documents based on their topic model feature representations, using K-means clustering

	Document	Category	ClusterLabel
0	The sky is blue and beautiful.	weather	2
1	Love this blue and beautiful sky!	weather	2
2	The quick brown fox jumps over the lazy dog.	animals	1
3	A king's breakfast has sausages, ham, bacon, eggs, toast and beans	food	0
4	I love green eggs, ham, sausages and bacon!	food	0
5	The brown fox is quick and the blue dog is lazy!	animals	1
6	The sky is very blue and the sky is very beautiful today	weather	2
7	The dog is lazy but the brown fox is quick!	animals	1

Source: https://towardsdatascience.com/understanding-feature-engineering-part-3-traditional-methods-for-text-data-f6f7d70acd41

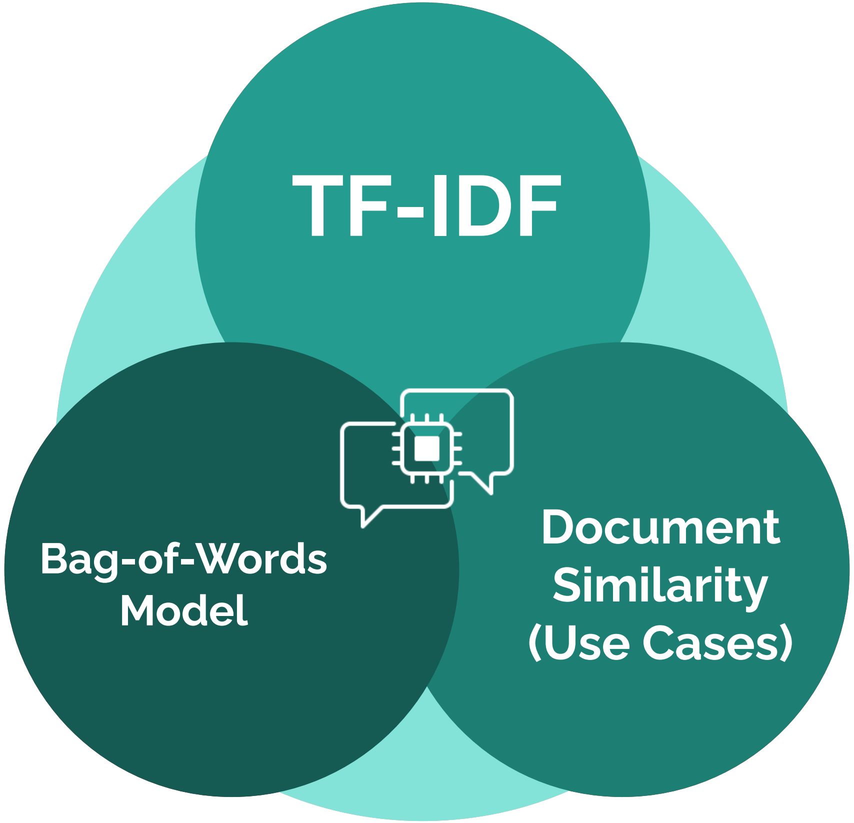
Document Similarity ✔️

5

Theory Wrap-up & Next Steps


PHEW!!

Recap



6

Google Colab Project

https://bit.ly/introtonlp-week3-notebook

Homework #1

Additional Resources

- Document Similarity using Python: https://bit.ly/2BaChK2
- DataMites - TF-IDF and Bag-of-Words Techniques: https://bit.ly/3jkGdZR
- Traditional Methods for Text Data: https://bit.ly/2ZEEh6T

WOMEN WHO CODE DATA SCIENCE

See you next week!

Questions?

Join us on Slack and post your questions to the #help-me channel