Natural Language Processing

Erfan Akhavan Azari

@erfan226

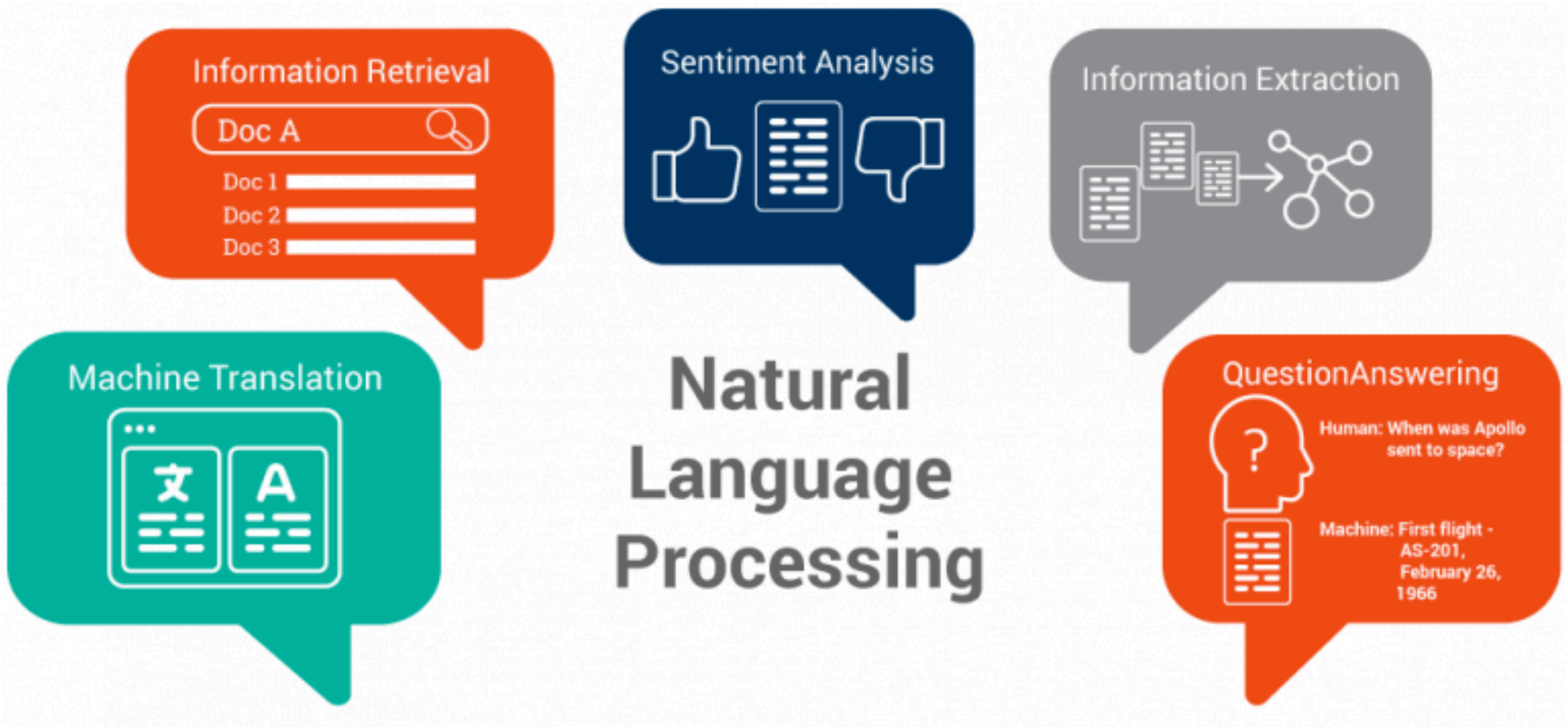
Outline

- What is Natural Language Processing?
- NLP Applications
  - ChatBots
  - OCR
- NLP Challenges & ML Solutions
- Sample NLP Pipeline
- Information Retrieval
- NLP Techniques
- Text Corpus
- Text Representation
- Language Modeling
- Named Entity Recognition
- POS Tagging
- Text Classification
- Kaggle Datasets
- Libraries & Toolkits
- Resources

What is Natural Language Processing?

Natural Language Processing, NLP for short, is an interdisciplinary field between linguistics, computer science, and artificial intelligence.

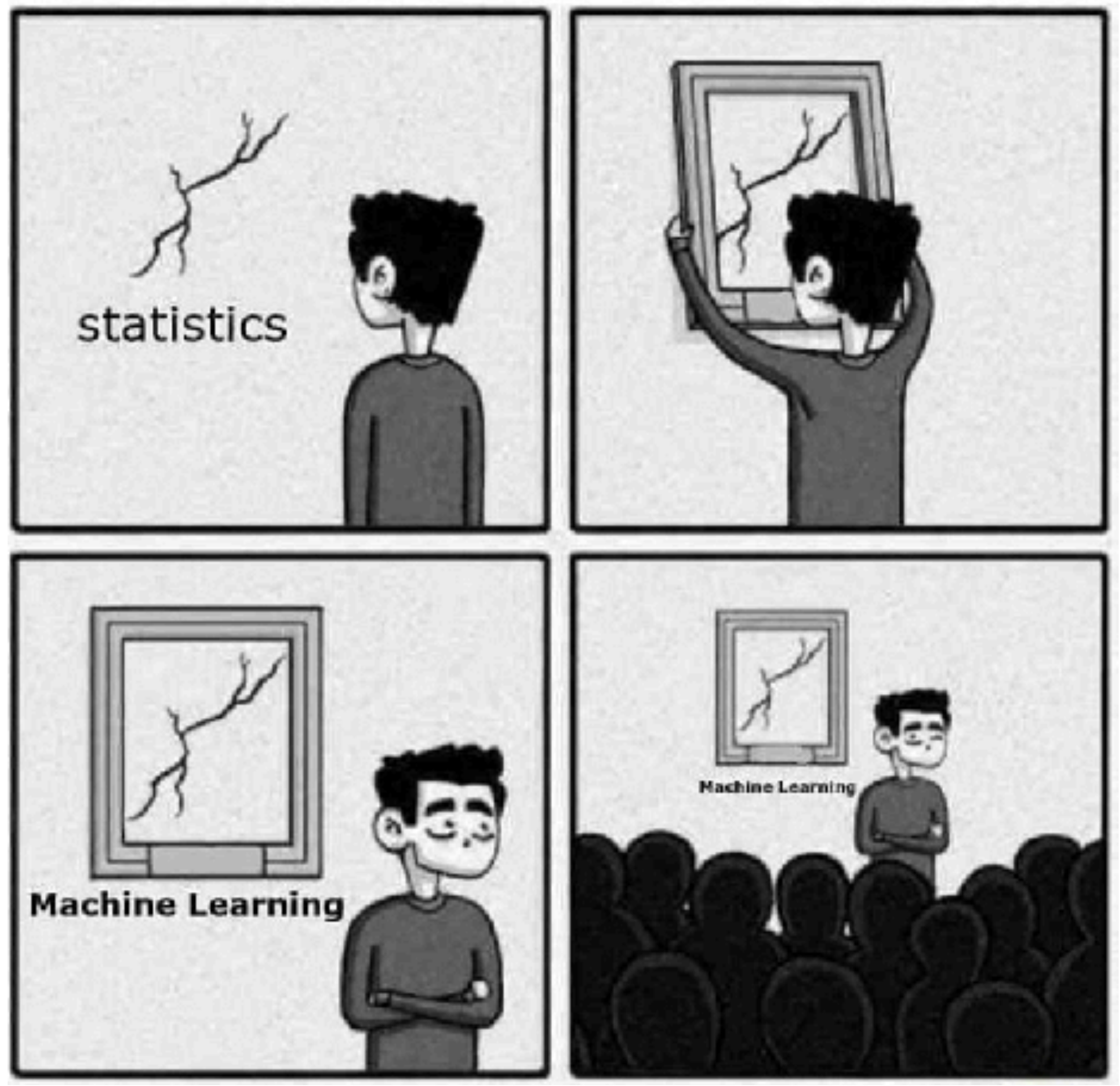
The ultimate goal of this branch is to enable computers to have full-fledged, human level capability of communication either in form of text, voice or both.

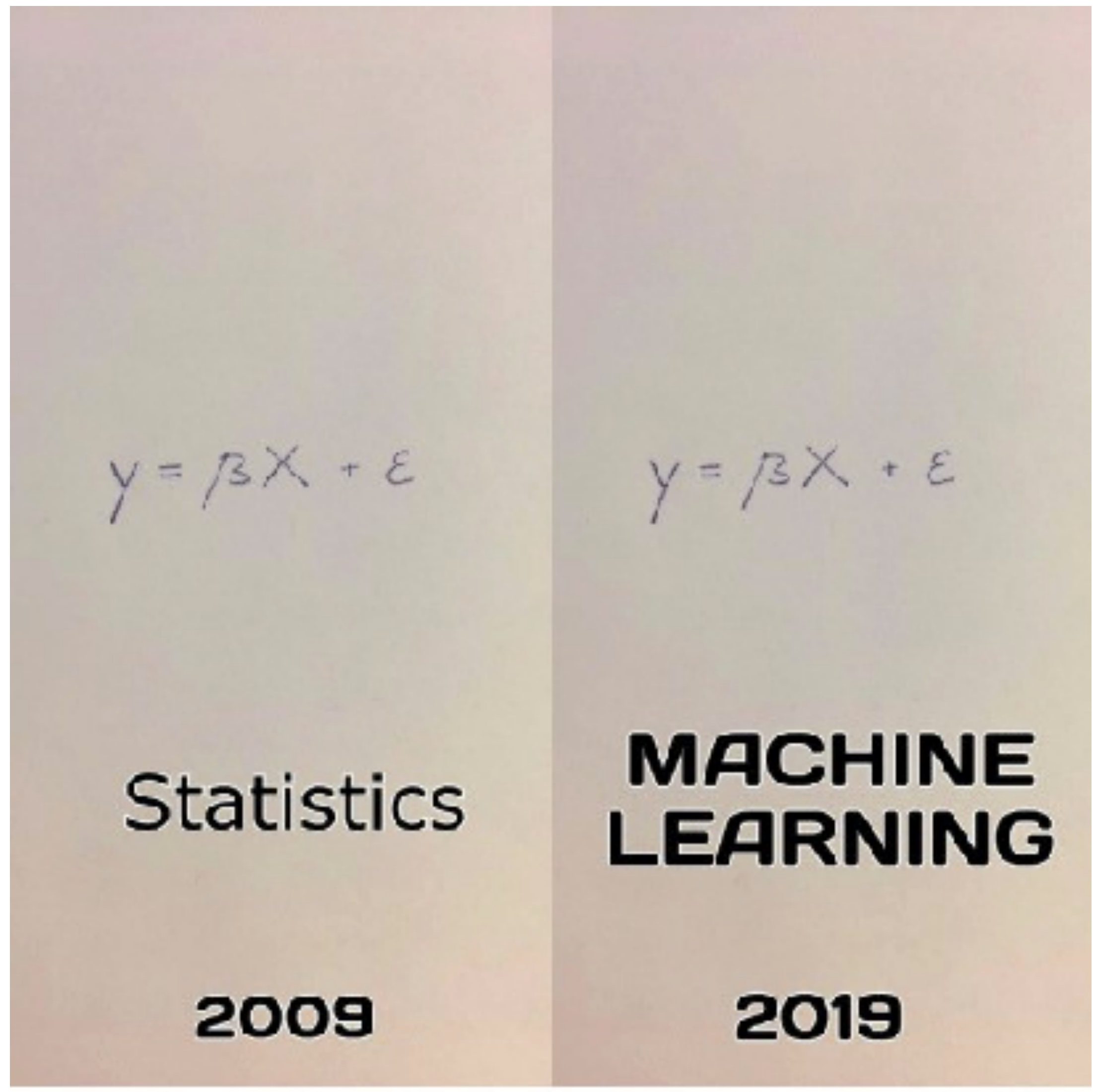


Source: medium.com/analytics-vidhya

Natural Language Processing: Ultimate Goal

Automatic captioning & generating text for memes. Obviously!





#10yearchallenge

Source: medium.com/nybles

NLP Applications

- **Document Processing:** Information Extraction, Summarization, Topic Identification, Document Clustering
- **Information Retrieval:** Text/Spoken Document Retrieval
- **Machine Translation**
- **Text Generation** (NLG)
- **Text Summarization**
- **Spell and Grammar Checking**
- **Speech Recognition** (or ASR)
- **Text-to-Speech Synthesis** (TTS)
- **Optical Character Recognition** (OCR)
- **Spoken Dialogue Systems & ChatBots**
- **Question-Answering Systems** (QA)

NLP Applications: Dialogue Systems & Chatbots

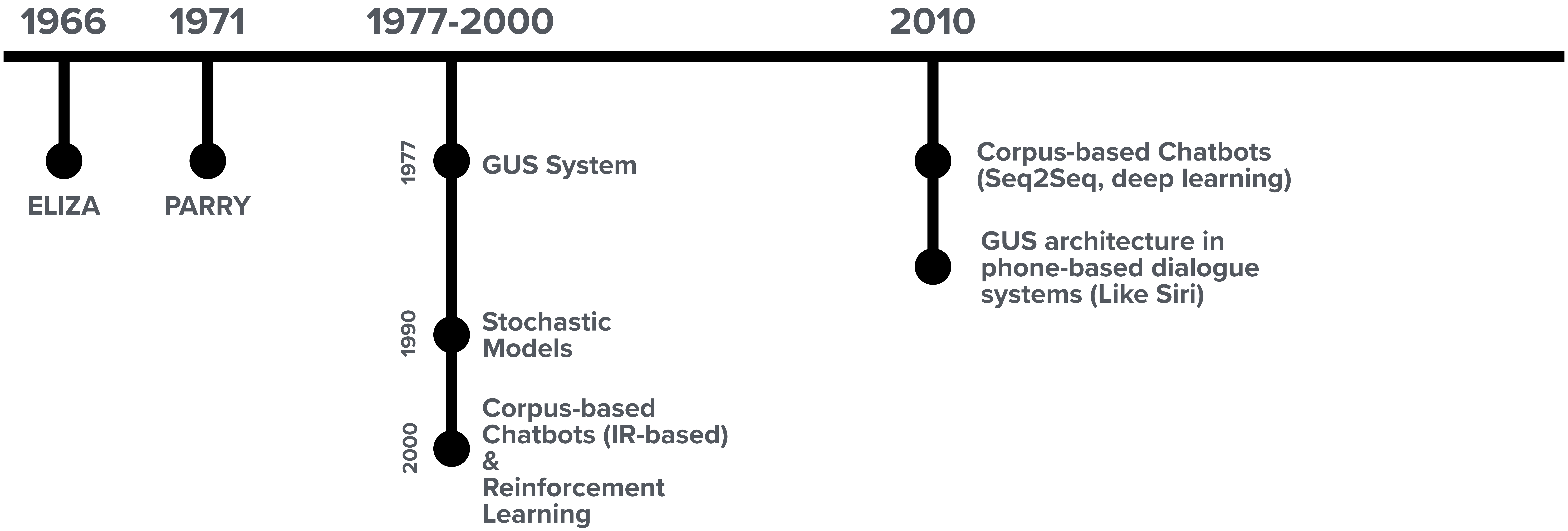
- **Task-oriented Dialogue Systems:** Converses with users to help complete tasks
  - **Tasks:** Giving directions, Controlling appliances, finding restaurants, or making calls
  - **Examples:** Digital assistants like Siri, Alexa, Google Assistant/Home, Cortana, etc.
- **Chatbots:** Systems designed for extended conversations, by mimicking the unstructured conversations or ‘chats’ characteristic of human-human interaction
  - **Goal:** Entertainment, or making task-oriented agents more natural
  - **Examples:** Eliza, Parry, Microsoft’s XiaoIce & Tay, Replika, etc.

Replika

Google Assistant

Tay

NLP Applications: Dialogue Systems & Chatbots Timeline



Natural Language Processing: and We Are Going to…

Design an intelligent voice assistant, with performance comparable to Google Assistant.

Googool Assistant!

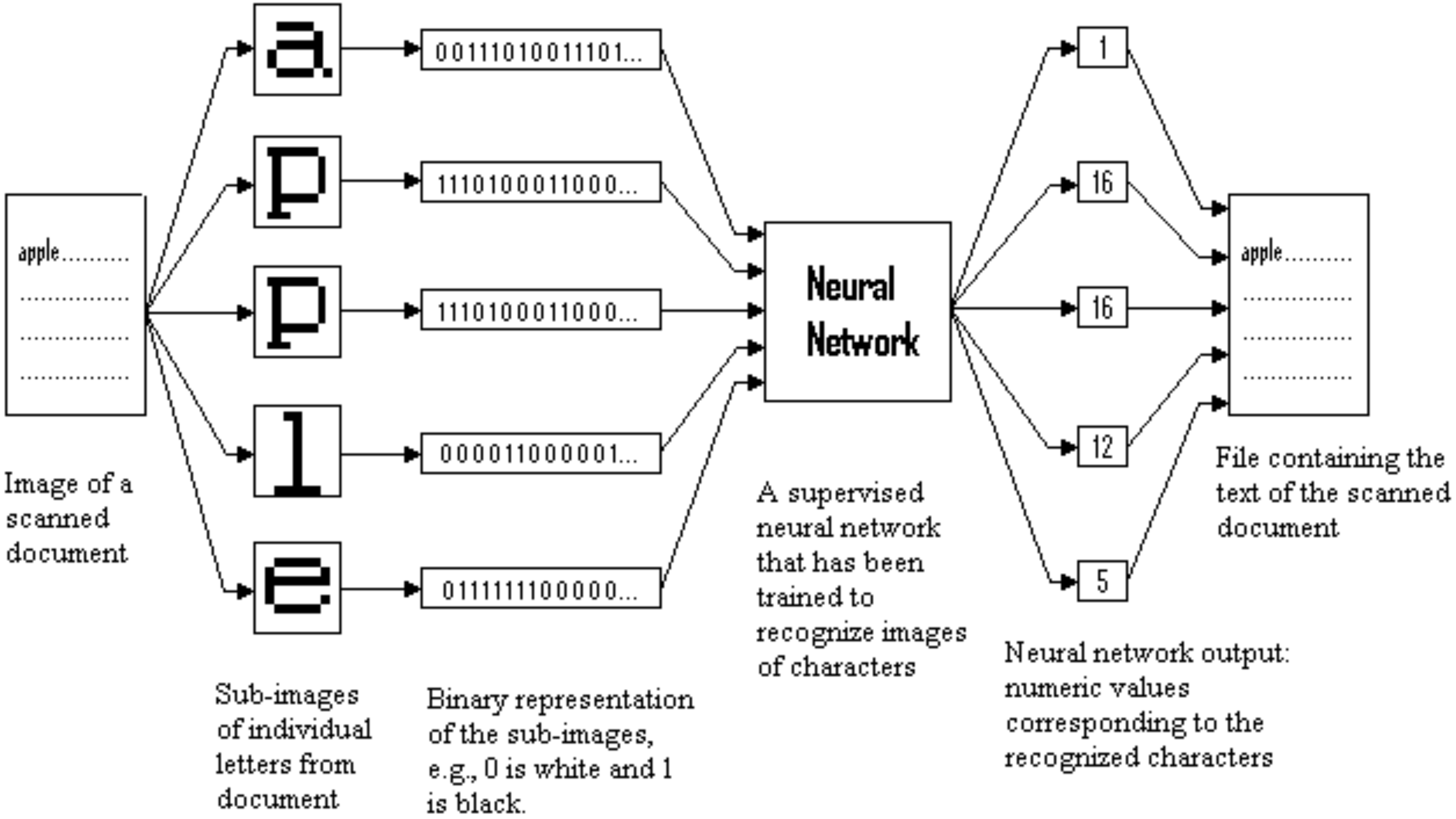
Natural Language Processing: and We Are Going to…

Design an intelligent voice assistant, with performance comparable to Google Assistant.

Googool Assistant!

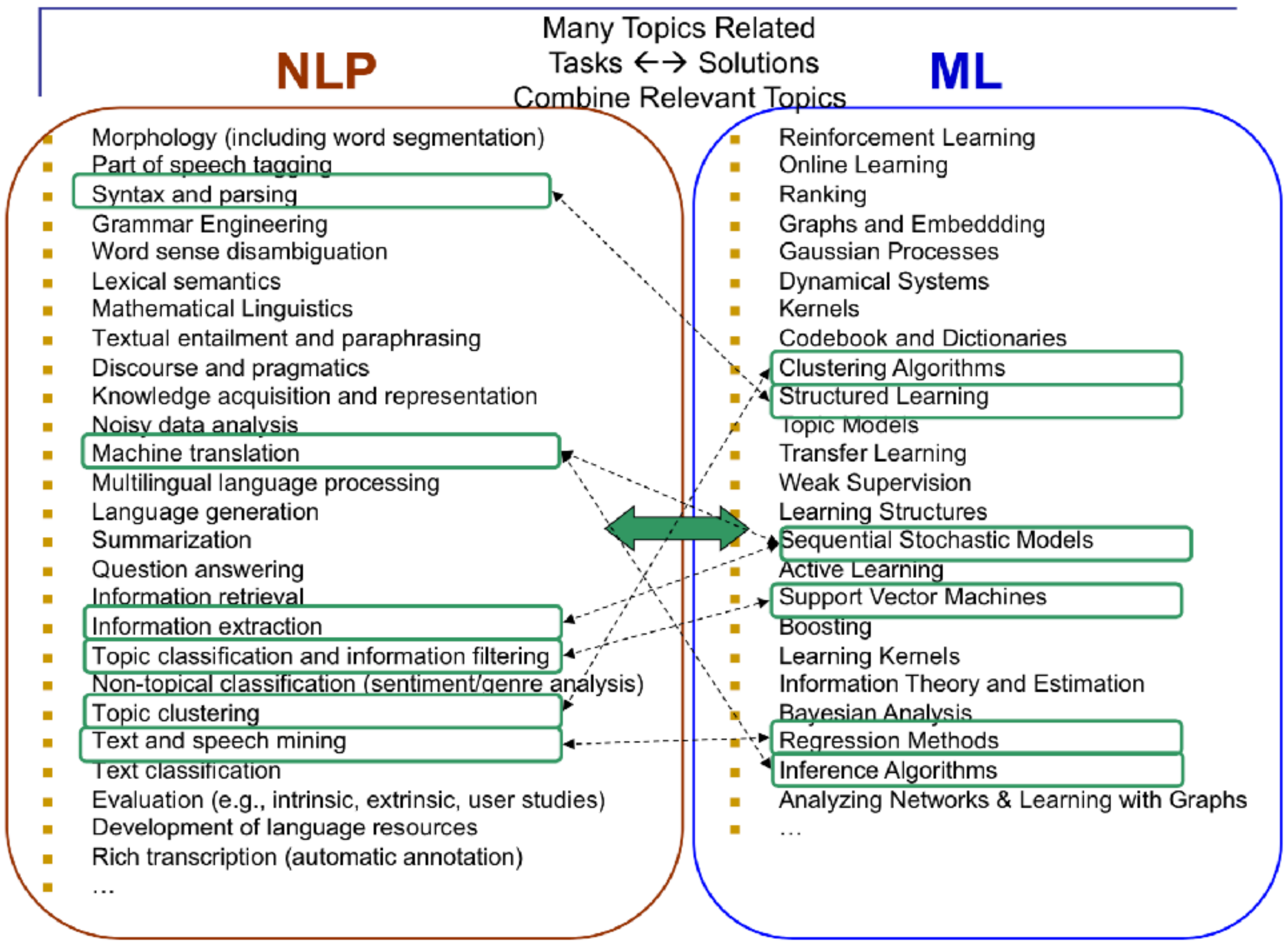
Nah, just kidding…

NLP Applications: Optical Character Recognition



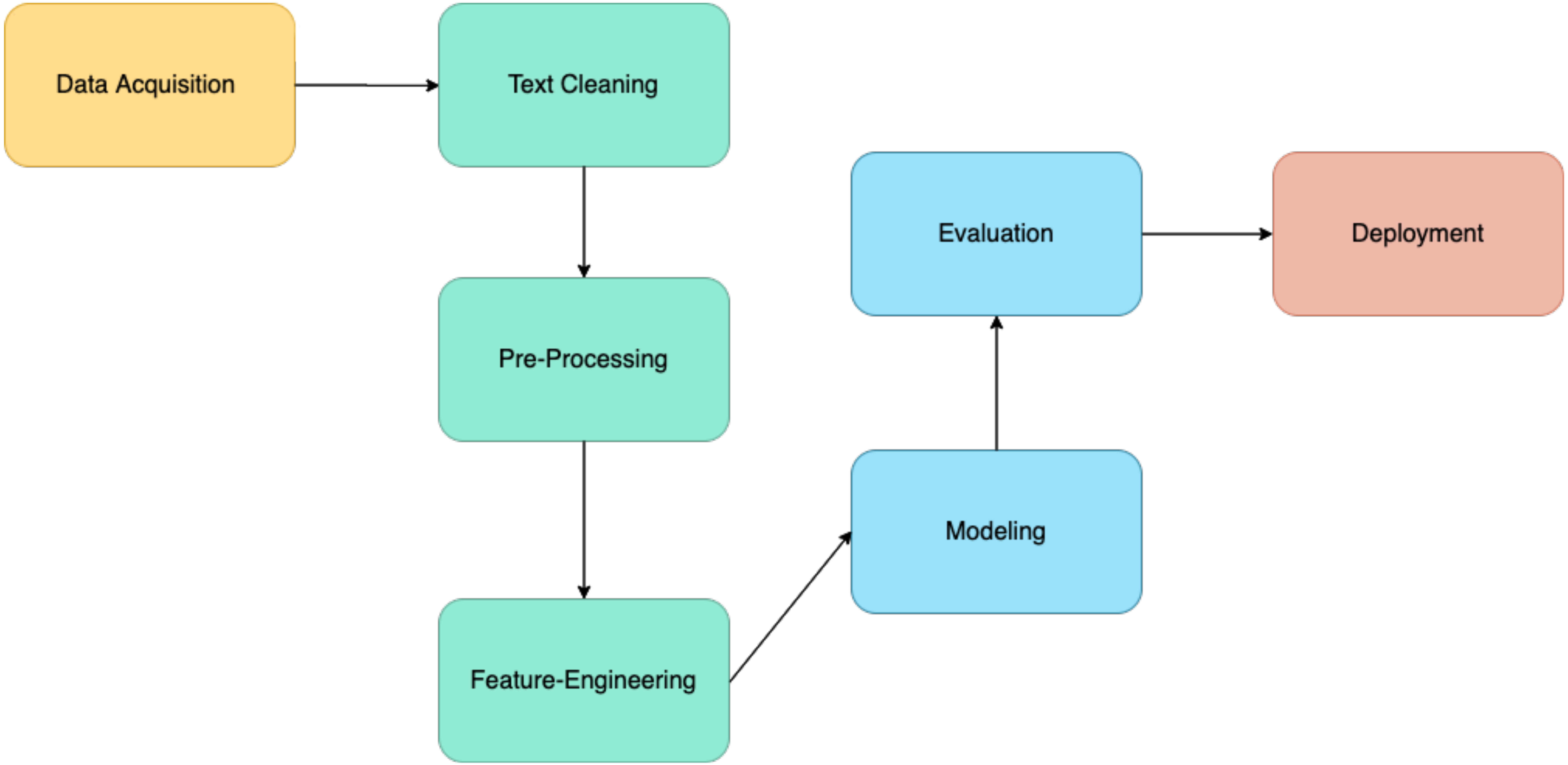
Source: osp.mans.edu.eg

NLP Challenges & ML Solutions



Source: Sameer Maskey, Columbia University

Sample NLP Pipeline



Sample NLP Pipeline

Capgemini invent | AI Garage

Input data → Textual data manipulation and transoformation → Analytical engine → Implementation

Unstructured texts

Examples:

- News
- Social media posts
- Letters from citizens
- Position papers of interest groups

Textual data manipualtion

Examples:

- OCR, From pictures to machine-readable texts
- Text cleansing, e.g., removal of html tags or footers, removal of language-specific stopwords (for instance „*a*“, „*the*“)
- Tokenization, e.g., selection of words out of a sentence

Textual data transformation

Examples:

- Feature generation, e.g., bag-of-words (each word as a variable)
- Feature reduction, e.g., through clustering of similar words

Analytics

Examples:

- Classification
- Sentiment analysis
- Clustering
- Topic Models
- Named Entity Recognition

Visualization/Reporting

Example:

- Dashboard for presenting the analytical insights with interactive visualization

Source: capgemini.com

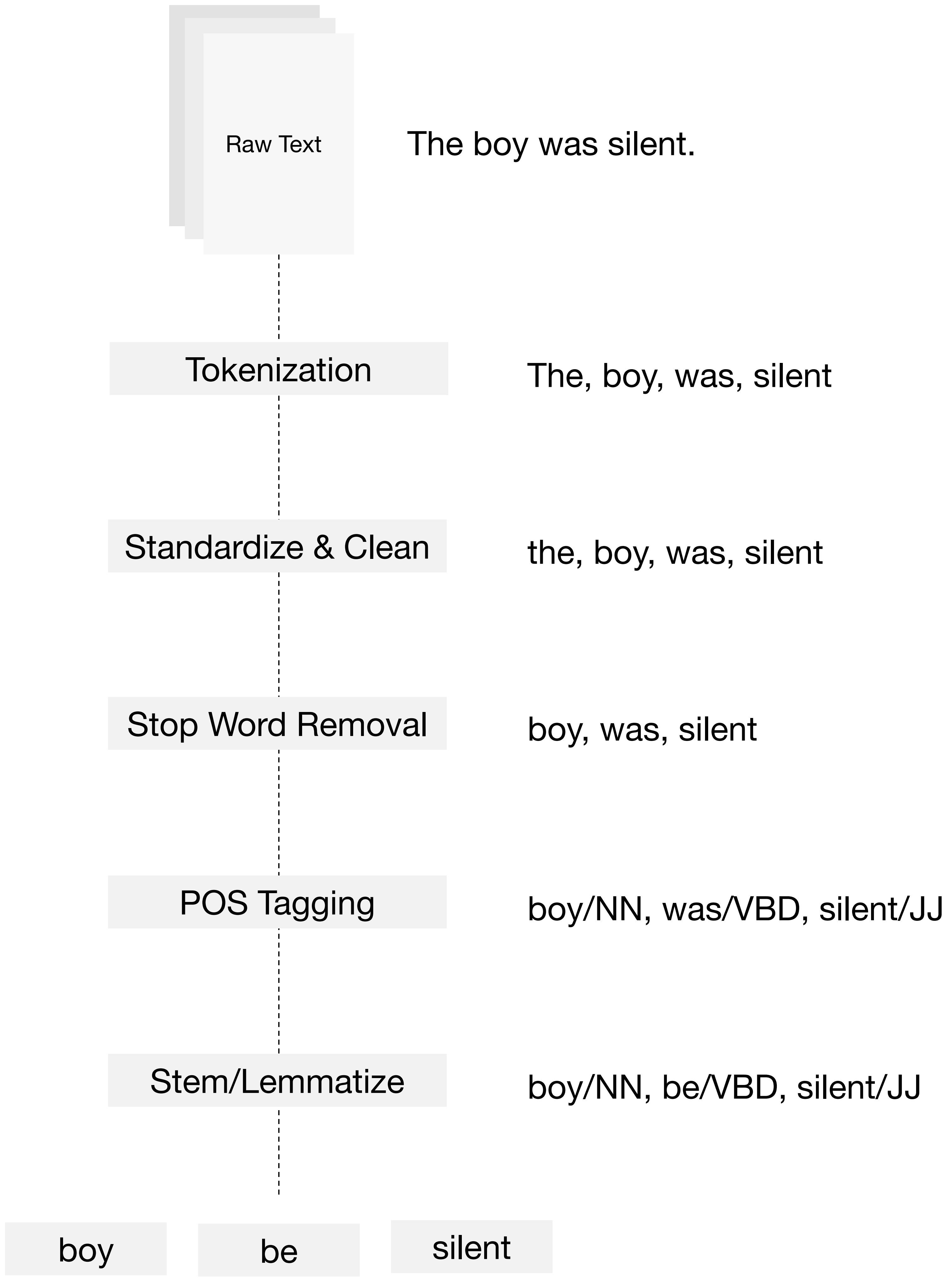
Information Retrieval: Web Scraping & Crawling

- **Scrapping:** The process of extracting structured information from a web page (A list of urls)
- **Crawler:** A computer program that automatically digs up information from the Internet

Source: blog.apify.com

NLP Techniques: Text-Document Processing

- Tokenizing texts (sentences), normalizing and cleaning text documents, deleting stop-words, NER and POS tagging stemming, lemmatization, etc.
- Stemming, prefixes / suffixes deletion, and punctuation removal may or may not be performed, depending on the task performed, so that the performance of the system does not decrease

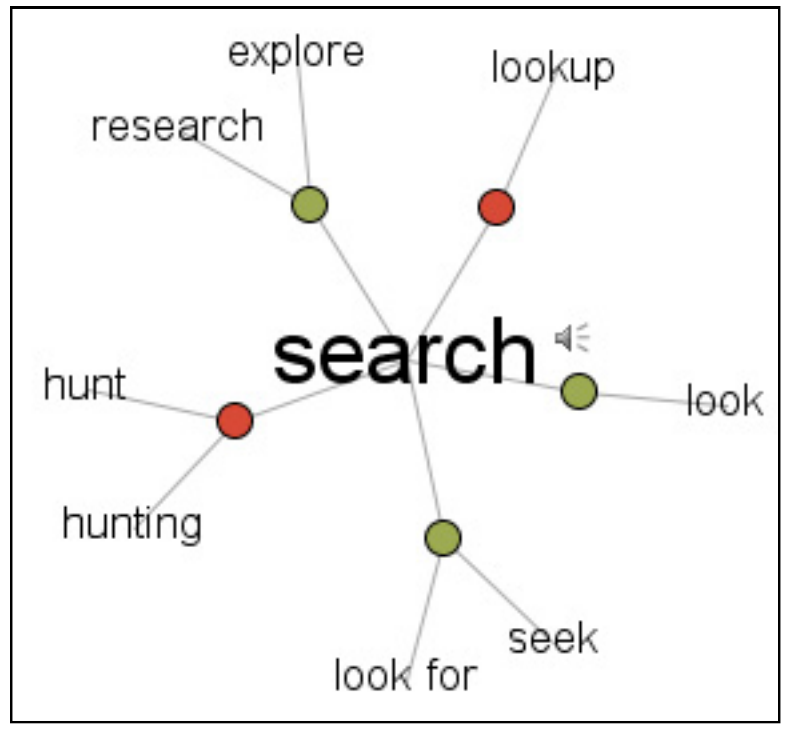


Text Corpus

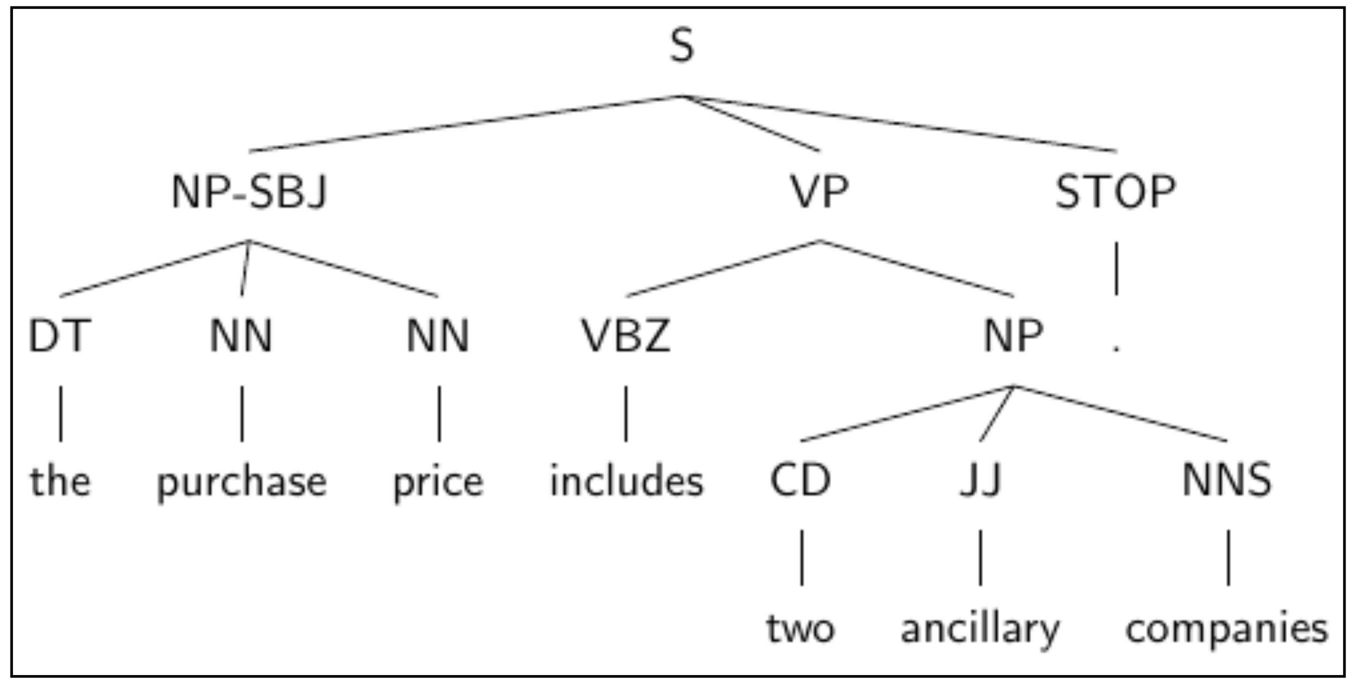
Machine-readable texts that have been produced, and probably processed, from authentic texts, such as books, newspapers, movie dialogue, etc.

Types of Corpus (in text processing tasks):

- **TreeBank Corpus:** Part-of-speech taggers, parsers, semantic analyzers and machine translation systems
- **PropBank Corpus:** Semantic role labeling
- **WordNet:** Word-sense disambiguation, word similarity, information retrieval, automatic text classification and machine translation



WordNet Graph



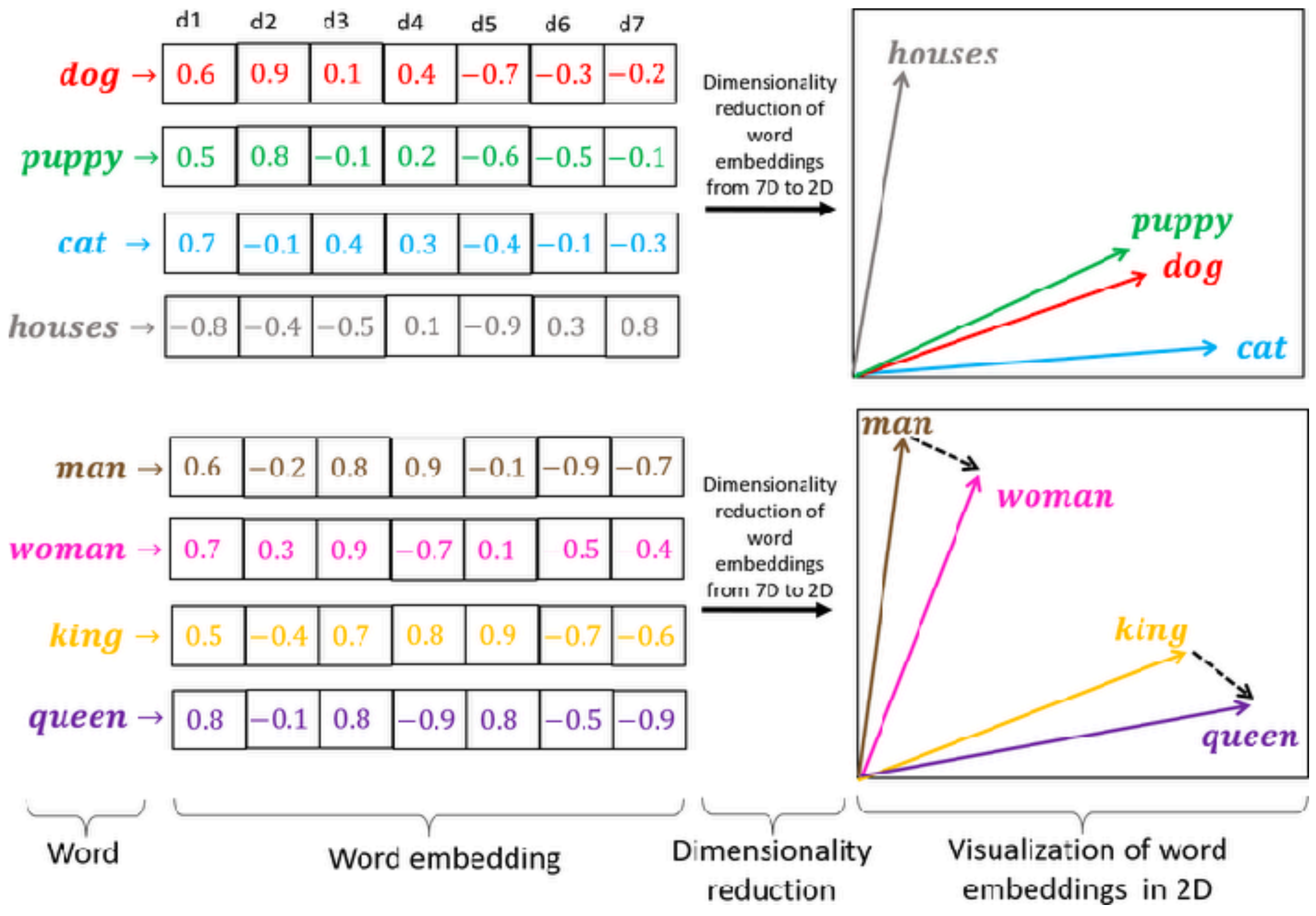
A TreeBank

Time	ArgM-TMP	In 2002,
Speaker –	Arg0 *(Informer)*	the U.S. State Department
Target –	REL	INFORMED
Addressee –	Arg1 *(informed)*	North Korea
Message –	Arg2 *(information)*	that the U.S. was aware of this program , and regards it as a violation of Pyongyang's nonproliferation commitments

A PropBank

Text (Word) Representation

- To work with texts, we have to convert them in a way that system can understand and work with it. We do it by mapping every possible word to a specific integer.
- **Techniques:**
  - One-hot Encoding
  - Token Counts
  - $w_{i,j} = tf_{i,j} \cdot log(\frac{N}{df_i})$ TF-IDF Vectorization
  - Vector representation (word embeddings)



Source: Wide Range Screening of Algorithmic Bias in Word Embedding Models Using Large Sentiment Lexicons Reveals Underreported Bias Types

Features	Converted Data					
Car	0	0	1	0	1	0
Laptop	1	0	1	0	0	1
Tower	0	0	0	0	1	1
Eat	1	1	0	0	0	0
University	0	1	1	0	0	0

One-hot Encoding Approach

Text (Word) Representation: Distributional Hypothesis

- **Distributional Hypothesis:** Words in similar context have similar meanings. So the meaning of a word is related to distribution of words around it.
- **Vector Semantics:** Distributional hypothesis + Vectors intuition
- **Word Embeddings:** A class of techniques where each word is mapped to a vector in predefined vector space. (Dense, i.e. 100-300 dimensions, in contrast to 100000+ dimensions in one-hot approach)
- A word embedding is a learned representation for text where words that have the same meaning have a similar representation, in one of the two form:
    - Words are expressed as vectors of co-occurring words
    - Words are expressed as vectors of linguistic contexts in which the words occur

Text (Word) Representation: Distributional Hypothesis

A term-document matrix

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	0	7	13
good	114	80	62	89
fool	36	58	1	4
wit	20	15	2	3

Source: Speech and Language Processing

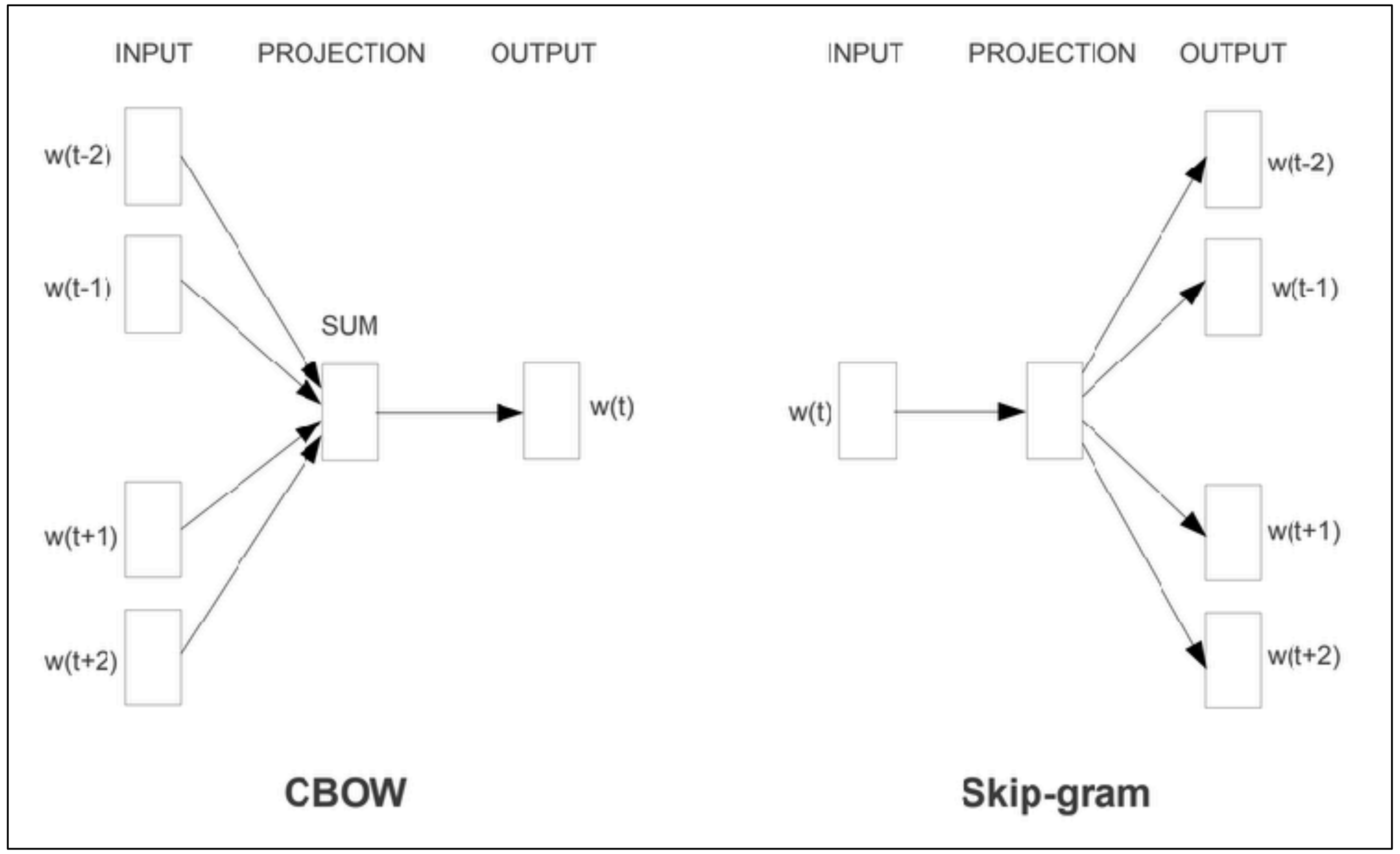
A term-term/term-context matrix

	aardvark	...	computer	data	pinch	result	sugar	...
apricot	0	...	0	0	1	0	1	
pineapple	0	...	0	0	1	0	1	
digital	0	...	2	1	0	1	0	
information	0	...	1	6	0	4	0	

Source: Speech and Language Processing

Text (Word) Representation: Word Embeddings

- It can be implemented in one of these ways:
  - **Embedding Layer:** Requires a large amount of text data
  - **Word2Vec:** Both methods are shallow neural networks which map word/words to the target variable (word/words). They both learn weights which act as word vector representations.
    - Continuous Bag-of-Words (CBOW model)
    - Continuous Skip-Gram Model
  - **GloVe**



Source: Efficient Estimation of Word Representations in Vector Space

Text (Word) Representation: Word Embeddings

- **Similarity Metrics:**
  - Cosine Similarity:
    - $cos(u, v) = \dfrac{u \cdot v}{||u \cdot v||}$
  - Distance Metrics (e.g. Euclidean distance):
    - $||u - v||^2$

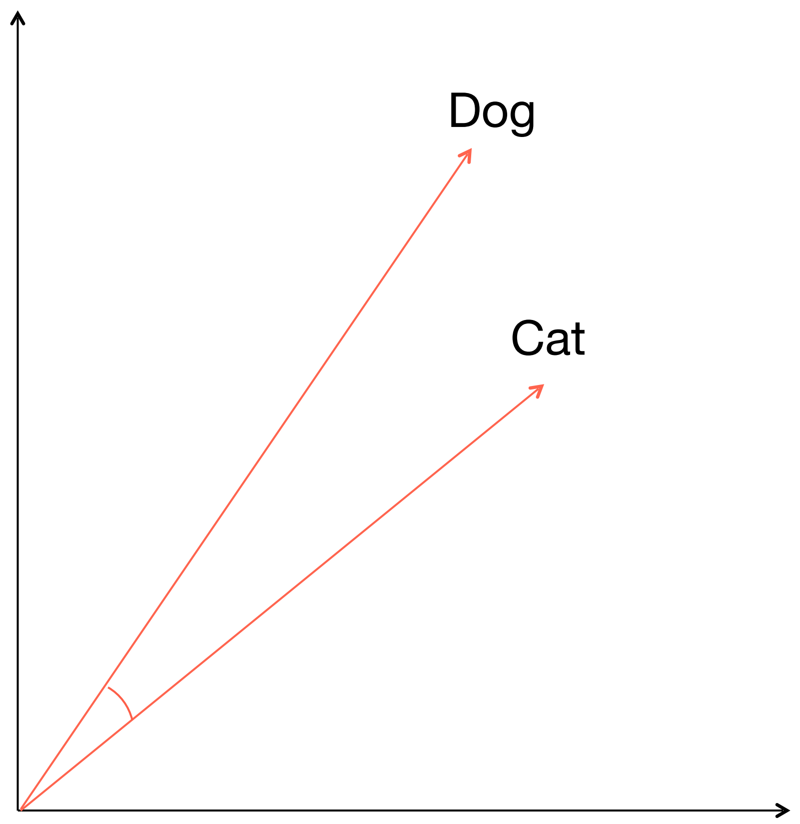
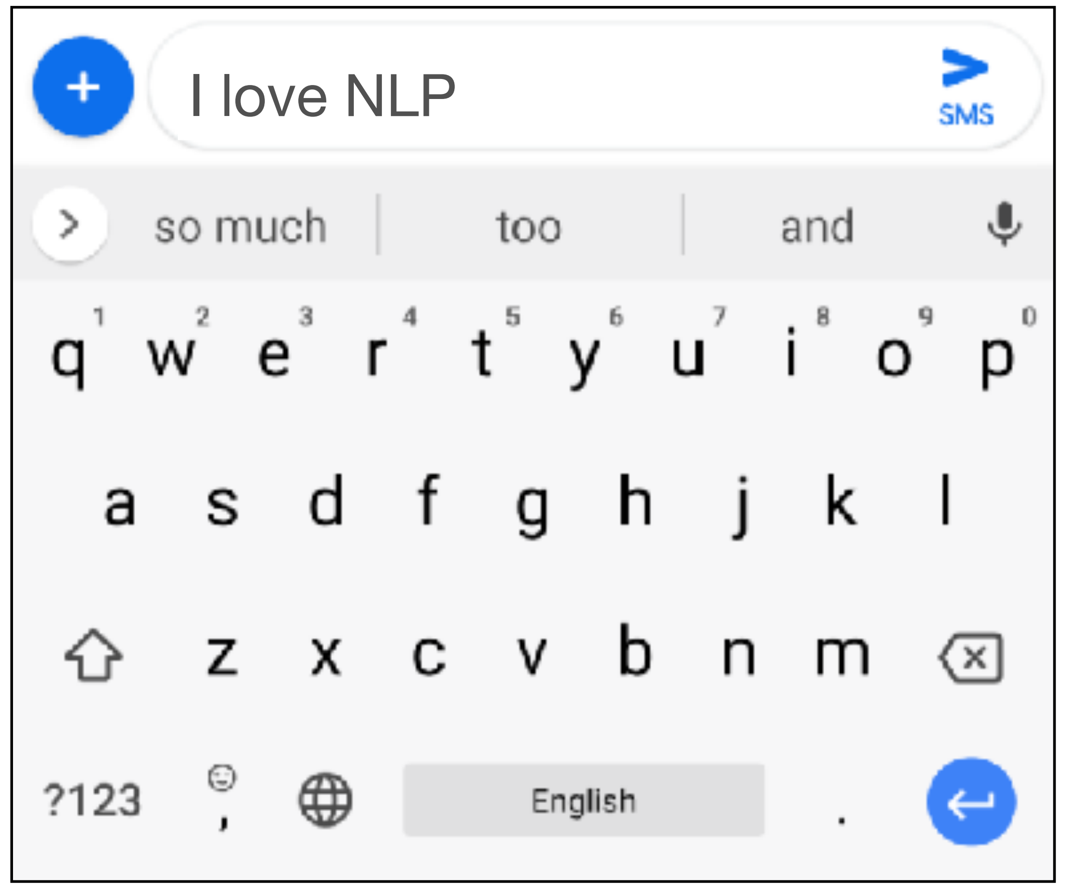


Figure 6.1 A two-dimensional (t-SNE) projection of embeddings for some words and phrases, showing that words with similar meanings are nearby in space. The original 60-dimensional embeddings were trained for sentiment analysis. Simplified from Li et al. (2015).

Source: Speech and Language Processing

Language Modeling

- The task of computing what word comes next (by computing the probability distribution over a sequence of words.)
- **Useful for:**
  - **Machine Translation:**
    - P(**high** winds tonight) > P(**large** winds tonight)
  - **Spelling Correction:**
    - The office is about fifteen minuets from my house
      - P(about fifteen **minutes** from) > P(about fifteen **minuets** from)
  - **Speech Recognition:**
    - P(I saw a van) >> P(eyes awe of an)
  - **Summarization, chat-bots, question-answering systems, etc.**



Language models used in next word prediction/suggestion

Language Modeling: Details

- Given a sequence of words $x^{(1)}, x^{(2)}, \ldots, x^{(t)}$ compute the probability distribution of the next word $x^{(t+1)}$ where $x^{(t+1)}$ can be any word in the vocabulary $V = \{w_1, \ldots, w_{|v|}\}$:
  - $P(x^{(t+1)} \mid x^{(t)}, \ldots, x^{(1)})$
- **Pre-Deep Learning:**
  - **N-grams**: A sequence of N words. Simplest model, yet works very well in many tasks including IR tasks.

    Unigram -> please, turn, the, machine, on
    $P(w_i) = C(w_i)/N, p(w_1) \cdot p(w_2) \cdot \ldots \cdot p(w_n)$
    *This one is actually a bag of words model

    Bigram -> please turn, turn the, the machine, machine on
    $$P(w_i \mid w_{i-1}) = \frac{C(w_{i-1}, w_i)}{C(w_{i-1})}, P(w_1) \cdot P(w_2 \mid w_1) \cdot P(w_3 \mid w_2) \cdot \ldots \cdot P(w_i \mid w_{i-1})$$

    Trigram -> please turn the, the machine on
    $$P(w_i \mid w_{i-2} w_{i-1}) = \frac{C(w_{i-2}, w_{i-1}, w_i)}{C(w_{i-2}, w_{i-1})}, P(w1) \cdot P(w2 \mid w1) \cdot P(w3 \mid w2,w1) \cdot \ldots \cdot P(w_i \mid w_{i-2} w_{i-1})$$

Named Entity Recognition

- Named Entity Recognition is essential for different NLP tasks to recognize information units like **names and person**, **organization**, **location**, **date** (or even drug names, genes, magazines, etc.)
- **Ambiguities:**
  - **Ambiguity of Segmentation:** Deciding what's an entity and what isn't, and where the boundaries are
  - **Ambiguity of Type:** For example *JFK* can refer to a person, the airport in New York, or any number of schools, bridges, and streets around the United States
- **Algorithms:**
  - Rule-based Methods
  - Sequence Labeling Methods
  - feature-based Methods

In a statement issued with France and UN chief António Guterres on Saturday, China committed to "update" its climate target "in a manner representing a progression beyond the current one". It also vowed to publish a long term decarbonization strategy by next year.

Name | Group | Place

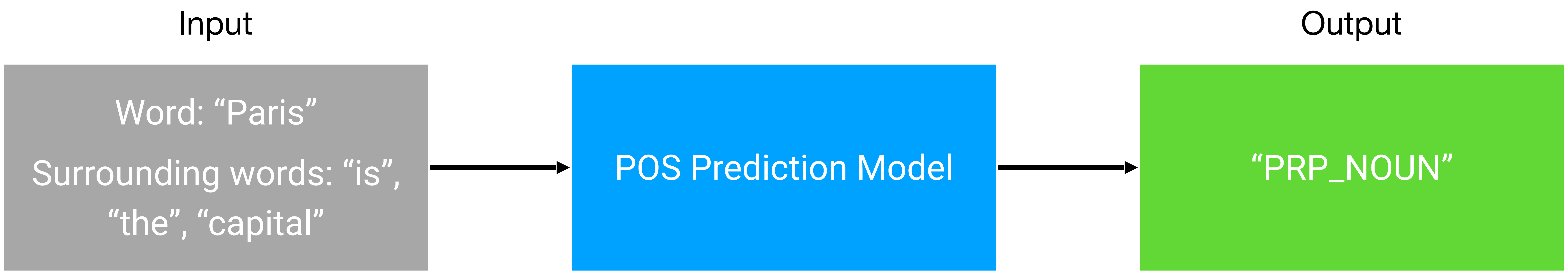
Source: paralleldots.com

POS Tagging

Corresponding a **word** to a part of **speech tag,** based on its *context* & *definition*.

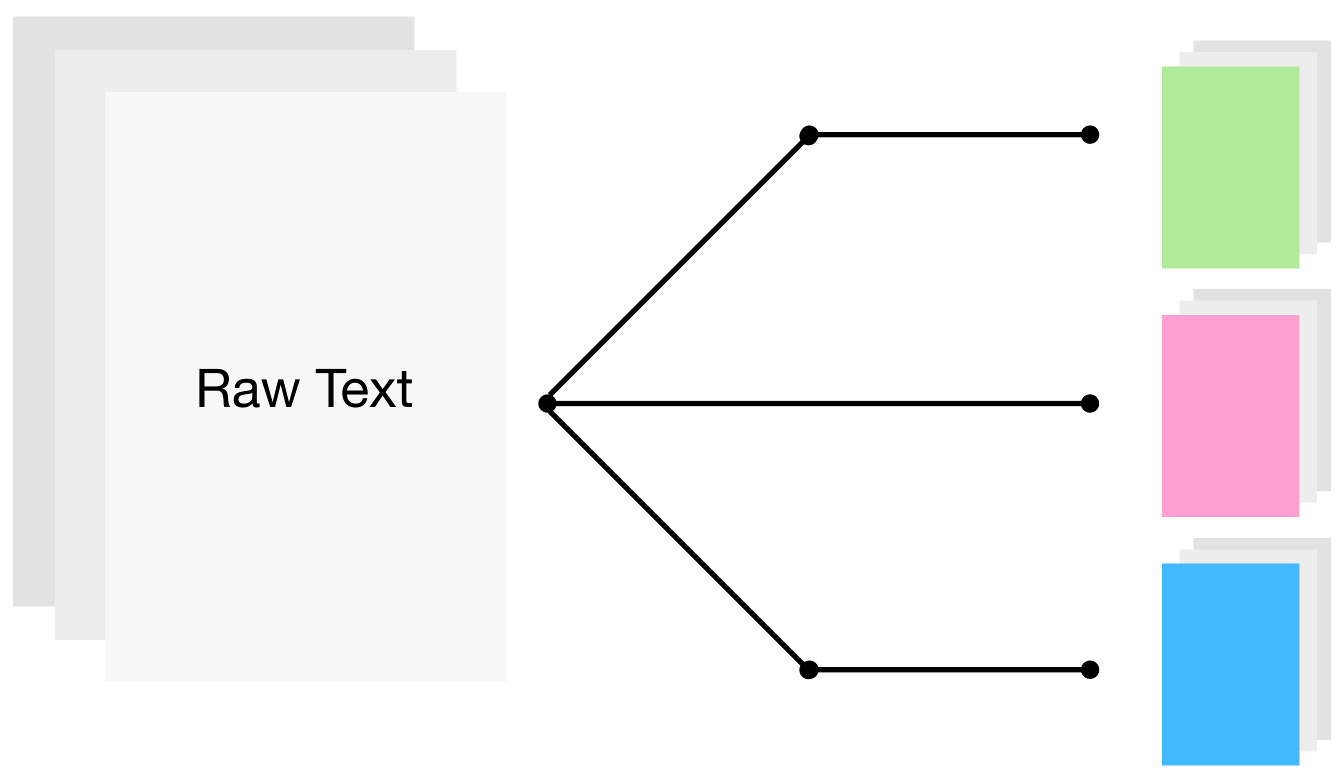
PRP VBD IN DT NN

I walked by the seashore



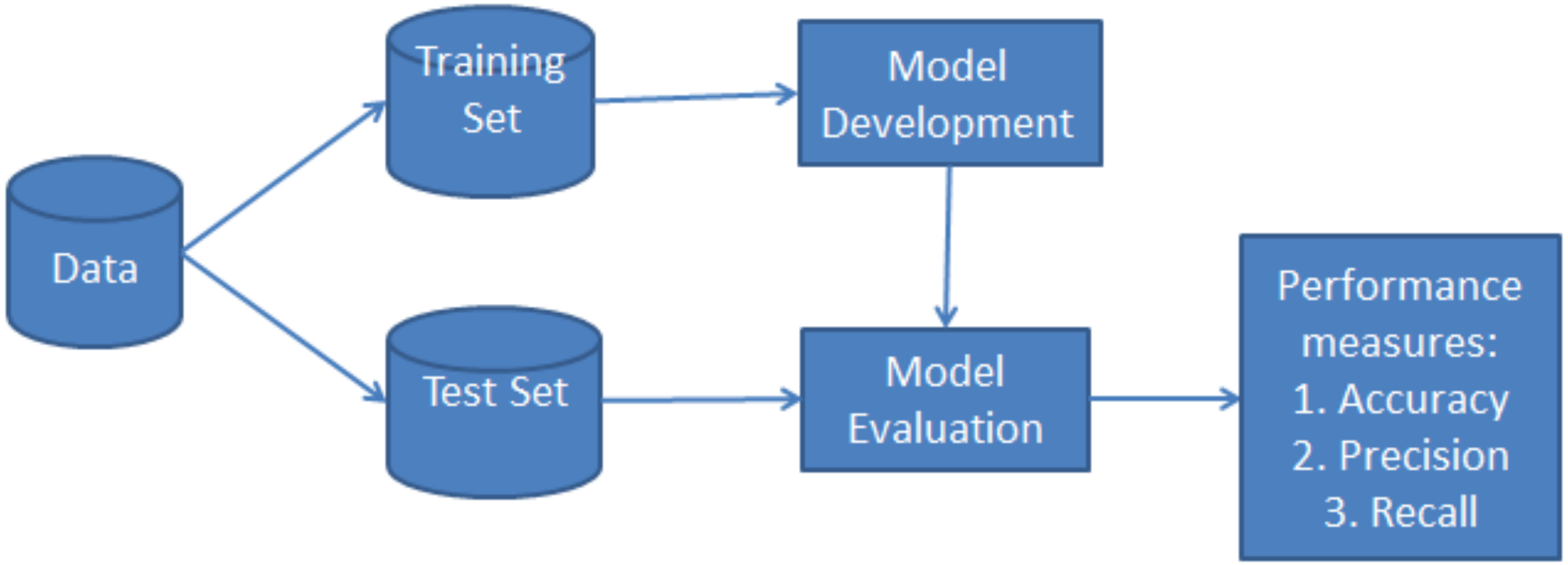
Classification

- Classifying a given input to an output (given input x, predict y from $Y = y_1, y_2, \ldots, y_n$). It could be a text, sentence or even a single word.
  - **Sentiment Analysis:** Classifying user comments about a product (Satisfied or not satisfied or even a range of values like Very satisfied to extremely dissatisfied)
  - **Spam Detection:** Filtering emails based on their contents
  - Even **POS tagging**, **NER** and **language modeling** can be seen as classifying tasks.



Classification: Types of Classifiers

- **Generative Classifiers:** Build classes that can generate some input data. Given an observation, they predict the class which has most likely generated that observation.
- **Discriminative Classifiers:** Learn features from input which are most useful in discriminating between different classes. More accurate than generative ones.

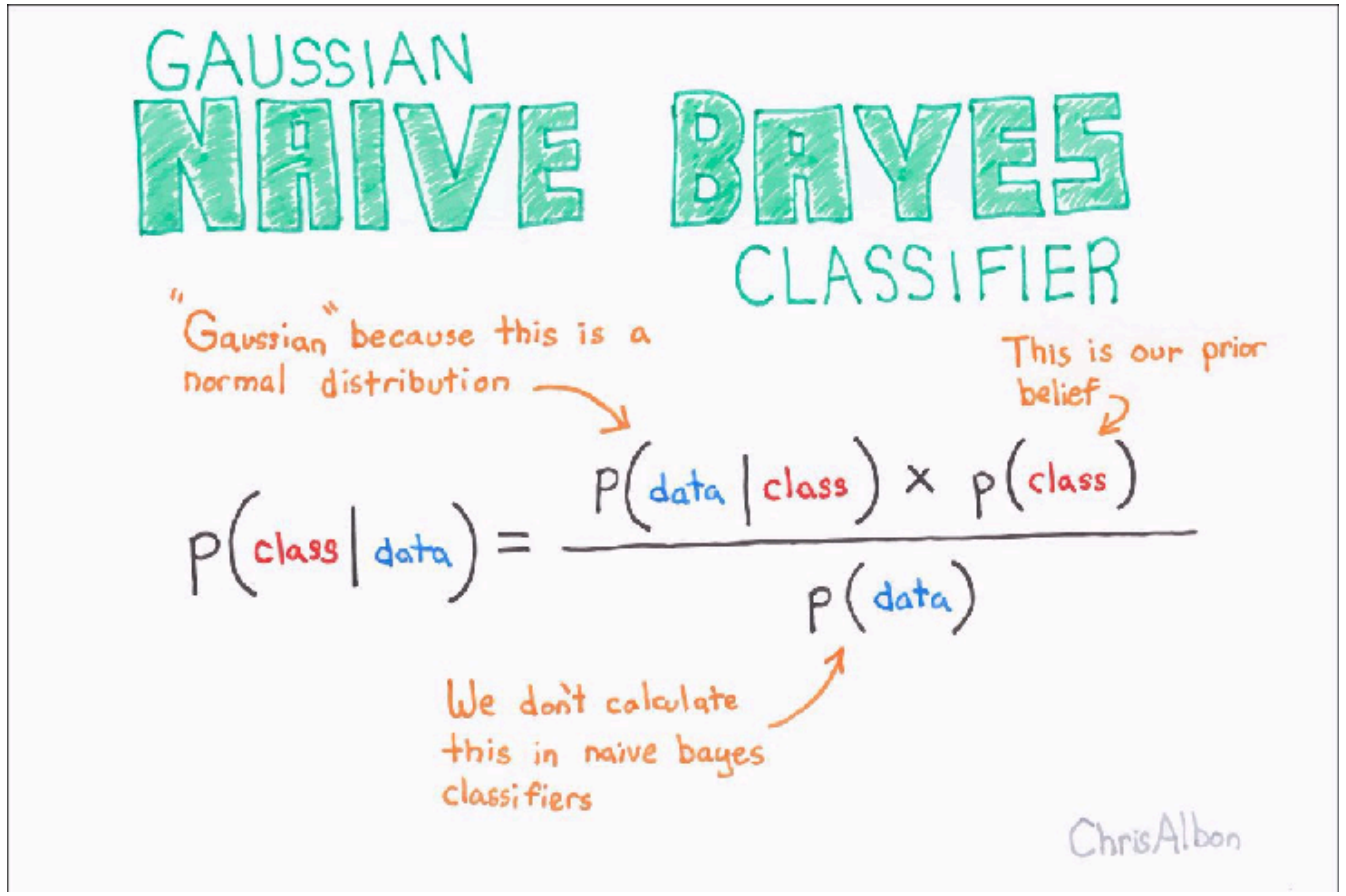


Source: datacamp.com

Classification: Naïve Bayes

- **Naïve Bayes:** A generative, probabilistic classifier based on Bayes theorem, though with the assumption that features are independent from each other.

$$c = argmax\, P(c\,|\,d) = argmax\, P(d\,|\,c) \cdot P(c) = argmax\, P(c) \prod P(w\,|\,c)$$



Source: towardsdatascience.com

Classification: Logistic Regression

- **Logistic Regression:** A discriminative classifier that tries to directly compute $P(c \mid d)$.
- **Predicting a new observation:**

$$P(c1) = \alpha(w \cdot x + b), P(c0) = 1 - \alpha(w \cdot x + b) \text{ \#sums to 1}$$

$$\alpha = \frac{1}{1 + e^{-(w.x+b)}}$$

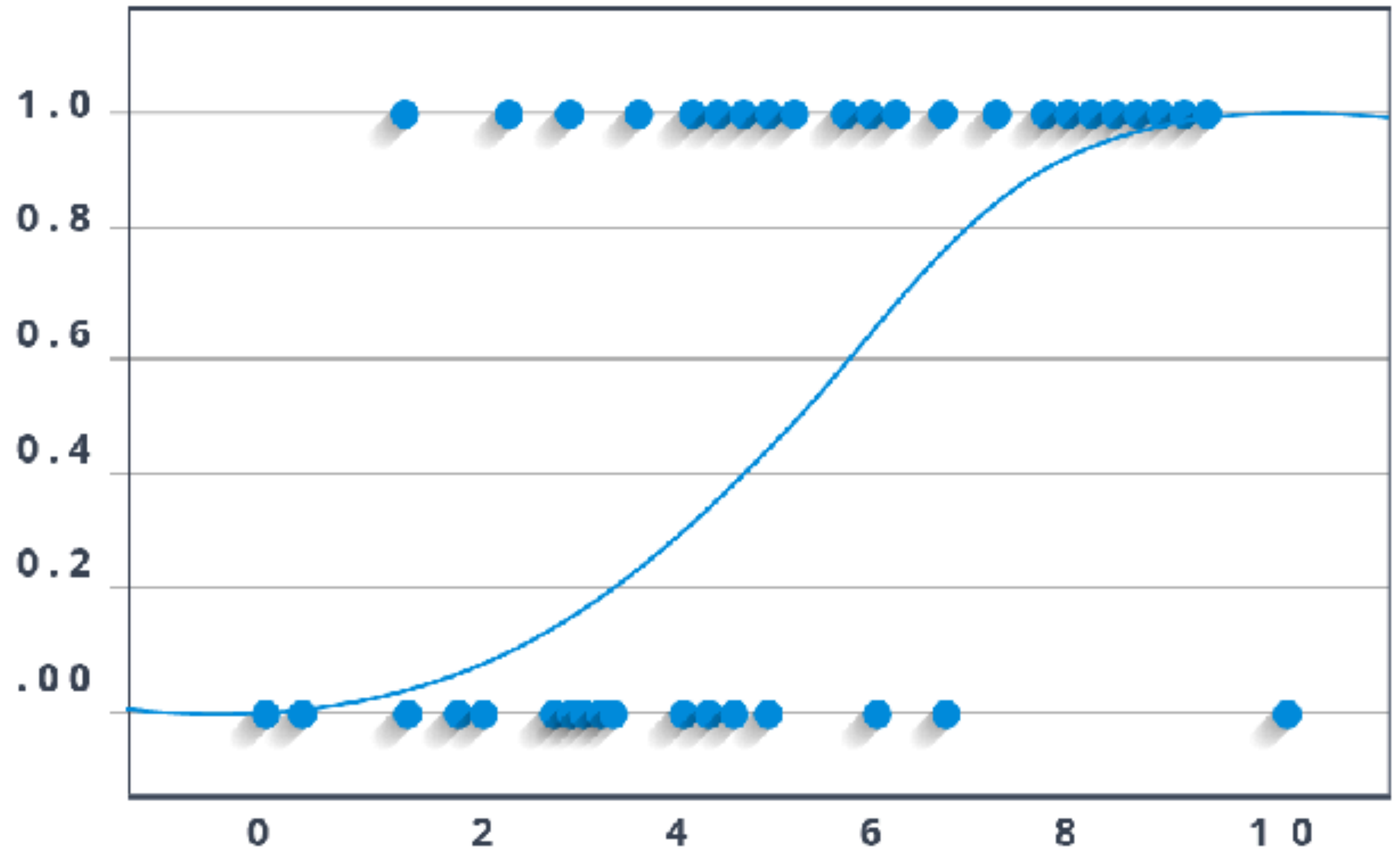
- **Cost function:** Cross-entropy loss

$$Lce(w,\ b) = -\left[y\log\alpha(w\,.\,x + b) + \left(1 - y\right)\log(1 - \alpha(w\,.\,x + b))\right]$$

- **Optimizing cost function:** Using gradient descent to find the optimal weights/minimizing the loss.

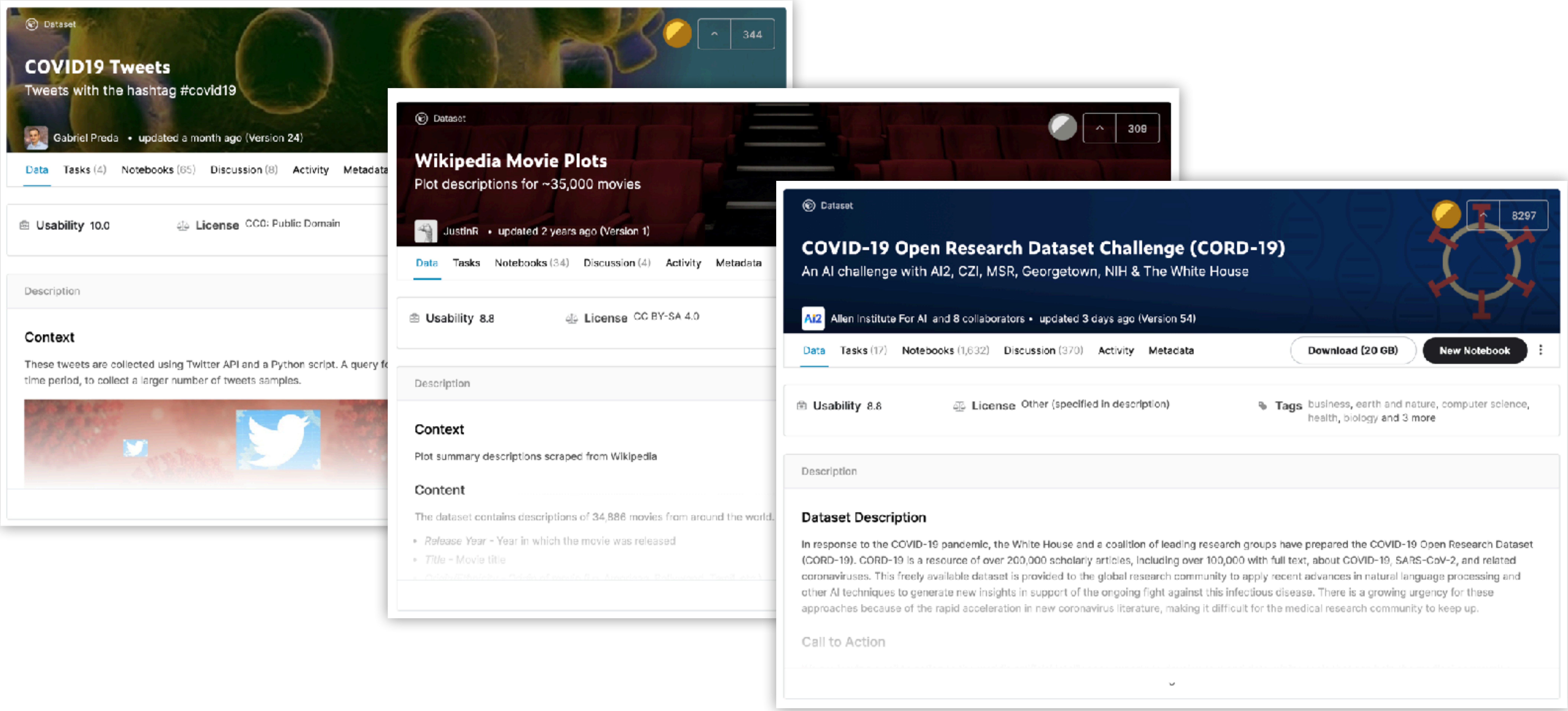
$$\theta t = \theta t - \eta \nabla L(f(x;\theta), y)$$

$$\frac{\partial LCE(w,b)}{\partial wj} = [\sigma(w\,.\,x + b) - y]x_j$$



Source: towardsdatascience.com

Kaggle Data-sets



Libraries & Toolkits: NLTK

- Install:
  - `pip install nltk`
- Usage:
  - import nltk

```
sentence = """At eight o'clock on Thursday morning
... Arthur didn't feel very good."""
tokens = nltk.word_tokenize(sentence)
tokens = ['At', 'eight', "o'clock", 'on', 'Thursday', 'morning',
'Arthur', 'did', "n't", 'feel', 'very', 'good', '.']
tagged = nltk.pos_tag(tokens)
tagged[0:6] = [('At', 'IN'), ('eight', 'CD'), ("o'clock", 'JJ'), ('on', 'IN'),
('Thursday', 'NNP'), ('morning', 'NN')]
```

Libraries & Toolkits: spaCy

- Install:
  - `pip install spacy`
- Usage:
  - import spacy
  - `nlp = spacy.load(“en_core_web_sm”)`

```
text = ("When Sebastian Thrun started working on self-driving cars at "
        "Google in 2007, few people outside of the company took him "
        “seriously.”)
doc = nlp(text)
for entity in doc.ents:
    print(entity.text, entity.label_)
Sebastian Thrun (PERSON), Google (ORG) 2007 (DATE)
```

Libraries & Toolkits: Gensim

- Install:
  - `pip install --upgrade gensim`
- Usage:
  - from gensim.summarization import summarize

```
text = (
    "Thomas A. Anderson is a man living two lives. By day he is an "
    "average computer programmer and by night a hacker known as "
    "Neo. Neo has always questioned his reality, but the truth is "
    "far beyond his imagination. Neo finds himself targeted by the "
    "police when he is contacted by Morpheus, a legendary computer "
    "hacker branded a terrorist by the government. Morpheus awakens "
    "Neo to the real world, a ravaged wasteland where most of "
    "humanity have been captured by a race of machines that live "
    "off of the humans' body heat and electrochemical energy and "
    "who imprison their minds within an artificial reality known as "
    "the Matrix. As a rebel against the machines, Neo must return to "
    "the Matrix and confront the agents: super-powerful computer "
    "programs devoted to snuffing out Neo and the entire human "
    "rebellion. "
)
print(summarize(text))
('Morpheus awakens Neo to the real world, a ravaged wasteland where most of '
 'humanity have been captured by a race of machines that live off of the '
 "humans' body heat and electrochemical energy and who imprison their minds "
 'within an artificial reality known as the Matrix.')
```

Libraries & Toolkits: Hazm

- Install:
  - `pip install hazm`
  - `sudo apt install python-pip # Windows`
- Usage:
  - `from hazm import *`
  - `normalizer = Normalizer()`
  - `sentence = 'اصلاح نویسه ها و استفاده از نیم‌فاصله پردازش را آسان می کند'`
  - `normalizer.normalize(sentence) => 'اصلاح نویسه‌ها و استفاده از نیم‌فاصله پردازش را آسان می‌کند'`
  - `tagger = POSTagger(model='resources/postagger.model')`
  - `tagger.tag(word_tokenize('ما بسیار کتاب می‌خوانیم'))`
  - [('ما', 'PRO'), ('بسیار', 'ADV'), ('کتاب', 'N'), ('می‌خوانیم', 'V')]

Resources for Further Reading

- **Books:**
  - Daniel Jurafsky, and James Martin, Speech and Language Processing, 3rd edition draft, 2019.
  - Manning, C. D., & Schütze, H. (1999). Foundations of statistical natural language processing. MIT Press.
  - J. Eisenstein. Introduction to Natural Language Processing. MIT Press, 2019.
  - James, G., Witten, D., Hastie, T., & Tibshirani, R. (Eds.). (2013). An introduction to statistical learning: With applications in R. Springer.
- **Courses:**
  - DeepLearningAI: Natural Language Processing
  - CS224n: Natural Language Processing with Deep Learning
  - COMS W4705: Natural Language Processing

References

- Daniel Jurafsky, and James Martin, Speech and Language Processing, 3rd edition draft, 2019.
- nlp.stanford.edu/IR-book/
- towardsdatascience.com (Various articles)
- web.stanford.edu/class/cs124/
- Rozado, D. (2020). Wide range screening of algorithmic bias in word embedding models using large sentiment lexicons reveals underreported bias types. PLOS ONE, 15(4), e0231189. https://doi.org/10.1371/journal.pone.0231189
- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. ArXiv:1301.3781 [Cs]. http://arxiv.org/abs/1301.3781

Q&A