



NANYANG
TECHNOLOGICAL
UNIVERSITY
SINGAPORE

NTU Academy for Professional
and Continuing Education

(SCTP) Advanced Professional Certificate

Data Science and AI



3.9 Natural Language Processing

Module Overview

3.1 Probability and Statistics for Machine Learning

3.2 Introduction to Machine Learning

3.3 Supervised Learning

3.4 Supervised Learning - Advanced

3.5 Unsupervised Learning

3.6 Time Series Data & Forecasting

3.7 Neural Network & Deep Learning

3.8 Computer Vision (CV)

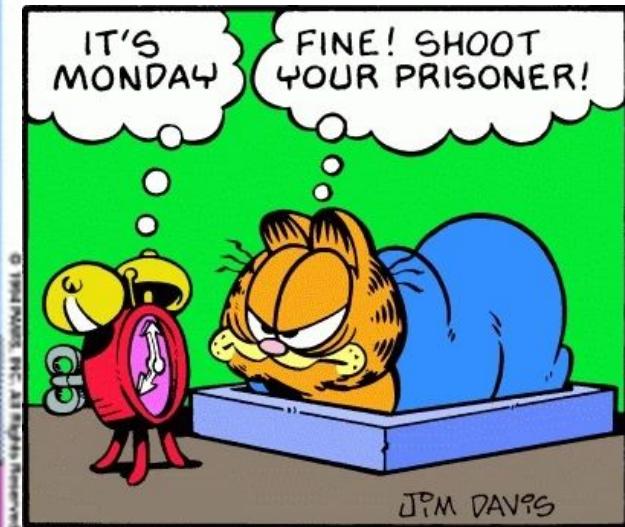
3.9 Natural Language Processing (NLP)

3.10 NLP - Advanced

Lesson Objectives

- Definition, Historical Context, Challenges and Applications of Natural Language Processing (NLP)
- Text Preprocessing
- Text Normalization
- Language Modeling
- Vector Space Model and Word Embeddings
- Text Classification

Garfield was trying to stay
cool



Why do we need NLP?

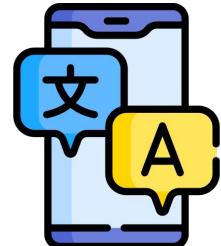
1. Makes Technology Understandable

NLP enables machines to understand, interpret, and respond to human language — whether written or spoken. This is what powers voice assistants (like Siri or Alexa), chatbots, and automated customer service.

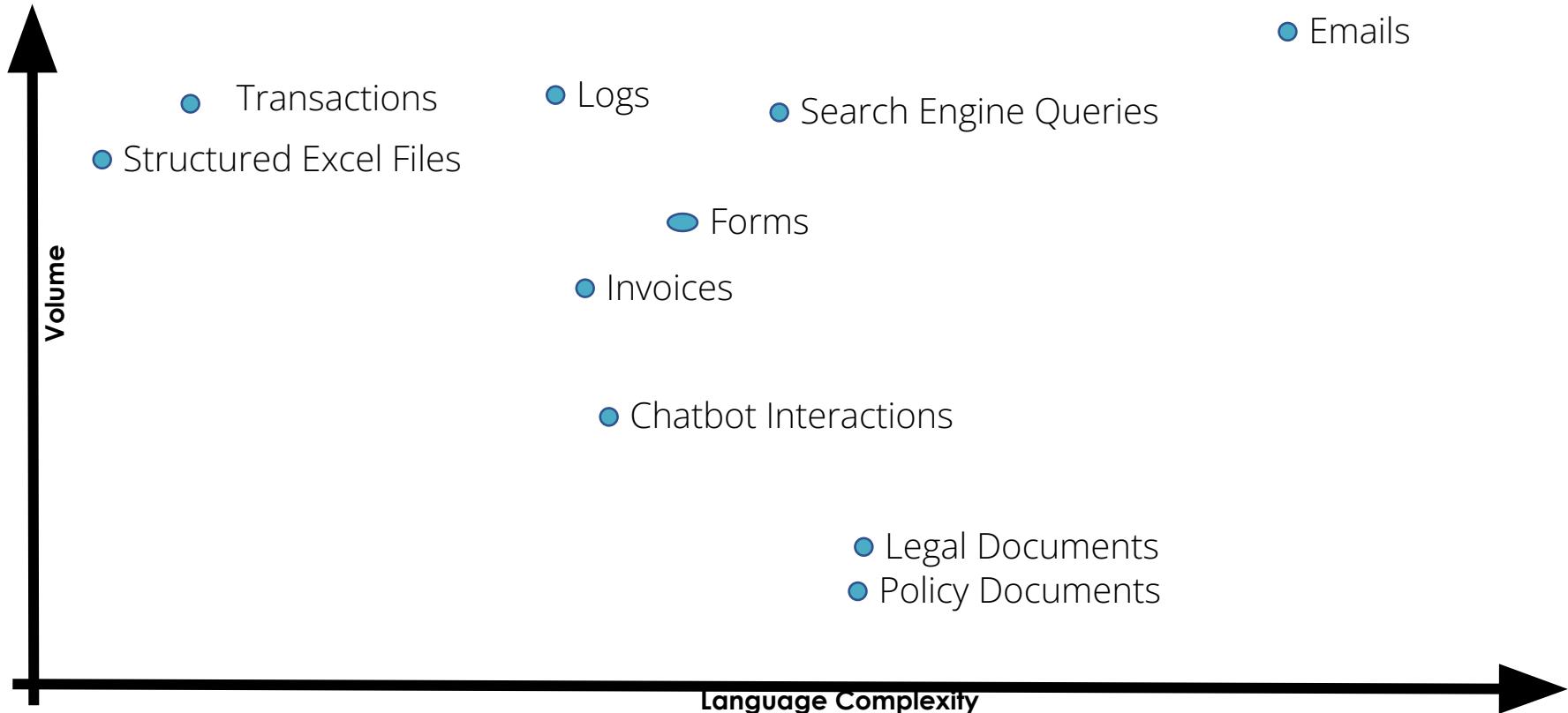
2. Drives Automation

NLP automates tasks that would otherwise require human input:

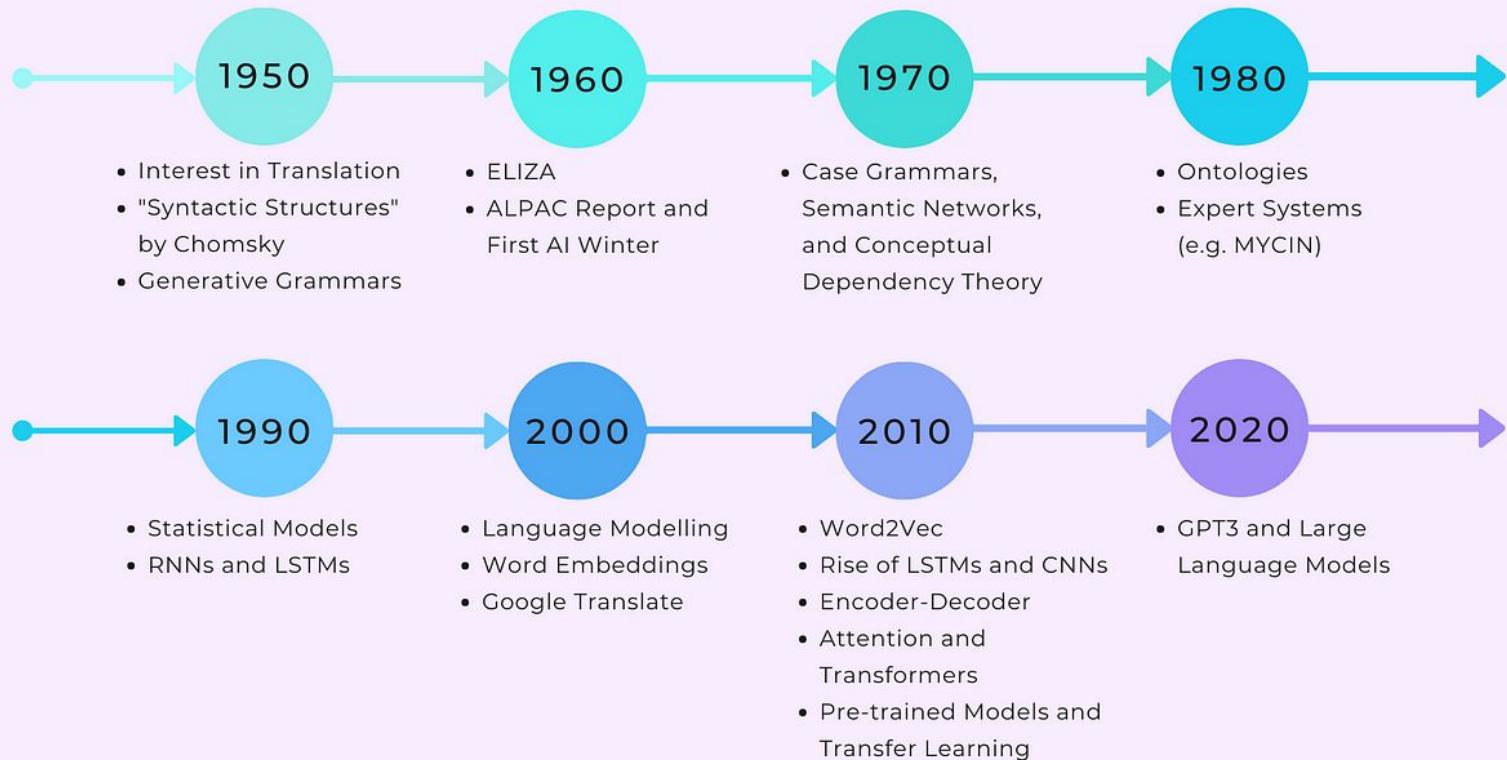
- Email filtering
- Document summarization
- Language translation
- Sentiment analysis (e.g., understanding customer feedback)



WHY



A Brief Timeline of NLP



<https://medium.com/nlplanet/a-brief-timeline-of-nlp-bc45b640f07d>

Welcome to

EEEEEE	LL	III	ZZZZZ	AAAAA
EE	LL	II	ZZ	AA AA
EEEEEE	LL	II	ZZZ	AAAAAAA
EE	LL	II	ZZ	AA AA
EEEEEE	LLLLL	III	ZZZZZ	AA AA

Eliza is a mock Rogerian psychotherapist.

The original program was described by Joseph Weizenbaum in 1966.

This implementation by Norbert Landsteiner 2005.

ELIZA: Is something troubling you ?

YOU: Men are all alike.

ELIZA: What is the connection, do you suppose ?

YOU: They're always bugging us about something or other.

ELIZA: Can you think of a specific example ?

YOU: Well, my boyfriend made me come here.

ELIZA: Is it important to you that your boyfriend made you come here ?

YOU: He says I'm depressed much of the time.

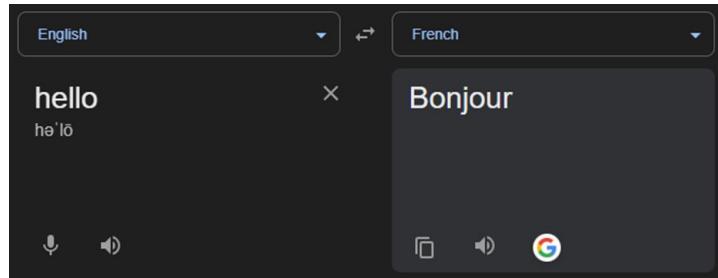
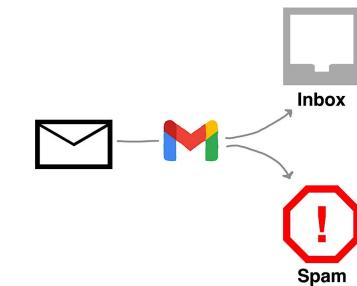
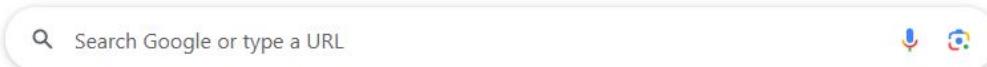
ELIZA: I am sorry to hear that you are depressed.

YOU: It's true. I am unhappy.

ELIZA: Can you explain what made you unhappy ?

YOU:

NLP in real life



This park will become the free, welcoming space that our community is in need of.

- Conciseness

is in need of → **needs**

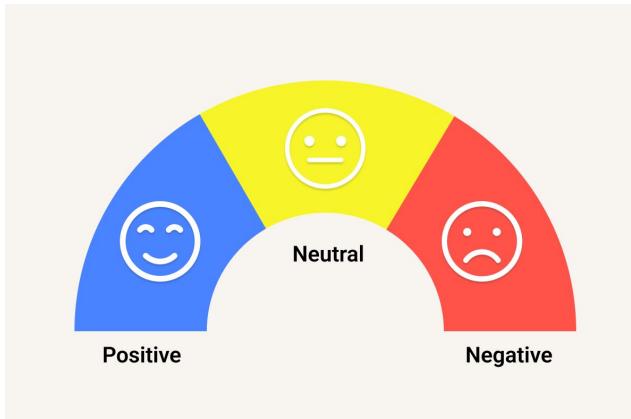
The phrase *is in need of* may be wordy. Consider changing the wording.



<https://www.grammarly.com/blog/company/transforming-writing-style-with-ai/>

NLP Applications

- **Text Classification:** Assigning categories or labels to text, such as spam detection in email services.
- **Machine Translation:** Translating text from one language to another, like Google Translate.
- **Sentiment Analysis:** Identifying the sentiment of text, used in social media monitoring and market research.
- **Chatbots and Virtual Assistants:** Powering conversational agents like Siri, Alexa, and customer service bots.
- **Information Extraction:** Extracting structured information from unstructured text, such as named entity recognition.
- **Summarization:** Generating a shortened version of a text, retaining its most important information.
- **Speech Recognition:** Translating spoken language into text, used in voice user interfaces.
- **Question Answering:** Building systems that automatically answer questions posed by humans in a natural language (ChatGPT).



```
Welcome to
      EEEEEE LL      IIII    ZZZZZZ   AAAA
      EE      LL      II      ZZ   AA  AA
      EEEEEE LL      II      ZZZ  AAAAAAA
      EE      LL      II      ZZ   AA  AA
      EEEEEE LLLLLL  IIII ZZZZZZ   AA  AA

Eliza is a mock Rogerian psychotherapist.
The original program was described by Joseph Weizenbaum in 1966.
This implementation by Norbert Landsteiner 2005.

ELIZA: Is something troubling you ?
YOU: Men are all alike.
ELIZA: What is the connection, do you suppose ?
YOU: They're always bugging us about something or other.
ELIZA: Can you think of a specific example ?
YOU: Well, my boyfriend made me come here.
ELIZA: Is it important to you that your boyfriend made you come here ?
YOU: He says I'm depressed much of the time.
ELIZA: I am sorry to hear that you are depressed.
YOU: It's true. I am unhappy.
ELIZA: Can you explain what made you unhappy ?
YOU:
```

NLP Applications - Semantic Search Engine

Places to go in Japan where there aren't too many tourists

X | 🔍 | ⌂ | ⌂ | ⌂

Reddit · r/JapanTravelTips
60+ comments · 1 year ago · ⌂

Itinerary for underrated and rarely visited gems of Japan
Agree on Onomichi, loved that town wish I had more time there, also any place in Shikoku seems quite tourist free.

Looking for an area in **Japan** where **tourists** don't usually go 45 posts 13 Feb 2024
Tokyo: Things to do that aren't crowded with **tourists** ... 48 posts 11 Mar 2024
More results from www.reddit.com

The Invisible Tourist
<https://www.theinvisibletourist.com/japan-off-the-beat...> · ⌂

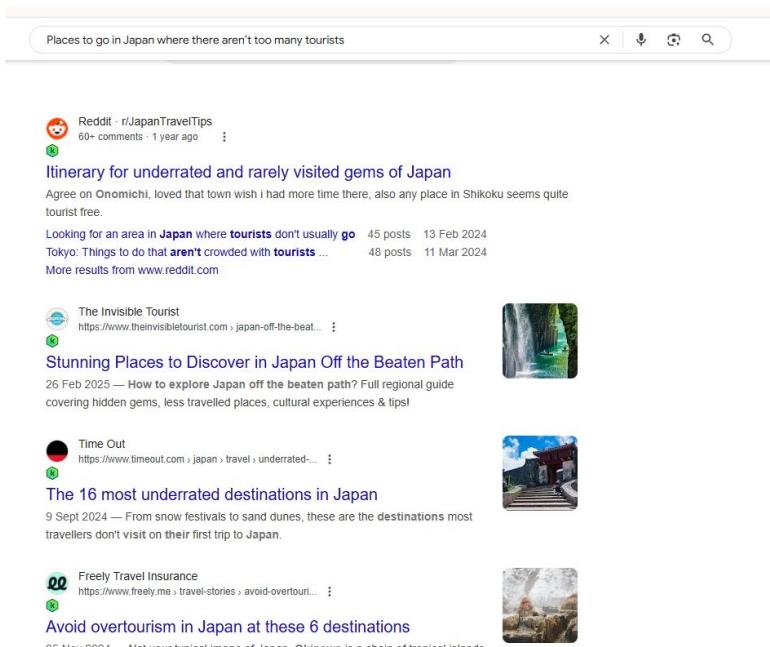
Stunning Places to Discover in Japan Off the Beaten Path
26 Feb 2025 — How to explore Japan off the beaten path? Full regional guide covering hidden gems, less travelled places, cultural experiences & tips!

Time Out
<https://www.timeout.com/japan/travel/underrated-...> · ⌂

The 16 most underrated destinations in Japan
9 Sept 2024 — From snow festivals to sand dunes, these are the destinations most travellers don't visit on their first trip to Japan.

Freely Travel Insurance
<https://www.freely.me/travel-stories/avoid-overtourism...> · ⌂

Avoid overtourism in Japan at these 6 destinations
25 Nov 2024 — Not your typical image of Japan, Okinawa is a chain of tropical islands located nearer to Taiwan than Japan's four main Islands. A popular ...



Healthy foods

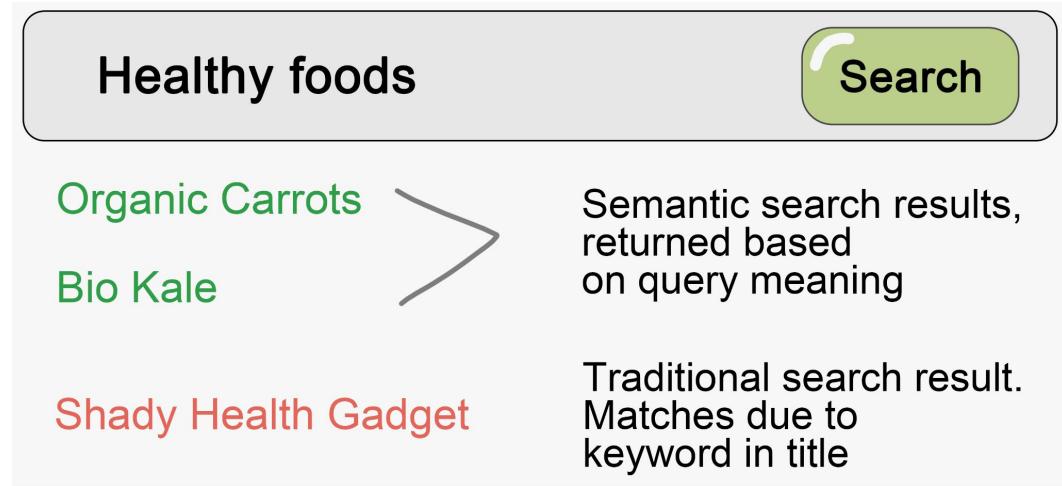
Search

Organic Carrots > Bio Kale

Semantic search results, returned based on query meaning

Shady Health Gadget

Traditional search result. Matches due to keyword in title

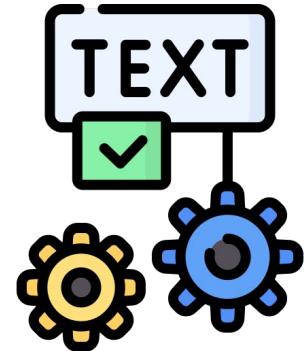


Text Processing

Text preprocessing is a critical step in NLP. It involves preparing and cleaning text data for further analysis and modeling. The goal is to simplify the text and remove any noise that might distract the machine learning algorithms from understanding the core content.

Raw text data is often messy and unstructured, with various issues:

- Irrelevant characters and symbols
- Inconsistent formatting
- Typos and spelling errors
- Diverse languages and slang
- Stopwords (commonly used words that may not be useful in analysis)



Tokenisation

Tokenization is the process of splitting text into smaller units called tokens.
These tokens can be:

- White Space (e.g. "I love cats" → ["I", "love", "cats"])
- Characters (e.g. "Hi" → ["H", "i"])
- Sentences (splitting a paragraph into sentences)

It's one of the first steps in text processing for most NLP pipelines.



Stemming

Stemming removes suffixes (and sometimes prefixes) from words to arrive at the stem, which may not always be a valid word in the language.

For example:

- "running" → "run"
- "flies" → "fli"
- "happiness" → "happi"

Note: The resulting stem might not be a real word! Stemming is rule-based, not always linguistically correct.



Lemmatization

Lemmatization is the process of grouping together the inflected forms of a word so they can be analyzed as a single item, identified by the word's lemma, or dictionary form.

While stemming often involves rule-based chopping of ends of words, lemmatization involves a linguistic approach to reduce a word to its base or root form. Lemmatization uses vocabulary and morphological analysis, often with the aid of part-of-speech tagging, to return the base or dictionary form of a word, known as the lemma.

Word	Lemma	Stemmed
running	run	run
better	good	better
studies	study	studi
was	be	wa

Stop Words

Stop words are the most frequently used words in a language that usually carry grammatical meaning but not much semantic value.

Examples (in English):

the, is, in, at, which, on, a, an, and, but

For instance, in the sentence:

"The cat is on the mat"

After stop word removal → "cat mat"

Removing stop words can help:

- Reduce noise in the data
- Improve performance of models by reducing dimensionality
- Focus on meaningful content (like nouns, verbs, key adjectives)
- However, it's not always beneficial—especially if word context or sentiment matters.

Part of Speech Tagging

Part-of-Speech tagging assigns tags (e.g., noun, verb, adjective) to each word in a text based on:

- Its definition
- Its context within the sentence

For example:

Sentence: "She can fish."

Tags: She/PRP (pronoun), can/MD (modal verb), fish/VB (verb)

Now compare:

Sentence: "I caught a can of tuna."

Tags: I/PRP, caught/VBD, a/DT, can/NN (noun), of/IN, tuna/NN

Same word, different tag depending on context!



Why we need POS tagging

It helps NLP systems understand the structure and meaning of text:

- **Named Entity Recognition (NER):** Distinguish between "**Apple**" as a noun vs. brand.
- **Text-to-speech:** Correct pronunciation
(e.g., "**read**" as past or present).
- **Machine translation:** Proper grammar conversion.
- **Dependency parsing:** Understand relationships between words.

When **Sebastian Thrun** PERSON started working on self-driving cars at **Google** ORG in **2007** DATE, few people outside of the company took him seriously. "I can tell you very senior CEOs of major **American** NORP car companies would shake my hand and turn away because I wasn't worth talking to," said **Thrun** PERSON, now the co-founder and CEO of online higher education startup Udacity, in an interview with **Recode** ORG earlier this week DATE.

<https://demos.explosion.ai/displacy-ent>

A little **less than a decade later** DATE, dozens of self-driving startups have cropped up while automakers around the world clamor, wallet in hand, to secure their place in the fast-moving world of fully automated transportation.

Code-along

Jupyter Notebook

Part 1: Text Processing



Language Models

A language model predicts the likelihood of a sequence of words.

“Given the previous words, what’s the most probable next word?”

For the sentence:

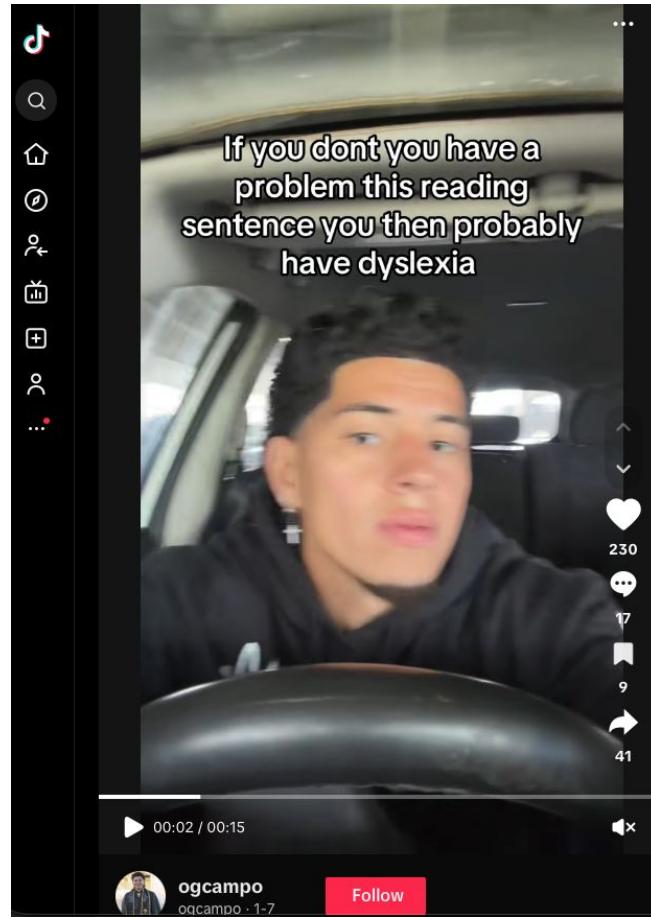
“The cat sat on the ___”

A language model might rank:

- “**mat**” → high probability
- “**refrigerator**” → lower
- “**banana**” → very low

The model has learned grammar, semantics, and word patterns.





Types of Language Models

Type	How it works	Example
N-gram	Uses fixed-size history (e.g., last 2 words)	“The cat” → predicts “sat”
RNN/LSTM	Learns longer-term dependencies using hidden states	Captures sequences and context better than N-grams
Transformer-based	Uses attention to look at all words at once (no fixed memory)	BERT, GPT, etc.

N grams

An n-gram is a sequence of n words (or tokens) in a text.

They're used to model word sequences and estimate how likely a word is to follow others.

This is Big Data AI Book

Uni-Gram

This	Is	Big	Data	AI	Book
------	----	-----	------	----	------

Bi-Gram

This is	Is Big	Big Data	Data AI	AI Book
---------	--------	----------	---------	---------

Tri-Gram

This is Big	Is Big Data	Big Data AI	Data AI Book
-------------	-------------	-------------	--------------

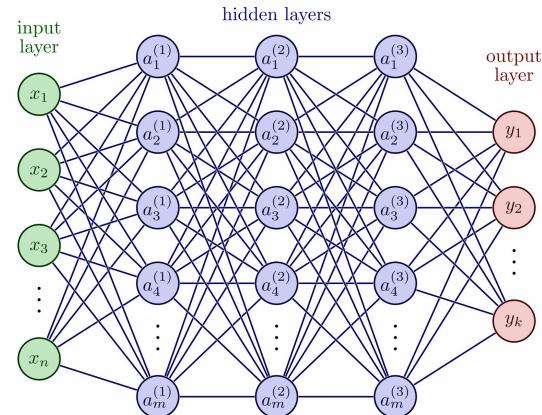
N grams

N-grams are now mostly replaced by neural language models (like RNNs and Transformers) that:

- Capture long-term dependencies
- Handle out-of-vocabulary words better
- Use attention to model context dynamically

BUT — n-grams are still useful for:

- Spell-checkers
- Query completion
- Simpler tasks or when interpretability is needed

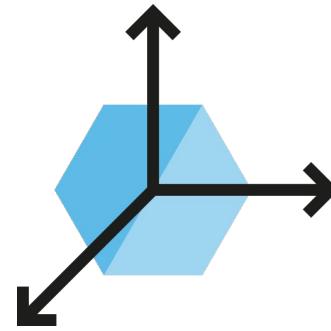


Vector Space Model

The Vector Space Model (VSM) is a mathematical model used to represent text documents as vectors of identifiers, such as index terms. It is used in information retrieval and text mining to measure the similarity between documents. In VSM, each dimension corresponds to a separate term, and the value in each dimension represents the significance of the term in the document.

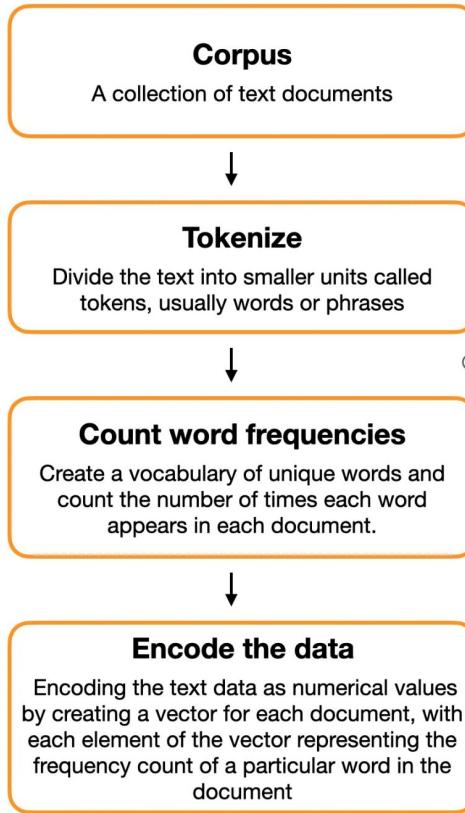
Term-Document Matrix

In VSM, a Term-Document Matrix is a mathematical representation of a text corpus. It describes the frequency of terms that occur in the collection of documents. In a Term-Document Matrix, rows correspond to terms in the corpus while columns correspond to documents. Each entry in this matrix denotes the frequency or the weight of a term in a document.



Bag of Words Model

Bag of Words Model explanation



Example

Corpus:
The dog is happy. The child makes the dog happy. The dog makes the child happy

Tokenization:
D1: [The] [dog] [is] [happy]
D2: [The] [child] [makes] [the] [dog] [happy]
D3: [The] [dog] [makes] [the] [child] [happy]

Documents	Counting word frequencies
D1	the: 1, dog: 1, is: 1, happy: 1
D2	the: 2, dog: 1, makes: 1, child: 1, happy: 1
D3	the: 2, child: 1, makes: 1, dog: 1, happy: 1

Encode	child	dog	happy	is	makes	the	BoW Vector representations
D1	0	1	1	1	0	1	[0,1,1,1,0,1]
D2	1	1	1	0	1	2	[1,1,1,0,1,2]
D3	1	1	1	0	1	2	[1,1,1,0,1,2]

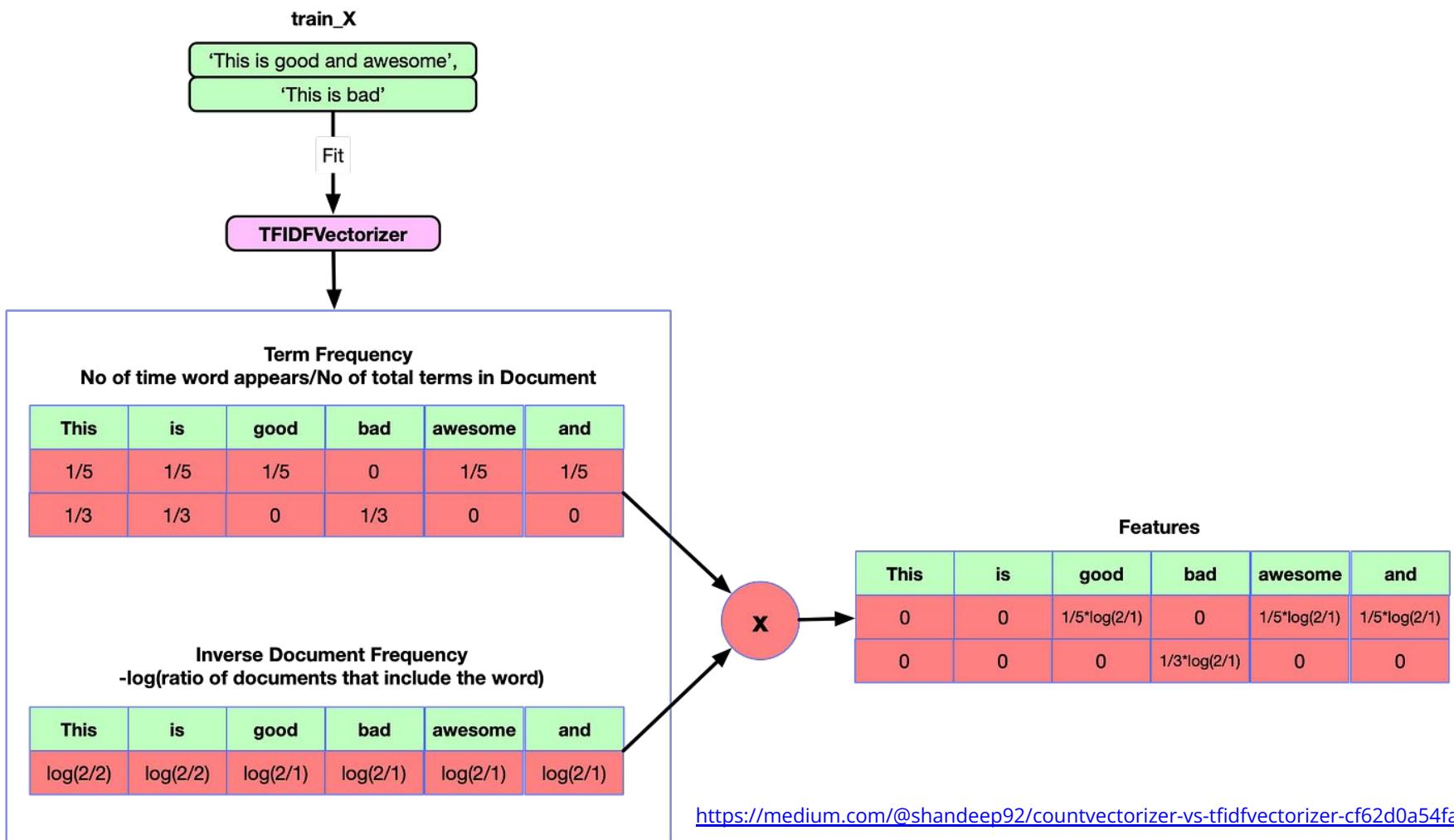
TF-IDF

Converts a collection of text into a matrix of features (like CountVectorizer), but it applies a weighting scheme to the word counts.

- **TF** (Term Frequency): How often a word appears in a document.
- **IDF** (Inverse Document Frequency): Measures how important a word is across all documents, with rare words given more weight.

$$TF(t, d) = \frac{\text{Number of times term } t \text{ appears in document } d}{\text{Total number of terms in document } d}$$

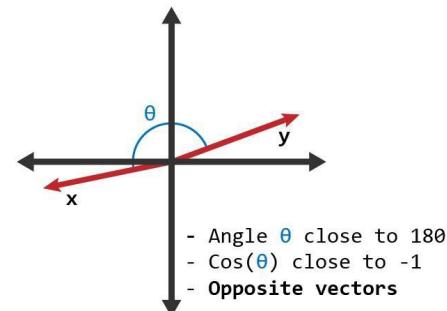
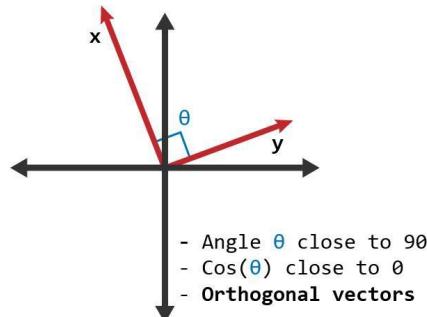
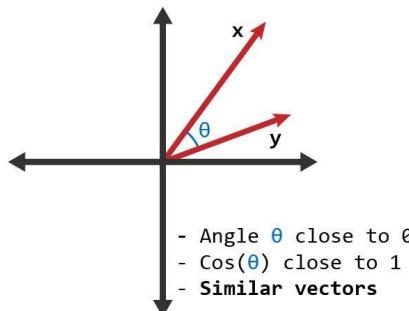
$$IDF(t, D) = \log \frac{\text{Total number of documents in corpus } D}{\text{Number of documents containing term } t}$$



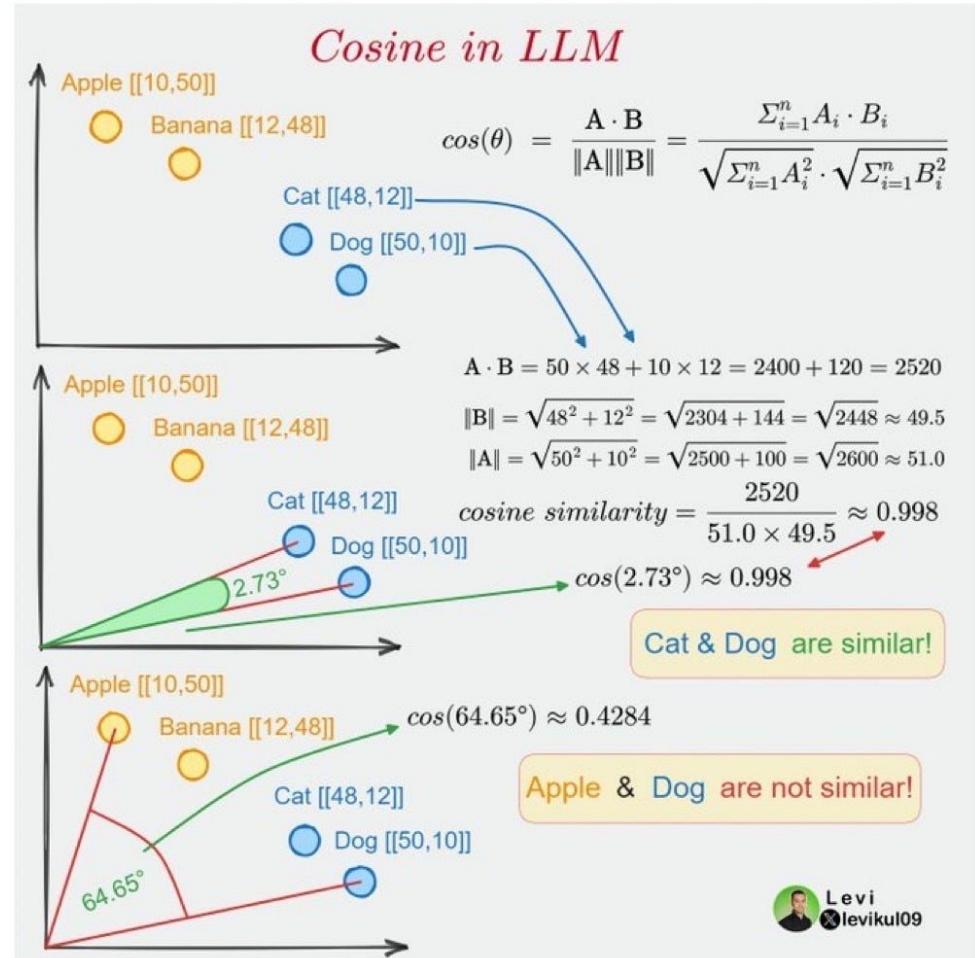
<https://medium.com/@shandeep92/countvectorizer-vs-tfidfvectorizer-cf62d0a54fa4>

Text Similarity - Cosine Similarity

- Cosine similarity measures the cosine of the angle between two vectors in an n-dimensional space.
- Think of each word, sentence, or document as a vector, and cosine similarity tells you how similar their directions are — regardless of their magnitude (length).
- In NLP, cosine similarity typically falls between 0 and 1 because vector values are usually non-negative.



Text Similarity - Cosine Similarity



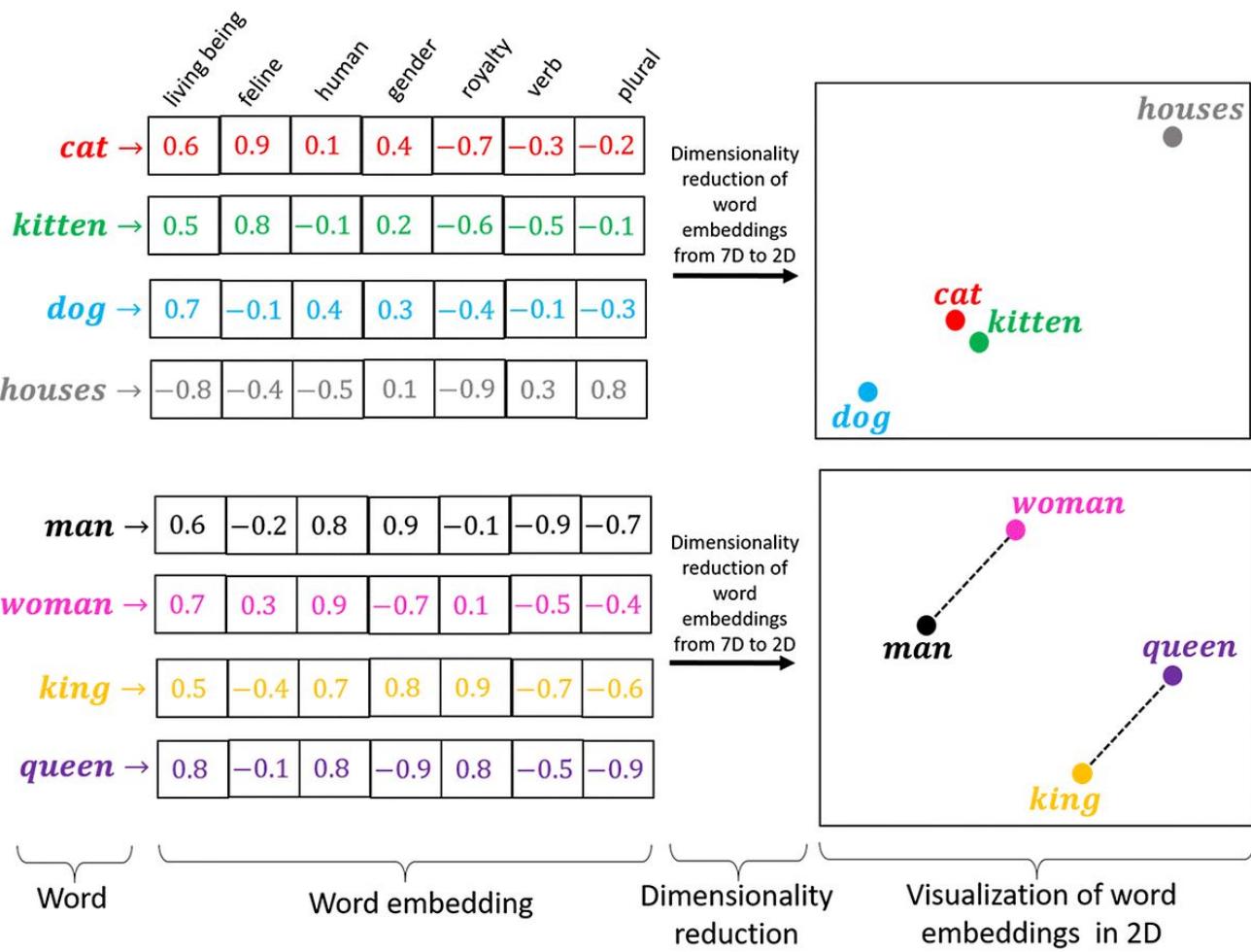
Code-along

Jupyter Notebook



Part 2: TF-IDF Vectorizer

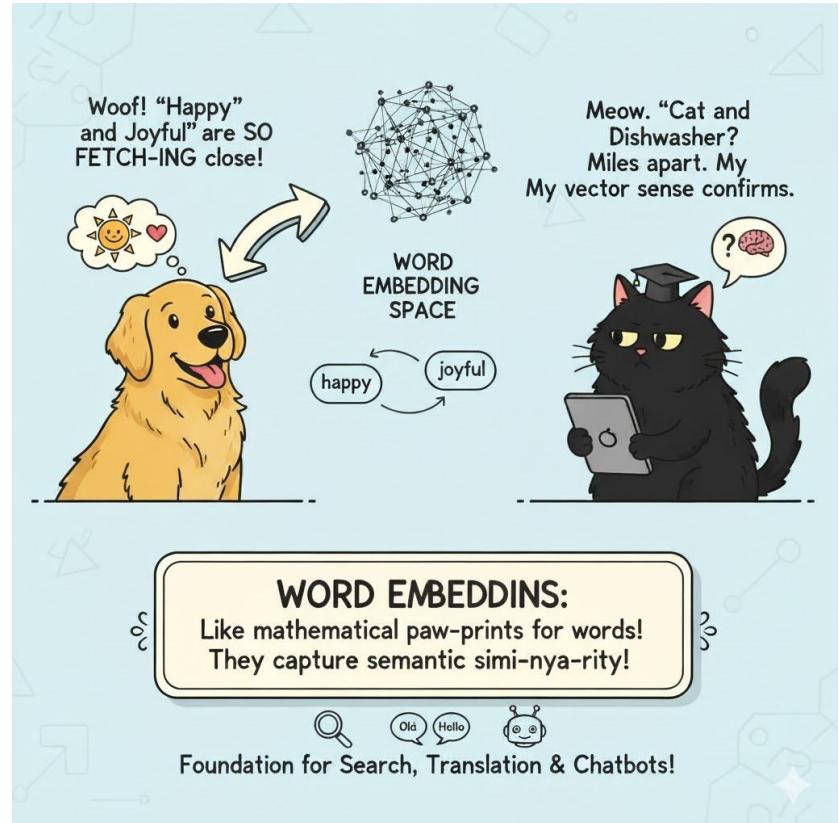
Word Embeddings



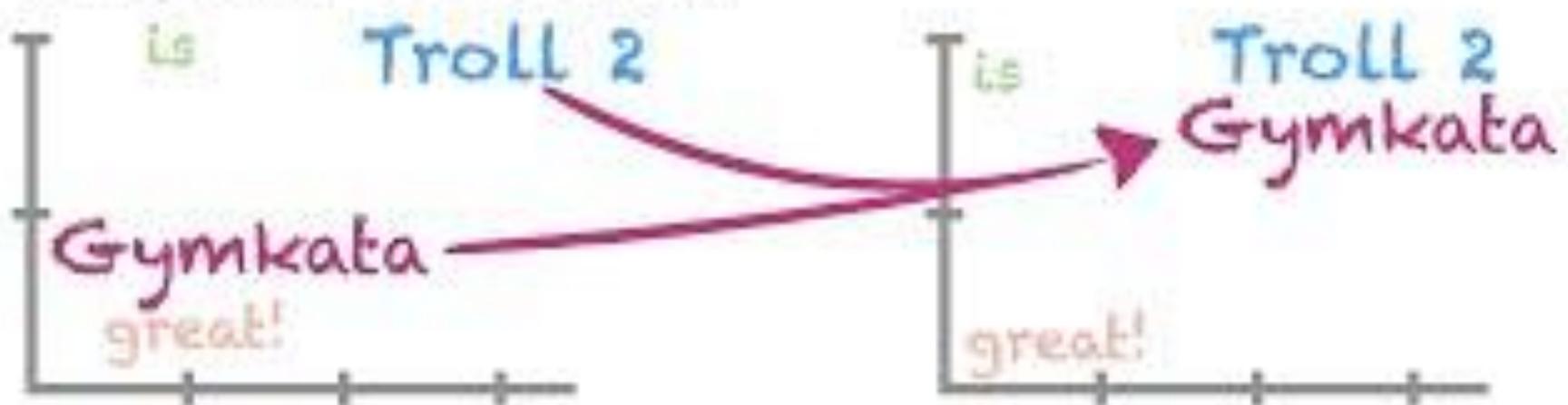
Word Embeddings

They're like mathematical fingerprints for words. Each word is mapped to a vector (e.g., 300 dimensions) where semantic relationships between words are captured by the distance and direction between their vectors.

- Capture semantic similarity: “**happy**” and “**joyful**” will be close in vector space.
- Input for models like classifiers, clustering algorithms, or neural networks.
- Foundation for tasks like sentiment analysis, machine translation, chatbots, search, and more.



Word Embedding and Word2Vec...

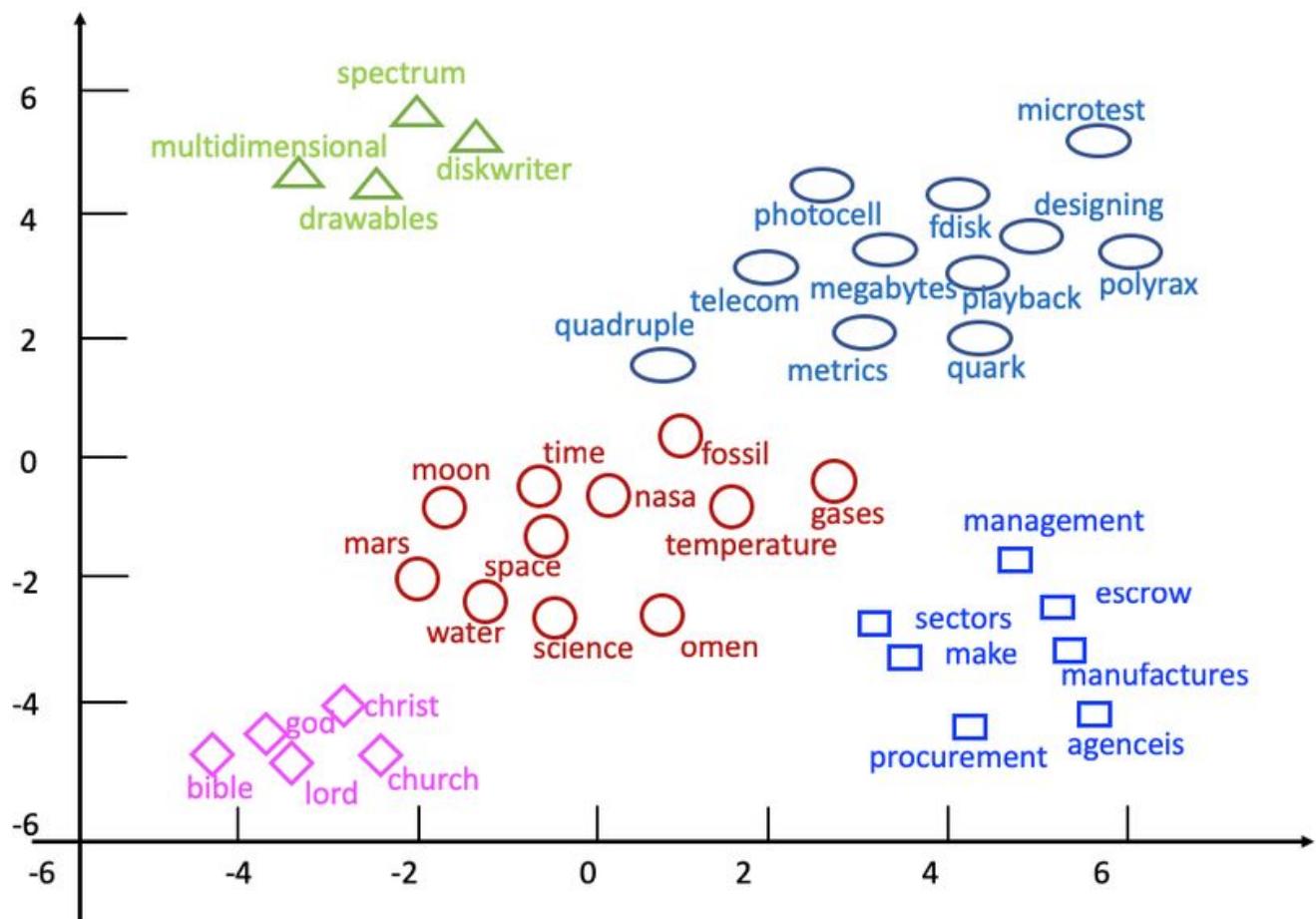


...Clearly Explained!!!

Word Embeddings

Feature	Word2Vec	GloVe	FastText	BERT
Embedding type	Static	Static	Static	Contextual
Handles OOV words?	No	No	Yes	Yes
Subword aware?	No	No	Yes	Yes
Dimensionality	100–300	50–300	100–300	768 (base), 1024 (large)

Word Embeddings & PCA



https://www.researchgate.net/figure/D-PCA-projection-of-word-embeddings-Five-different-word-clusters-are-shown_fig2_332892222

Code-along

Jupyter Notebook



Part 3: Word Embeddings

End of Lesson - Exit Ticket

Survey Link

<https://www.menti.com>

