Deloitte.

Deloitte Data Science Academy

Day 4:

Natural Language Processing

13/10/2022

Natural Language Processing

Natural language processing (NLP) is a subfield of linguistics, computer science, and artificial intelligence concerned with enabling computers to **understand human languages**.

The terms **text analytics**, **text mining**, or **unstructured data analytics** are also commonly used.

NLP is an integral part of Artificial Intelligence

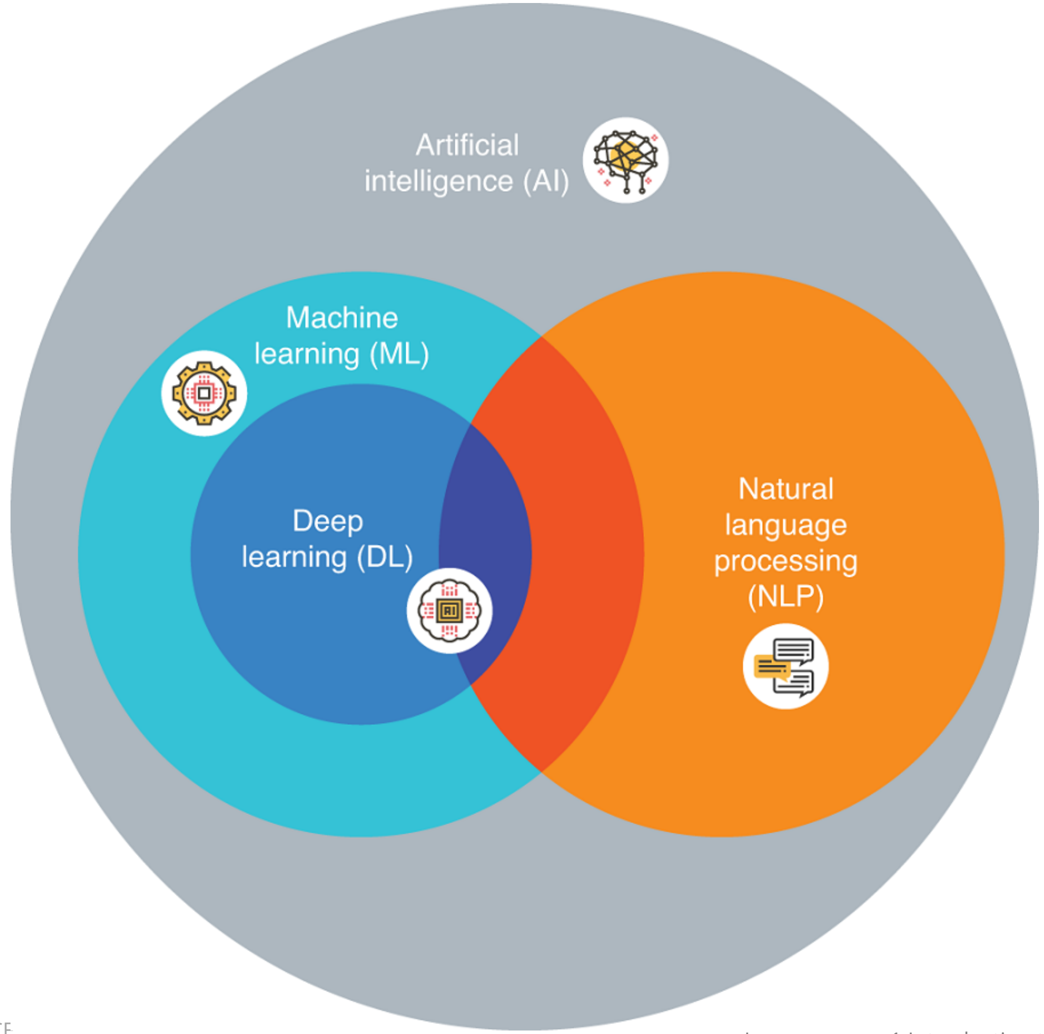


Image source: [1 Introduction to natural language processing - Real-World Natural Language Processing (manning.com)]

NLP is an integral part of Artificial Intelligence

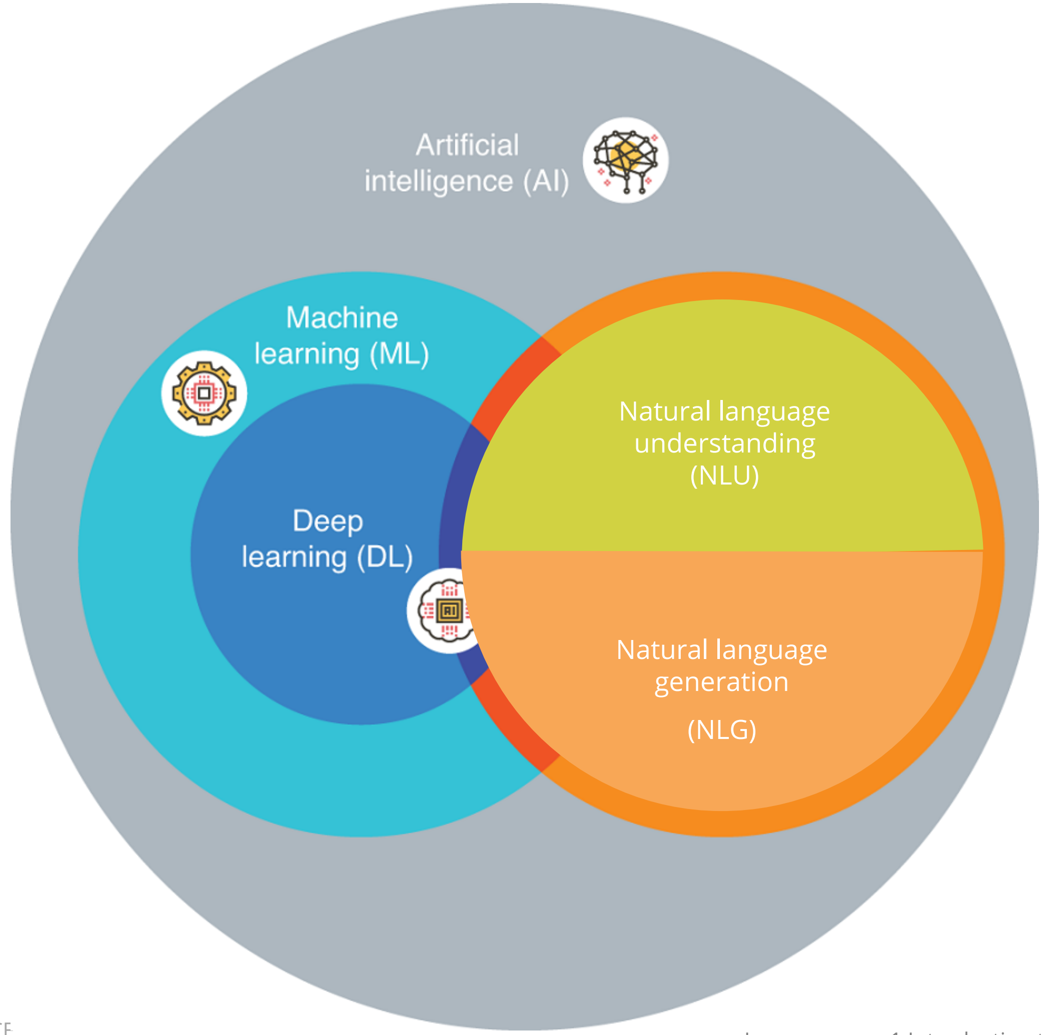
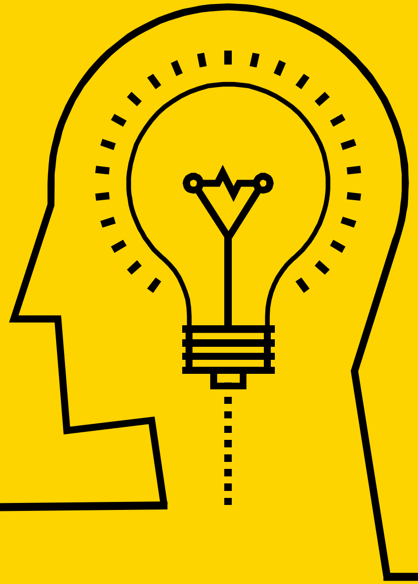


Image source: [1 Introduction to natural language processing - Real-World Natural Language Processing (manning.com)]

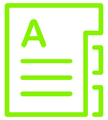
“A computer could be considered intelligent if it could carry out a conversation with a human being without the human realizing they were talking to a machine.”

– Alan Turing



Where can you meet NLP?

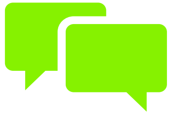
E-mail classification & filtering



Spell checker & auto-correct

Search engines

Understanding consumer feedback & media monitoring



Chatbots & voicebots

Translation

Data sources used in NLP

- Transaction notes
- Client's emails
- Call centre calls
- Logs
- Internal documents
- User reviews
- Online discussions
- Social networks

Working with individual words

Working with text – representing words

- For computer processing, it is necessary to **translate** (unstructured) **words into** (structured) **numerical values**.

- Most intuitive way is to create embeddings - a **word embedding** is a learned representation for text typically in the form of a real-valued **vector that encodes the meaning** of the word such that the words that are closer in the vector space are expected to be similar in meaning

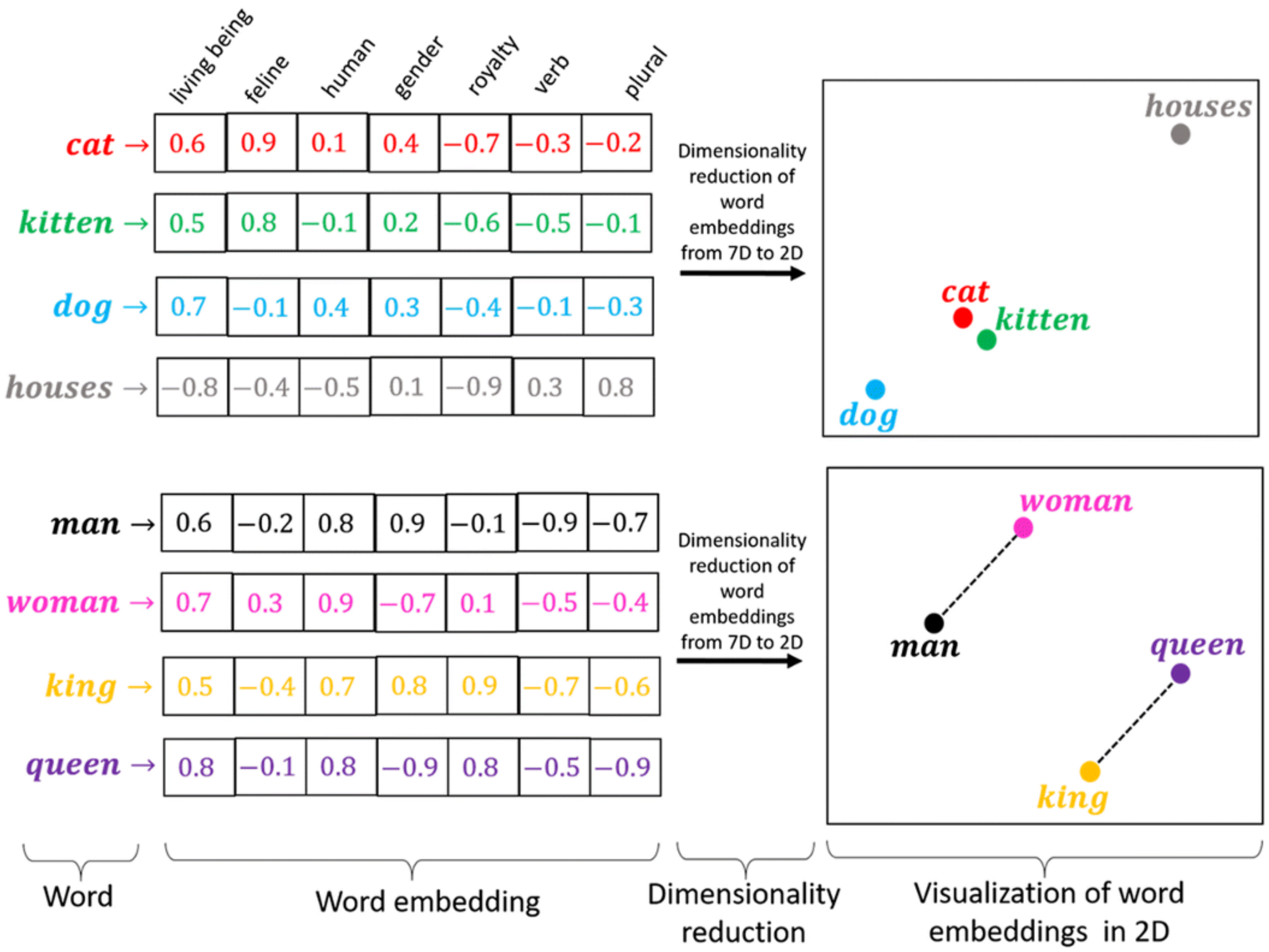
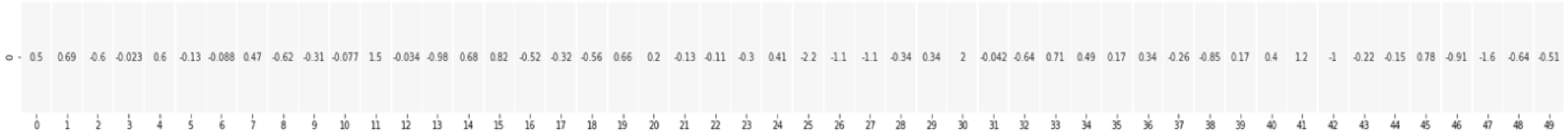


Image source: Word Embedding: Basics. Create a vector from a word | by Hariom Gautam | Medium

Working with text – representing words

“king”



Models like Word2Vec or GloVe: generating embeddings to group words by semantic similarity

“king”

“Man”

“Woman”

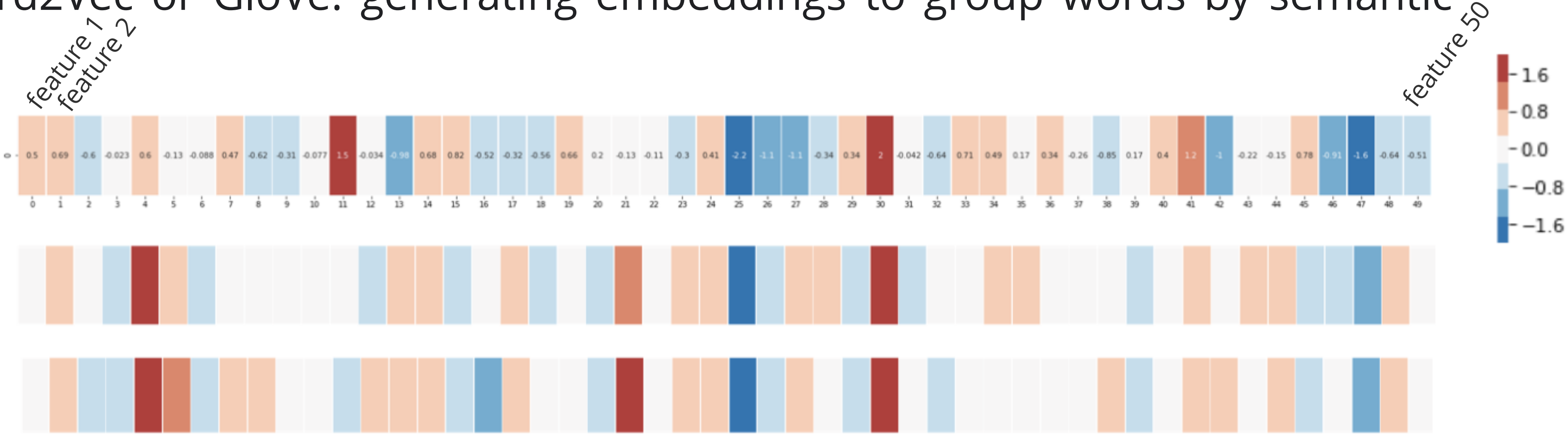


Image source: The Illustrated Word2vec – Jay Alammar

Working with text – representing words

Word embeddings allow for arithmetic operations with words

king – man + woman ~= queen

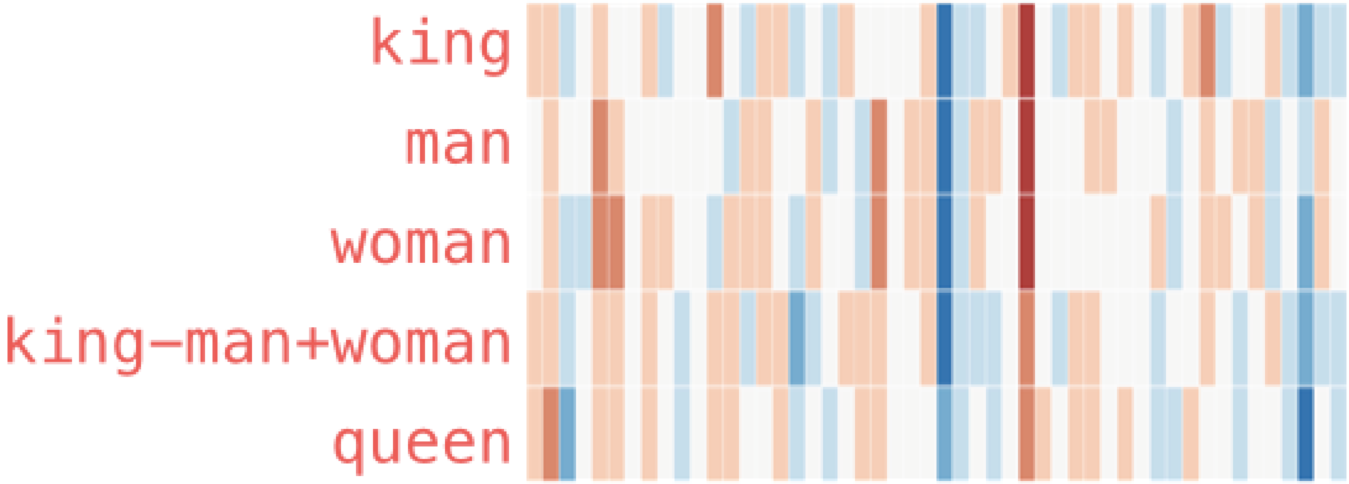


Image source: The Illustrated Word2vec – Jay Alammar

The resulting vector from "king-man+woman" doesn't exactly equal "queen", but "queen" is the closest word to it from the 400,000 word embeddings we have in this collection.

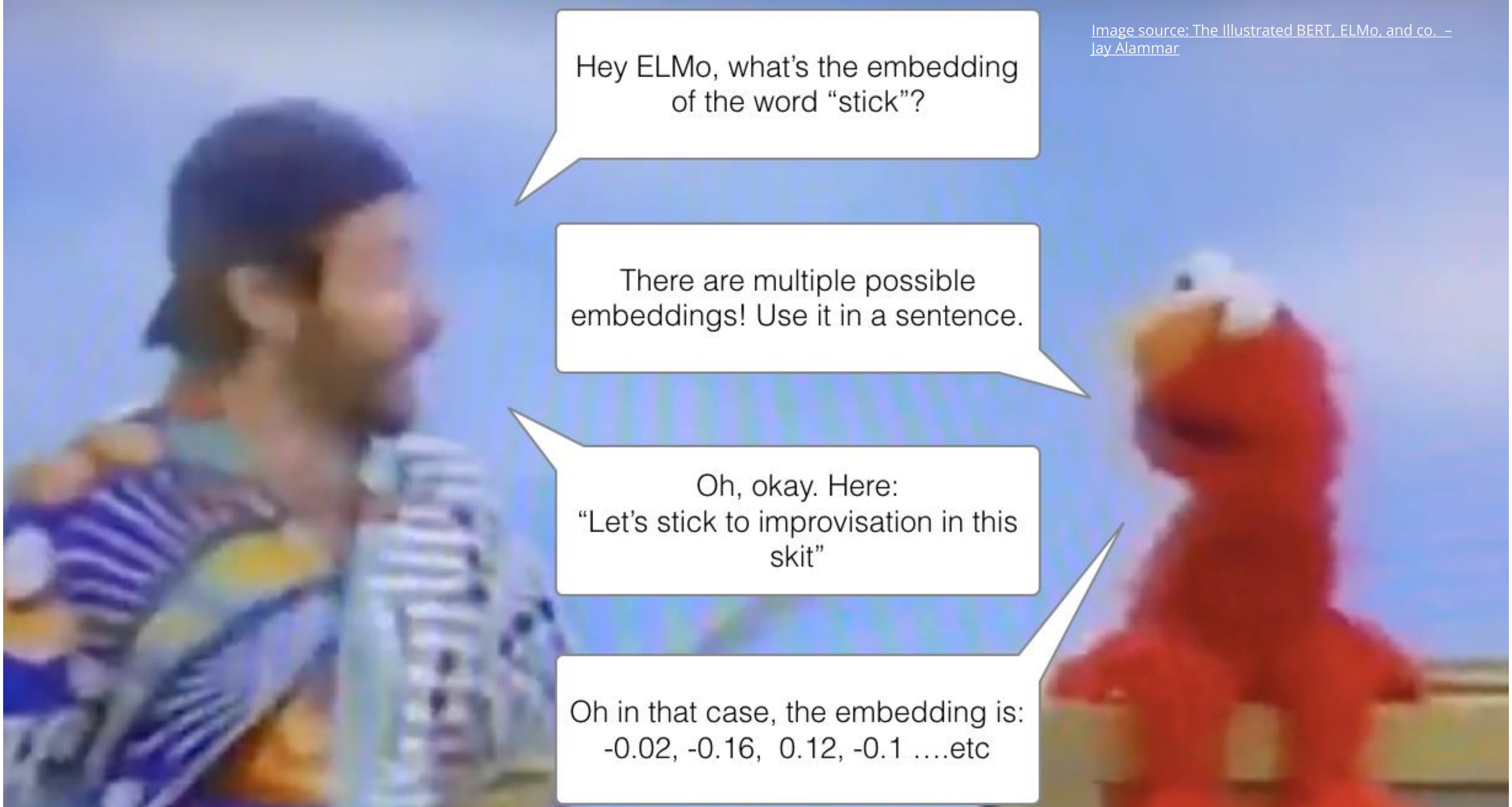


Image source: The Illustrated BERT, ELMo, and co. – Jay Alammar

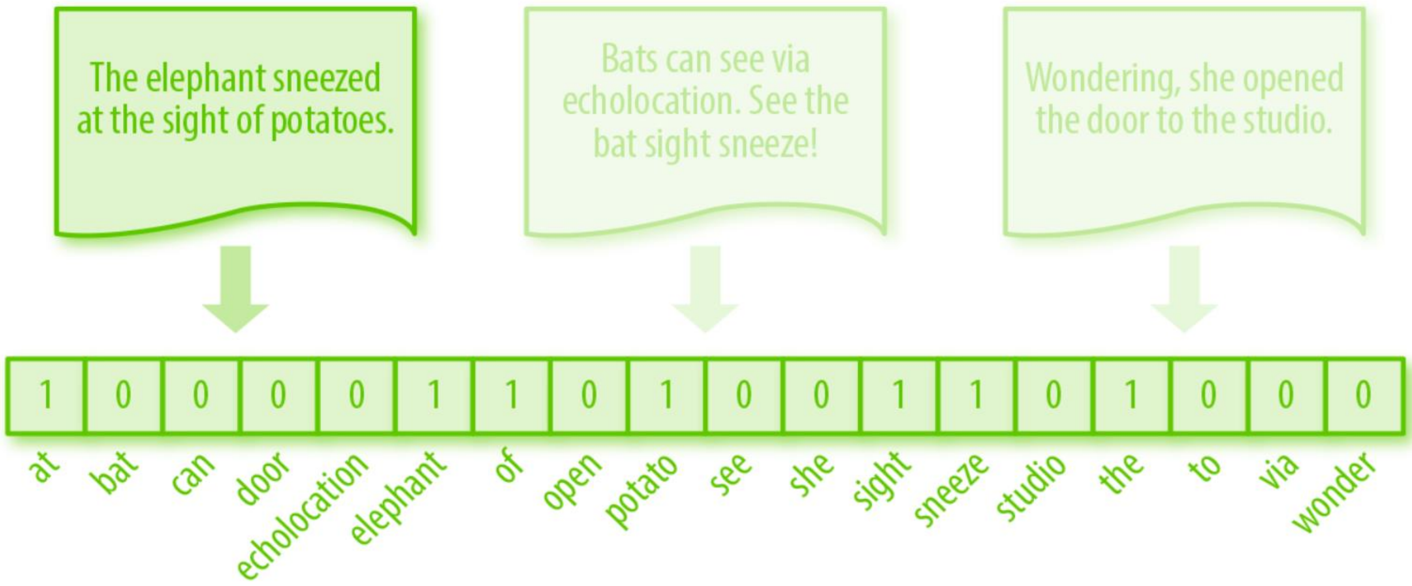
ELMo – a first language model providing contextualized embeddings, trained bi-directionally on sequences of words

Working with documents

Working with text – representing whole documents

Vectorization of documents

- ➢ represent each text document with a **structured numeric vector** (e.g binary vectorization)



Problem of **excessive dimensionality**, as each form of a word or phrase represents a dimension

- ➢ problem eliminated by using NLP techniques for **text pre-processing**

Text pre-processing

Main goals:

- Represent each text document with a **list of relevant terms/tokens**
- **Dimensionality reduction** of the original documents

Sentence Segmentation

Breaking text into a list of sentences

Word Tokenization

Breaking sentences into a list of words/tokens

Stopwords & Punctuation Removal

Elimination of meaningless words/tokens

Normalization

Lemmatization or Stemming: conversion of words to their base forms

(Possible other steps: language detection, spelling correction, emojis and special characters handling)

Document representation: Bag of Words

A **bag of words (BoW)** is a representation of text that **describes the occurrence of words** within a document.

- Disregarding grammar (-> normalizing words to e.g. lemmas) and order
- Keeping track of occurrence frequency

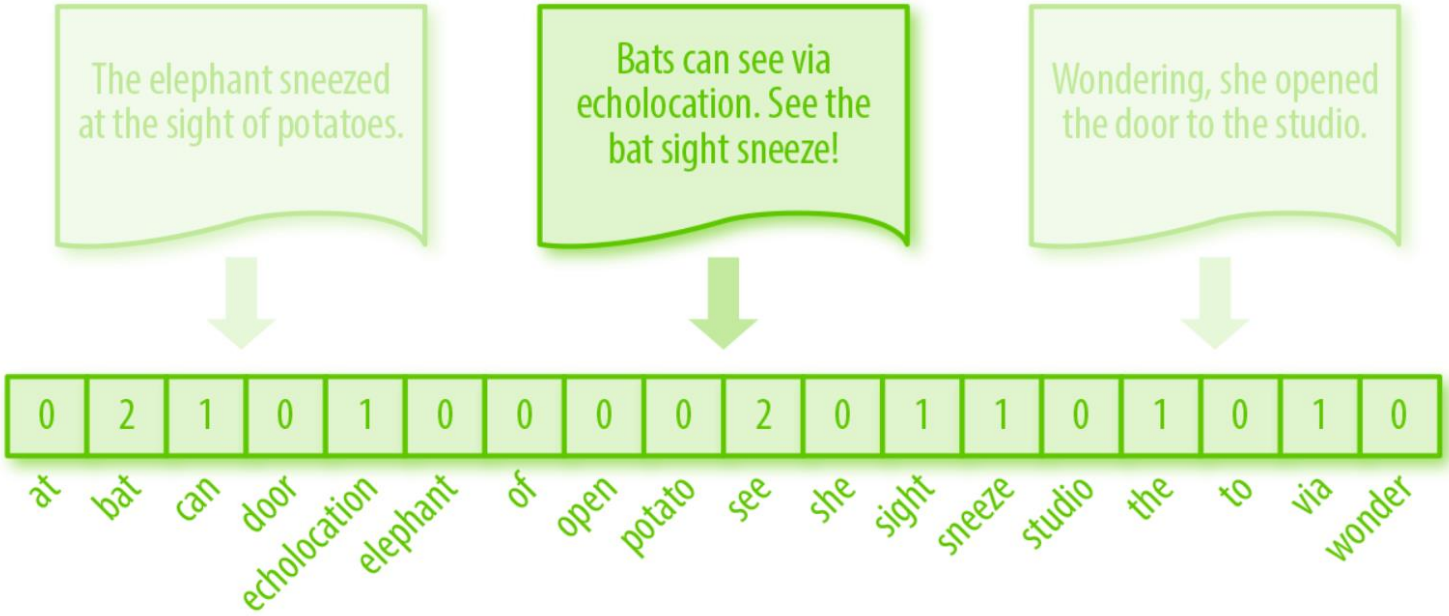


Image source: [Applied Text Analysis with Python [Book] (oreilly.com)]

Document representation: TF-IDF

TF-IDF (Term frequency–Inverse document frequency) reflects also **relevancy of each term** in a document

- **term frequency (TF)** : count of a word in a document
- **inverse document frequency (IDF):** inverse document frequency of the word across a set of documents. This suggests how common or rare a word is in the entire document set. Words appearing across all documents have low scores, words unique for a particular document have high scores.

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

$tf_{i,j}$ = number of occurrences of i in j

df_i = number of documents containing i

N = total number of documents

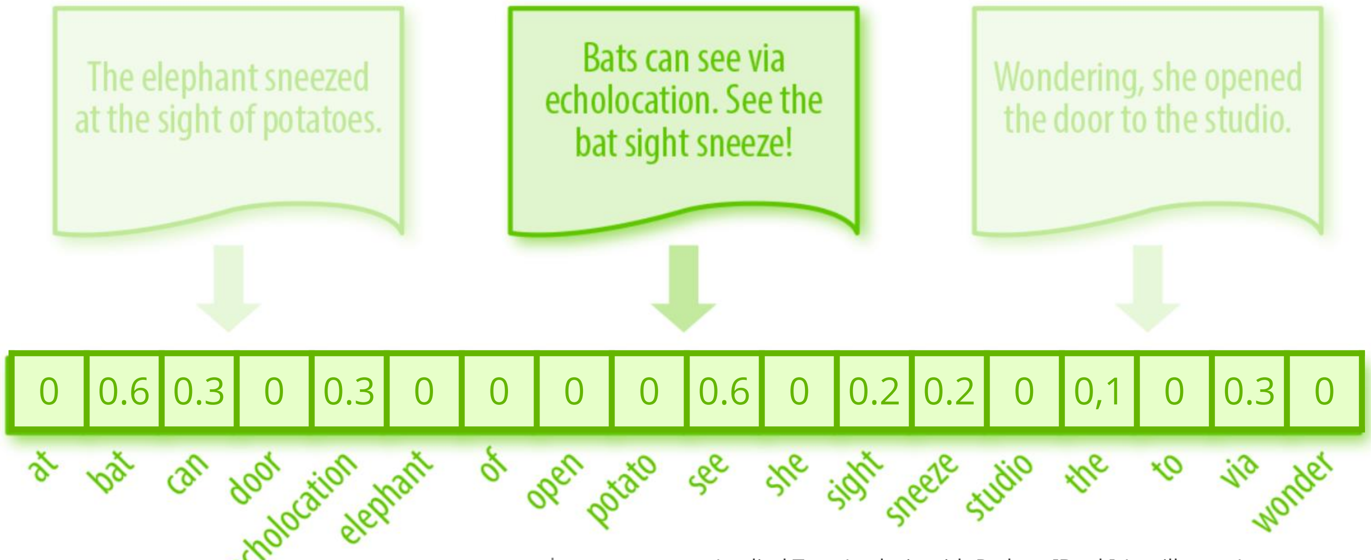
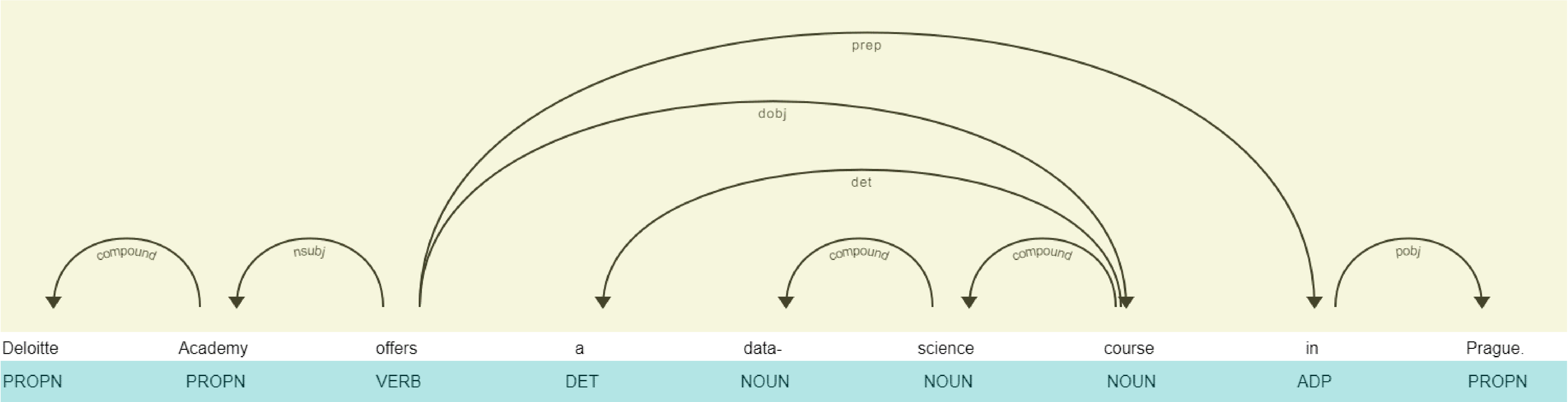


Image source: [Applied Text Analysis with Python [Book] (oreilly.com)]

Dependency parsing & Part-of-Speech tagging

Dependency parsing processes grammatical structures in a sentence and defines relationships between words and phrases.

Part-of-Speech (POS) tagging is the process of categorizing words in correspondence with a particular part of speech based on their definition and context.



Text Enrichment: Named Entity Recognition

Named Entity Recognition (NER) identifies entities (country, people names, locations, organisations…) and type of information such as money, date, …

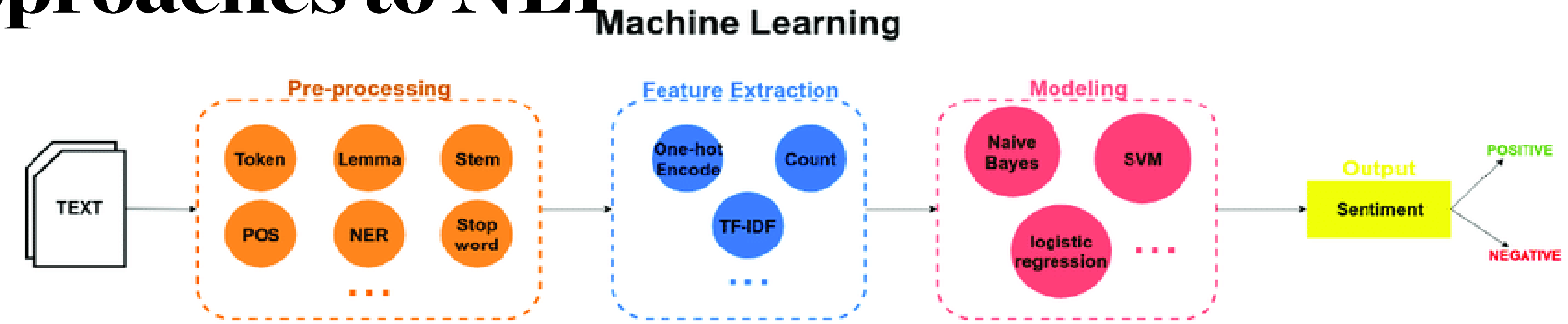
Deloitte Academy **ORG** offers a data-science course in Prague **GPE** .

Three basic methods:

- Gazetteer methods (list of NEs)
  - search all occurrences of NEs from lists in a target strings
- Rule–based extraction
  - linguistic grammar-based techniques
- Semi-supervised/supervised machine learning
  - NER is solved as **classification task** for each token in a sequence
  - many pre-trained NER models available

Approaches to NLP

Approaches to NLP



Transforming NLP

Attention is all you need.

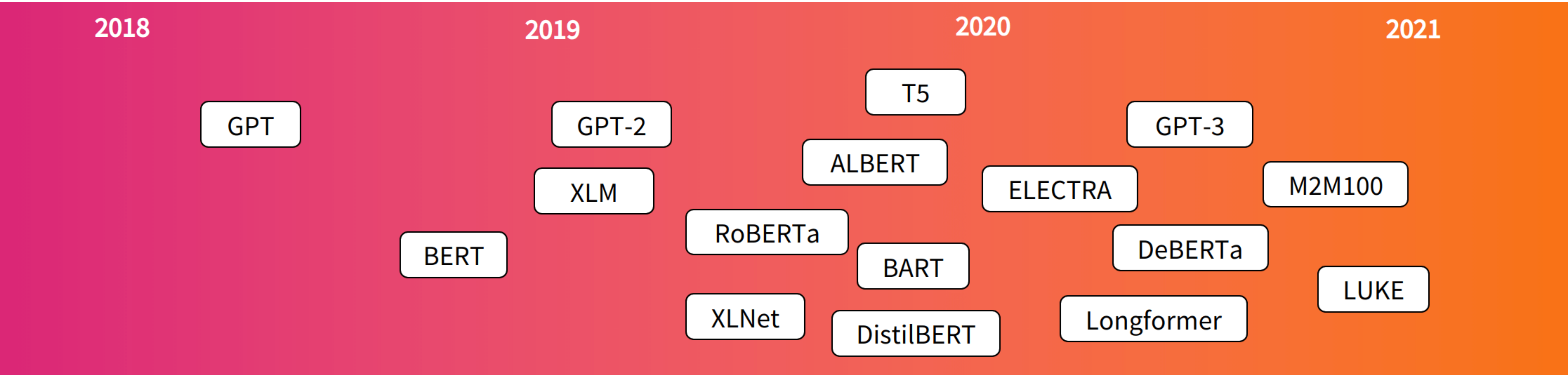


Image source: https://huggingface.co/course/chapter1/4

Transformers in NLP: BERT

BERT = Bidirectional Encoder Representations from Transformers

- Powerful generic ML model for NLP
- Working with the encoder part of transformers
- Setting new benchmarks on most typical NLP tasks, surpassing also humans
- Developed by Google in 2018
- Open-source
    - → many trained models fine-tuned to specific tasks are readily available

- Great BERT explanation by Jay Alammar available here

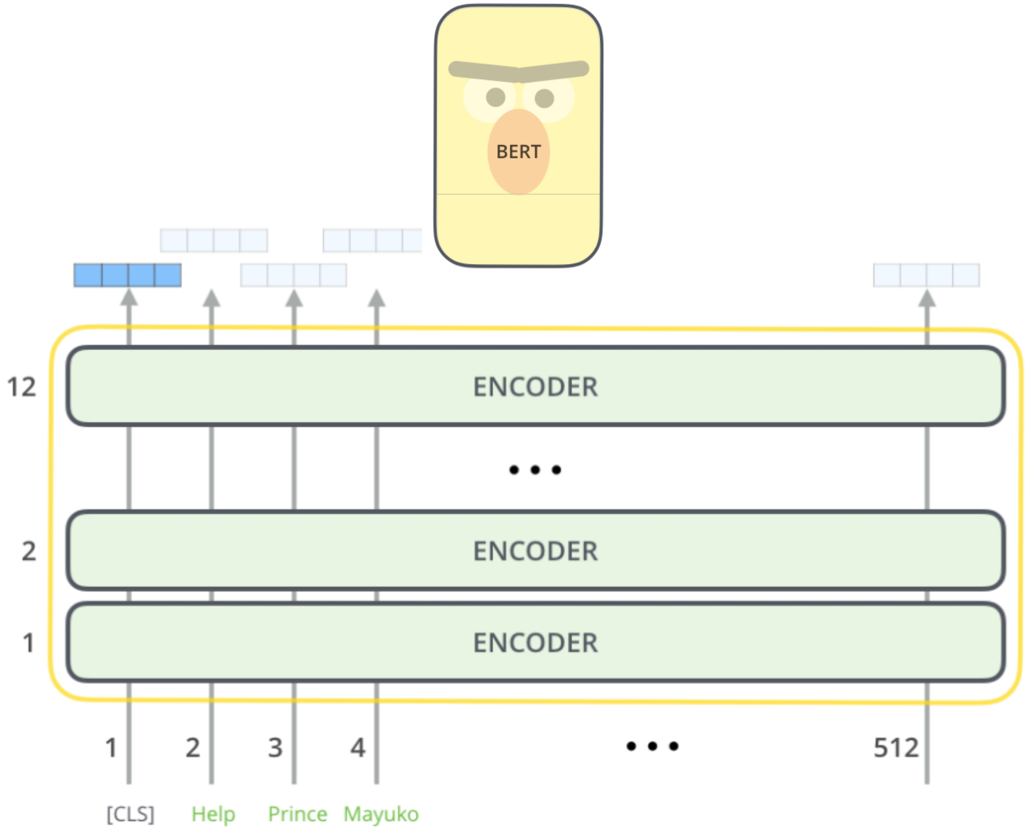


Image source: The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning) – Jay Alammar – Visualizing machine learning one concept at a time. (jalammar.github.io)

BERT Foundations

Large amounts of training data	• Wikipedia + Google's Book Corpus (over 3B words together)
Masked Language Model	• Masking a word (15% random words in training) & bidirectionally learning from words around to predict the masked word → use of context

BERT – Masked Language Model

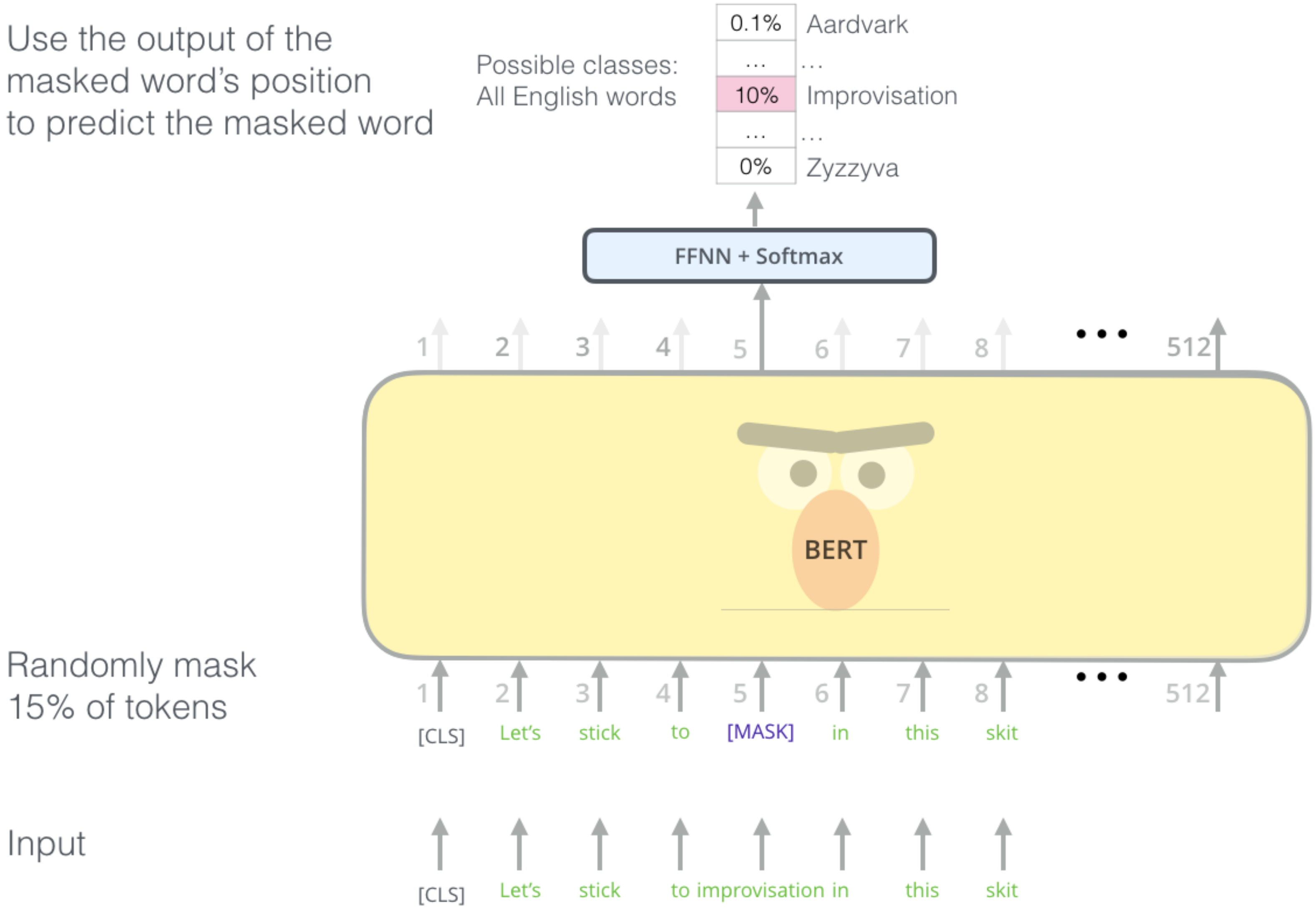


Image source: The Illustrated BERT, ELMo, and co. (jalammar.github.io)

BERT Foundations

Large amounts of training data	• Wikipedia + Google's Book Corpus (over 3B words together)
Masked Language Model	• Masking a word (15% random words in training) & bidirectionally learning from words around to predict the masked word → use of context
Next Sentence Prediction	• Mix of correct and random sentence pairs (50%-50%) & predicting which are the right sequences

BERT – Next Sentence Prediction

Predict likelihood that sentence B belongs after sentence A

Tokenized Input

Input

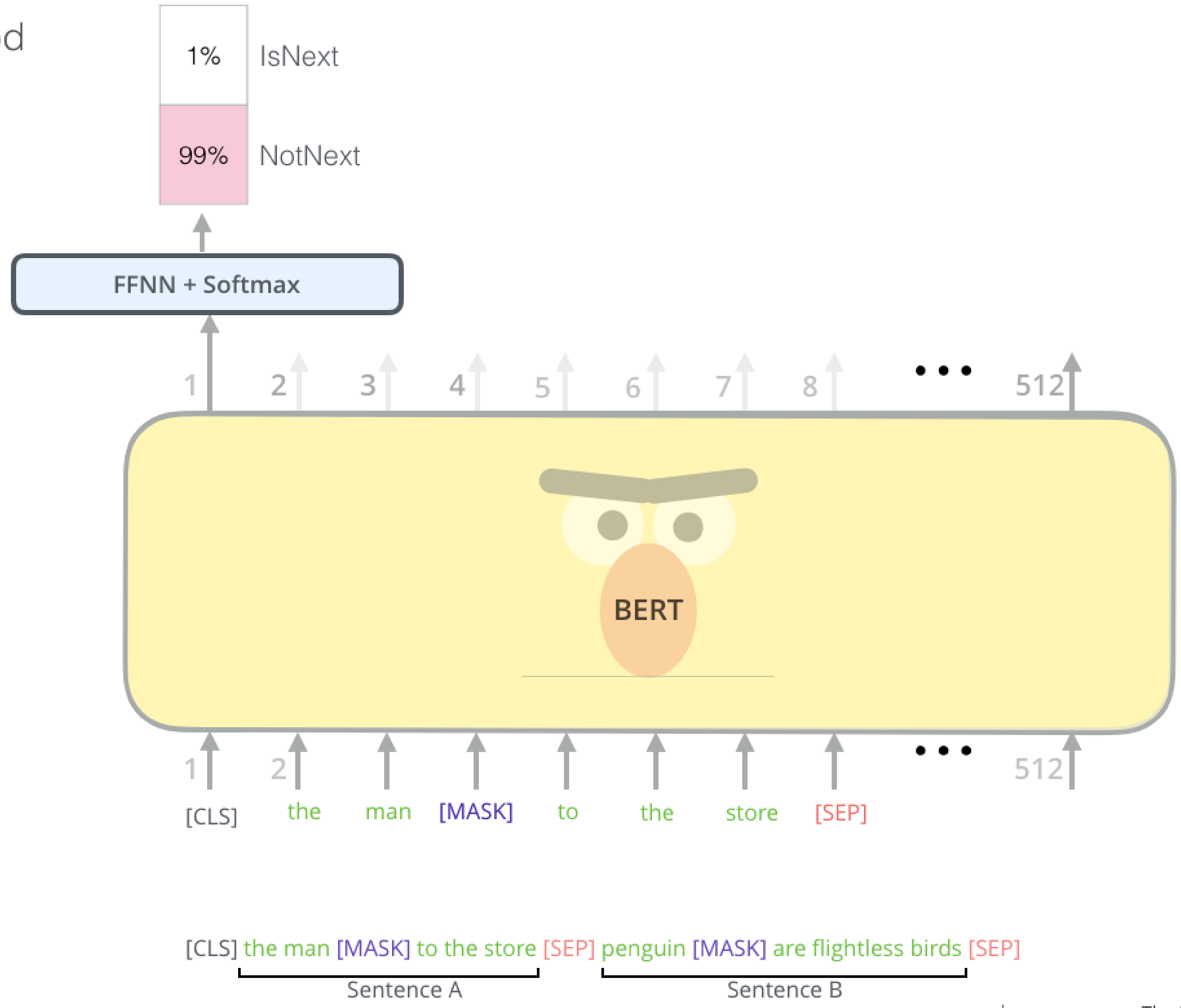
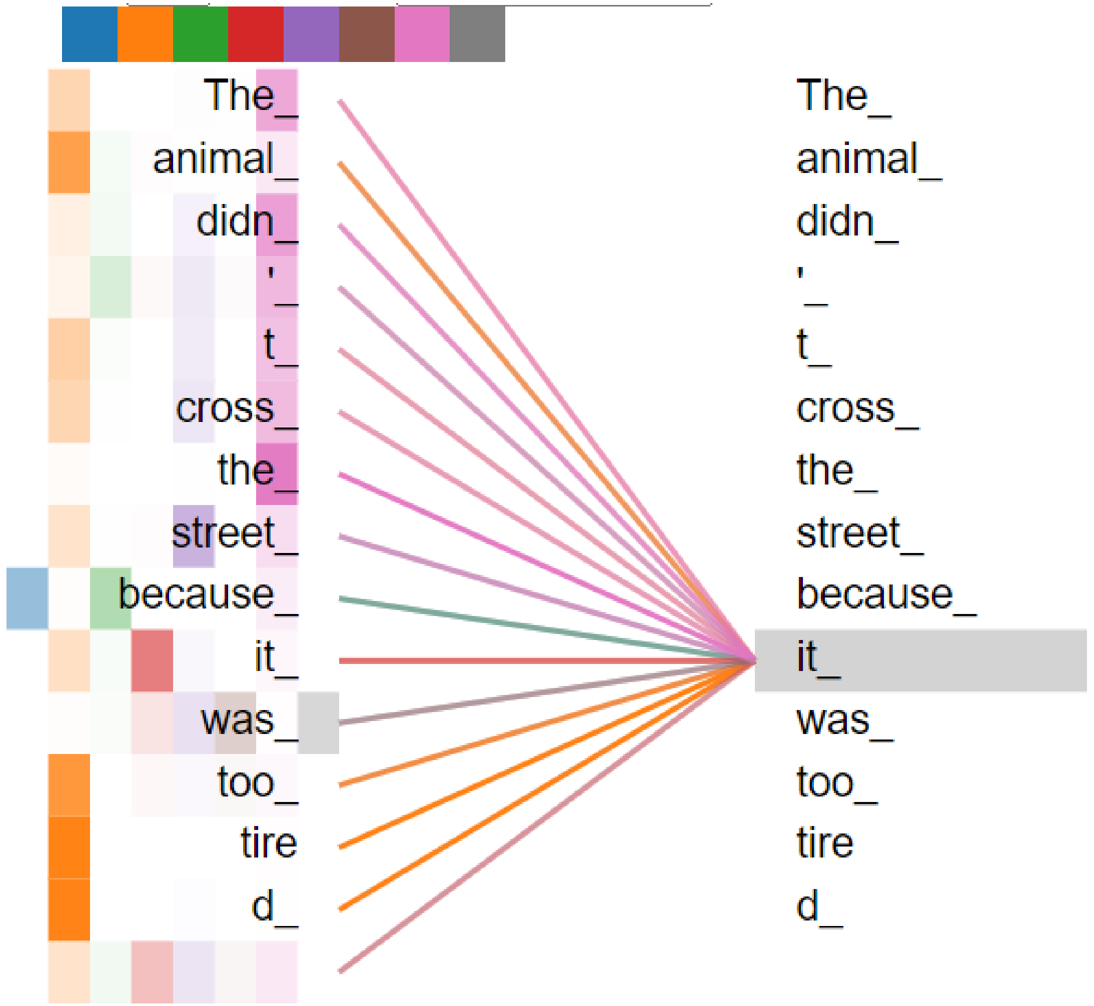


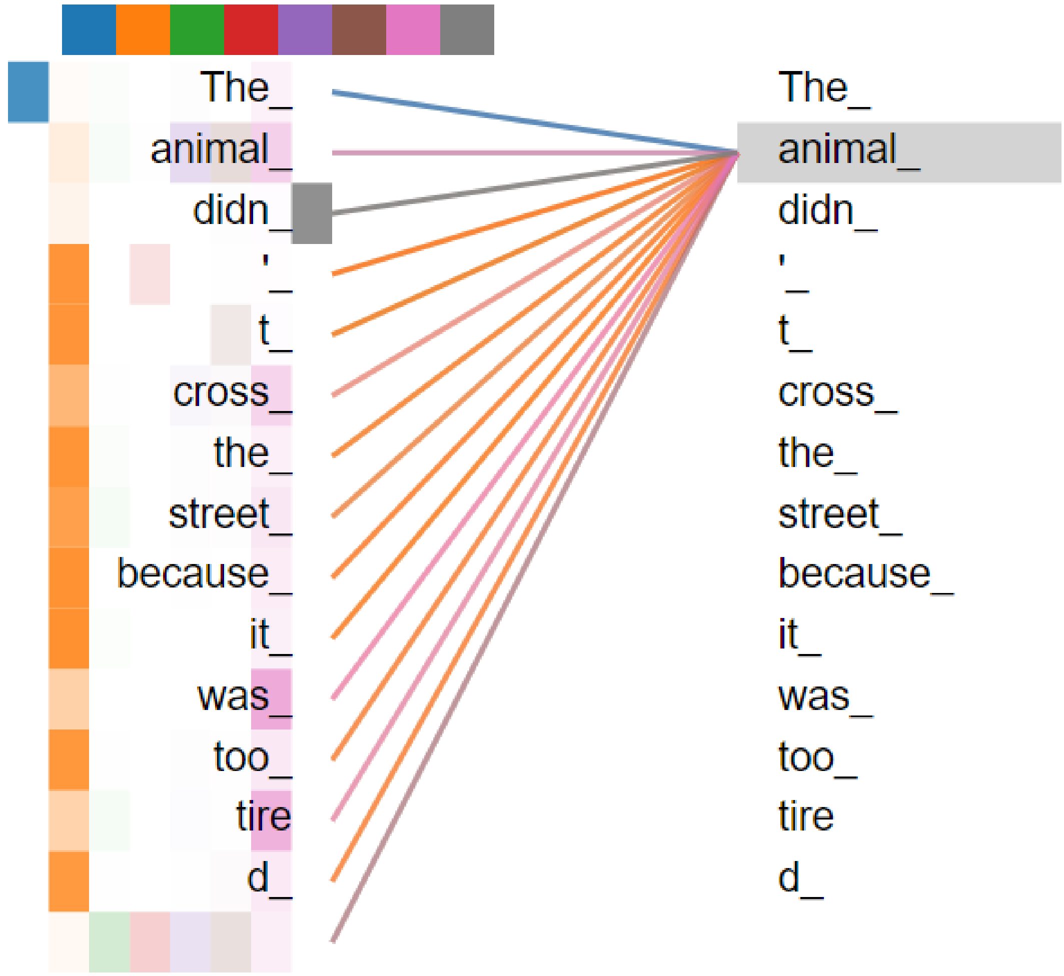
Image source: The Illustrated BERT, ELMo, and co. (jalammar.github.io)

BERT Foundations

Large amounts of training data	• Wikipedia + Google's Book Corpus (over 3B words together)
Masked Language Model	• Masking a word (15% random words in training) & bidirectionally learning from words around to predict the masked word → use of context
Next Sentence Prediction	• Mix of correct and random sentence pairs (50%-50%) & predicting which are the right sequences
Transformer architecture & attention mechanism	• Massive parallelization • Attention mechanism: assigning weights based on how critical individual words in a sentence are for further processing

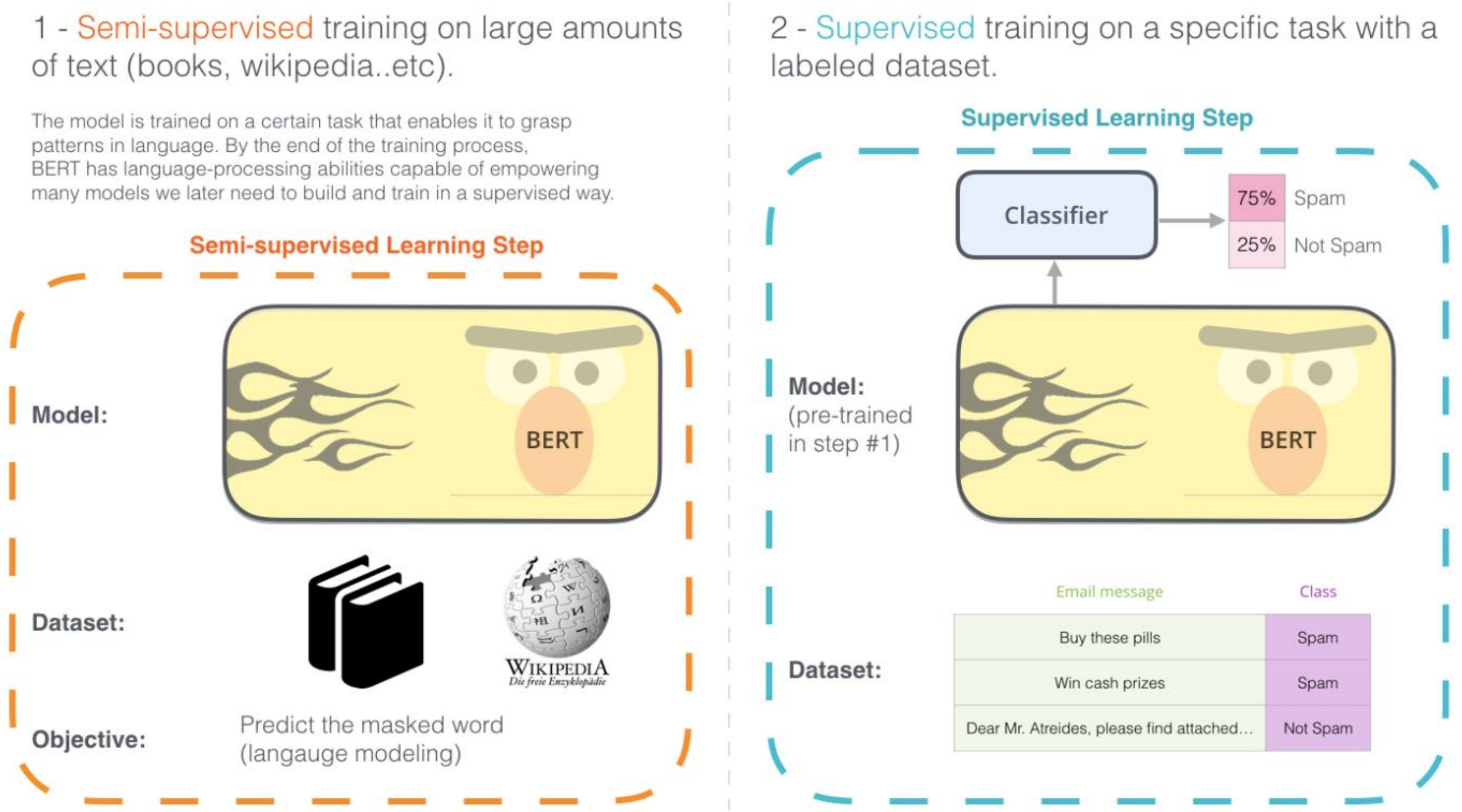
BERT – Attention Mechanism





Source: Tensor2Tensor Intro - Colaboratory (google.com)

Transformers in NLP: BERT



The two steps of how BERT is developed. You can download the model pre-trained in step 1 (trained on un-annotated data), and only worry about fine-tuning it for step 2. [Source for book icon].

Image source: The Illustrated BERT, ELMo, and co. (jalammar.github.io)

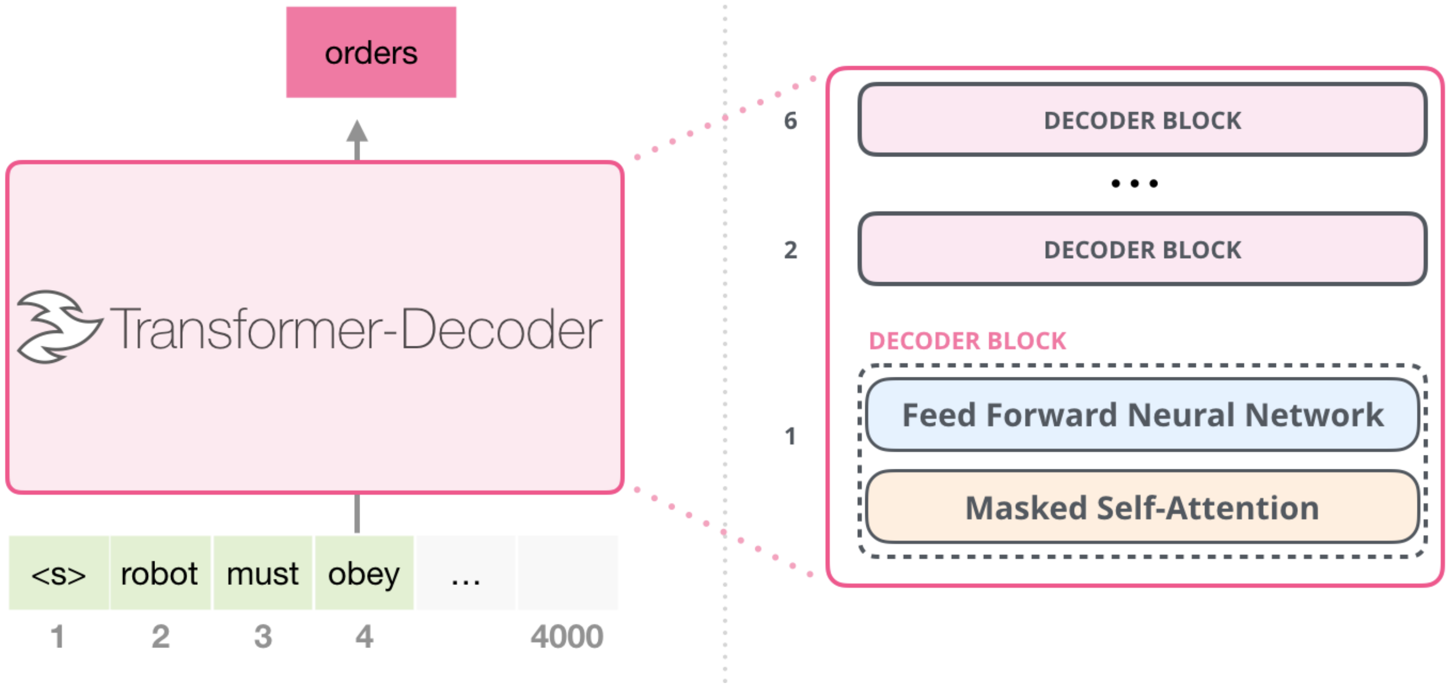
Transformers in NLP: GPT-3

GPT = Generative Pre-trained Transformer

- Powerful generic ML model for NLP, working with decoder part of transformers
- Developed by OpenAI, three releases: GPT, GPT-2 (open-source), GPT-3
- Basis for ChatGPT

GPT-3
- No supervised training needed – minimal fine-tuning, just a few examples
- 175B parameters
- Accessible through paid API

- Detailed explanation here and here



 |

Transformers in NLP: GPT-3

Sample use cases

Q&A
Answer questions based on existing knowle...

Grammar correction
Corrects sentences into standard English.

Mood to color
Turn a text description into a color.

Write a Python docstring
An example of how to create a docstring for ...

Summarize for a 2nd grader
Translates difficult text into simpler concep...

Natural language to OpenAI API
Create code to call to the OpenAI API usin...

Analogy maker
Create analogies. Modified from a communi...

JavaScript one line function
Turn a JavaScript function into a one liner.

Text to command
Translate text into programmatic commands.

English to other languages
Translates English text into French, Spanish...

Micro horror story creator
Creates two to three sentence short horror ...

Third-person converter
Converts first-person POV to the third-pers...

Natural language to Stripe API
Create code to call the Stripe API using nat...

SQL translate
Translate natural language to SQL queries.

Notes to summary
Turn meeting notes into a summary.

VR fitness idea generator
Create ideas for fitness and virtual reality g...

Parse unstructured data
Create tables from long form text

Classification
Classify items into categories via example.

ESRB rating
Categorize text based upon ESRB ratings.

Essay outline
Generate an outline for a research topic.

Python to natural language
Explain a piece of Python code in human un...

Movie to Emoji
Convert movie titles into emoji.

Recipe creator (eat at your own risk)
Create a recipe from a list of ingredients.

Chat
Open ended conversation with an AI assist...

For more see: https://beta.openai.com/examples

NLP is not perfect

- Biases

  Models might contain implicit biases (stereotypes or negative sentiment) towards certain groups coming from training data

- Interpretability

  Bigger and more complex models like GPT work like a black box, not providing much insights that could explain their outputs

- Domain-specific

  Models should be used on the same domains they were trained on

- Processing-heavy & dependent on vast volumes of training data

  Training of new models is heavy on resources and suitable data

- Worse results for smaller languages or dialects
- Limited options for processing of longer documents
- Complex question answering / text understanding is hard
- Lack of world knowledge & context

Complexity of language processing

- Large quantities of unstructured data
- Multitude of languages
- Different channels – added complexity for voice that needs to be converted to text first
- Actual meaning of voice/text is often difficult to interpret
    - Sarcasm and irony
    - Usage of slang and local dialects
    - Different words/expressions can have the same meaning

      car – vehicle; I don't want that. – I would rather take something else. – No, thanks.
    - Same words/expressions can have different meaning in different context

      a branch of a tree VS a branch of a bank
    - Incomplete sentences, interruptions
    - Implicit world knowledge

NLP Tasks and their Applications

Common NLP tasks

- Topic analysis
- Machine translation
- Sentiment analysis
- Language detection
- Named entity recognition
- Next word prediction
- Text generation
- Text summarization
- Question answering

Topic Analysis: Topic Modelling & Topic Classification

Topic analysis enables exploration of recurring topics within documents.

- Applications: Medical industry, Scientific research review, Recommender systems, Customer support (tagging and routing to the right resource), Opinion or meeting summarization, ….

- Approaches: unsupervised modelling or supervised/rule-based classification

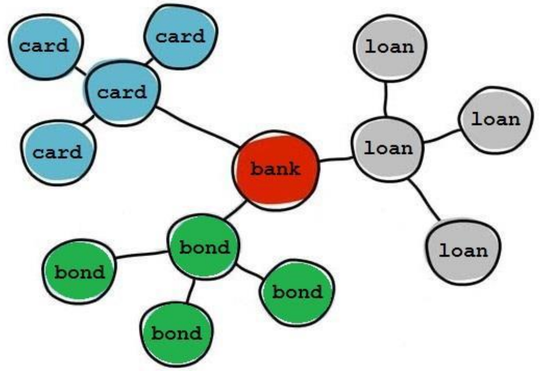
FDA Approves Pfizer's COVID-19 Vaccine…	Health
U.S. President Barack Obama makes remarks to a Democratic National Committee fundraiser at the historic Warner Theatre, in Washington	Politics
France vs. Croatia odds, picks, prediction: Soccer expert reveals UEFA Nations League bets for Monday, June 13	Sport
Barack Obama Attends Bulls-Celtics Game During Visit to …	Politics or Sports?

Topic Modelling

Topic modeling is a statistical modeling method for **discovering** the abstract "**topics**" that occur in a collection of documents (the topics are not pre-defined).

Allow grouping of texts into groups based on their similarity

- without training data (clusters are recognized automatically)
- used for **Content analysis, clustering**



Calculation of similarity

- based on comparing vector representations of texts
- **Similarity of texts == similarity of numeric vectors of texts**
- metrics: Cosine similarity, Euclidean distance,...

Methods

- unsupervised ML methods for clustering (k-means,...)

Topic Modelling: Latent Dirichlet Allocation

In LDA, we take documents at the input and produce (unlabeled) topics at the output

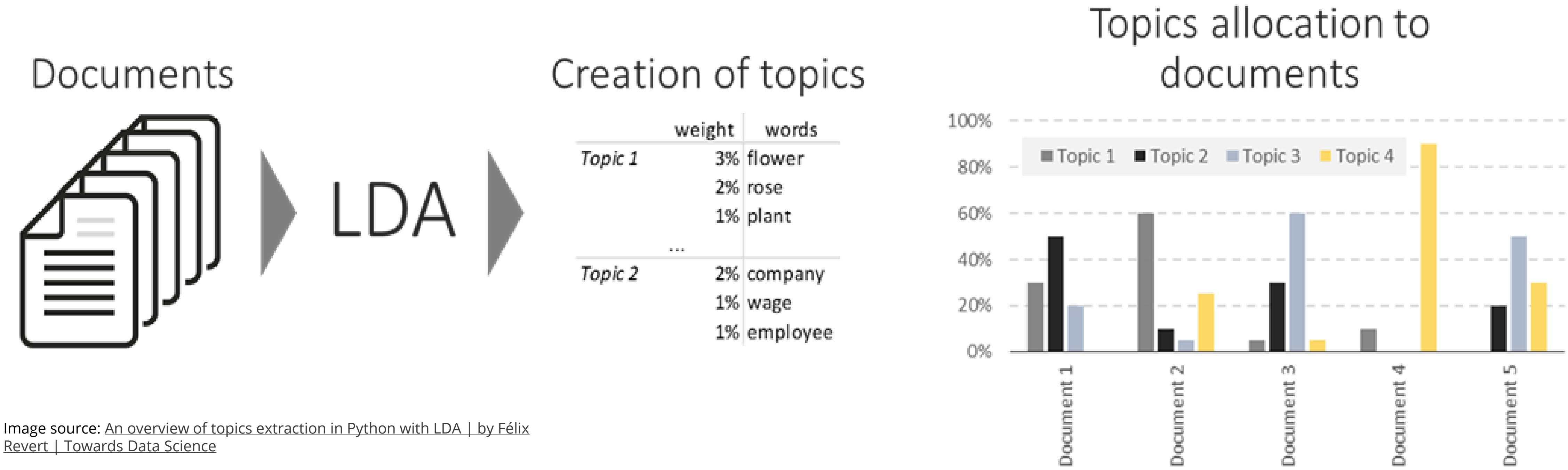


Image source: An overview of topics extraction in Python with LDA | by Félix Revert | Towards Data Science

A **topic** is a bunch of **weighted words**, and **documents** are represented as a **mixture of topics**

Topic Classification

Topic Classification **categorizes** text **into** one of **pre-defined topics**.

1. Keyword/rule-based approach
   - Get a list of keywords related to your categories

     Sports= [football, basketball, LeBron James]
     Politics = [Barack Obama, Donald Trump, Hillary Clinton, Putin]

   - Use the keywords in rules for classification

     Text = “When is LeBron James' first game with the Lakers? “

     Sports→ 1
     Politics→ 0
     AND/OR
     Vector_similarity(Text, Sports) > Vector_similarity(Text, Politics)
     ⬇
     Topic: Sports

2. ML-based approach
   - Get annotated training data split into topics
   - Train a classification model (based on embeddings or tfidf/BoW)

Machine Translation

Machine translation is a sub-field of computational linguistics that investigates the use of software to **translate text or speech from one language to another**

- Applications: translation across industries and use cases
- Approaches:
    - Statistical MT
    - Rule-based MT
    - Hybrid MT
    - Neural

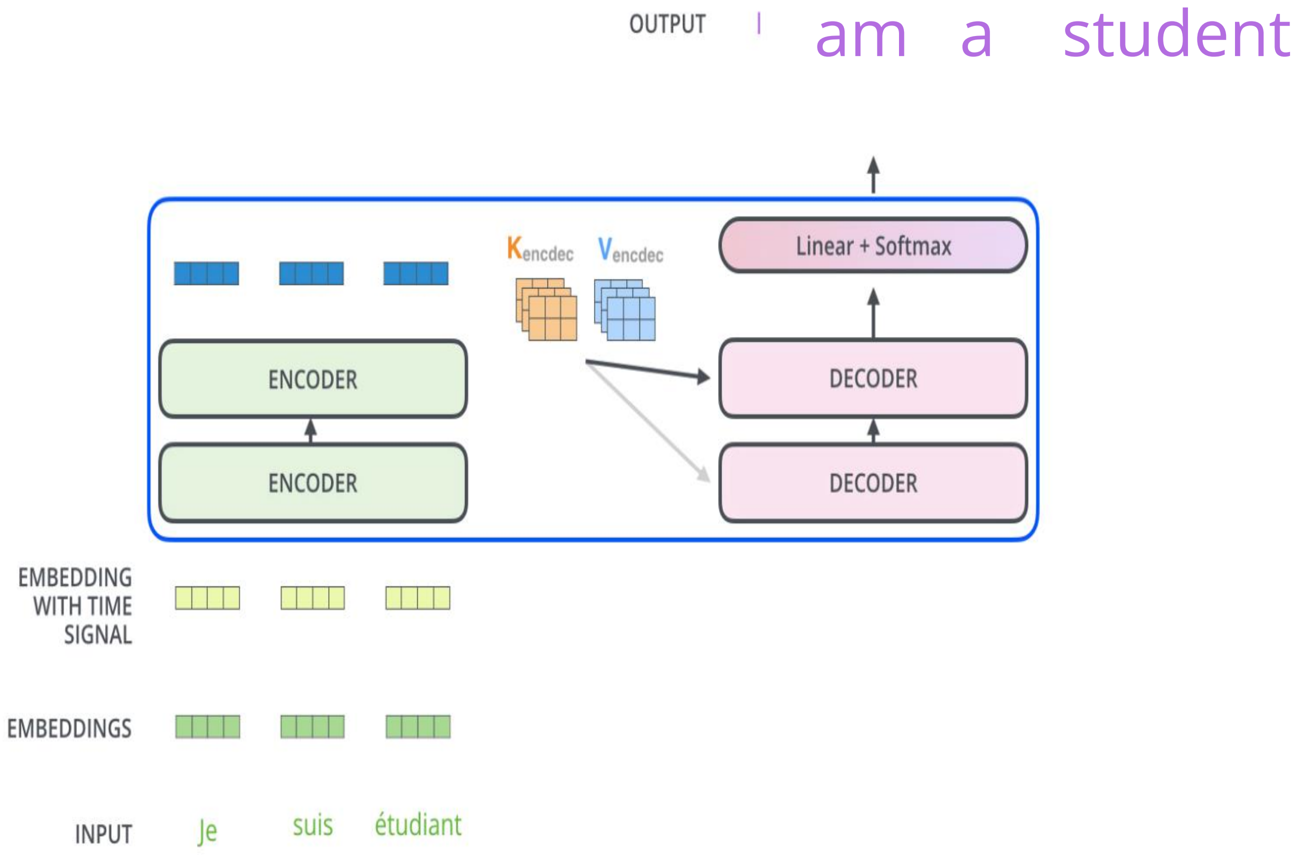


Image source: The Illustrated Transformer (jalammar.github.io)

Sentiment Analysis

Sentiment analysis (or Opinion Mining) is used to **systematically identify, extract, quantify, and study affective states** and subjective information.

- Applications: marketing analyses, product & market research, product reviews, reputation monitoring, social media monitoring, customer feedback, request prioritization…
- What makes it tricky: negation, sarcasm/irony, sentiment of a phrase vs whole sentence
- Approaches: Lexicon- & rule-based or ML/DL-based

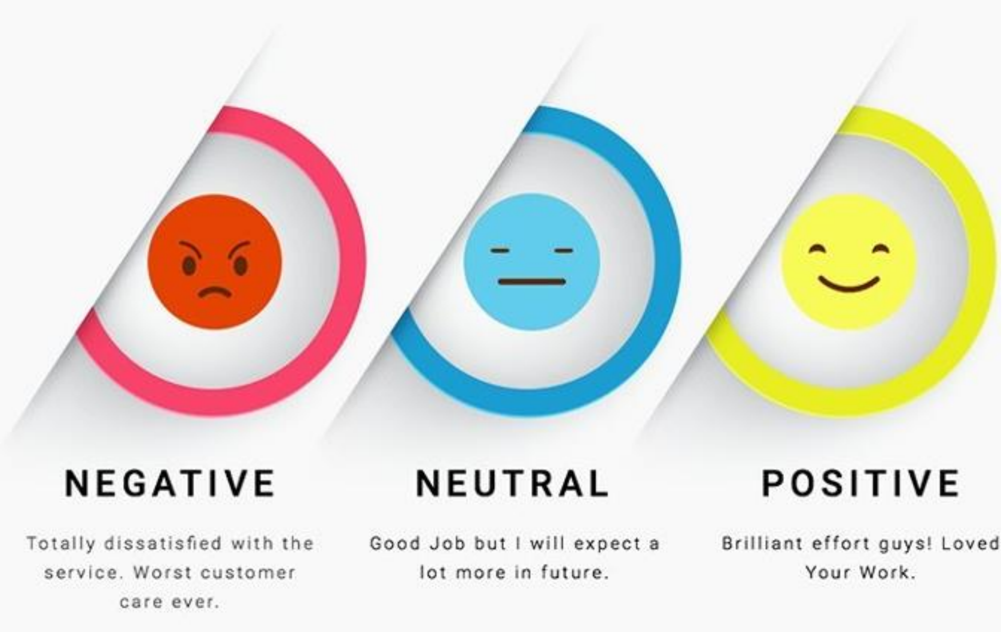
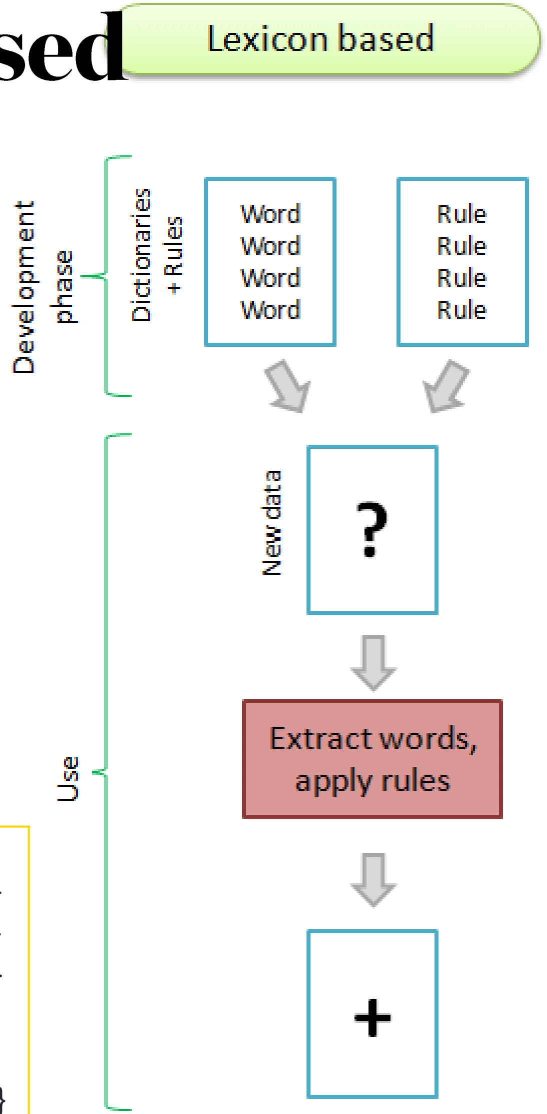


Image source: A Guide to Customer Sentiment Analysis (and Why It Matters) (revechat.com)

Sentiment Analysis: Lexicon- & rule-based

- Easy to interpret, but labour-intensive to create, limited to a crafted lexicon of weighted words
- Open-source VADER tool based on extensive lexicon and complex linguistic rules
  - Works also with common emoticons, acronyms (LOL), punctuation
  - Provides scores for positive, negative and neutral emotions + a compound score




```
VADER is smart, handsome, and funny.--------------------------------- {'pos': 0.746, 'compound': 0.8316, 'neu': 0.254, 'neg': 0.0}
VADER is smart, handsome, and funny!--------------------------------- {'pos': 0.752, 'compound': 0.8439, 'neu': 0.248, 'neg': 0.0}
VADER is very smart, handsome, and funny.--------------------------- {'pos': 0.701, 'compound': 0.8545, 'neu': 0.299, 'neg': 0.0}
VADER is VERY SMART, handsome, and FUNNY.----------------------- {'pos': 0.754, 'compound': 0.9227, 'neu': 0.246, 'neg': 0.0}
VADER is VERY SMART, handsome, and FUNNY!!!------------------- {'pos': 0.767, 'compound': 0.9342, 'neu': 0.233, 'neg': 0.0}
VADER is VERY SMART, uber handsome, and FRIGGIN FUNNY!!!- {'pos': 0.706, 'compound': 0.9469, 'neu': 0.294, 'neg': 0.0}
VADER is not smart, handsome, nor funny.---------------------------- {'pos': 0.0, 'compound': -0.7424, 'neu': 0.354, 'neg': 0.646}
```

Image source: Machine learning and lexicon-based approaches to sentiment analysis

Sentiment Analysis: ML based

- Requires labelled training data
- Different classification algorithms possible, commonly Naïve Bayes or LSTM

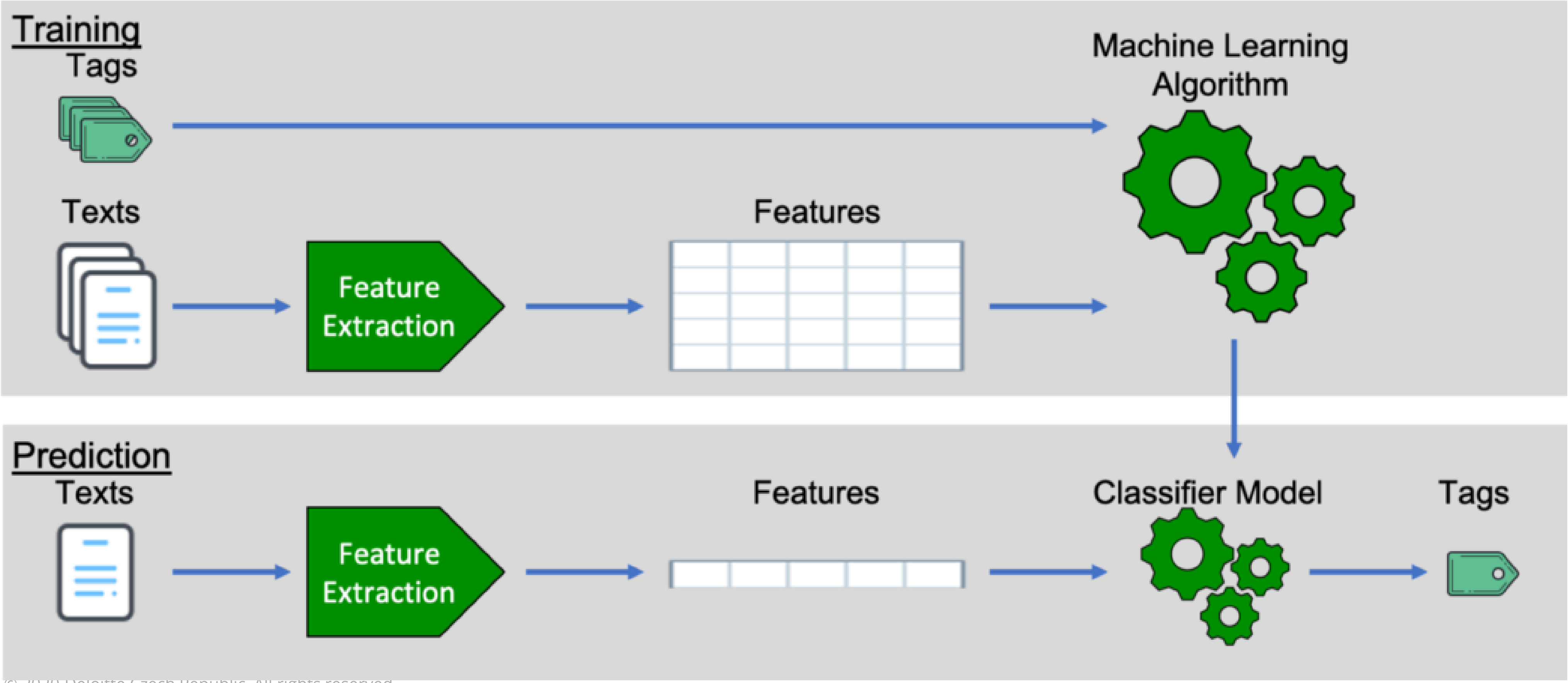


Image source: Sentiment Analysis — Comparing 3 Common Approaches: Naive Bayes, LSTM, and VADER | by Kevin C Lee | Towards Data Science

Language Detection

Process of **classifying** incoming text according to its **language**

- Applications: Homogenization of a large dataset (e.g Google Play app reviews), routing purposes (e.g route support tickets to the appropriate team)
- Doesn’t work very well on short texts/fractions of texts
- Variety of models and APIs readily available

Text

“Text mining je vědecká disciplína na pomezí dolování z dat, strojového učení a počítačové lingvistiky.”

Language categories

Czech
English
German
Dutch

Label: **Czech**
Confidence: **92 %**

 |

Next word prediction

Next word prediction is **the task of predicting what word comes next.**

- Applications: Search engines' auto suggest, Services such as Gmail-SmartCompose
- Basic next word prediction possible directly with the pre-trained BERT and its Masked Language Model, placing the mask as the last sequence item.

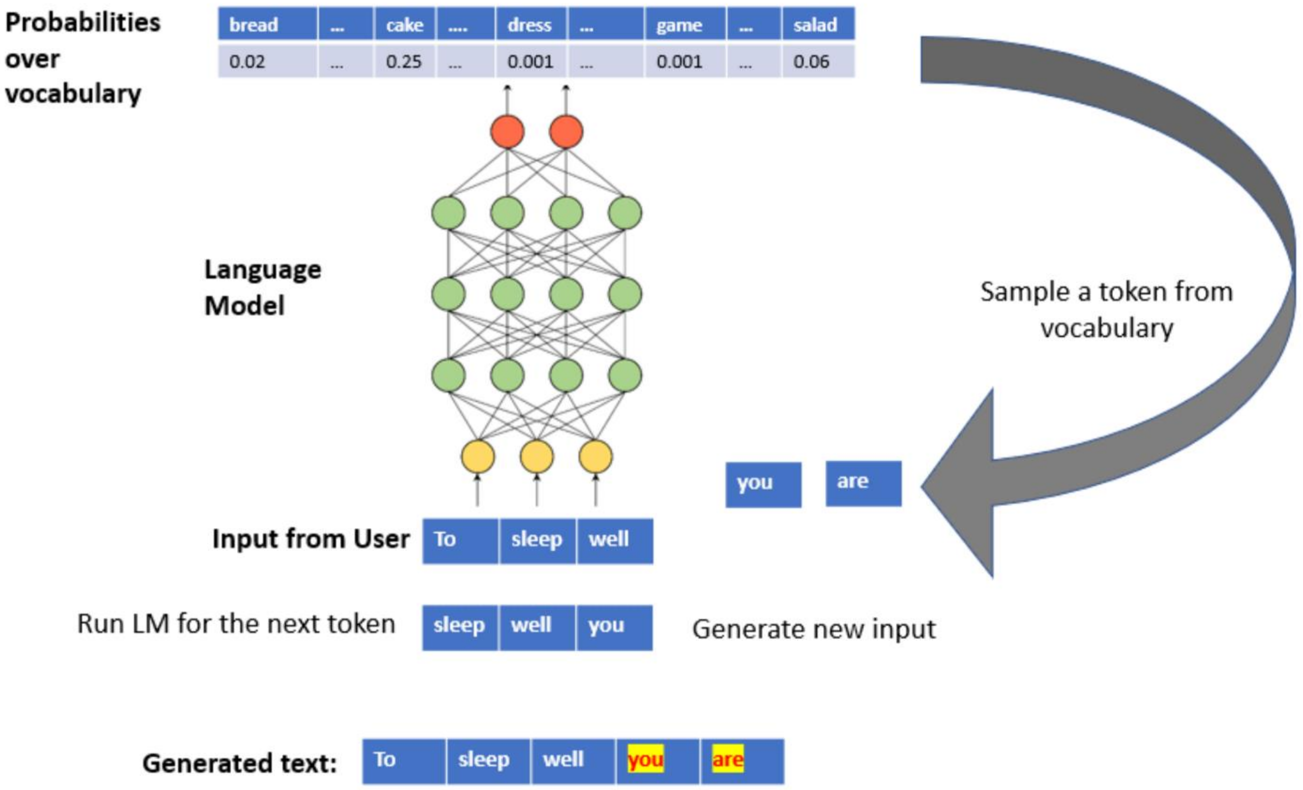


Image source: An illustration of next word prediction with state-of-the-art network architectures like BERT, GPT, and XLNet | by Ajay Arunacha

Text Generation

Natural Language Generation / Text Generation is a process that **automatically produces natural language output**. The goal is to have output indistinguishable from human text.

Prompt: "Explain to a team of data scientists, how they can benefit from a NLP training."

Generated response:
A NLP (Natural Language Processing) training can benefit data scientists in several ways:

- Improved Data Analysis: NLP techniques can help data scientists extract information from unstructured data sources like text, audio and video, making it easier to analyze and gain insights.
- Better Predictive Models: NLP techniques can also be used to enhance predictive models by including text-based features as inputs, leading to improved predictions.
- Improved Text Generation: Data scientists can use NLP techniques to generate text automatically, for example, to create summaries, generate product descriptions, or write reports.
- Enhanced Customer Interaction: NLP can be used to create chatbots and conversational agents, providing customers with more natural and efficient ways to interact with a company.
- Increased Automation: NLP can automate many manual tasks such as sentiment analysis, summarization, and document classification, freeing up data scientists to focus on more complex tasks.

In summary, NLP is a valuable tool for data scientists to extract insights from large volumes of unstructured data, improve predictive models, automate repetitive tasks, and enhance customer interactions.

Source: ChatGPT

Text Summarization

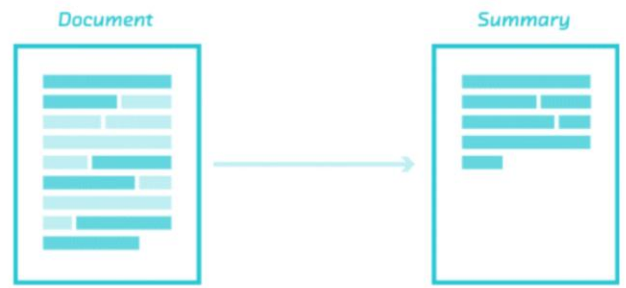
Text summarization is the technique for **generating a** concise and precise **summary** of voluminous texts while focusing on the sections that convey useful information, and **without losing the overall meaning**.

Extractive Summarization extracts several parts, such as phrases and sentences, from a piece of text and stack them together to create a summary.

The summary obtained contains **exact sentences from the original text**

Abstractive Summarization uses advanced NLP techniques to **generate** an entirely new summary.

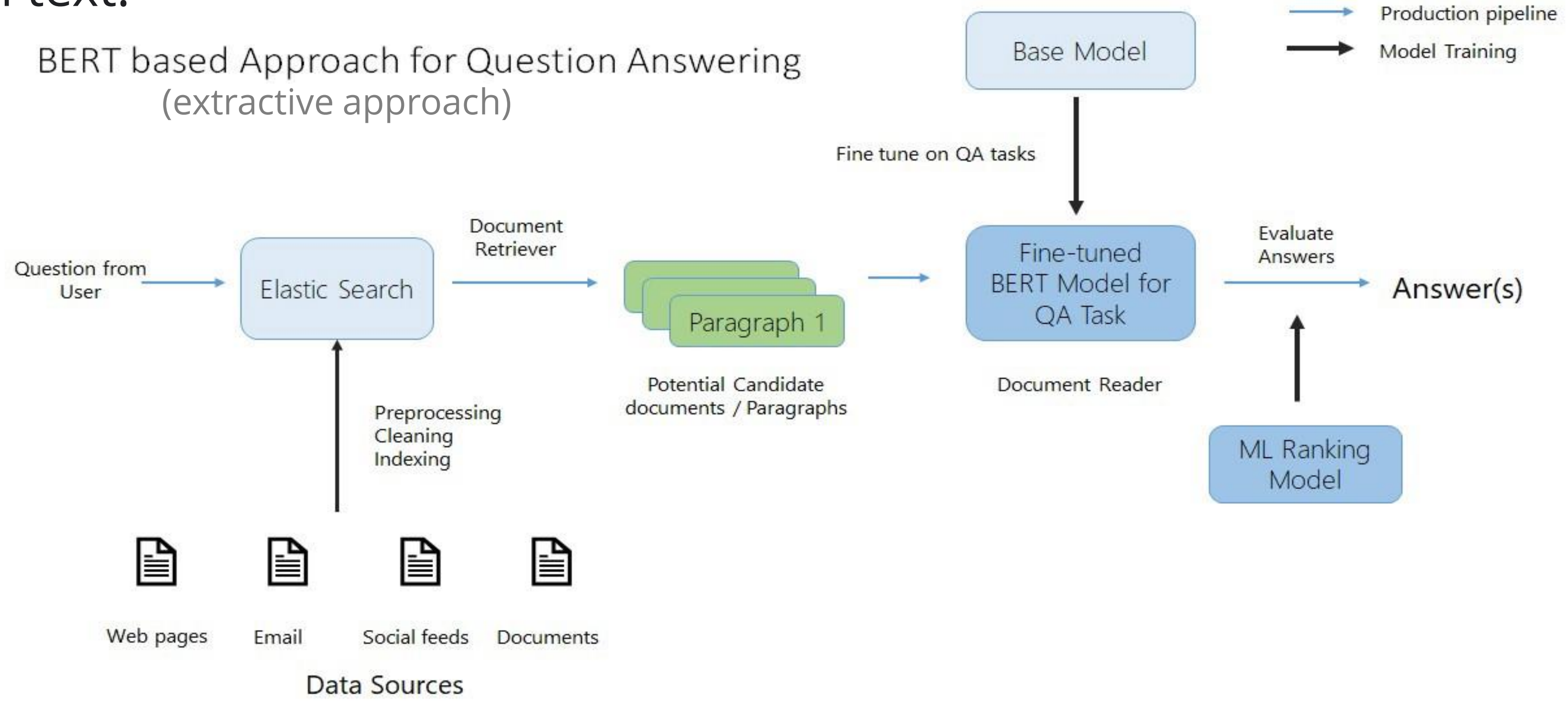
The summary obtained **does not contain exact sentences** from the original text



https://medium.com/@ondenyi.eric/extractive-text-summarization-techniques-with-sumy-3d3b127a0a32

Question Answering

Question Answering is a computer science discipline within the fields of information retrieval and natural language processing, which is concerned with building systems that **automatically answer questions posed by humans in a natural language**.
Within NLP, Question Answering refers to the ability to retrieve the answer to a question from a given text.



NLP hands-on