Sentiment Analysis with BERT

Advanced Software-Engineering

Dr. Harald Stein

Prof. Dr. Selcan Ipek-Ugay

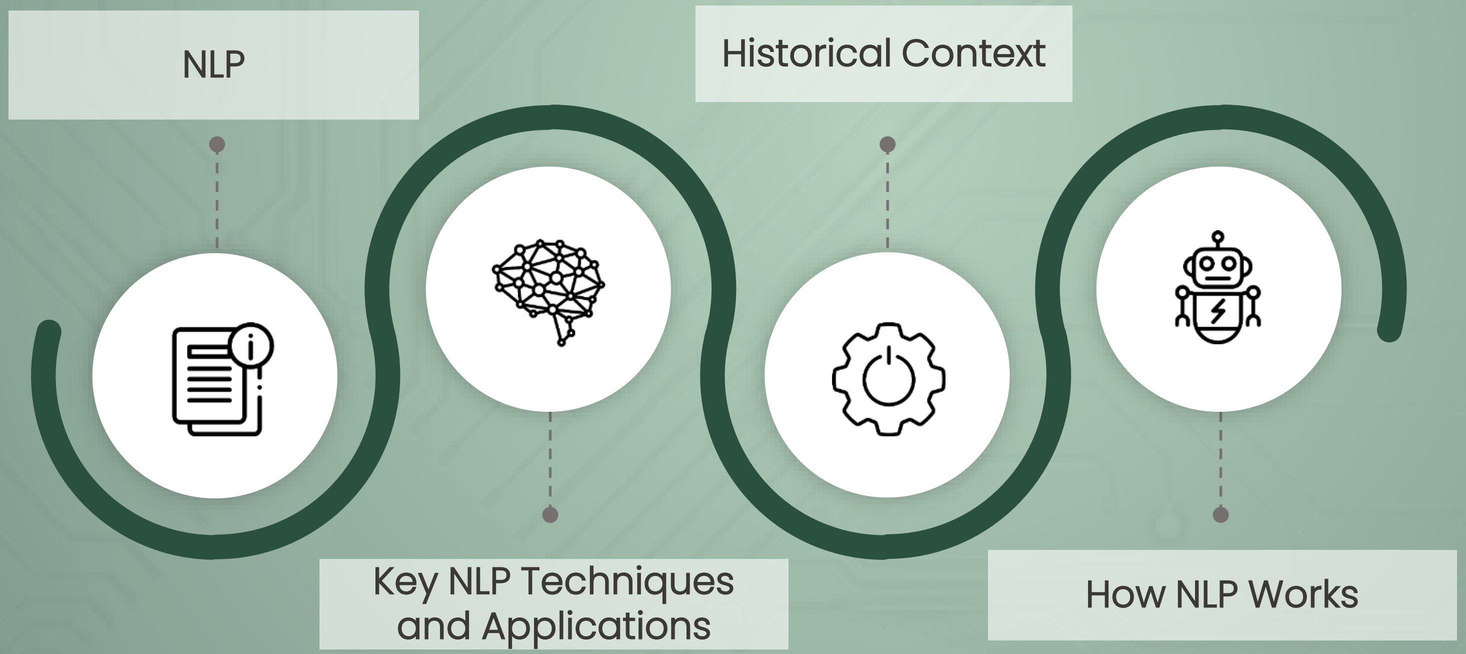
WS 2024/25

Agenda

- Understanding Natural Language Processing
- Vector Representations
- Transformers: Basics and usage for sentiment analysis
- Programming Tools
- Example: Sentiments of movie comments

Natural Language Processing

… is a technology that bridges the communication gap between human language and computer understanding.



NLP

Key NLP Techniques and Applications

Historical Context

How NLP Works

Natural Language Processing (NLP)

The primary goal of NLP is to enable computers to understand, interpret, and generate human language in a way that is both meaningful and useful.

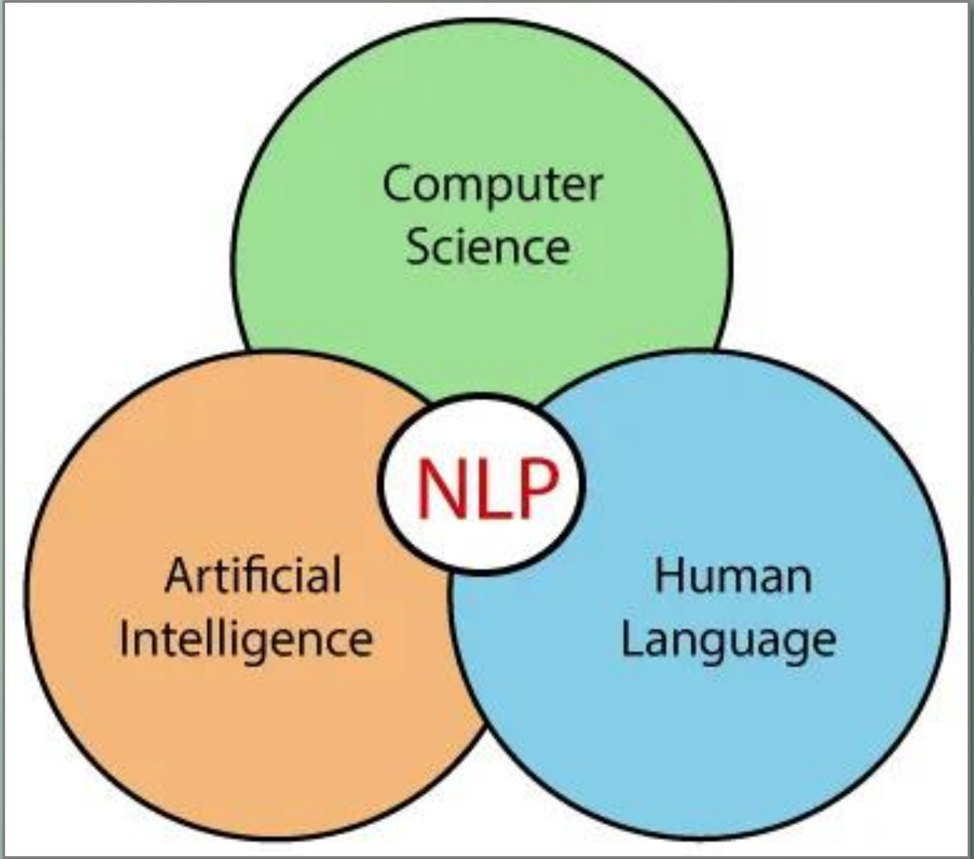
Interdisciplinary Nature

It encompasses areas of

- computer science
- artificial intelligence
- linguistics
- to interpret, recognize, and generate human language in a way that is valuable.

Real-World Applications

- Voice assistants
- Chatbots
- Translation services
- Sentiment analysis
- Customer service.



Key NLP Techniques and Applications

Diverse Applications of NLP

Focus

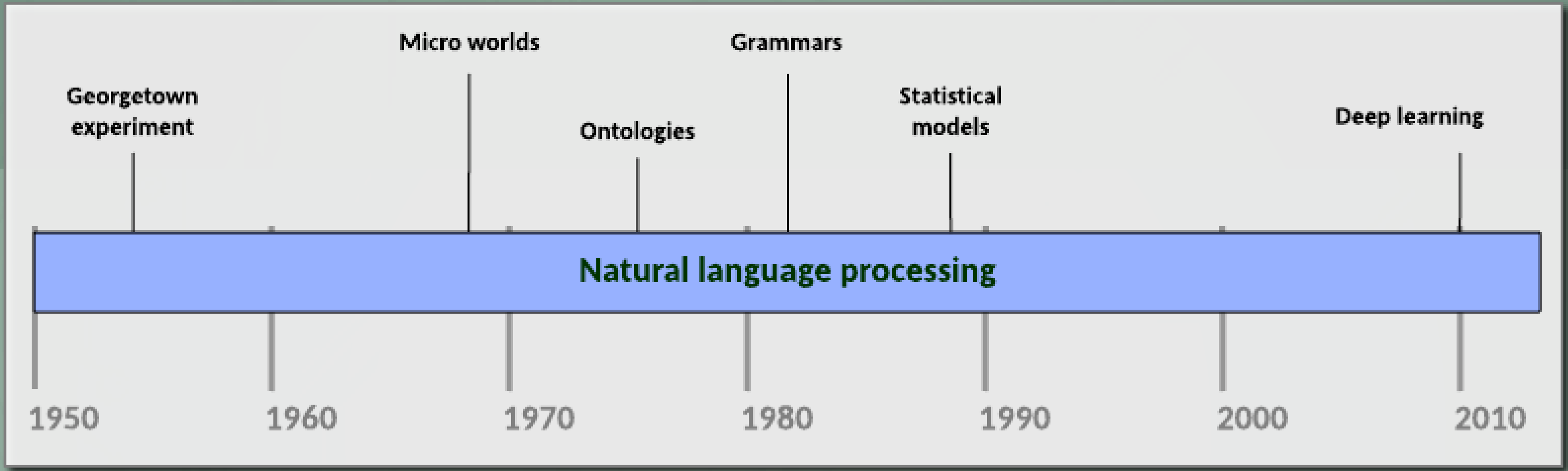
	Description	Kind of task
Sentiment Analysis	Identifying emotions in text to gauge sentiments like positive, negative, or neutral.	Classification
Text/Document Classification:	Assigning categories to text based on content. Utilizes supervised learning on labeled data.	Classification
Part-of-speech (POS) Tagging	Assigning grammatical categories to words in sentences to identify their syntactic roles.	Classification
Language Detection & Machine Translation	Identifying a text's language and translating text between languages	Translation
Information Retrieval	Retrieving relevant information from vast text datasets in response to user queries.	Text Generation

Key NLP Techniques and Applications

Diverse Applications of NLP

	Description	Kind of task
Text Summarization	Condensing long texts while preserving key information and context.	Text Generation
Knowledge Graph & QA System	Organizing information in a structured form and answering questions using that knowledge.	Text Generation
Topic Modeling	Uncovering hidden topics in text collections using unsupervised learning.	(Unsupervised) latent class identification
Speech to Text	Converting spoken language into written text.	Text Generation
Text to image by diffusion model	Converting text to image	Image generation

Historical context



1950s–1960s: Early Days

- Initial experiments in machine translation and automated reasoning.
- Example: Georgetown-IBM experiment, 1954, first machine translation from Russian to English

1980–2010: Statistical Revolution

- Statistical models, algorithms like Hidden Markov Models.
- Machine Learning, Language Processing, Word vectorization (Bag-of-Words, TF-IDF, etc.)

2010s: Deep Learning Breakthroughs

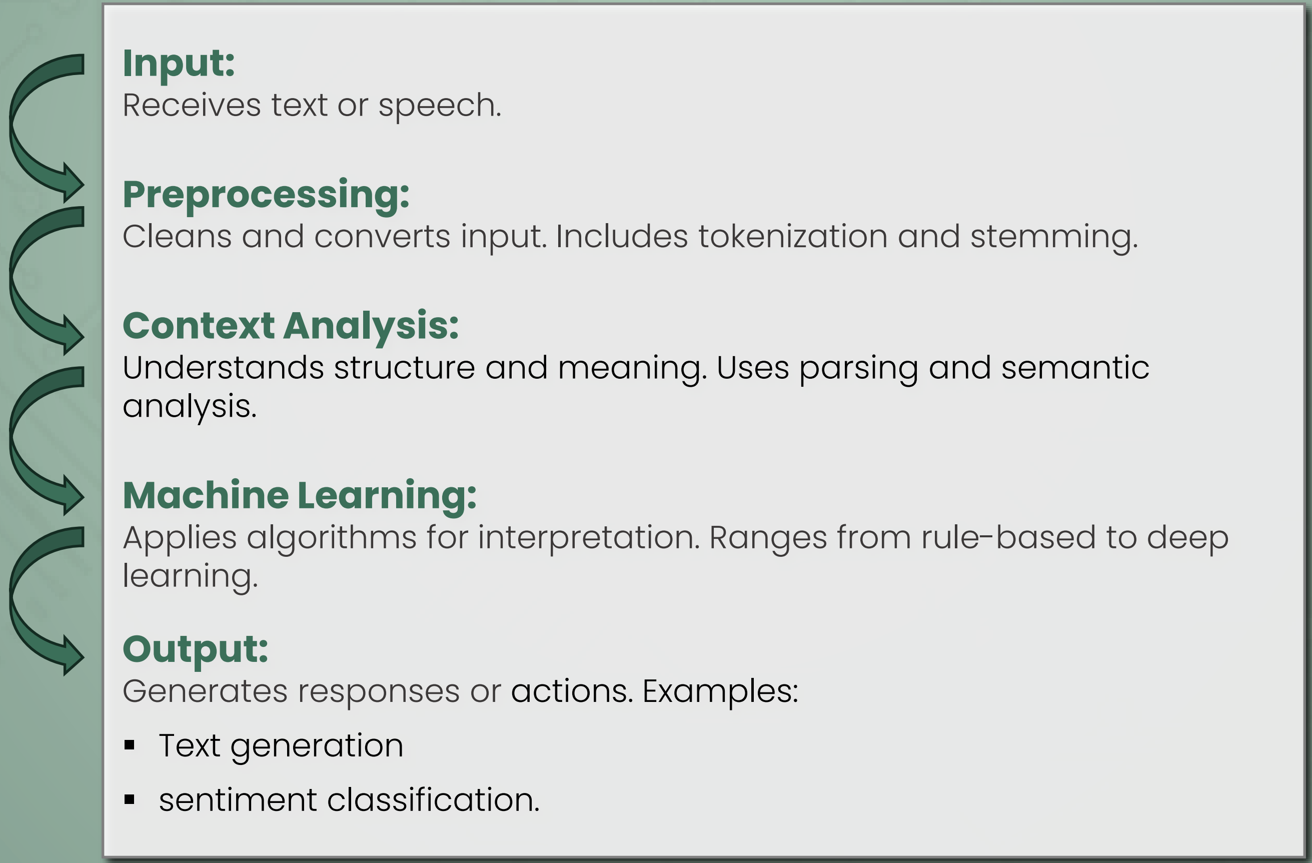
- Adoption of deep learning and neural networks.
- Emergence of models like Word2Vec and BERT.

2020s: Advanced Language Models

- State-of-the-art models like GPT and Transformer architectures.
- Unprecedented capabilities in language generation and understanding.

How NLP Works

Generic workflow of Natural Language Processing



Input:
Receives text or speech.

Preprocessing:
Cleans and converts input. Includes tokenization and stemming.

Context Analysis:
Understands structure and meaning. Uses parsing and semantic analysis.

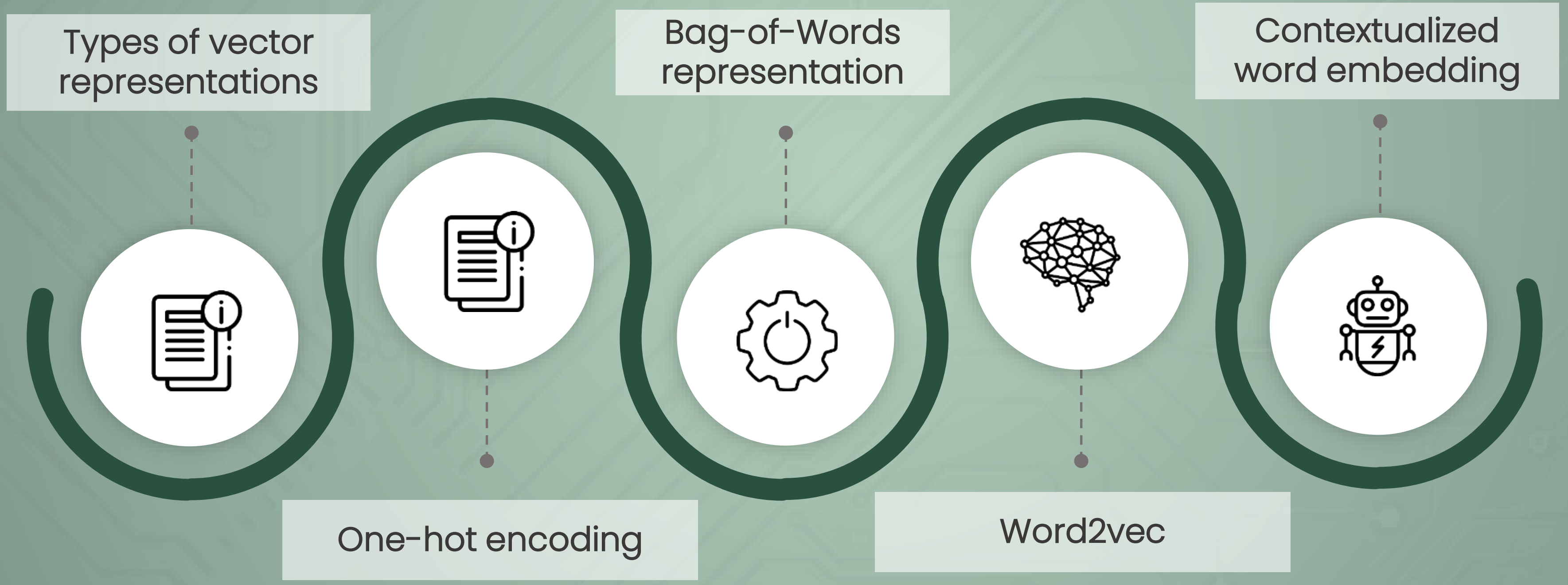
Machine Learning:
Applies algorithms for interpretation. Ranges from rule-based to deep learning.

Output:
Generates responses or actions. Examples:

- Text generation
- sentiment classification.

Vector Representations

…involves the organization, summarization, and visualization of data. It provides simple summaries about the sample and the measures.



Types of vector representations

One-hot encoding

Bag-of-Words representation

Word2vec

Contextualized word embedding

Vector Representations

…enable us to convert textual data into numerical forms that can be processed by machine learning models.

One-Hot Encoding:

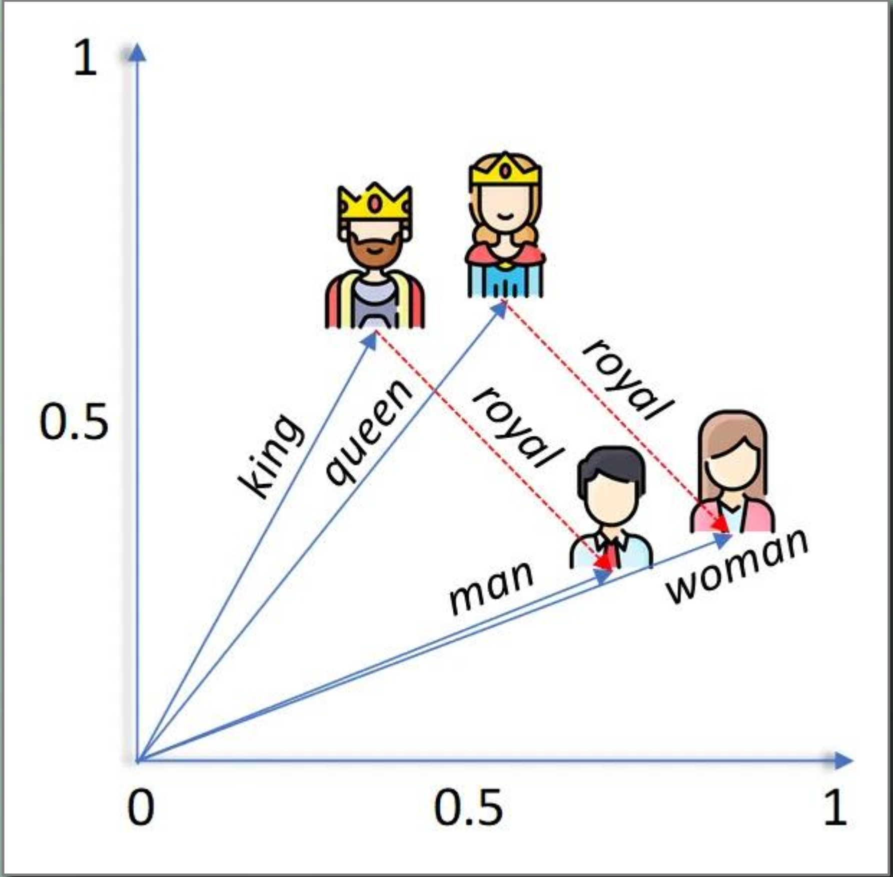
- process that converts categorical variables into binary vector representation
- only one element is "hot" (set to 1)
- all other elements are "cold" (set to 0), uniquely representing each category.

Word2Vec:

- models that are used to produce word embeddings
- where words are represented as vectors in continuous vector space
- based on their context and semantic similarity

Contextualized Word Embedding:

- meaning of a word is dynamically encoded
- based on the word's context within a sentence
- leading to different embeddings for the same word in different contexts
- thereby capturing nuances in usage and meaning



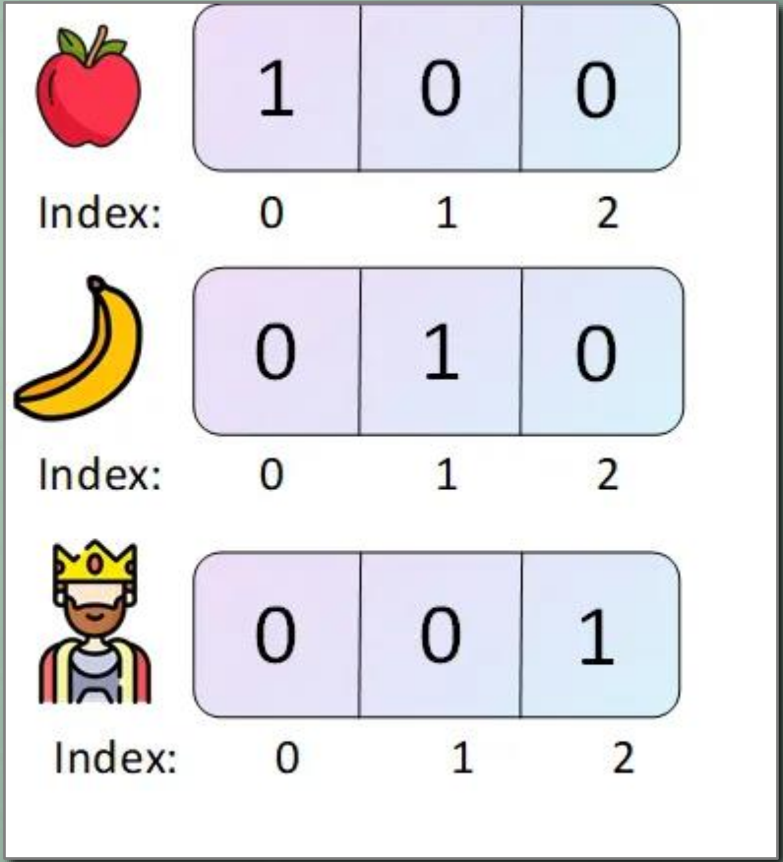
One-hot encoding

… is basic method for vectorizing words in NLP

- Each word in a vocabulary is represented as a binary vector:
  - A vector of all zeros except for a single 1
  - indicating the word's presence.

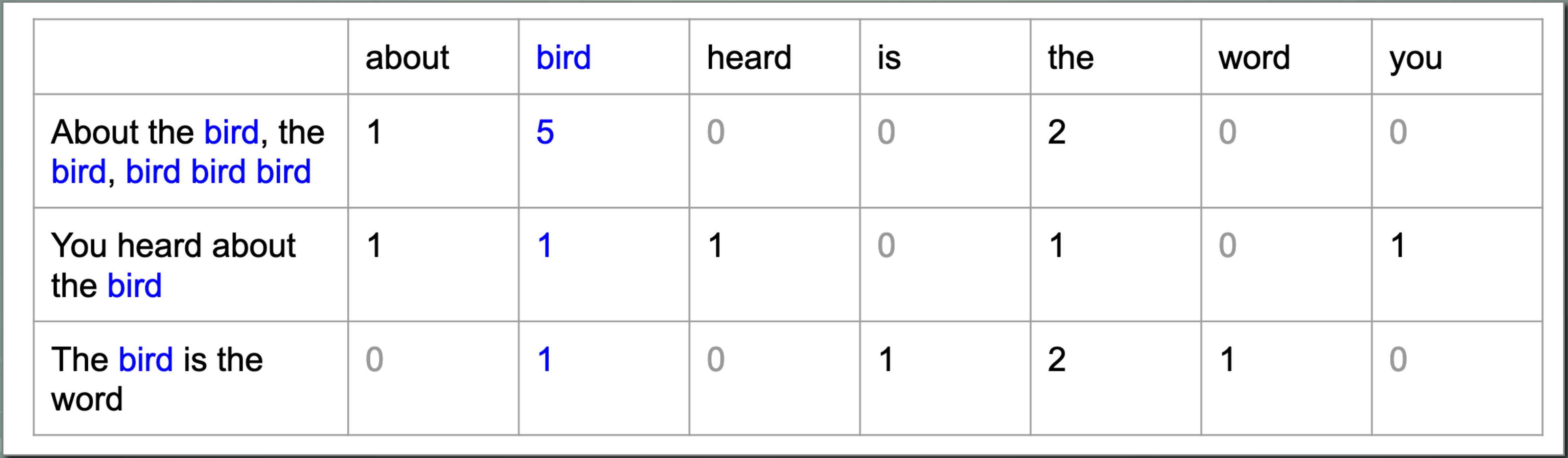
- Simple and intuitive, but has limitations:
  - Doesn't capture semantic relationships between words.
  - High dimensionality in large vocabularies.

- Often used as a starting point for more advanced techniques.



Bag-of-Words representation

… transforms text into numerical vector, where each unique word is represented by feature and value indicates frequency of word in the document, disregarding grammar, word order.



	about	bird	heard	is	the	word	you
About the bird, the bird, bird bird bird	1	5	0	0	2	0	0
You heard about the bird	1	1	1	0	1	0	1
The bird is the word	0	1	0	1	2	1	0

Word2Vec

…is a popular word embedding technique that represents words in a continuous vector space.

Key Features:

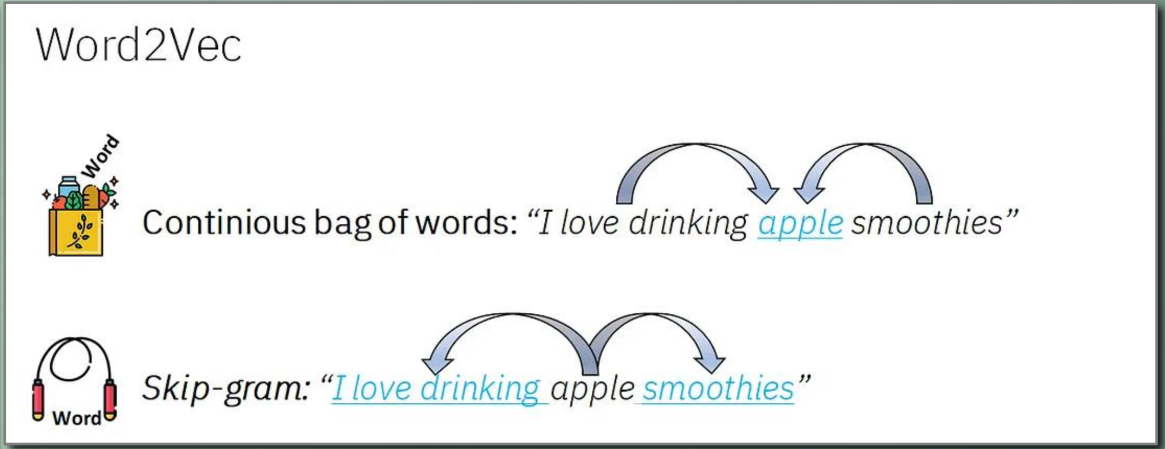
- Captures semantic meaning
- Words with similar meanings are closer in the vector space

Two Training Methods:

- Continuous Bag of Words (CBOW): Predicts a word given its context
- Skip-Gram: Predicts context words from a given target word

Benefits:

- Enables better performance in NLP tasks
- Helps in capturing semantic relationships (e.g., "king" - "man" + "woman" ≈ "queen")


Word2Vec

Continious bag of words: *“I love drinking apple smoothies”*

Skip-gram: “I love drinking apple smoothies”

Contextualized Word Embedding

Unlike traditional embeddings, contextualized embeddings generate word representations based on their specific context within a sentence, allowing for dynamic meanings.

Key Features:

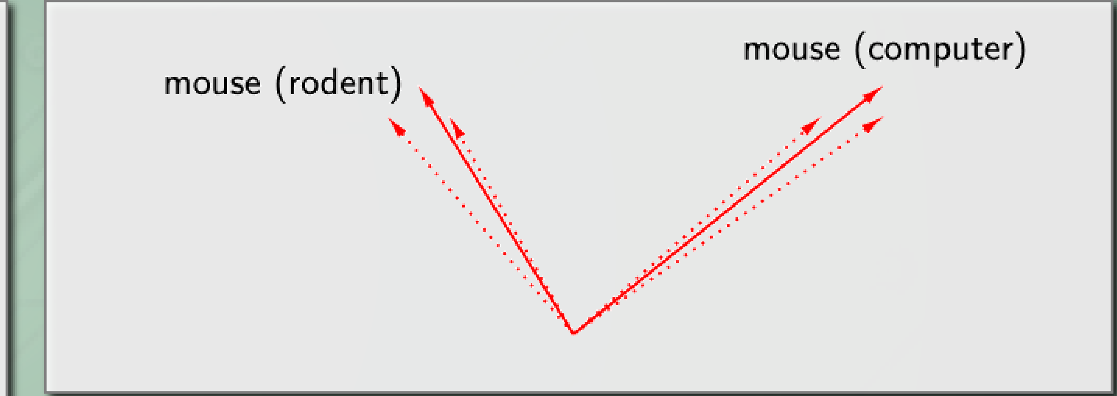
- Words can have different vectors in different contexts.
- Captures polysemy: a word's ability to have multiple meanings.

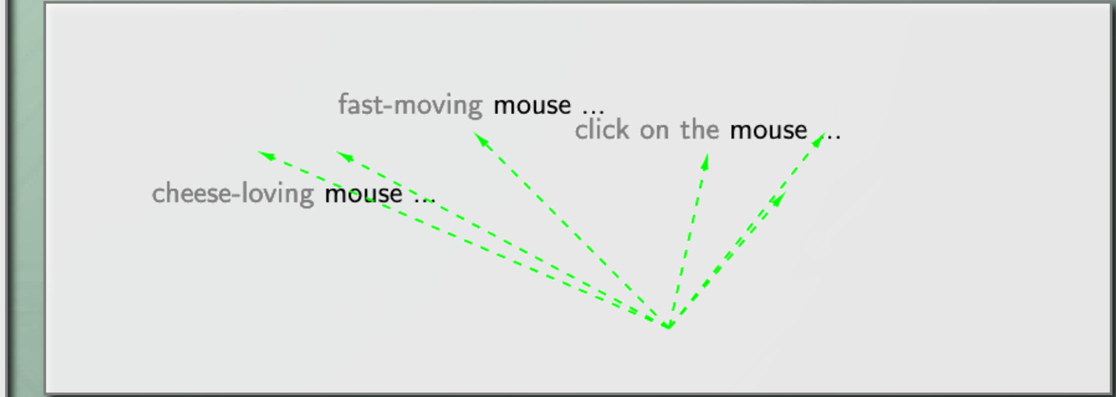
Popular Models:

- ELMo (Embeddings from Language Models)
- BERT (Bidirectional Encoder Representations from Transformers)

Benefits:

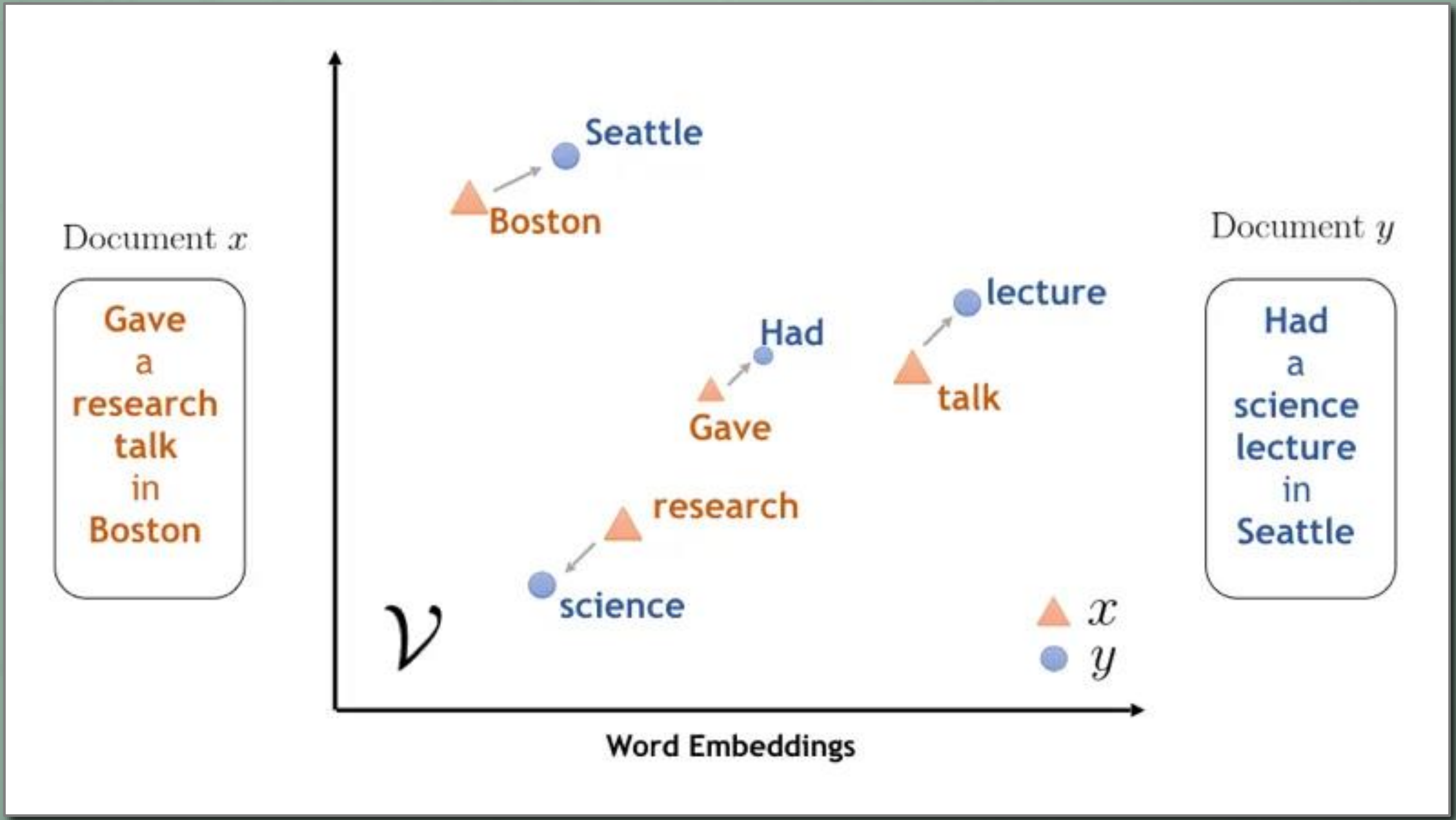
- Enhanced understanding of word nuances and meanings.
- Improved performance on downstream NLP tasks.





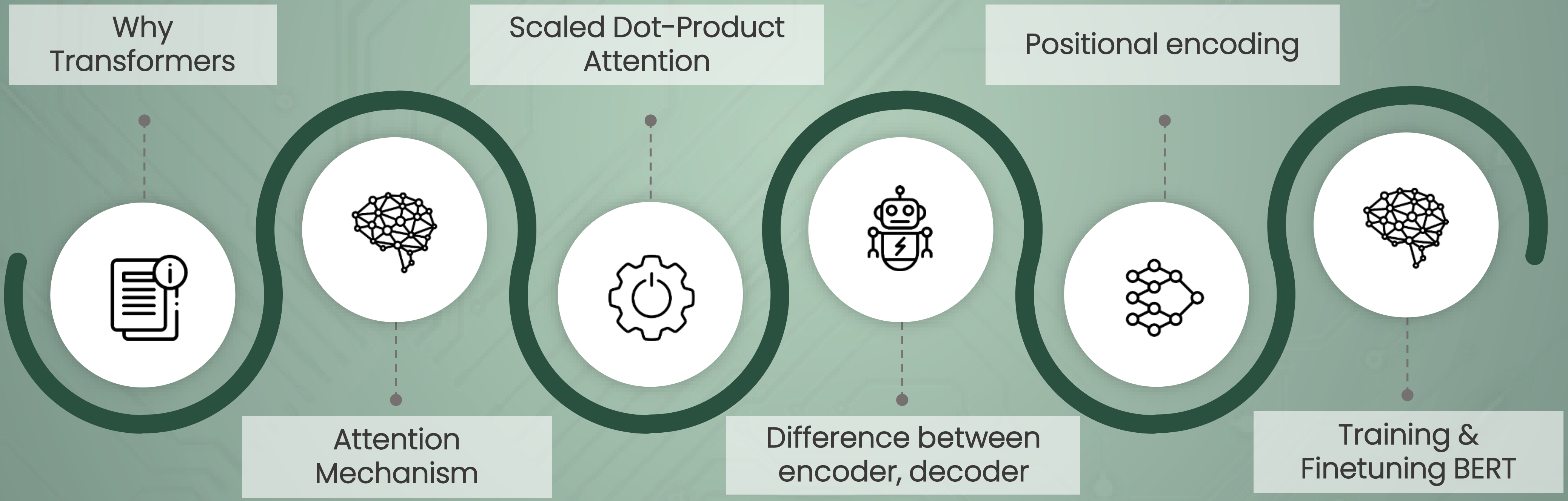
Contextualized Word Embedding

… allows you to calculate contextual distances of sentences, paragraphs, etc.



Transformers: Basics and usage for sentiment analysis

…involves the organization, summarization, and visualization of data. It provides simple summaries about the sample and the measures.



Why Transformers

Attention Mechanism

Scaled Dot-Product Attention

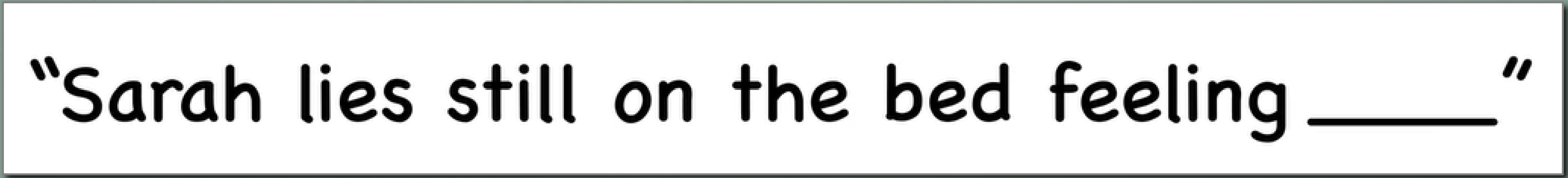
Difference between encoder, decoder

Positional encoding

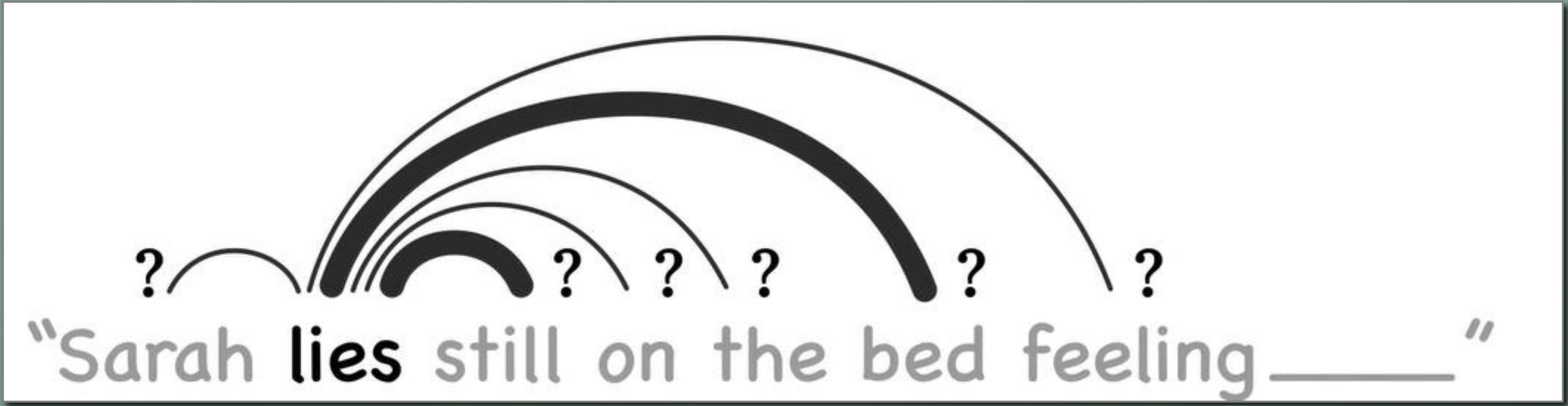
Training & Finetuning BERT

Attention Mechanism

What is the most probable missing word? It is determined by putting the other words into context to each other.



“Sarah lies still on the bed feeling ____“



Why Transformers are significant

Transformers excel at modeling sequential data like natural language.

Nowadays encoders, decoders are used separately.

Encoder:

- Sentiment Analysis
- Text classification

Decoder:

- Conversation (ChatGPT)
- Translation

Comparison with RNNs:

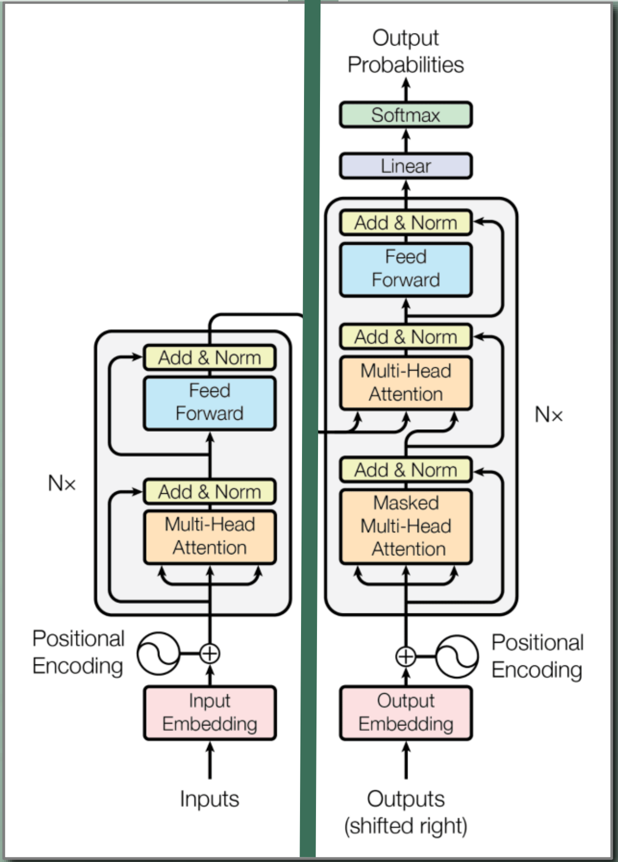
- Parallelizable and efficient on GPUs & TPUs.
- Replaces recurrence with attention for simultaneous computations.
- Outputs computed in parallel, unlike RNNs' series.

Advantages Over RNNs and CNNs:

- Captures distant or long-range contexts in data.
- Connects distant positions in sequences for longer connections.
- Uses attention to access entire input at each layer, unlike RNNs, CNNs.

Unique Characteristics:

- No assumptions on temporal/spatial relationships.
- Ideal for processing sets of objects (e.g., StarCraft units).



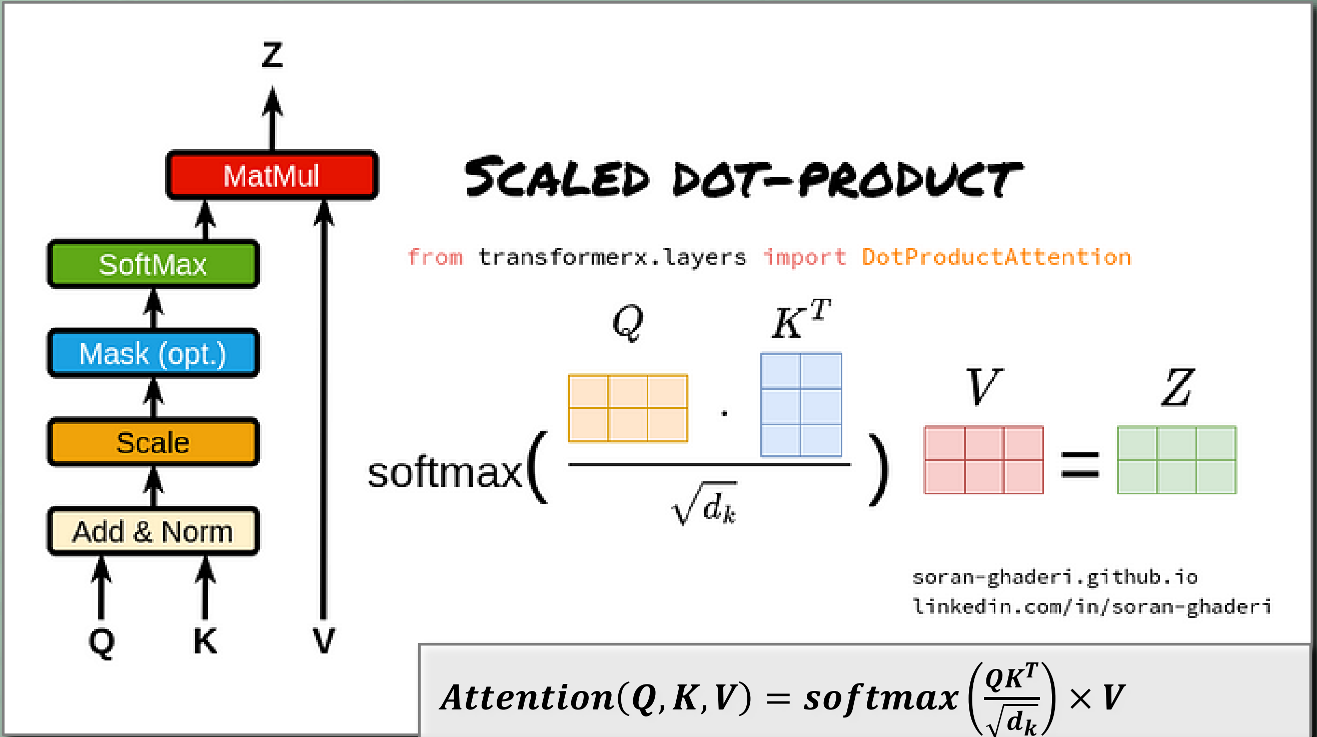
Scaled Dot-Product Attention

…is a mechanism used in attention models that calculates attention scores based on dot product of query and key, scaled down by square root of their dimensionality.

- Queries: derived from input data, represent focus of model
- Keys: also derived from input data, interpretation: "labels" for the input data
- Values: are weighted based on compatibility of query and corresponding key

How it works

- For each query, attention mechanism computes score with each key in input. Score represents how well query aligns with particular key.
- Scores are used to create a weighted combination of the values.
- If key aligns well with query, value associated with that key gets larger weight in final output.



```
from transformerx.layers import DotProductAttention
```

$$\text{softmax}\left(\frac{Q \cdot K^T}{\sqrt{d_k}}\right) V = Z$$

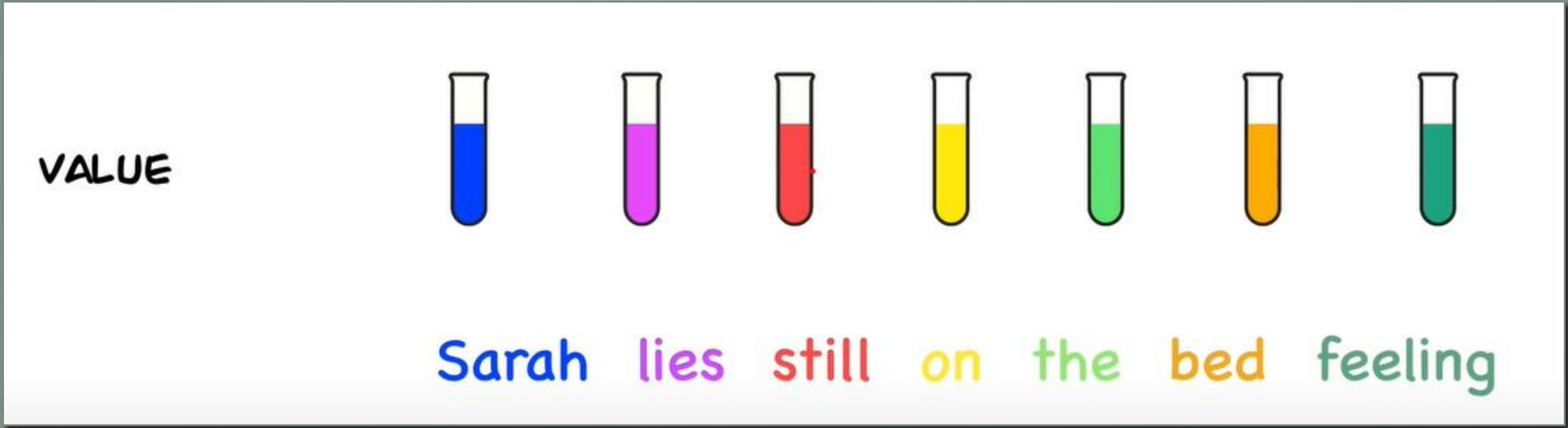
soran-ghaderi.github.io
linkedin.com/in/soran-ghaderi

$$\boldsymbol{Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right) \times V}$$

- Q, K, V: Query, Key, and Value vectors respectively
- $\boldsymbol{d_k}$: dimensionality of query/key vectors

Attention Mechanism: Value (V) vectors for all words

These contain actual information of each word that will be aggregated to form output

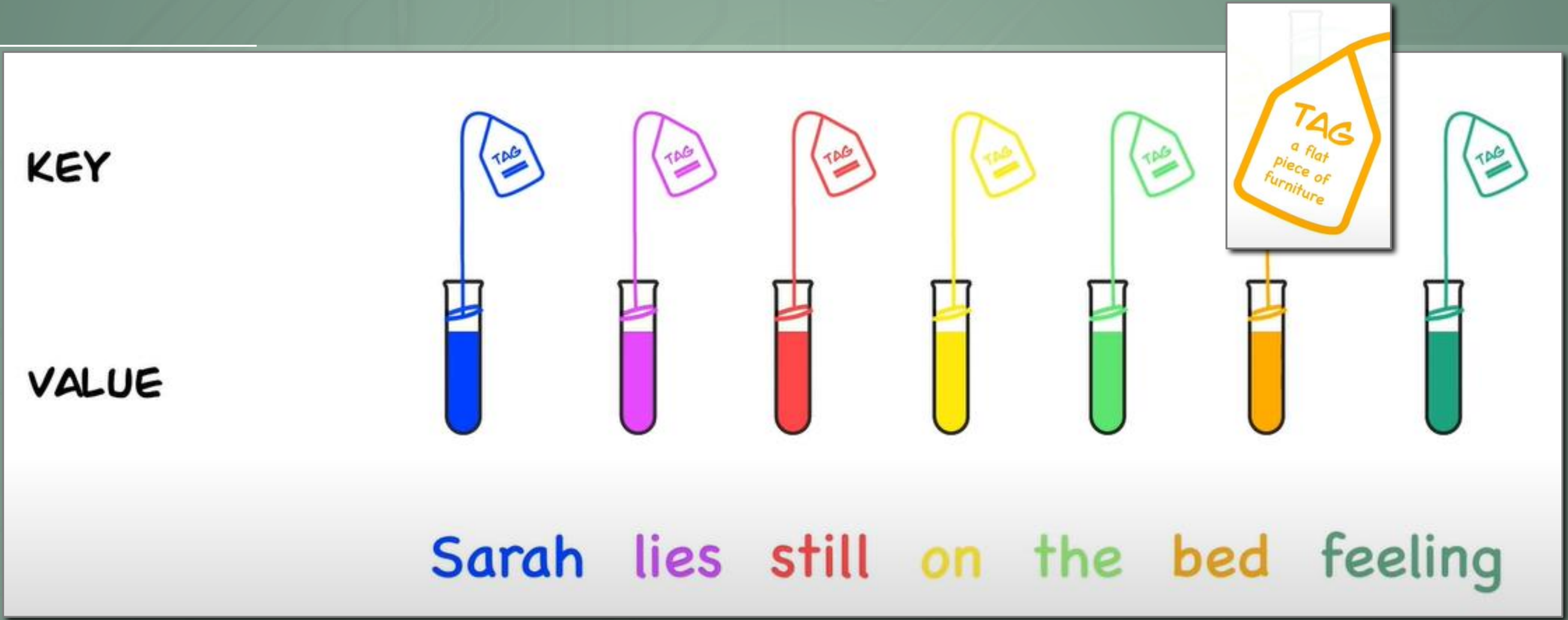


Position is not important for coding, if word occurs several times it receives the same code

[0.1, 0.2] [0.3] [0.4] [0.5] [0.6] [0.7] [0.8, 0.9]

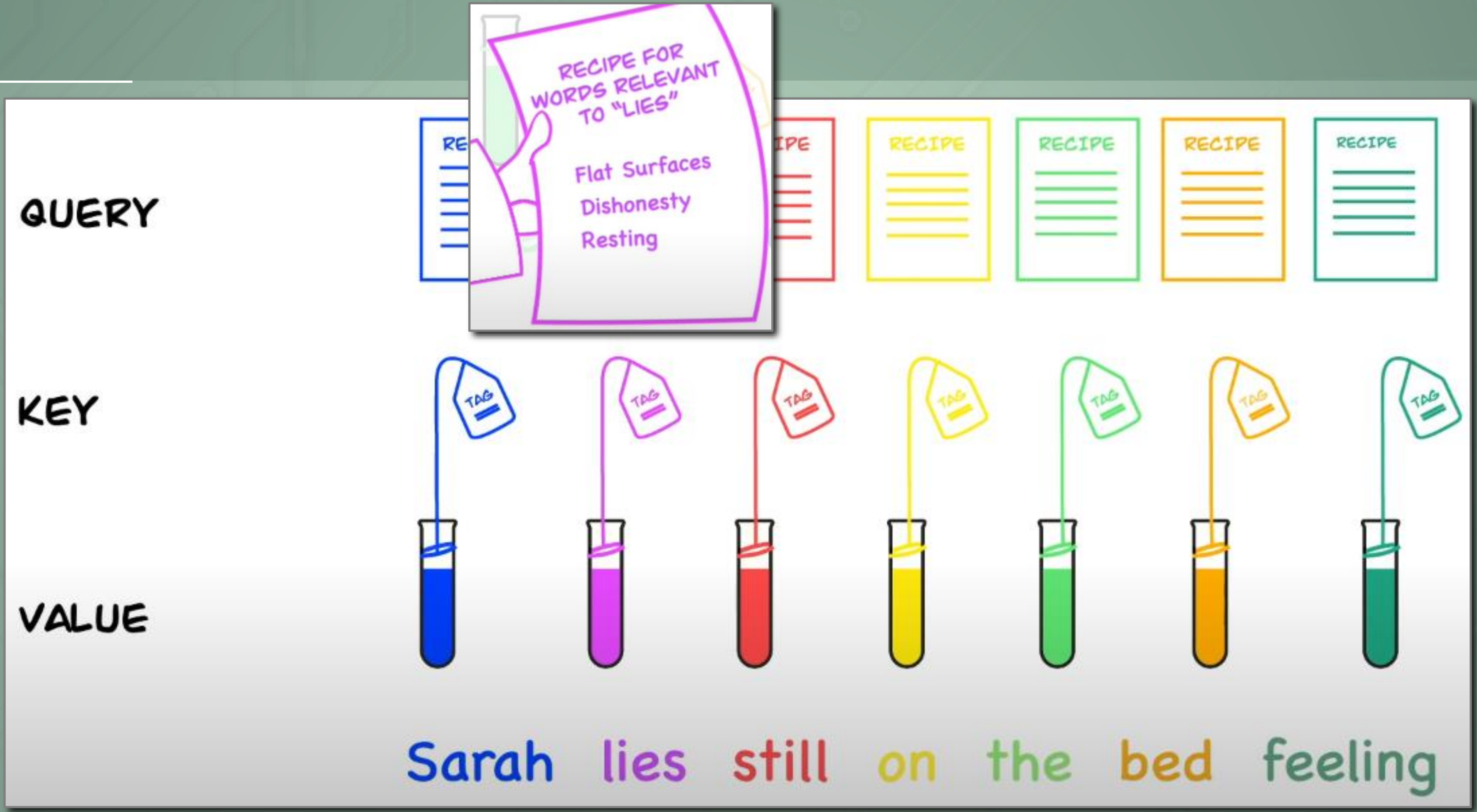
Attention Mechanism: Key (K) vectors for all words

These represent how each word in the sentence can be queried



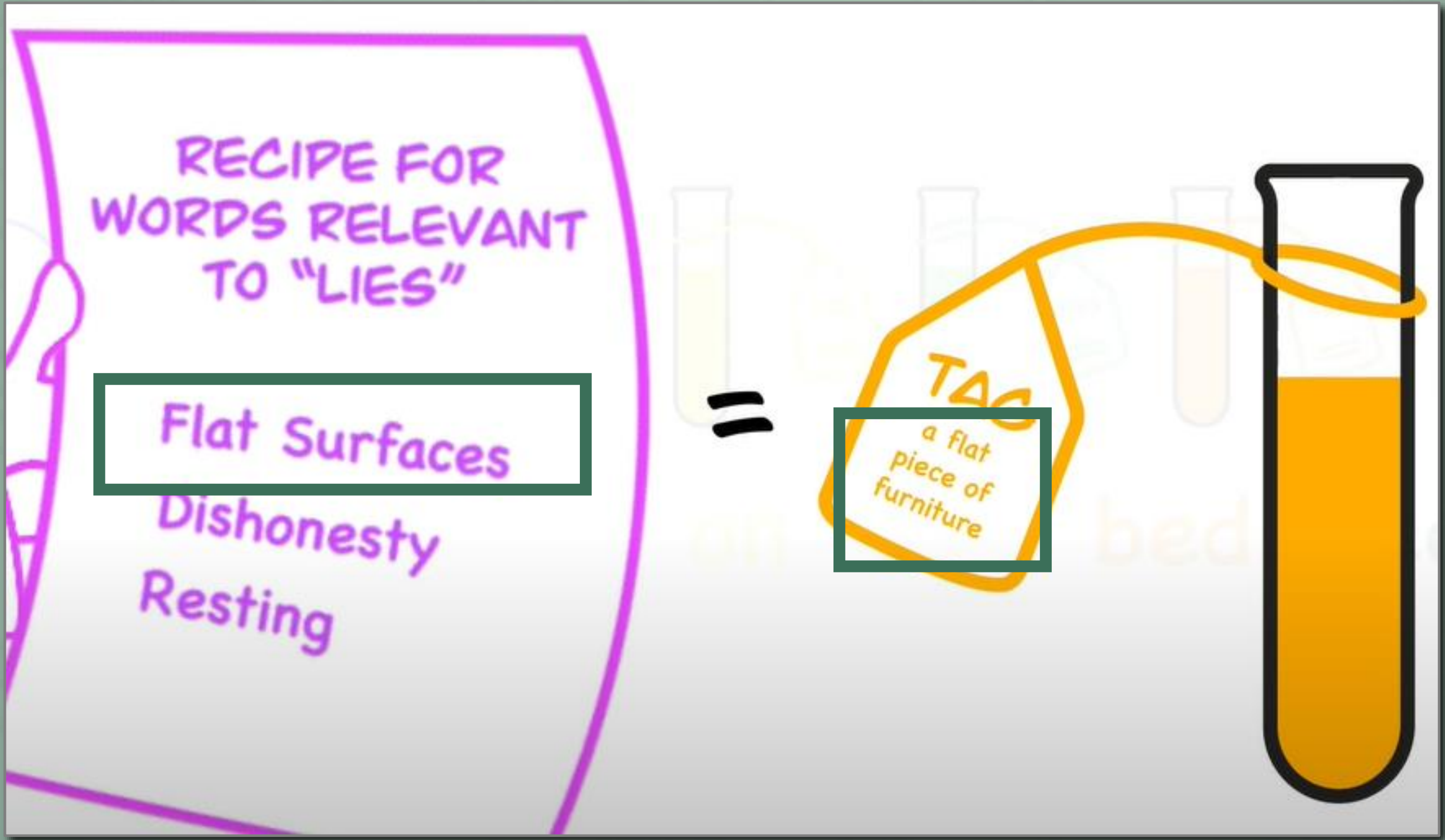
Attention Mechanism: Query (Q) vectors

... are used to represent the word in question, seeking information from other words.



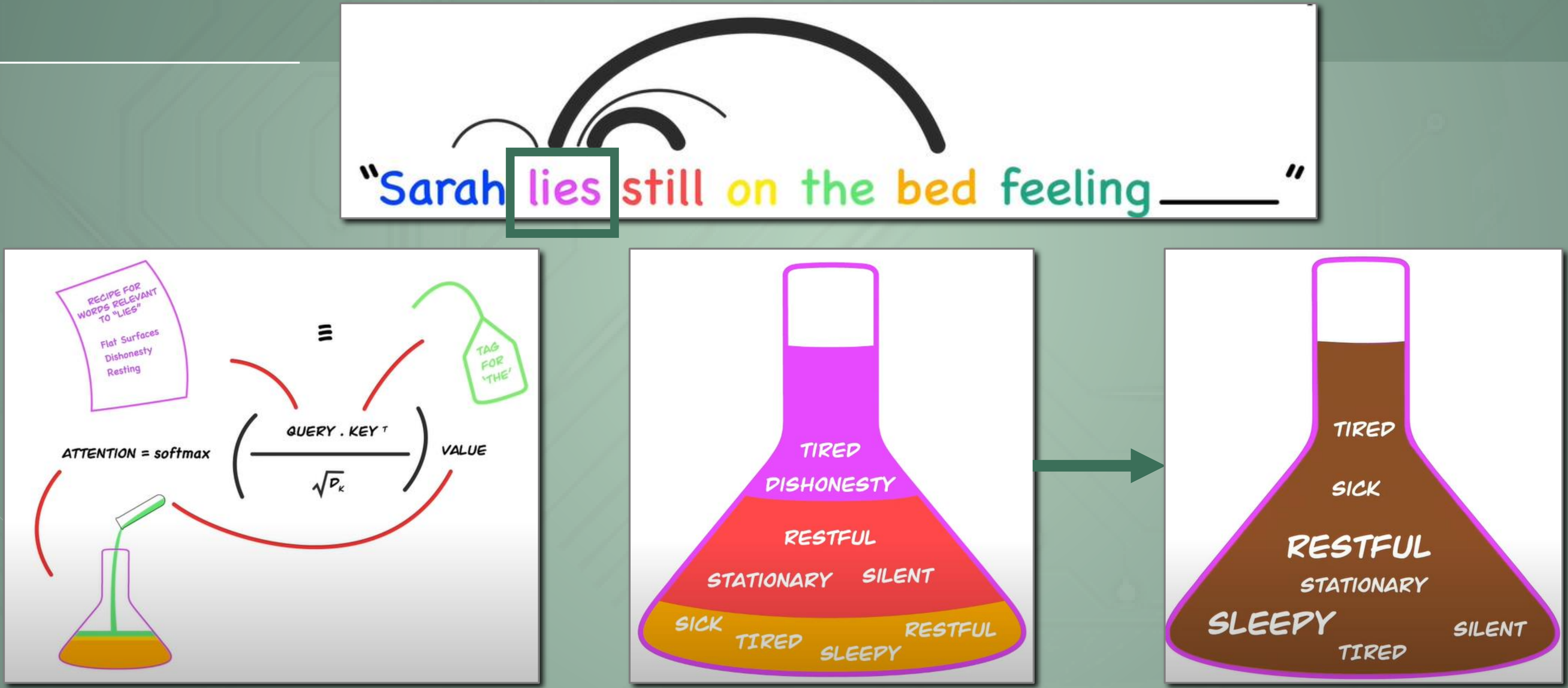
Attention Mechanism: Connecting query and key

… "Flat Surfaces" from Recepe and "a flat piece of furniture" are similar


RECIPE FOR WORDS RELEVANT TO "LIES"
Flat Surfaces
Dishonesty
Resting
=
TAG
a flat piece of furniture

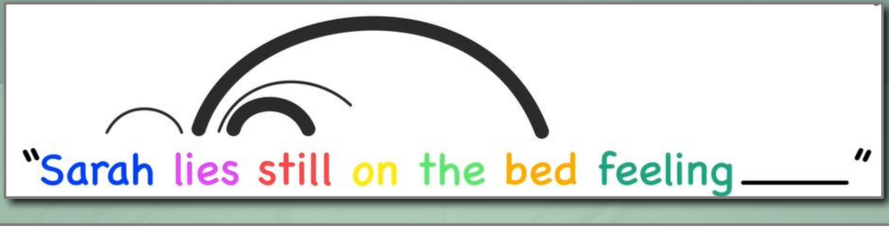
Attention

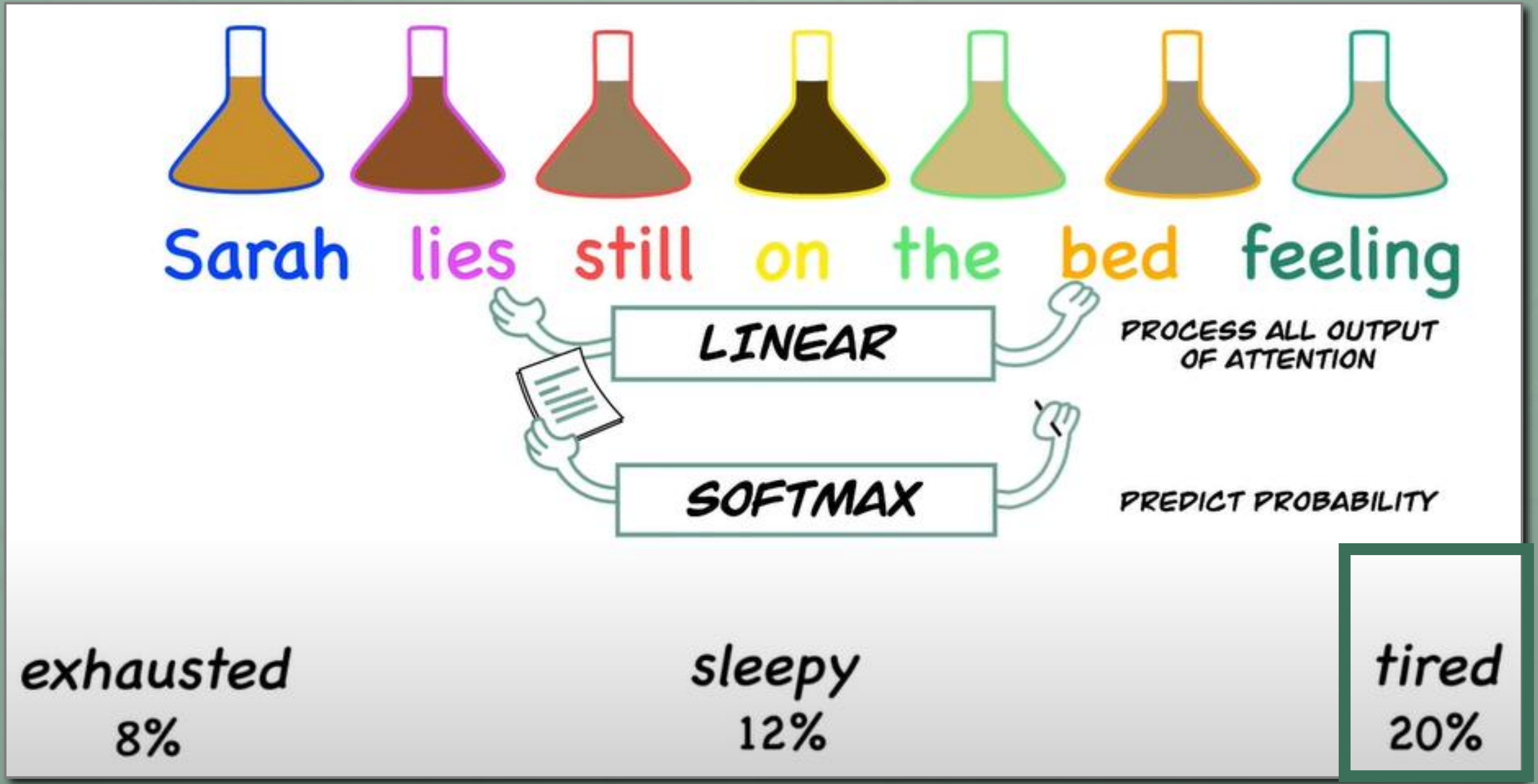
… of the word “lies”


“Sarah lies still on the bed feeling ____”
RECIPE FOR WORDS RELEVANT TO "LIES"
Flat Surfaces
Dishonesty
Resting
≡
TAG FOR 'THE'
ATTENTION = softmax $\left(\frac{QUERY . KEY^{T}}{\sqrt{D_K}} \right)$ VALUE
TIRED
DISHONESTY
RESTFUL
STATIONARY
SILENT
SICK
TIRED
SLEEPY
RESTFUL
TIRED
SICK
RESTFUL
STATIONARY
SLEEPY
SILENT
TIRED

Attention Mechanism

… calculates the word with highest probability through linear transformation & softmax





Attention Mechanism

A technique in deep learning models, especially in sequence-to-sequence tasks, that allows the model to focus on specific parts of the input when producing an output.

Key Features:

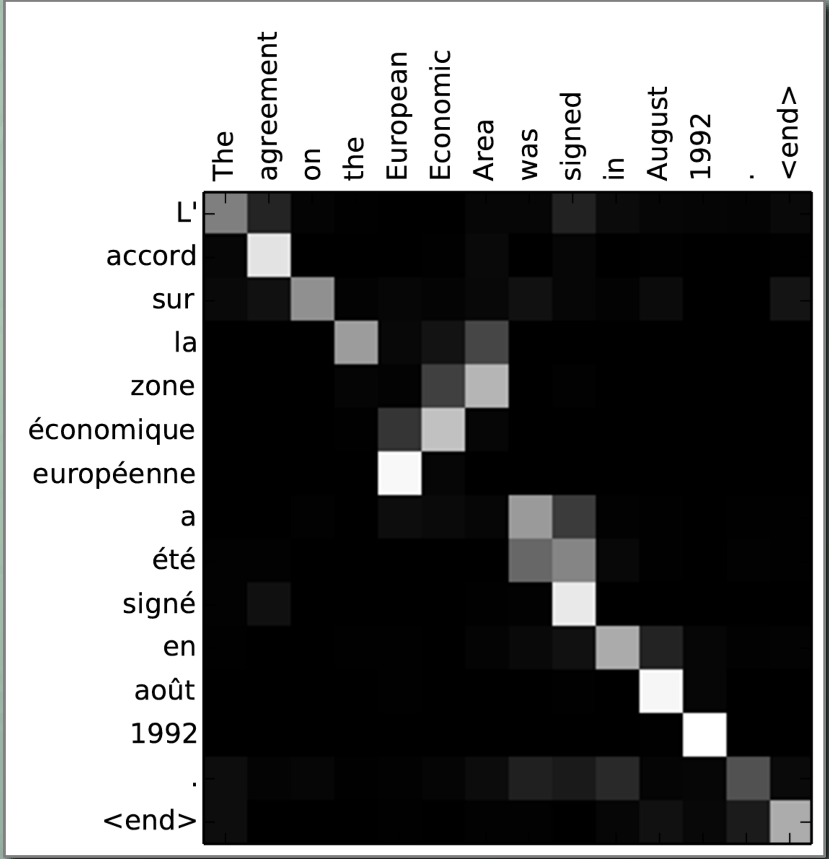
- Dynamically weighs input elements.
- Enhances the capturing of long-range dependencies in sequences.

Usage:

- Machine Translation: Helps in aligning words in source and target languages.
- Text Summarization: Prioritizes crucial parts of the content.

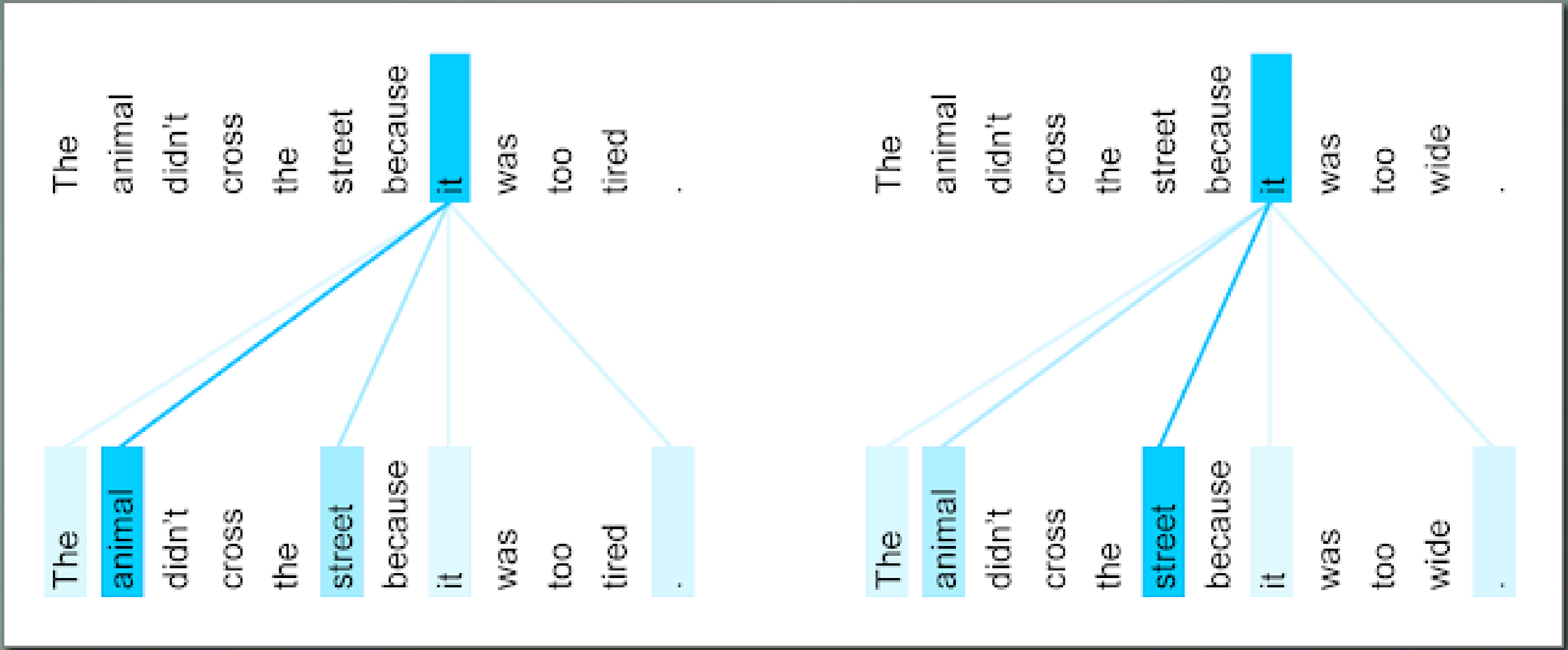
Benefits:

- Improves model's ability to remember long sequences.
- Enhances accuracy in tasks like translation and summarization.



Encoder self-attention distribution: word “it”

Meaning of the word „it“ depends on context. Any complex task like translation or even Sentiment Analysis of multi-sentence comments needs context awareness.



The animal didn't cross the street because it was too tired .
The animal didn't cross the street because it was too tired .
The animal didn't cross the street because it was too wide .
The animal didn't cross the street because it was too wide .

“it” means animal

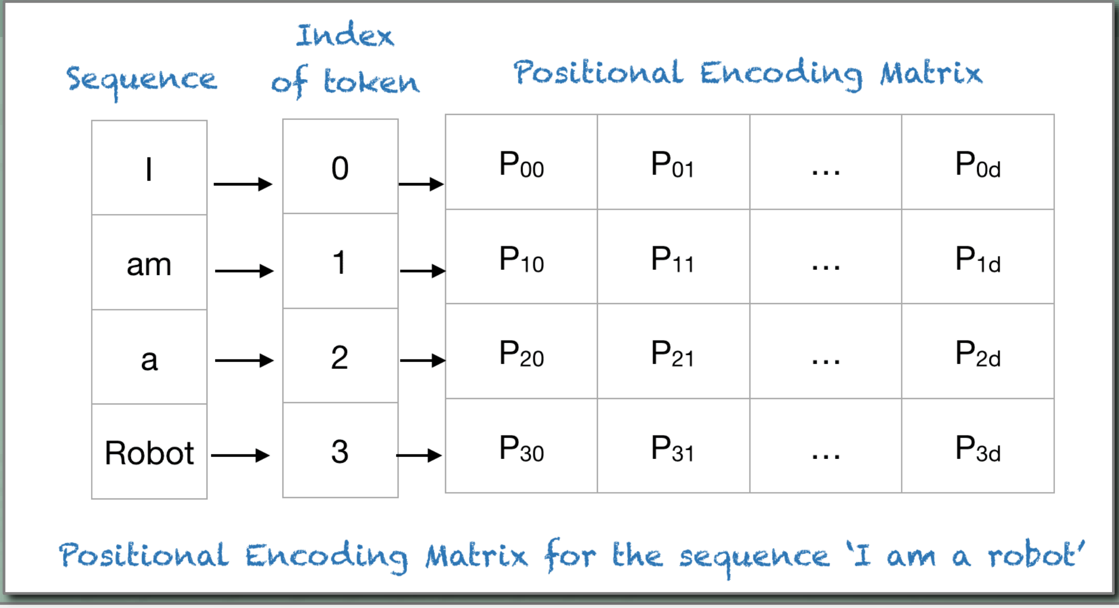
“it” means street

Difference between encoder, decoder

Initially, i.e. in 2014, transformer based translators have used both encoders and decoders. Encoder connectes words (tokens) in both directions, decoder predicts next word.

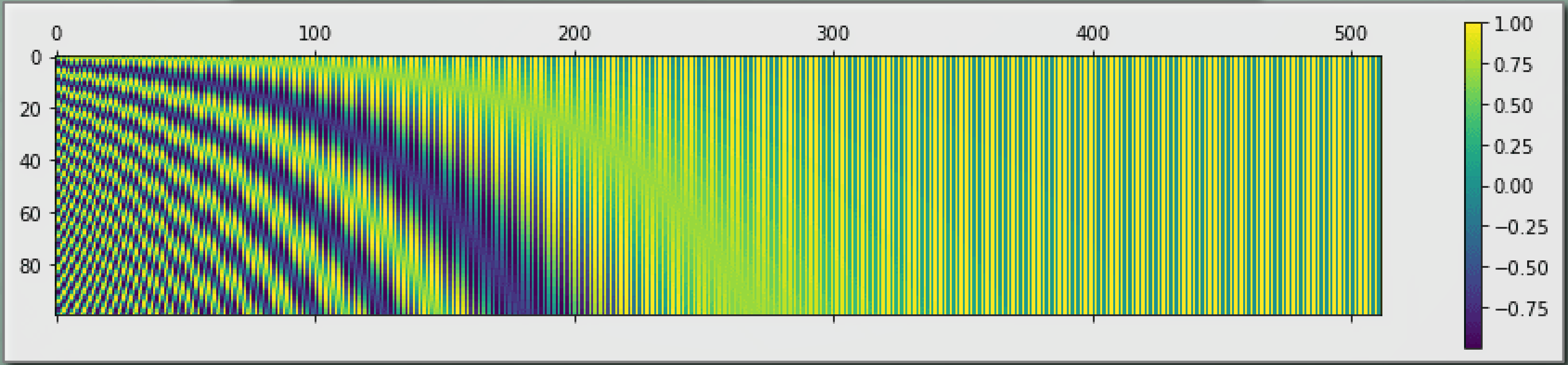
Positional Encoding

… provides sense of position for input tokens. Essential for models like Transformers that do not inherently process sequences in order. Overlapping trigonometric functions are used



Sequence		Index of token		Positional Encoding Matrix			
I	→	0	→	P_{00}	P_{01}	…	P_{0d}
am	→	1	→	P_{10}	P_{11}	…	P_{1d}
a	→	2	→	P_{20}	P_{21}	…	P_{2d}
Robot	→	3	→	P_{30}	P_{31}	…	P_{3d}

Positional Encoding Matrix for the sequence 'I am a robot'

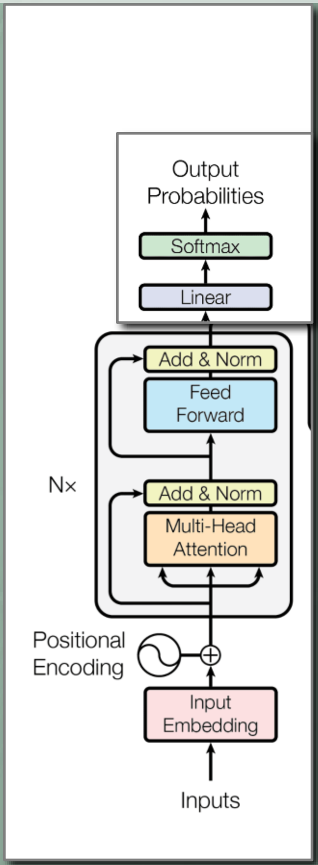


Training & Finetuning BERT for Sentiment Analysis

… Bidirectional Encoder Representations from Transformers (BERT) is a groundbreaking NLP model developed by Google. It classifies text data.

Encoder:

- Sentiment Analysis
- Text classification



Training BERT from Scratch (Optional & expensive)

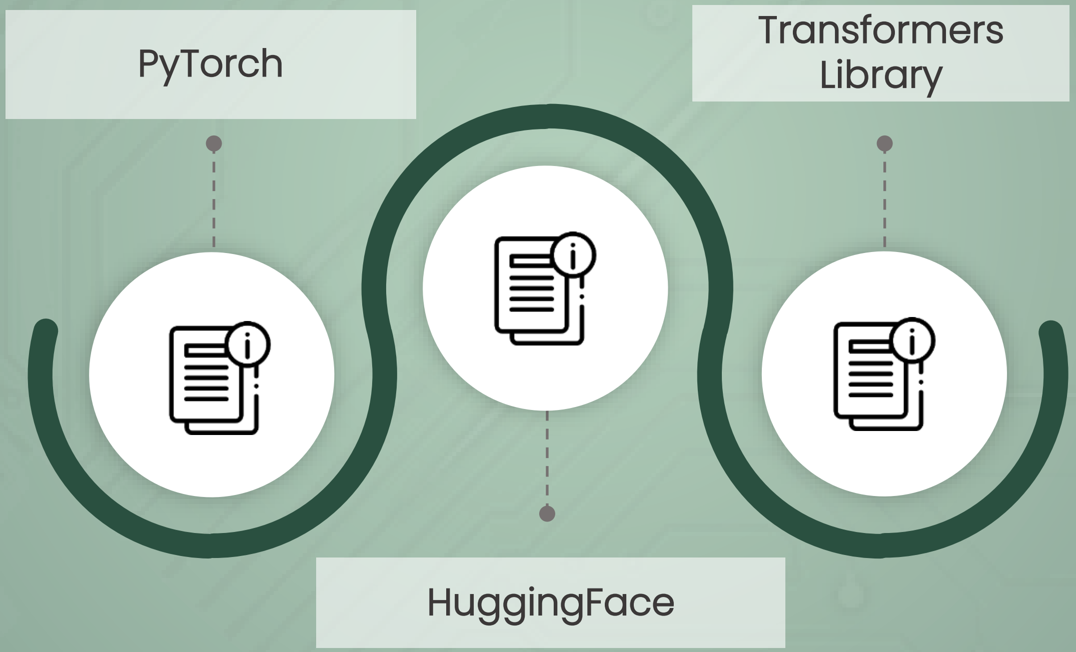
- Corpus Preparation: Gather a large, diverse corpus of text data.
- Pre-training Tasks: Masked Language Model (MLM) and Next Sentence Prediction (NSP).
- Computational Requirements:
  High, due to complexity and size of the model
  → better use HuggingFace

Finetuning Pre-trained BERT

- Objective: Adapt BERT to accurately classify sentiments in text (positive, negative, neutral).
- Dataset:
  Utilize sentiment-labeled dataset specific to the target domain
  (e.g., product reviews, social media posts, movie comments).
- Process:
  - Modify BERT's final layer to output sentiment categories.
  - Train model on sentiment dataset, adjusting parameters for optimal performance.

Programming Tools

…involves the organization, summarization, and visualization of data. It provides simple summaries about the sample and the measures.



PyTorch

HuggingFace

Transformers Library

What is PyTorch?

It is an open-source deep learning framework developed by Facebook's AI Research lab.

Key Features:

- Dynamic computational graph, offering flexibility in model design.
- Native support for GPU acceleration, enhancing performance.
- Intuitive tensor library similar to NumPy but with GPU support.

Applications:

- Research prototyping, offering ease and speed.
- Building deep learning models, from neural networks to complex architectures.

Community & Ecosystem:

- Active community contributing to its growth.
- Rich ecosystem with pre-trained models, tools, and libraries.

TensorFlow, PyTorch and Keras

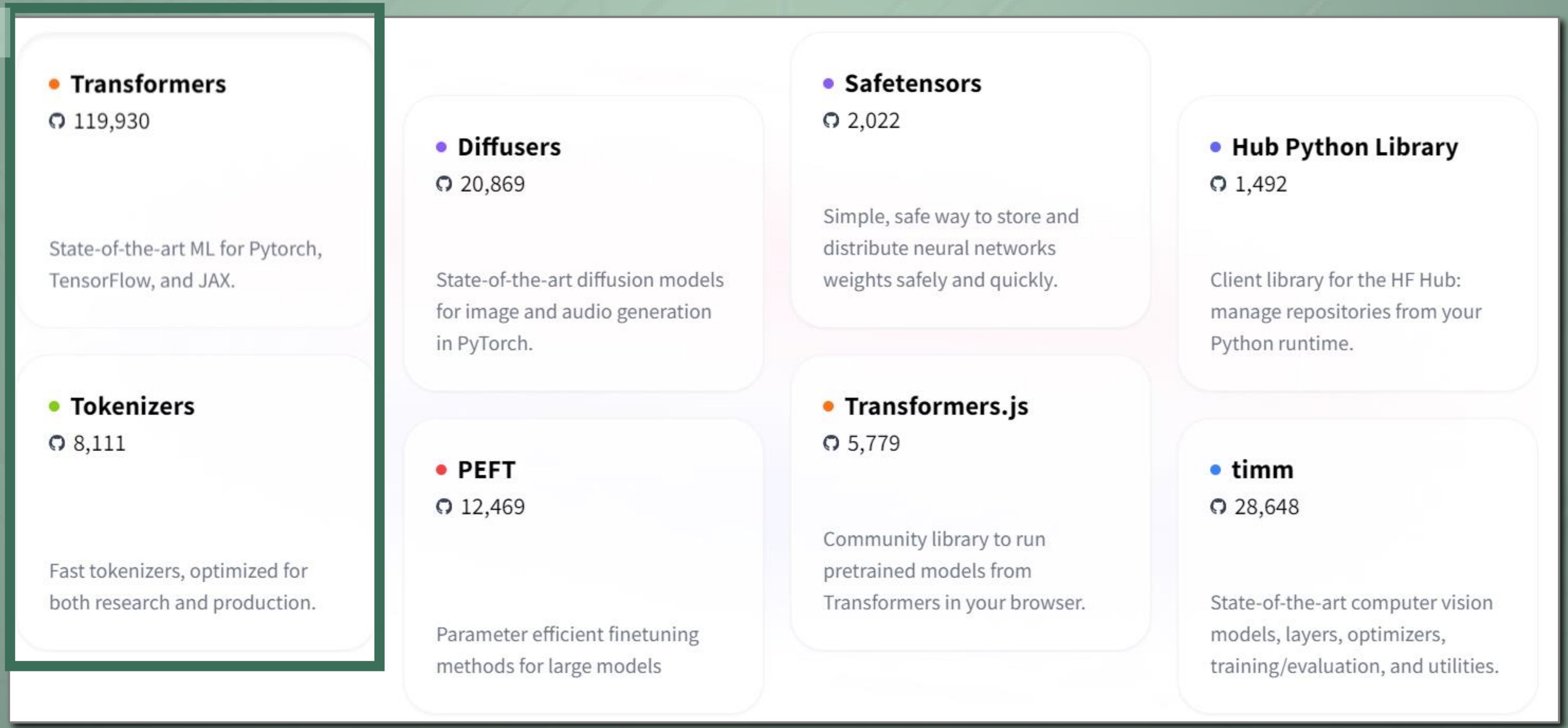
… are popular open-source libraries used for machine learning and deep learning.
Keras is API built on top of TensorFlow, PyTorch and others

	1 TensorFlow	2 PyTorch Focus	3 Keras
DEVELOPED BY	Google Brain team	Facebook's AI Research lab	François Chollet
EASE OF USE	Steeper learning curve	Easier to learn	Extremely user-friendly
FLEXIBILITY / CUSTOMIZATION	Very flexible	Very flexible	Limited flexibility
PERFORMANCE / SPEED	Excellent performance	Excellent performance	Performance is dependent on the backend used
USE CASES	Suitable for both production and research, especially in large-scale systems	Preferred for research, rapid prototyping, and academic purposes	Best for quick development of standard neural networks, less suited for research

Hugging Face

… offers 100s of 1000s NPL models, datasets, tokenizers, etc. and has vibrant open source community of AI researchers, engineers, enthusiasts contributing to AI advancement

Focus



● **Transformers**
119,930

State-of-the-art ML for Pytorch, TensorFlow, and JAX.

● **Tokenizers**
8,111

Fast tokenizers, optimized for both research and production.

● **Diffusers**
20,869

State-of-the-art diffusion models for image and audio generation in PyTorch.

● **PEFT**
12,469

Parameter efficient finetuning methods for large models

● **Safetensors**
2,022

Simple, safe way to store and distribute neural networks weights safely and quickly.

● **Transformers.js**
5,779

Community library to run pretrained models from Transformers in your browser.

● **Hub Python Library**
1,492

Client library for the HF Hub: manage repositories from your Python runtime.

● **timm**
28,648

State-of-the-art computer vision models, layers, optimizers, training/evaluation, and utilities.

HuggingFace: Transformers Library

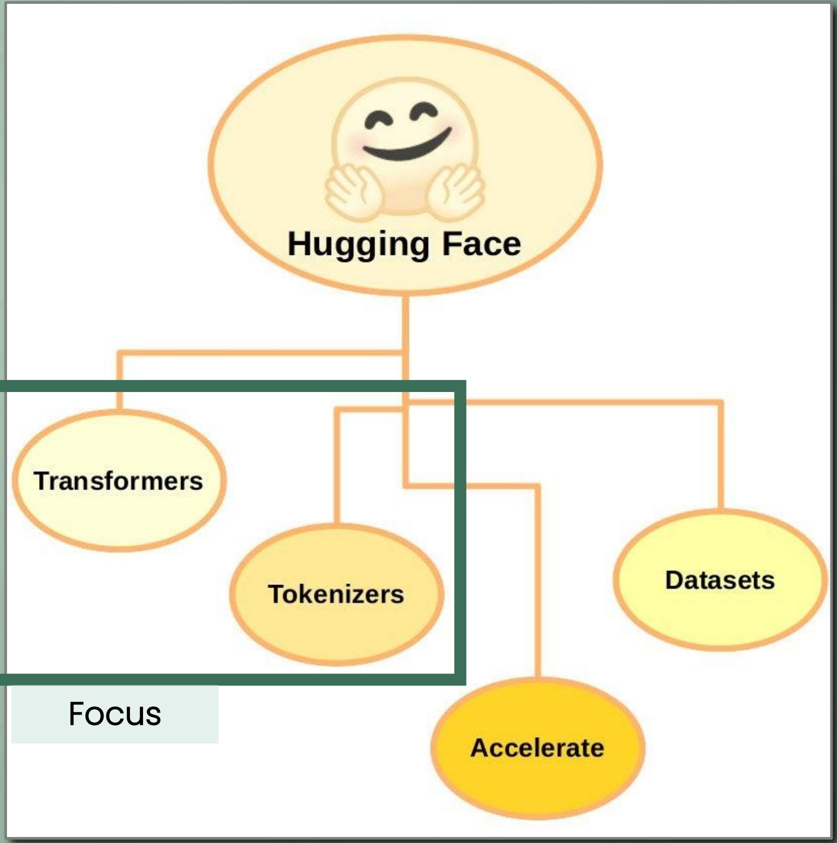
… provides tools to easily download, train state-of-the-art pretrained models. Using pretrained models can reduce time and resources required to train a model from scratch

Key Features:

- Comprehensive and Modular:
  Offers models like BERT, GPT, T5, etc, with simple interface for NLP tasks.
- Ease of Use:
  Enables easy model downloading, training, deployment with minimal code.
- Community-Driven:
  Actively supported by vast community, ensuring continuous updates, enhancements.

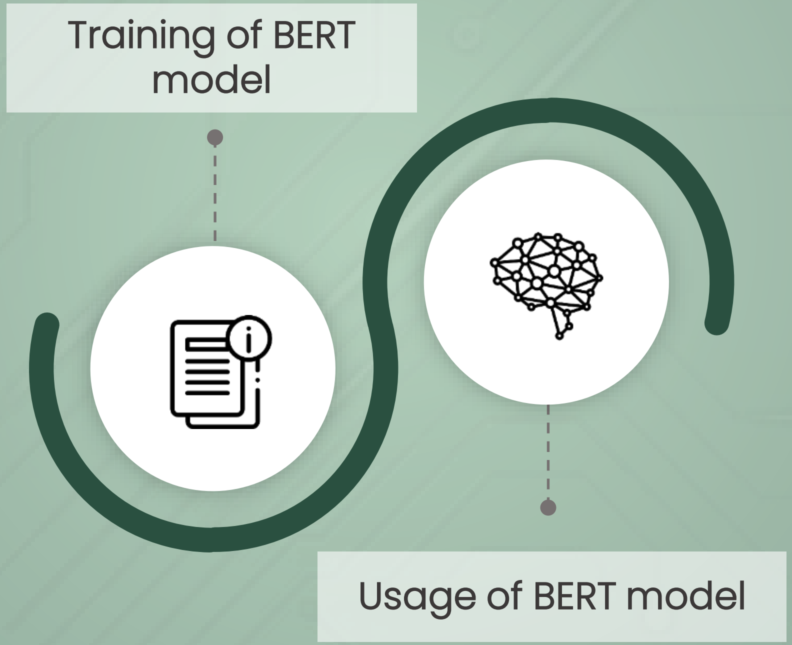
Applications:

- Research prototyping, offering ease and speed.
- Building deep learning models, from neural networks to complex architectures.



Example: Sentiments of movie comments

… Python example



Training of BERT model

Usage of BERT model

Links

i.e. sources for self-learning

	Title	Link
Basics of Natural Language Processing	An Introduction to Bag of Words (BoW) | What is Bag of Words?	An Introduction to Bag of Words (BoW) | What is Bag of Words?
	Understanding TF-IDF: A Traditional Approach to Feature Extraction in NLP	https://towardsdatascience.com/understanding-tf-idf-a-traditional-approach-to-feature-extraction-in-nlp-a5bfbe04723f
	Word Embedding and Word2Vec, Clearly Explained!!!	https://www.youtube.com/watch?v=viZrOnJclY0
	Word2Vec Explained	https://towardsdatascience.com/word2vec-explained-49c52b4ccb71
	Word2vec : NLP & Word Embedding	https://datascientest.com/de/word2vec
	Stanford XCS224U: NLU I Intro & Evolution of Natural Language Understanding	https://www.youtube.com/watch?v=K_Dh0Sxujuc&list=PLoROMvodv4rOwvldxftJTmoR3kRcWkJBp
	Introduction | NLP Tutorial For Beginners In Python	https://www.youtube.com/watch?v=R-AG4-qZs1A&list=PLeo1K3hjS3uuvuAXhYjV2lMEShq2UYSwX

Links

i.e. sources for self-learning

	Title	Link
Transformer based NLP	How GPT Works by Archerman Capital	https://www.youtube.com/watch?v=Mhy7I4E6eXs
	The math behind Attention: Keys, Queries, and Values matrices	https://www.youtube.com/watch?v=UPtG_38Oq8o
	The Attention Mechanism in Large Language Models	https://www.youtube.com/watch?v=OxCpWwDCDFQ
	Let's build GPT: from scratch, in code, spelled out.	https://www.youtube.com/watch?v=kCc8FmEb1nY
	State of GPT | BRK216HFS	https://www.youtube.com/watch?v=bZQun8Y4L2A
	Stanford CS224N: Natural Language Processing with Deep Learning	https://www.youtube.com/playlist?list=PLoROMvodv4rMFqRtEuo6SGjY4XbRIVRd4
	Data Science: Transformers for Natural Language Processing	https://www.udemy.com/course/data-science-transformers-nlp/learn/lecture/32714220
	Awesome-LLM-KG	https://github.com/RManLuo/Awesome-LLM-KG
	Transformers in Action: Attention Is All You Need	https://towardsdatascience.com/transformers-in-action-attention-is-all-you-need-ac10338a023a
	Attention Is All You Need	https://arxiv.org/abs/1706.03762
	BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding	https://arxiv.org/abs/1810.04805
	What exactly happens when we fine-tune BERT?	https://towardsdatascience.com/what-exactly-happens-when-we-fine-tune-bert-f5dc32885d76
	Sentiment Analysis in 10 Minutes with BERT and TensorFlow	https://towardsdatascience.com/sentiment-analysis-in-10-minutes-with-bert-and-hugging-face-294e8a04b671

ChatGPT/Dall-E3 Prompts

Illustration of a digital tablet lying flat, displaying a chat application with emojis representing various moods. Around the tablet, there are holographic projections of diverse faces showing emotions ranging from joy to sorrow. The entire scene is bathed in a gradient transitioning from green to gray.

Photo of a split screen showcasing on one side, a close-up of a computer screen displaying text inputs with highlighted positive and negative words, and on the other side, diverse faces of people with varying moods from happy to sad. Overlaying this scene, a translucent gradient flows from vibrant green at the top left corner to muted gray at the bottom right.

Berliner Hochschule für Technik

About me

Dr. Harald Stein

Data Scientist	~ 8 years experience
Algotrader	~ 4 years experience
Ph.D.	in Economics, Game Theory
LinkedIn:	https://www.linkedin.com/in/harald-stein-phd-1648b51a
ResearchGate:	https://www.researchgate.net/profile/Harald-Stein