

# Natural Language Understanding

## From BoW to GPT-3 and Beyond

---

Team NLU

# Content

---

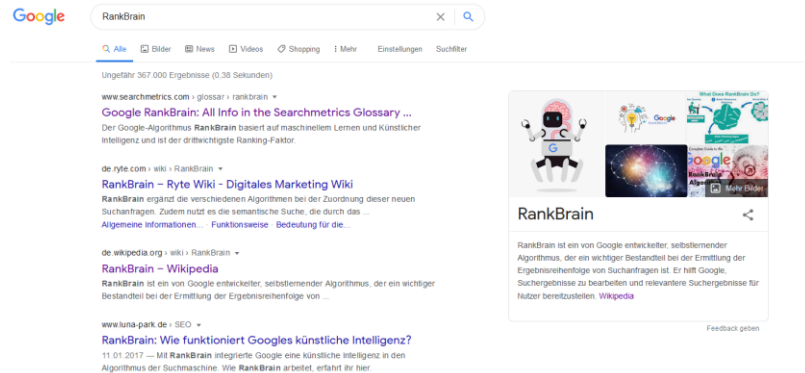
1. **Artificial Intelligence**
2. **Natural Language Understanding**
3. **„A brief history of NLU representations and applications“**
4. **Become a student assistant at IAIS.NLU**

# Artificial Intelligence

# AI in Everyday Use - I

## Search Engines, Speech Recognition, Translation, Virtual Assistants

### Smart Search Results



Source: [www.google.de](http://www.google.de)

### Live-Transcriptions

#### TRANSKRIPTION

#### Skype schreibt künftig Gespräche mit

Skype bietet künftig eine Transkription von Gesprächen an, damit gehörlose oder schwerhörige Menschen Audio- und Videokonferenzen folgen können. 2019 sollen auch bei Powerpoint Echtzeituntertitel eingeführt werden, die bei der Präsentation eingeblendet werden.

4. Dezember 2018, 7:34 Uhr, Andreas Donath



Skype mit automatischen Untertiteln

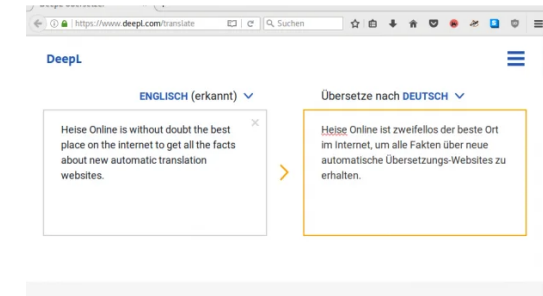
Source: <https://www.golem.de/news/transkription-skype-schreibt-kuenftig-gespraechе-mit-1812-138047.html>

### Translation

#### Maschinelle Übersetzer: DeepL macht Google Translate Konkurrenz

Linguee heißt jetzt DeepL und geht mit einem neuen maschinellen Übersetzer auf den Markt. Der basiert auf tiefen Convolutional Networks und dürfte effizienter übersetzen als Google Translate, das rekurrente Netze einsetzt.

Leszeit: 2 Min. In Pocket speichern



An dieser Übersetzung haben wir nichts auszusetzen.

29.08.2017 07:31 Uhr

Von Pina Merkert

Source:

<https://www.heise.de/newsticker/meldung/Maschinelle-Uebersetzer-DeepL-macht-Google-Translate-Konkurrenz-3813882.html>

# AI/NLU in the headlines

## GPT-3 and DALL-E

Comic, pencil drawing, cartoon style, researcher talking about large language models in front of students

Generate



Source: <https://labs.openai.com/>

# What is (hybrid) artificial intelligence?

One (of many) definitions

## Machine Learning

- E.g. neural networks

## Hardware

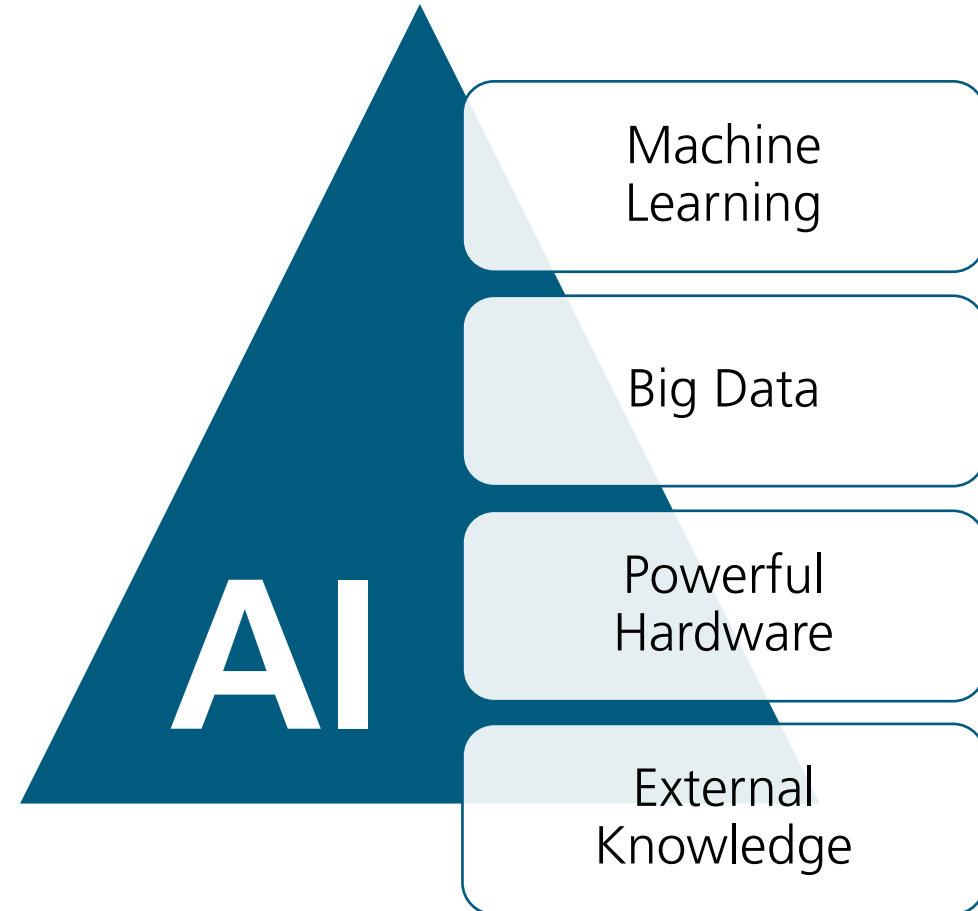
- Rising power and lowering cost of hardware such as GPUs/TPUs

## Big Data

- E.g. posts in social media, sensor data etc.

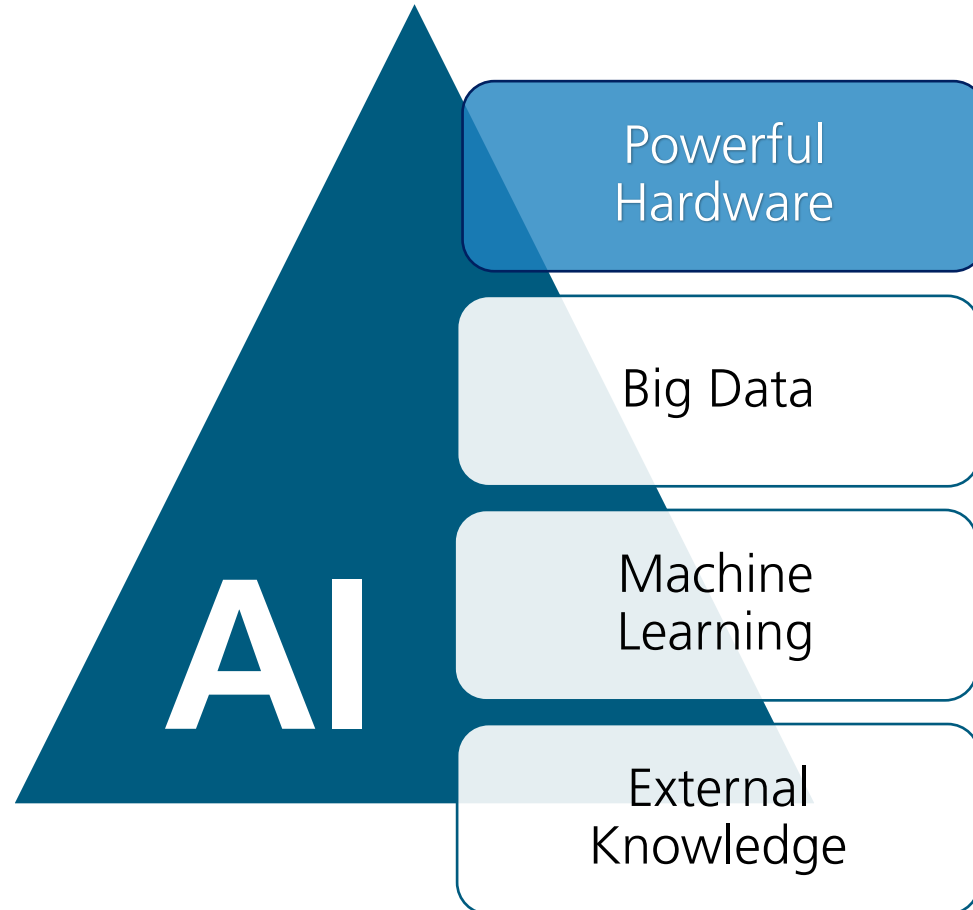
## Integration of external knowledge

- E.g. rules, ontologies, Probabilistic Soft Logic, etc.



# Components of Artificial Intelligence

Example: GPT-3 – Powerful Hardware



We are waiting for OpenAI to reveal more details about the training infrastructure and model implementation. But to put things into perspective, GPT-3 175B model **required 3.14E23 FLOPS of computing for training**. Even at theoretical 28 TFLOPS for V100 and lowest 3 year reserved cloud pricing we could find, this will take **355 GPU-years and cost \$4.6M for a single training run**. Similarly, a single RTX 8000, assuming 15 TFLOPS, would take 665 years to run.

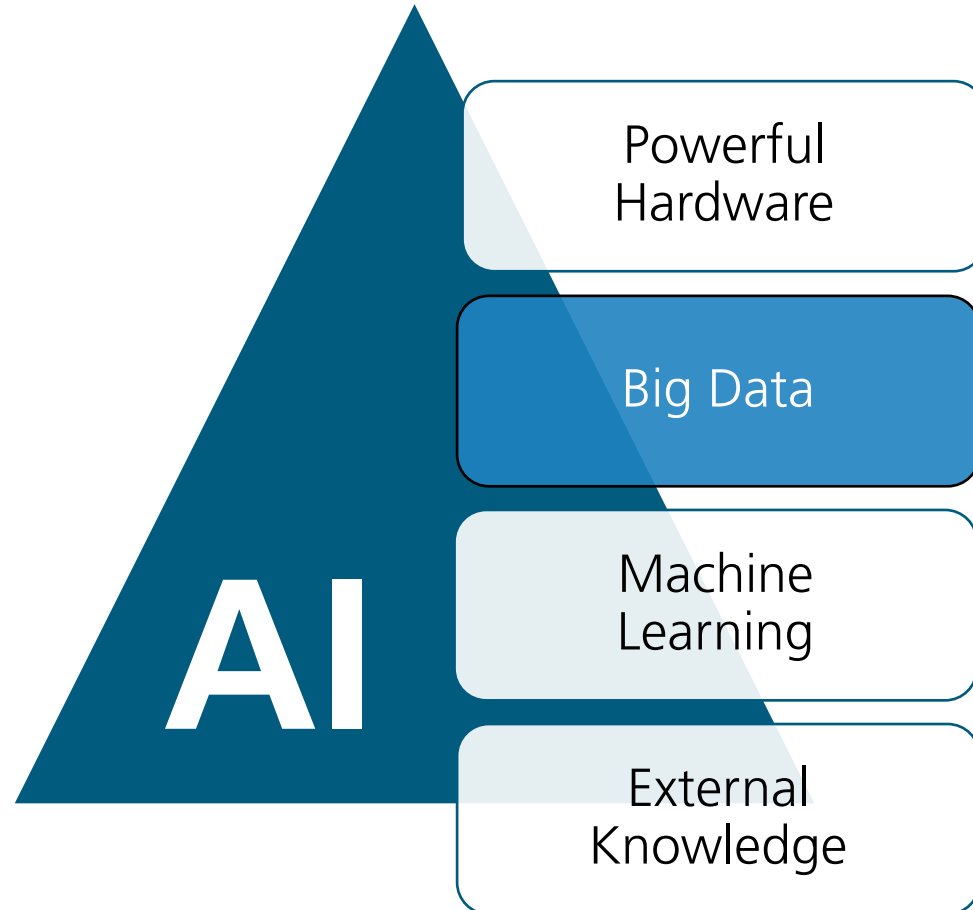
Source: <https://lambdalabs.com/blog/demystifying-gpt-3/>

“The supercomputer developed for OpenAI is a single system with more than **285,000 CPU cores, 10,000 GPUs and 400 gigabits per second** of network connectivity for each GPU server,” the companies stated in a [blog](#).

Source: <https://news.developer.nvidia.com/openai-presents-gpt-3-a-175-billion-parameters-language-model/>

# Components of Artificial Intelligence

Example: GPT-3 – Big Data



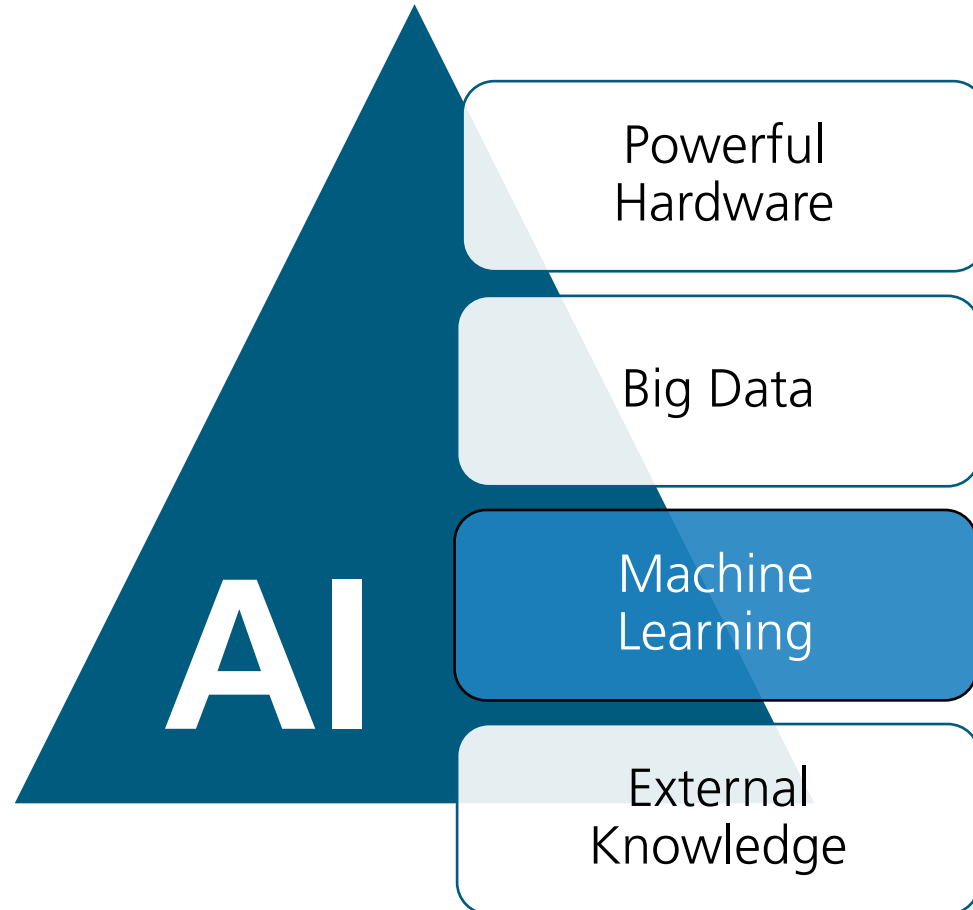
- GPT-3 was trained on **570 GB of texts**
  - Number of documents per language:

English	235.987.420
German	3.014.597
French	2.568.341
  - Paper shows that the quality of the model is correlated to the amount of training data and the amount of passes through the training data
  - The texts consist of approximately 500 billion tokens
    - (Caution: 1 Token != 1 Word)



# Components of Artificial Intelligence

Example: GPT-3– Machine Learning



- GPT-3 is based on the „**Decoder**“ of the „**Transformer**“ - neural network architecture
  - The biggest variant of this network contains 96 „Attention“-layers and about **175 billion parameters**
  - The Original Transformer-Paper „**Attention is all you need**“ (Vaswani et al., 2017) has around **60.502 citations**

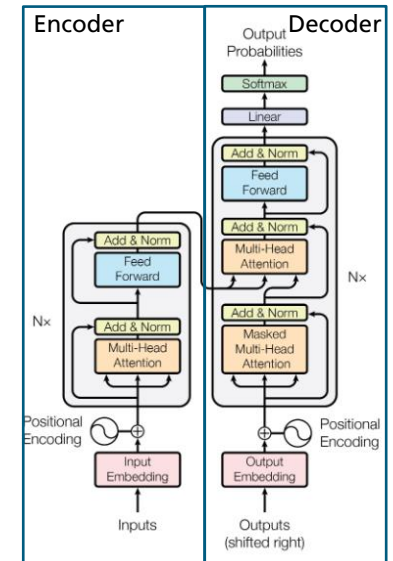
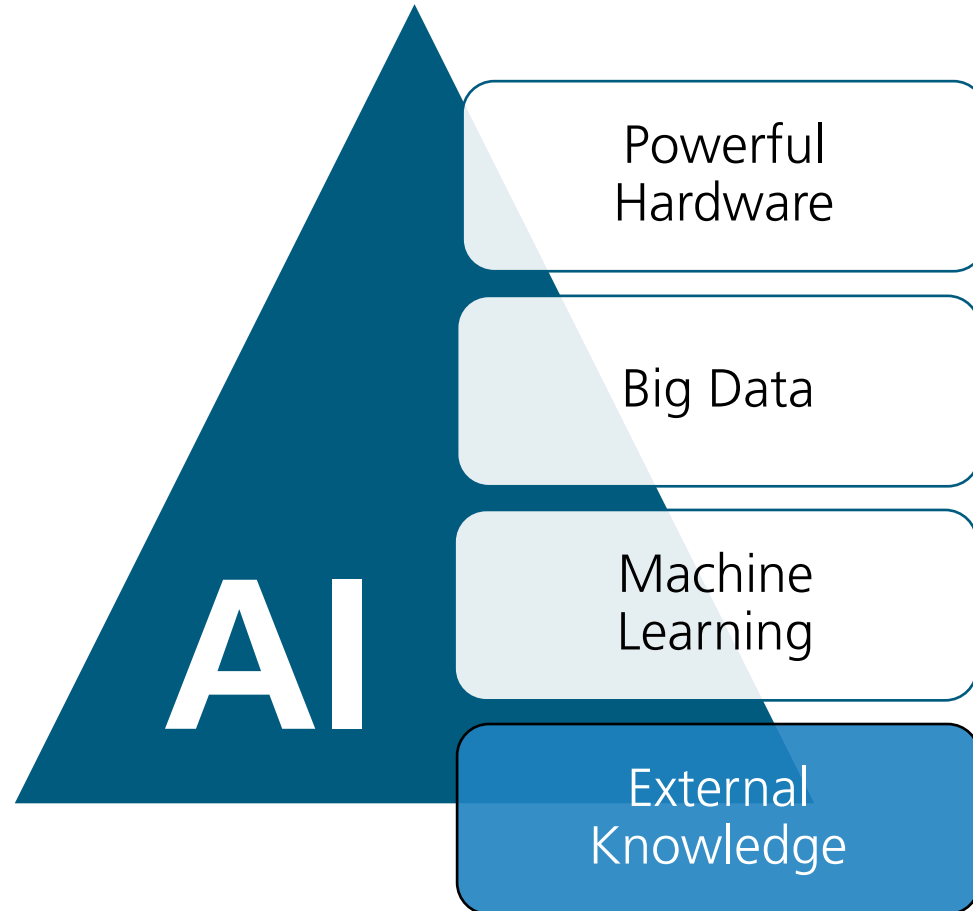


Figure 1: The Transformer - model architecture.

Source: Vaswani et al. – Attention is all you need (2017)

# Components of Artificial Intelligence

## Example: GPT-3– External Knowledge



- GPT-3 does not use any knowledge other than what is written in the texts that it saw for training, this leaves room for potential:
  - „[...] it still **sees** much **more text** [...] than a human sees in their lifetime [...]” – (Brown et al. 2020 – Language Models are Few Shot Learners)
  - “[...] apparently **simple** problems require humans to integrate knowledge across vastly disparate sources [...] entirely different sorts of tools are needed, along with deep learning, if we are to reach human-level **cognitive flexibility**.” – (Gary Marcus 2018 – Deep Learning – A Critical Appraisal”)

# Natural Language Understanding

# Natural Language Understanding

## What is it?

---

- Natural Language Understanding (NLU) is a part of Natural Language Processing (NLP) and deals with Understanding natural Language. NLU is a field at the intersection of
  - computer science,
  - artificial intelligence,
  - and linguistics
- **Goal:** for computers to process or “understand” natural language to perform tasks that are useful, e.g.,
  - making appointments, buying things
  - answering questions
- Fully understanding and represent the meaning of language
  - is a difficult goal
  - perfect language understanding is AI-complete

# Machine Learning: Key Ingredients

- training data instances  $(x_1, y_1), \dots, (x_n, y_n)$ 
  - $x_i$  is a numeric vector representing input data, e.g.
  - $y_i$  is a number / vector representing the desired output data e.g.
- A model function  $f: x \rightarrow y$  mapping input data to output data
  - a parameter vector  $w$  whose value has to be determined
  - for each  $x$  and  $w$  an output  $\hat{y} = f(x; w)$  is predicted
- A real-valued loss function  $L(x_i, y_i)(w)$  measuring the difference between the observed  $y_i$  and the predicted  $\hat{y}_i = f(x_i; w)$  outputs
  - For the training data we have  $L(x, y)(w) = L(x_1, y_1)(w) + \dots + L(x_n, y_n)(w)$

Two teen couples go to a church party, drink and then drive. They ...

$x_i = (0.1, -3.1, \dots, 1.2)$

bad movie

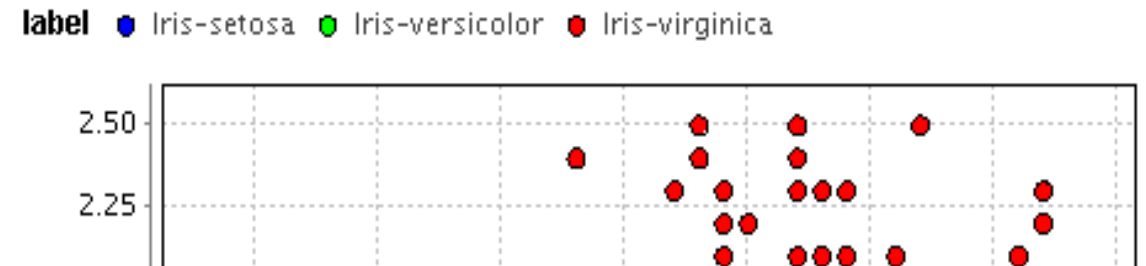
$y_i = 0$

modify  $w$  such that  $L_{\text{train}}(w)$  gets minimal

# Let this be the collected data

## The Iris Flower Data Set

Row	Sepal length	Petal width	Iris type
1	5.4	0.4	Setosa
2	5.5	0.2	Setosa
3	6.3	1.6	Versicolor
4	6.1	1.4	Versicolor
5	6.3	1.8	Virginica
6	7.7	2.2	Virginica
..	..	..	..



The **Iris flower data set** or **Fisher's Iris data** set is a multivariate data set introduced by the British statistician and biologist Ronald Fisher in his 1936 paper

„It is the **„Hello World!“** of Machine Learning“

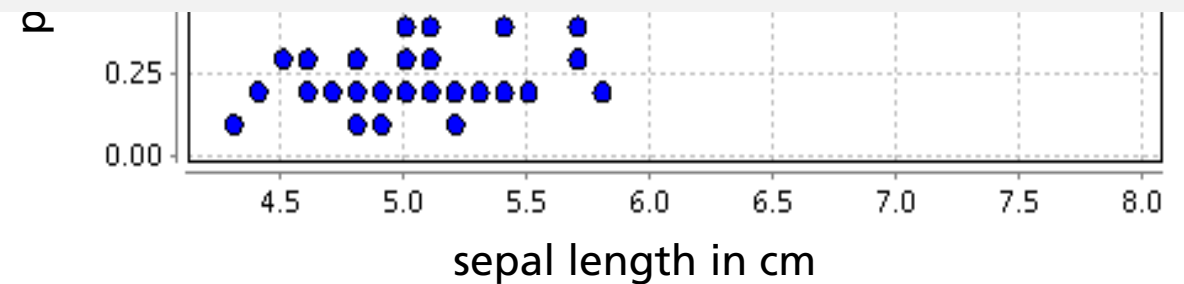


Image rights are held by: Sven Giesselbach, Fraunhofer IAIS

# Different analytical questions come to mind

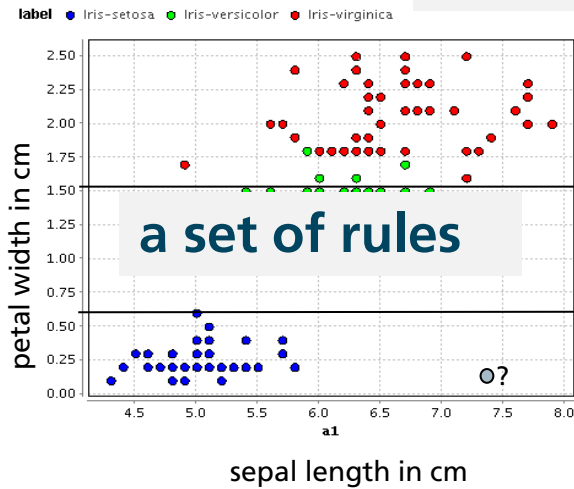
Such as..

Can we predict the iris type?

Can we predict the sepal length given the petal length?

Do the flowers naturally fall into groups?

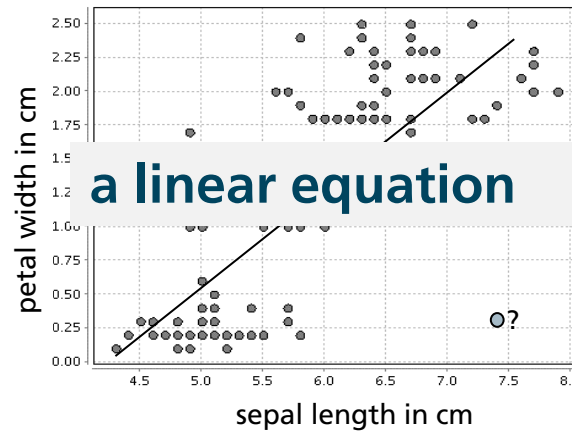
**We model a mathematical aspect of the data**



**a set of rules**

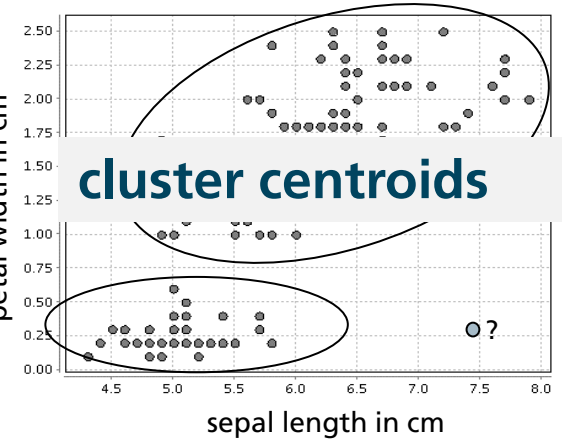
**Classification**

Supervised Learning



**a linear equation**

**Regression**



**cluster centroids**

**Clustering**

Unsupervised Learning

# What Is Unique in NLP/NLU?

- Differences to other fields
- Very large number of features (~ 300.000) that represent each of the documents
- Need to take into account long-range dependencies between words
- Patterns supported by small number of documents may be significant
  - Huge number of relevant patterns
  - Huge size of potential data:
    - text databases
    - the Web!

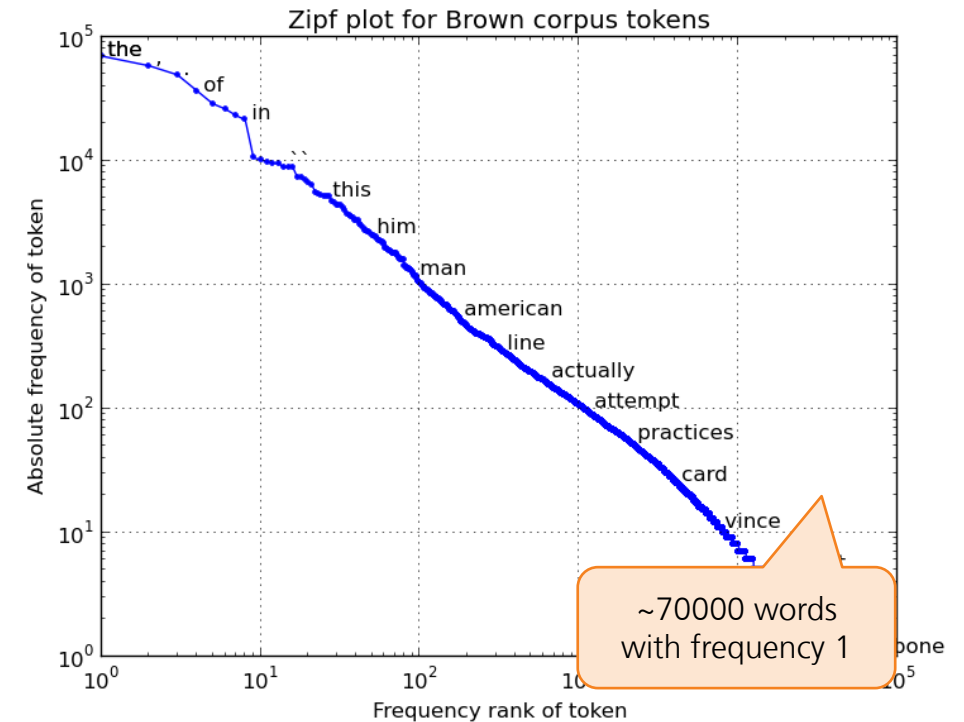
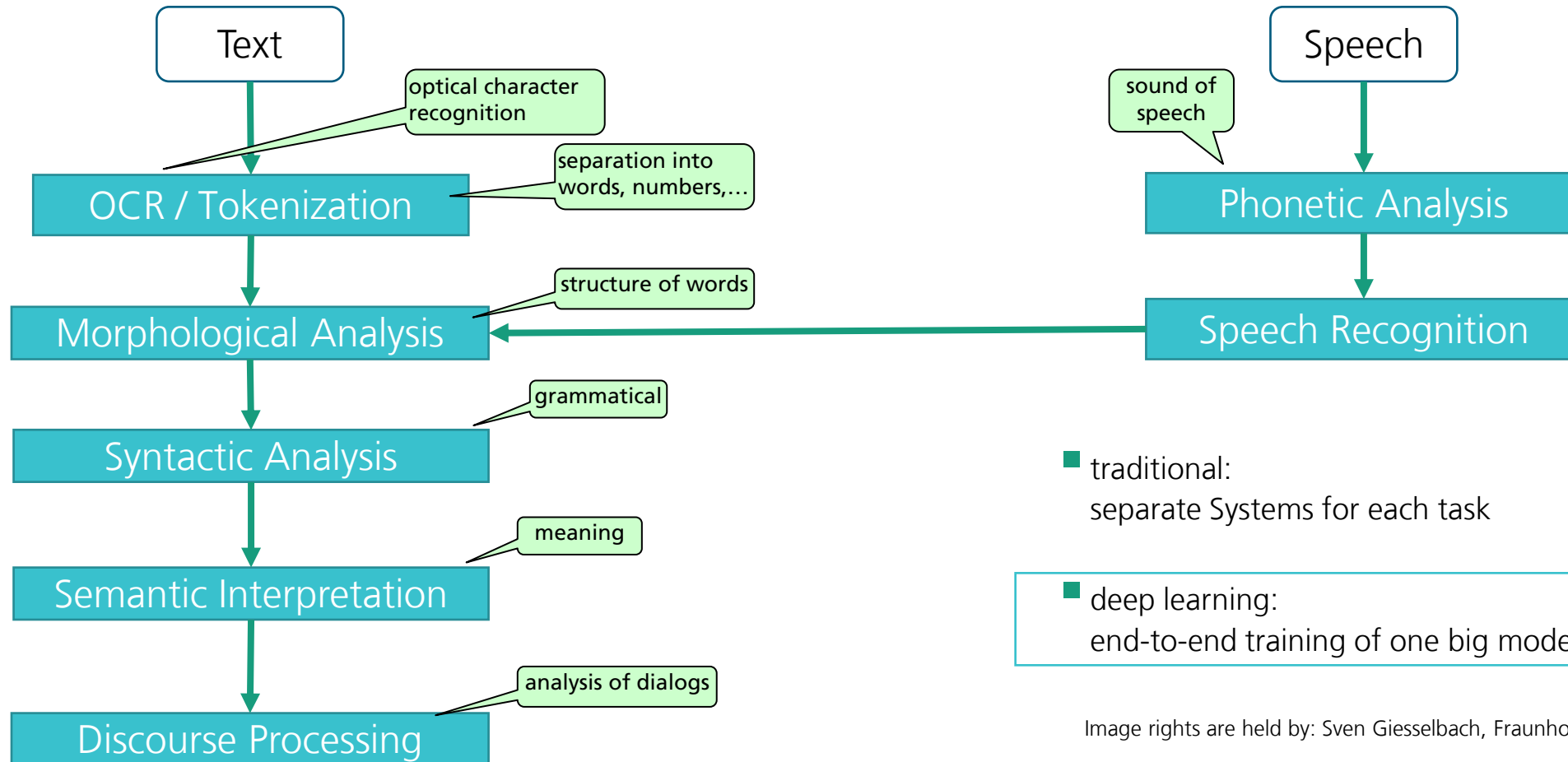


Image rights are held by: Sven Giesselbach, Fraunhofer IAIS



# NLP Subtasks: Traditional Pipeline



- traditional: separate Systems for each task

- deep learning: end-to-end training of one big model

Image rights are held by: Sven Giesselbach, Fraunhofer IAIS

# A sample of NLP Applications

Applications range from simple to complex:

---

- Spell checking, keyword search, finding synonyms, writing support
- **Classifying:**
  - overall content type of texts
  - positive/negative sentiment of longer documents
- **Extracting information** from websites such as
  - product price, dates, location, people or company names
  - relations between entities, e.g. person X lives in location Y
- Machine translation
- **Speech:** Recognition, Generation
- Complex question answering
- Spoken dialog systems

# Booming Industry Applications

---

- Search (written and spoken)
- Online advertisement matching
- Automated/assisted translation
  - product descriptions, hotel reviews
- Sentiment analysis for marketing or finance/trading
- Speech Recognition / Generation for customer service, chatbots
- Chatbots / Dialog agents
  - Automating customer support
  - Controlling devices
  - Ordering goods

# „A brief history of NLU representations and applications“

# Vector Space Representation

- Represent the document by a numeric vector of fixed length -> can be processed by statistical methods
- Example: Bag-of-Words (BOW) A vector with the count for each different word in the document.

Barack and Michelle Obama went to Paris and Bonn.

Sorted words of collection

Barack Bonn Michelle Obama Paris and to went ...

Document vector

1 1 1 1 1 2 1 1 ...

Image rights are held by: Sven Giesselbach, Fraunhofer IAIS

- Determine the **semantic similarity of documents**
- Approach: compare the vector representations of documents
  - Many identical words è high similarity

# Similarity Measure between Documents

- Document representation: counts  $c_{d,w}$  for every word  $w \in W$  of document  $d$

		length	cosine
the mouse eats the cheese	cheese:1 eats:1 he:0 mouse:1 on:0 puts:0 table:0 the:2	2.64	0,54
he puts the mouse on the table	cheese:0 eats:0 he:1 mouse:1 on:1 puts:1 table:2 the:2	3,46	

- high-dimensional space: (dim > 100.000)
- normalize length to 1

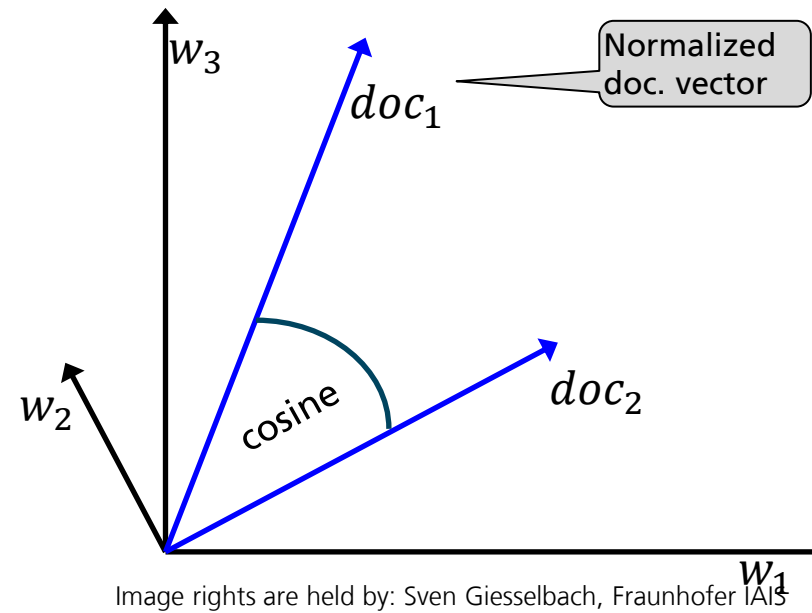
## Cosine Similarity

angle between vectors

$$u_{d,w} = c_{d,w} / \sqrt{\sum_{v \in W} c_{d,v}^2}$$

$$sim(d_i, d_j) = \sum_{w \in W} u_{d_i,w} * u_{d_j,w}$$

- value 1.0: very similar documents, 0.0: unrelated documents



# Problems with Vector Space Similarity

---

- **Synonyms:** there exist many ways to express the same object / fact
  - e.g.: car, automobile, convertible, limousine, motorcar, ...
  - -> cosine similarity too low: if different words with similar meanings
- **Homonyms:** the most words have several meanings
  - e.g.: mouse, bank, model, ...
  - -> cosine similarity too high: if same word has different meanings
- No grouping of words belonging to the same semantic field

Need a different representation for similarity computation

# Topic models: Clusters of Semantically Similar Words

- Each topic is a cluster of words, e.g., Neuroscience = brain, neuron, nerve, ...
- Words belong to a topic with a **weight** between 0 and 1 (=probability)
- Each document is a **mixture** of a few topics

Topics  
(word weights)

gene	0.04
dna	0.02
genetic	0.01
...	

life	0.02
evolve	0.01
organism	0.01
...	

brain	0.04
neuron	0.02
nerve	0.01
...	

data	0.02
number	0.02
computer	0.01
...	

Image rights are held by: Sven Giesselbach, Fraunhofer IAIS



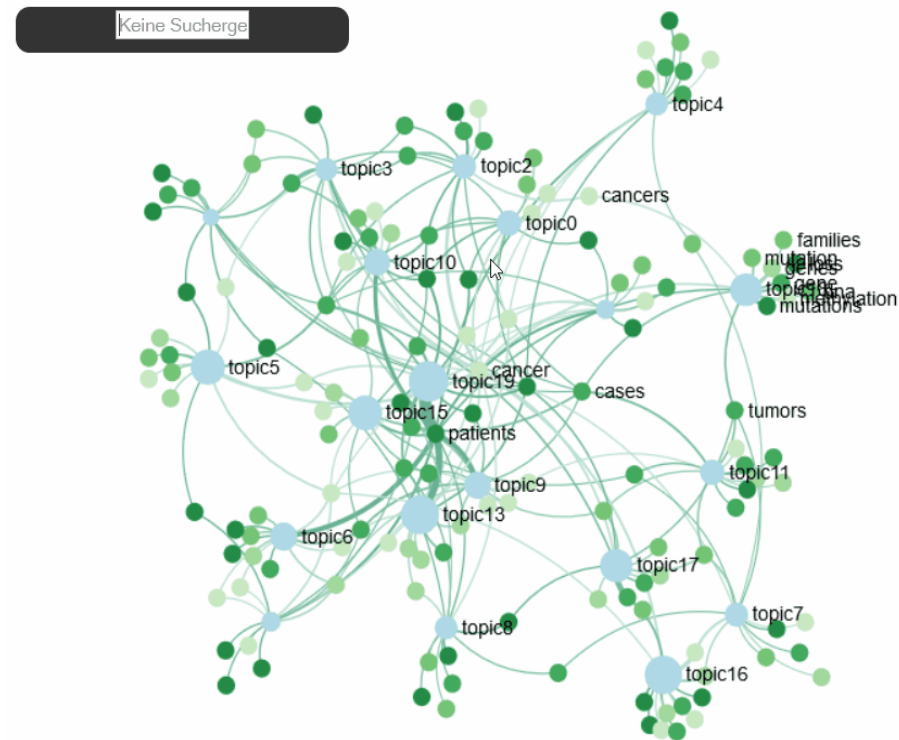
# Properties of the Solution

---

- Each topic  $z=i$  corresponds to a distribution  $p(w|z=i)$  over the words
- A words occurs usually only in a few topics with high probabilities. Probabilities for other topics are usually very low.
- For each word in a document the “corresponding” topic can be computed. May be different for the same word in other documents.
- Topic distribution of a document:  $p(z | d)$ ,  $z=1, \dots, n_z$  characterizes document
- **Synonyms:** Words with similar meaning (e.g. car – automobile) usually occur in the same topic
- **Homonyms:** The same word with different meanings (e.g. bank – river, bank – money) usually occur in different topics

# Application: Interactive Index of Document Collections

- Blue nodes: Topics
- Green nodes: Relevant Topic Words
- Topics trained on ~200.000 PubMed abstracts
- Other Applications include:
  - Social media analysis
  - Analysis of patents
  - Analysis of tech publications



**topic18**

digest

Lynch syndrome is the most common inherited colon cancer syndrome. Patients with Lynch syndrome develop a range of cancers including colorectal cancer (CRC) and carry a mutation on one of the mismatched repair (MMR) genes. Although CRC usually occurs after the fourth decade in patients with Lynch syndrome harboring a heterozygous MMR gene mutation it can occur in children with Lynch syndrome who have a compound heterozygous or homozygous MMR gene mutation. We report a case of CRC in a 13-year-old patient with Lynch syndrome and congenital heart disease. This patient had a heterozygous mutation in MLH1 (an MMR gene) but no compound MMR gene defects and a K-RAS somatic mutation in the cancer cells.

BACKGROUND: Changes in the number of alleles of a chromosome may have an impact upon gene expression. Loss of heterozygosity (LOH) indicates that one allele of a gene has been lost and knowing the exact copy number of the gene would indicate whether duplication of the remaining allele has occurred. We

Source: Fraunhofer IAIS

# Context and Meaning

---

- Similar words usually occur in similar contexts
- a word is characterized by the company it keeps [Firth, 1957]

The kid/child is playing in the sand.

The mother loves her kid/child

The kid/child is crying.

- Without problems we may exchange kid and child. -> very high similarity
- Embeddings: Only words close to target word are “neighborhood”

# Embeddings

- We represent each word by a small vector of numbers
- Words have similar meanings è their Vectors are close together [Wikipedia: Distributional semantics]
- Close: very similar words
- related words: a bit further
- Dissimilar words far apart
  - there are different directions of similarity = difference vector
- If we have vectors of length  $> 100$ 
  - -> very many possible directions of similarity

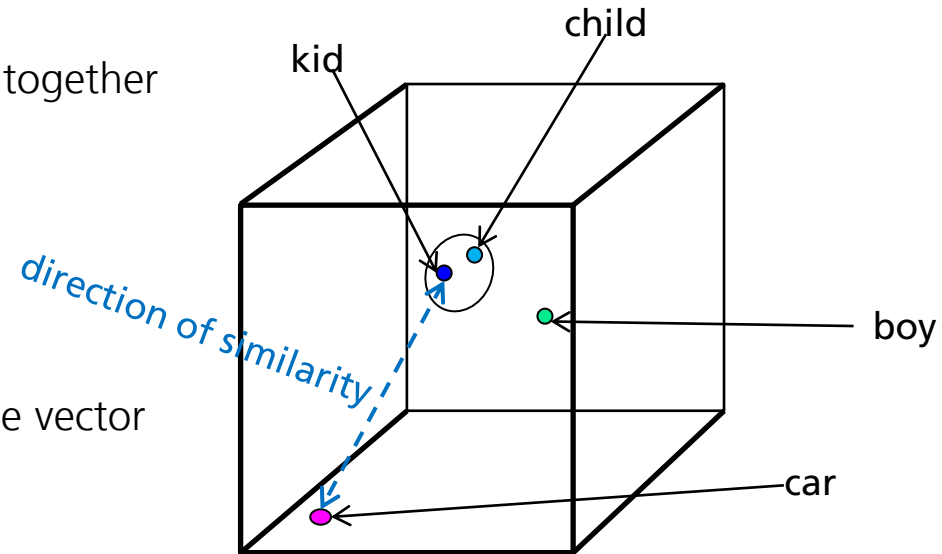


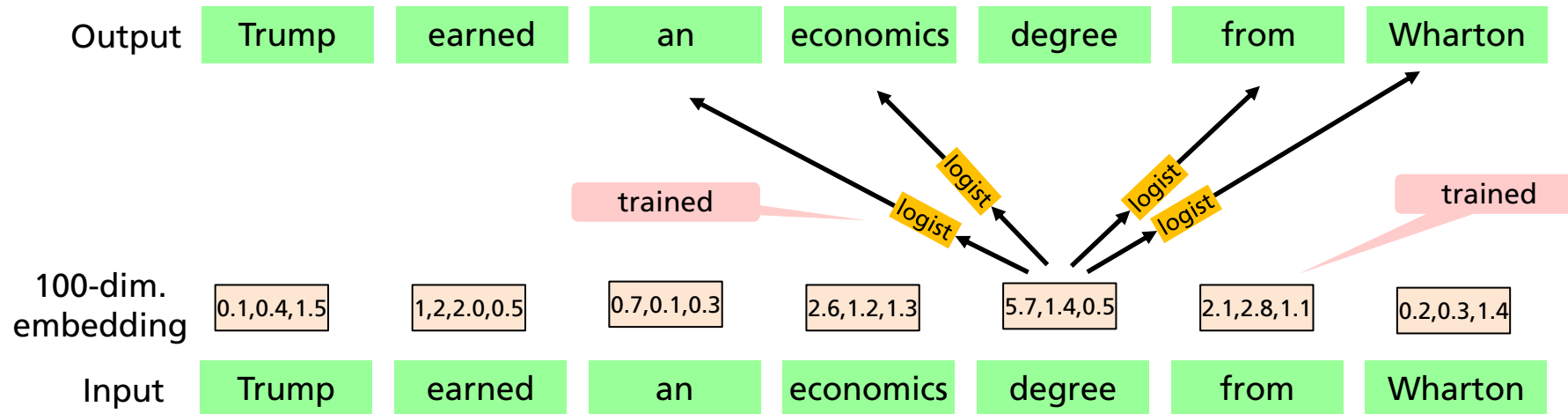
Image rights are held by: Sven Giesselbach, Fraunhofer IAIS

can cope with many dimensions of meaning

# Word2Vec

- To each word an embedding of size  $k$  is associated, e.g.  $k = 100$
- Predict neighbor words from the embedding of the central word
- Use a simple model: linear logistic model

no annotation required



Source: Mikolov et al. 2013

- Repeat this for all words of the collection

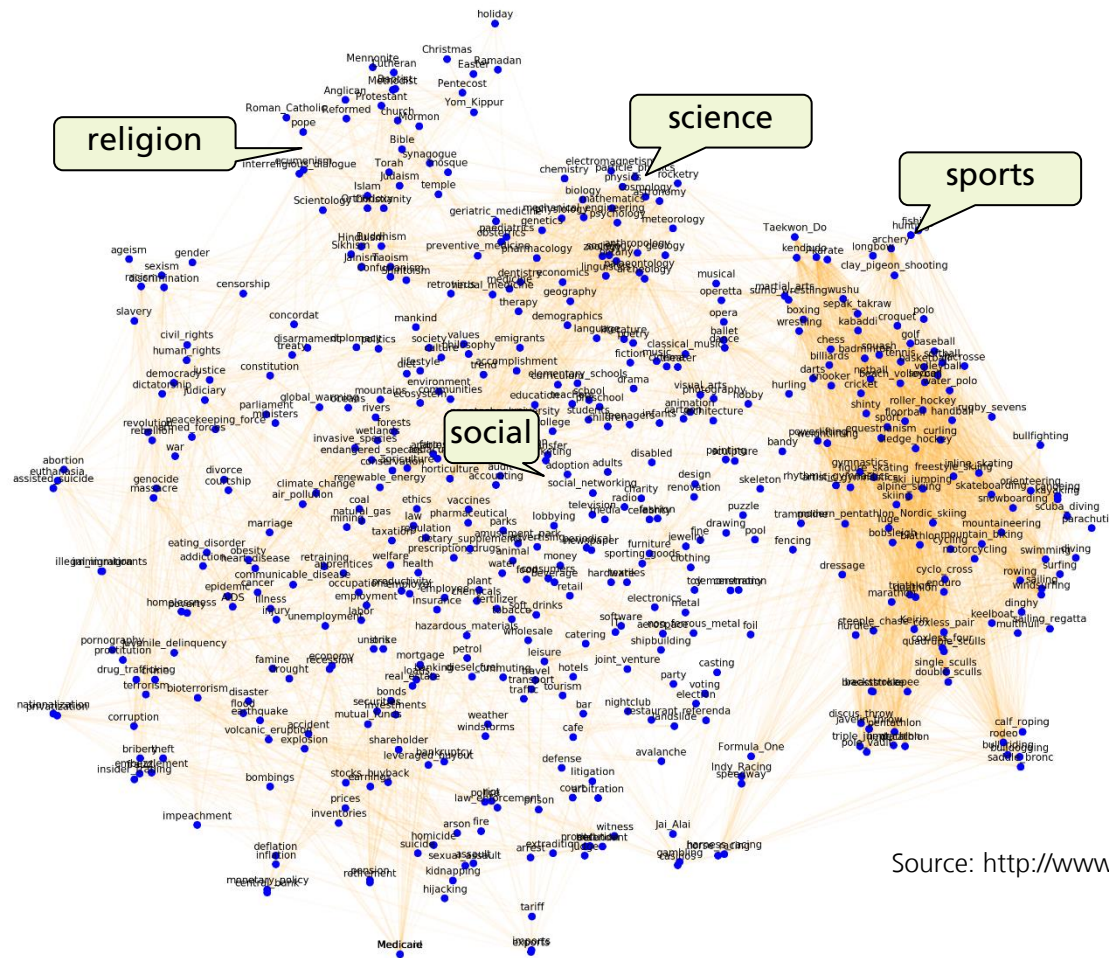
# A new type of machine learning

## Self-supervised learning

---

- A mixture of supervised and unsupervised learning
  - A supervised task is created by omitting parts of the training data and giving a model the task of reconstructing it
  - In the case of word2vec: omit neighboring words and let the model predict them
- Self-supervised learning requires no human-made annotations
- This form of learning is really important for almost all models which are currently developed

# Evaluation



2-d representation of word embeddings

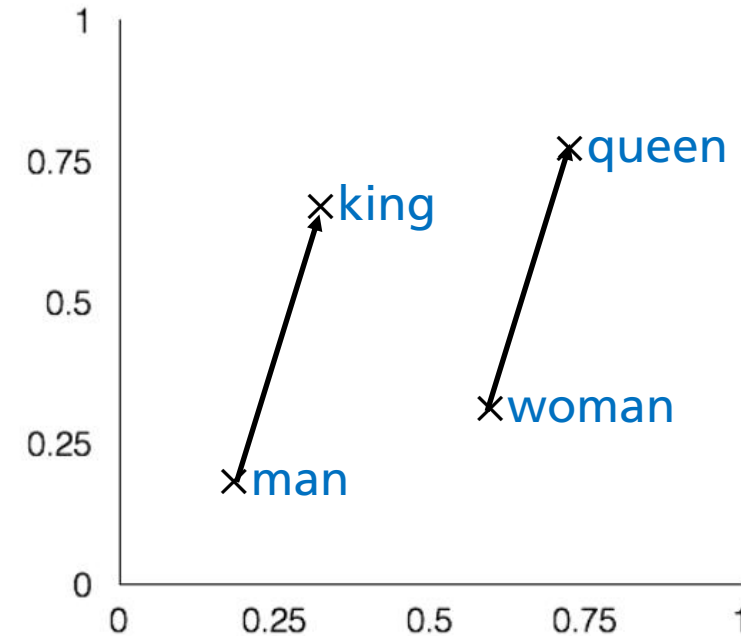
Source: <http://www.trivial.io/word2vec-on-databricks/>

Unsupervised Learning with Tensorflow

# Embeddings Capture some Relations

## Test for linear relationships [Mikolov, Yih & Zweig 2013]

	Word	Embedding
+	king	[0.30, 0.70]
-	man	[0.20, 0.20]
+	woman	[0.60, 0.30]
=	queen	[0.70, 0.30]



- Interactive analogy evaluation

Enter option:

usa dollar uk

gbp

34

Image rights are held by: Sven Giesselbach, Fraunhofer IAIS



# Application I: Building Terminologies

- Given a query word, yield related words
- Suggested words can be added to a terminology

Wort

Tumor

Ausführen

w2v	glove	fasttext
Muskel 0.76	Neuroendokriner 0.61	Tumors 0.9
Lymphknoten 0.76	Tumoren 0.6	Tumore 0.89
Abszess 0.74	Tumors 0.54	Lymphom 0.89
Gendefekt 0.73	Metastasen 0.54	Karzinom 0.88
Schilddrüse 0.73	neuroendokriner 0.53	maligner_Tumor 0.88

Image rights are held by: Sven Giesselbach, Fraunhofer IAIS

# Application II: Document Retrieval for News Evidence

- Given an article, rank articles from „reliable“ sources

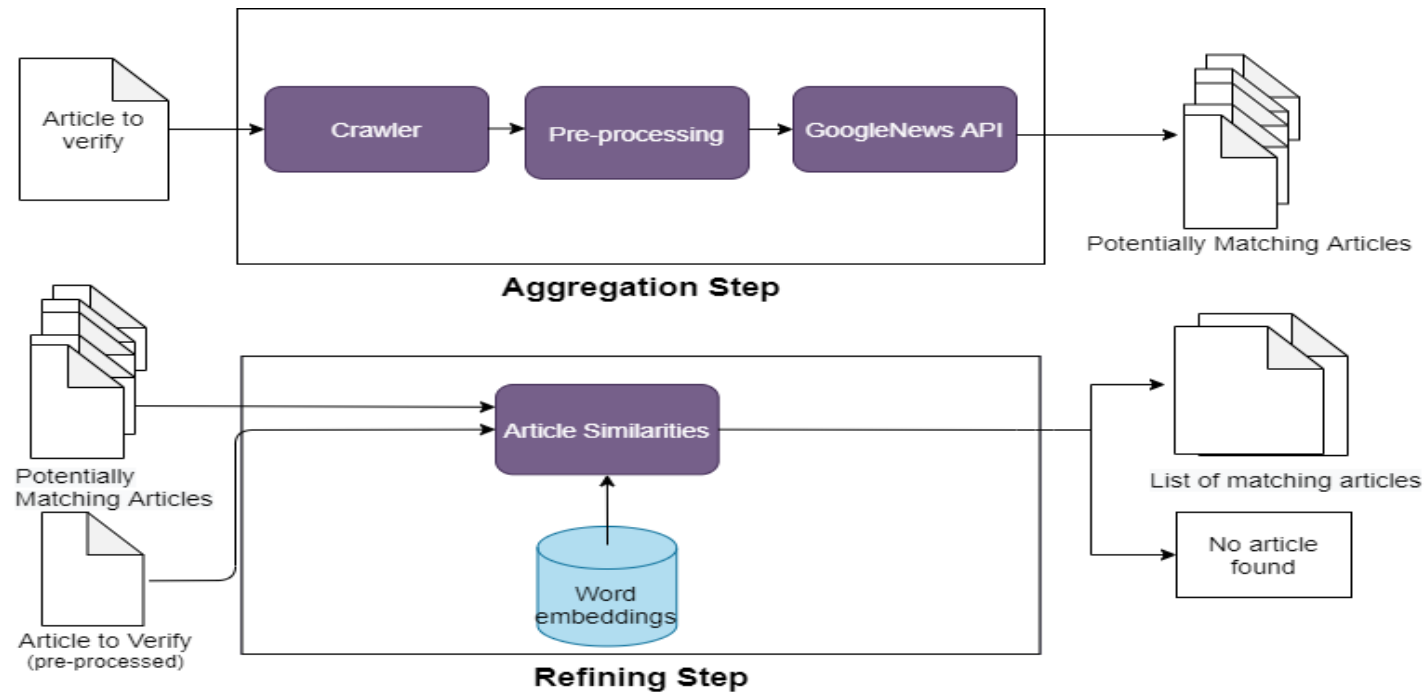


Image rights are held by: Sven Giesselbach, Fraunhofer IAIS

- Supporting verification of news articles with automated search for semantically similar articles – Gupta et al. - ROMCIR 2021

# Distributed Embeddings vs. Contextualized Embeddings

Embedding methods like **Word2Vec**, **FastText** and **BPE** are referred to as „**distributed embeddings**“

---

- A single embedding per word is learned over all contexts
- Good and efficient in terms of nearest neighbor retrieval and relations
- Not ideal for downstream tasks
  
- A „newer“ type of embeddings are „contextualized“ embeddings
- They learn embeddings per context in which words appear
- Tailor the embeddings towards the context (good for example in case of homonyms)

# Transformer

- Encoder self-attention layer:
  - Each position encoder can attend to all positions in the previous layer of the encoder.
- Decoder self-attention layer:
  - masking: each position in the decoder can attend to current and past positions in the previous layer of the decoder.

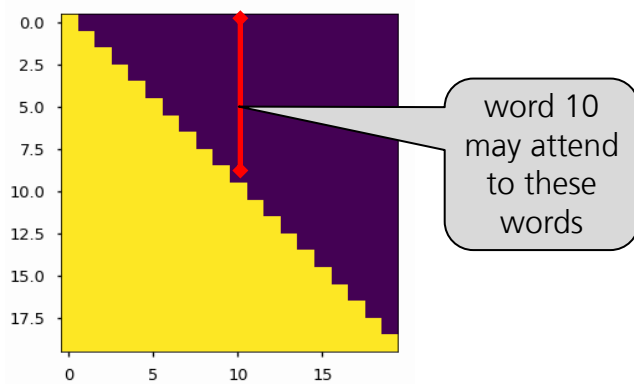
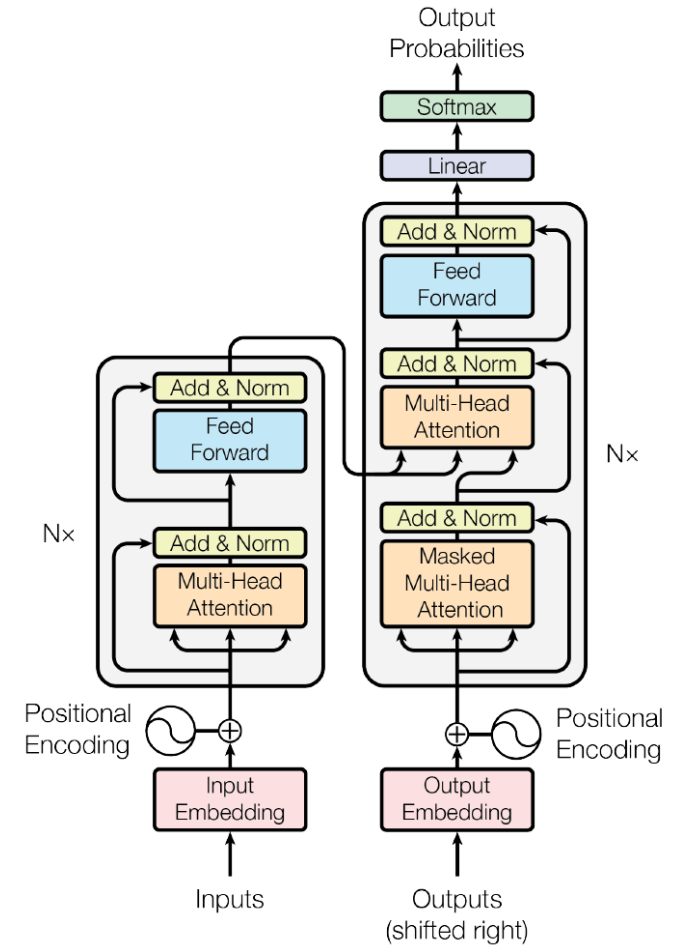


Image rights are held by: Sven Giesselbach, Fraunhofer IAIS



Source: Vaswani et al. 2017

# Self-Attention

$$z_r = \sum_j \alpha_j * v_j$$

$$\alpha = \text{softmax}(s_1, \dots)$$

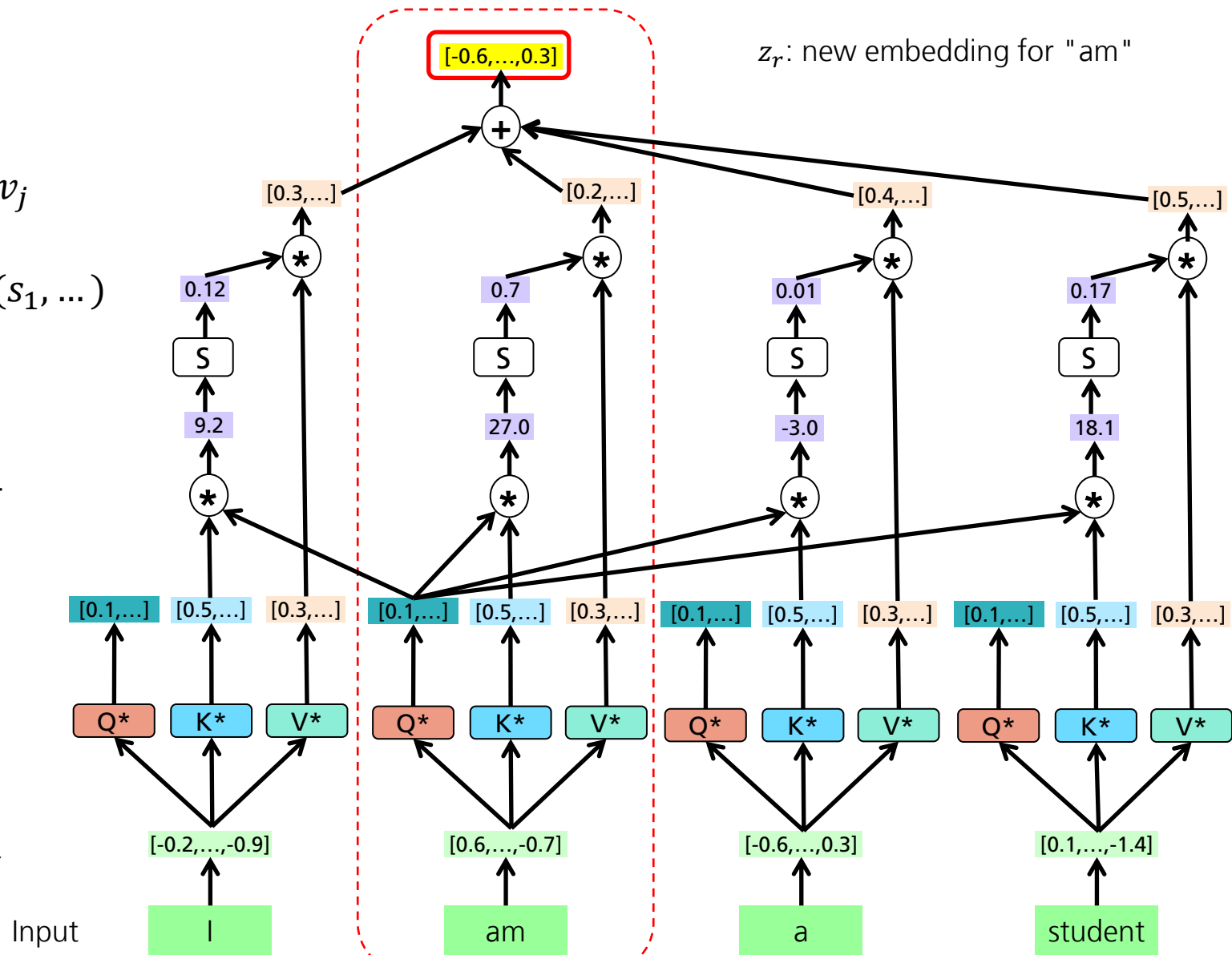
$$s_t = \frac{k_r' * q_t}{\sqrt{d_k}}$$

$$v_t = V * u_t$$

$$k_t = K * u_t$$

$$q_t = Q * u_t$$

Embeddings  $u$

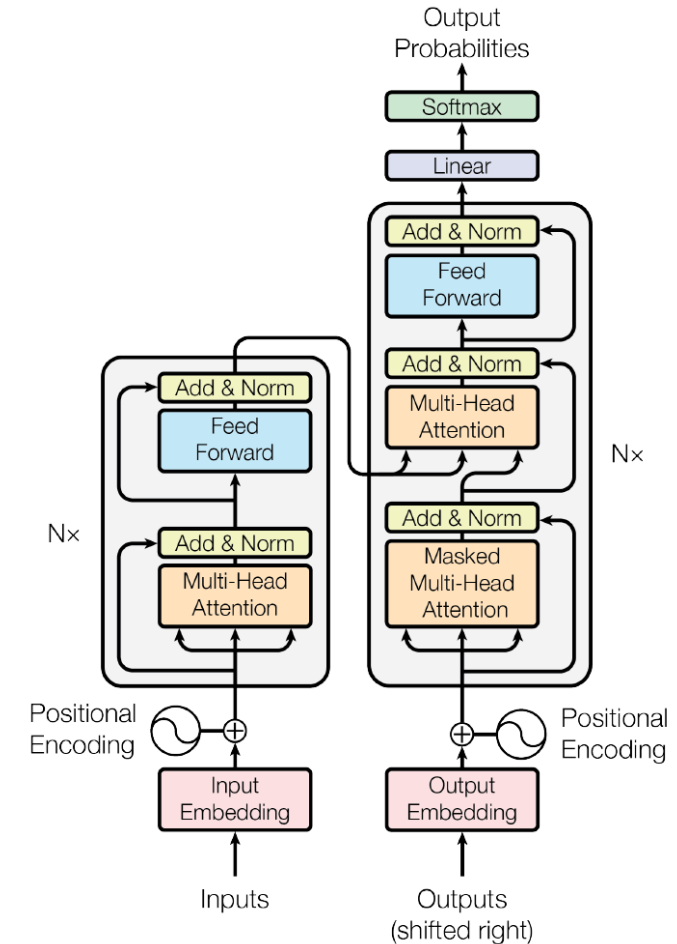


$z_r$ : new embedding for "am"

Image rights are held by: Sven Giesselbach, Fraunhofer IAIS

# Transformer

- Encoder self-attention layer:
  - Each position encoder can attend to all positions in the previous layer of the encoder.
- Decoder self-attention layer:
  - masking: each position in the decoder can attend to current and past positions in the previous layer of the decoder.
- Encoder-decoder attention
  - every position in the decoder attends all positions in the highest layer of the encoder.

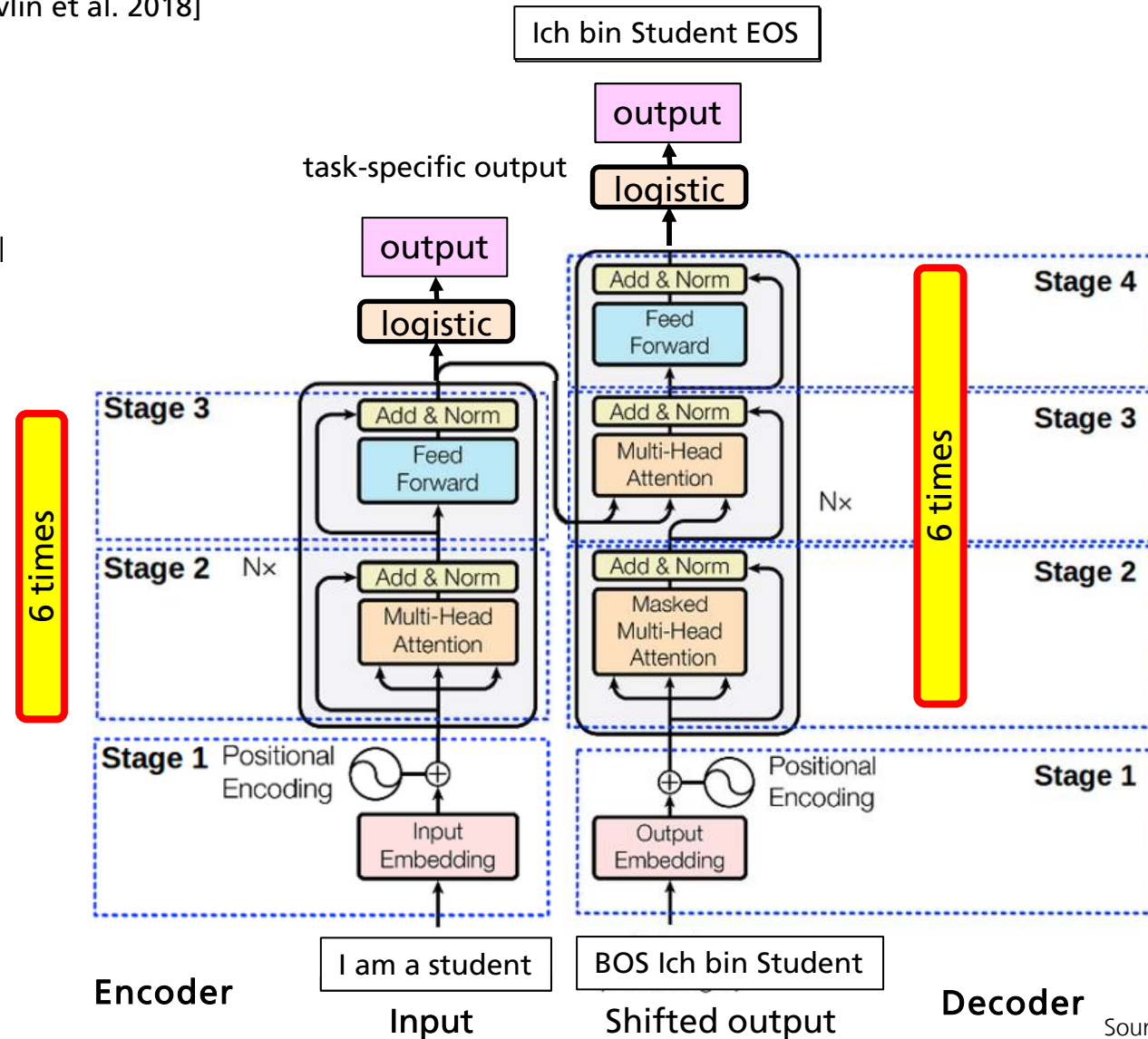


Source: Vaswani et al. 2017

# BERT: Transformer Encoder [Devlin et al. 2018]

- use encoder-layers of transformer as language model
- for some tokens: add a specific output prediction tasks
- train on large dataset

BERT  
Bidirectional Representations  
from Transformers



Source: Vaswani et al. 2017

# BERT: Pretraining Details

- Tokenization: Subwords
  - Start with letters -> subwords. Merge frequent subwords:
  - Representation: **play**, **##ing**, and **play**, **##ed** → Byte-Pair-Encoding, SentencePart
- Input embeddings: Sum of 3 embeddings

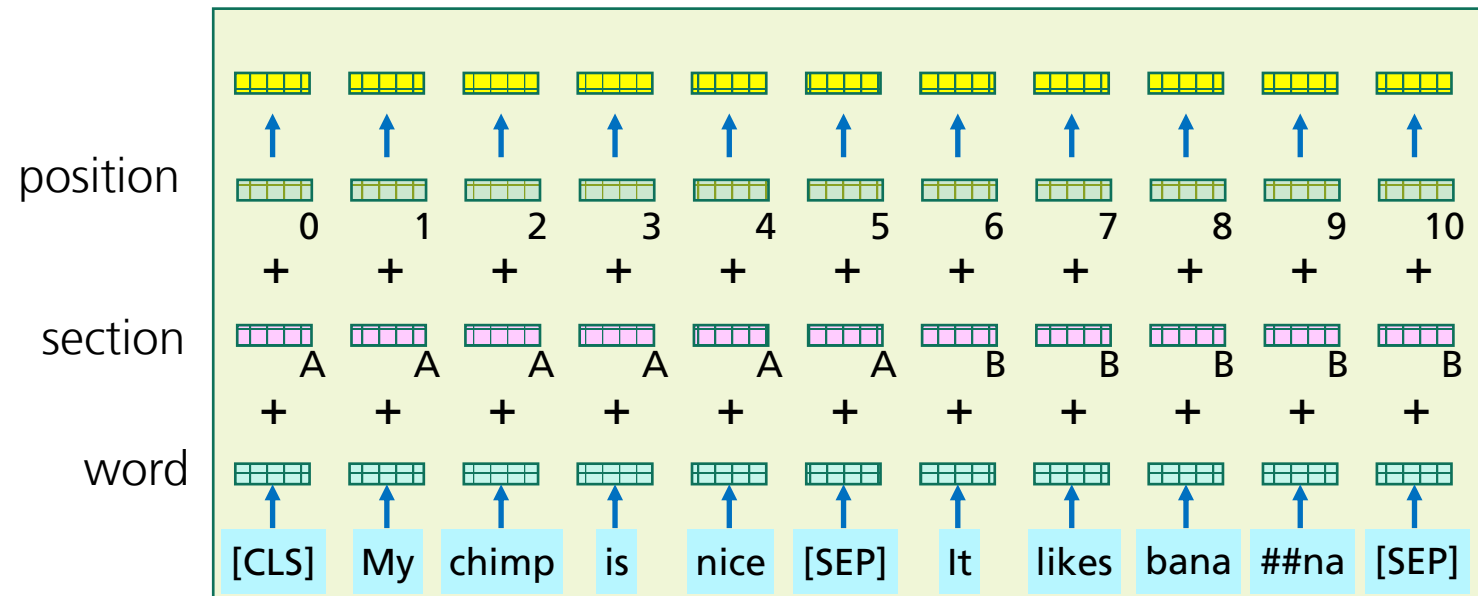


Image rights are held by: Sven Giesselbach, Fraunhofer IAIS



# Pretraining: Predict Masked Words

- Start token [CLS]. finish first and second text by [SEP]
- Randomly replace 15% of words by [MASK]
- predict the masked tokens by logistic regression use last layer embeddings
- 80% of the time tokens are actually replaced with the token [MASK].
- 10% of the time tokens are replaced with a random token.
- 10% of the time tokens are left unchanged.

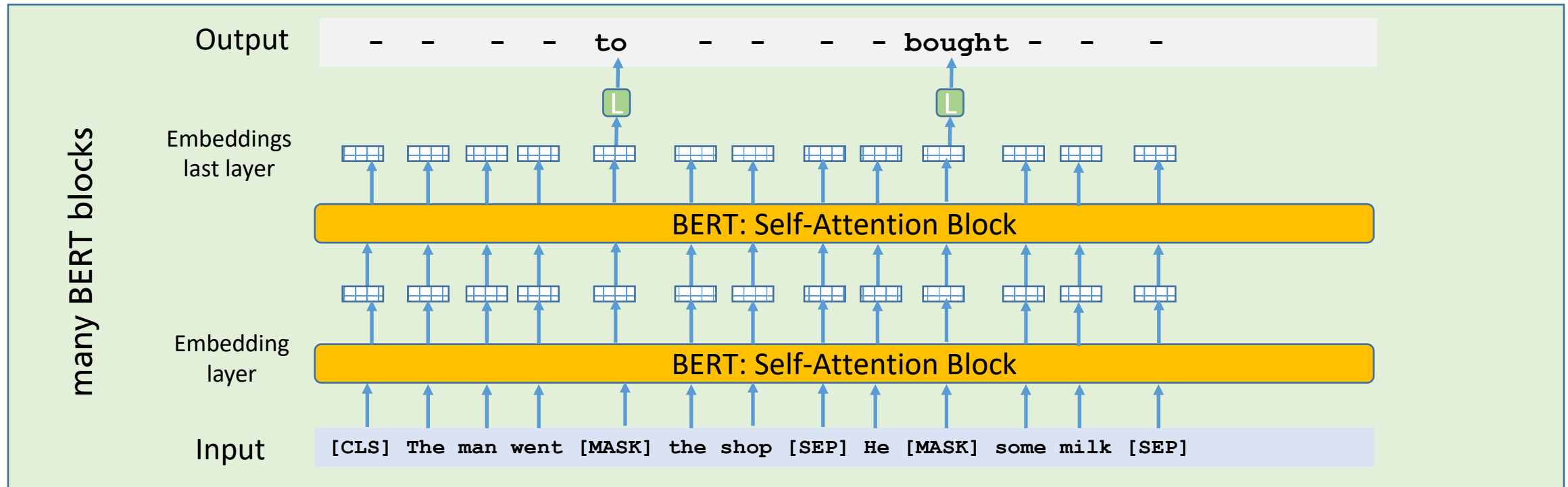


Image rights are held by: Sven Giesselbach, Fraunhofer IAIS

# Pretraining: Next Sentence Detection

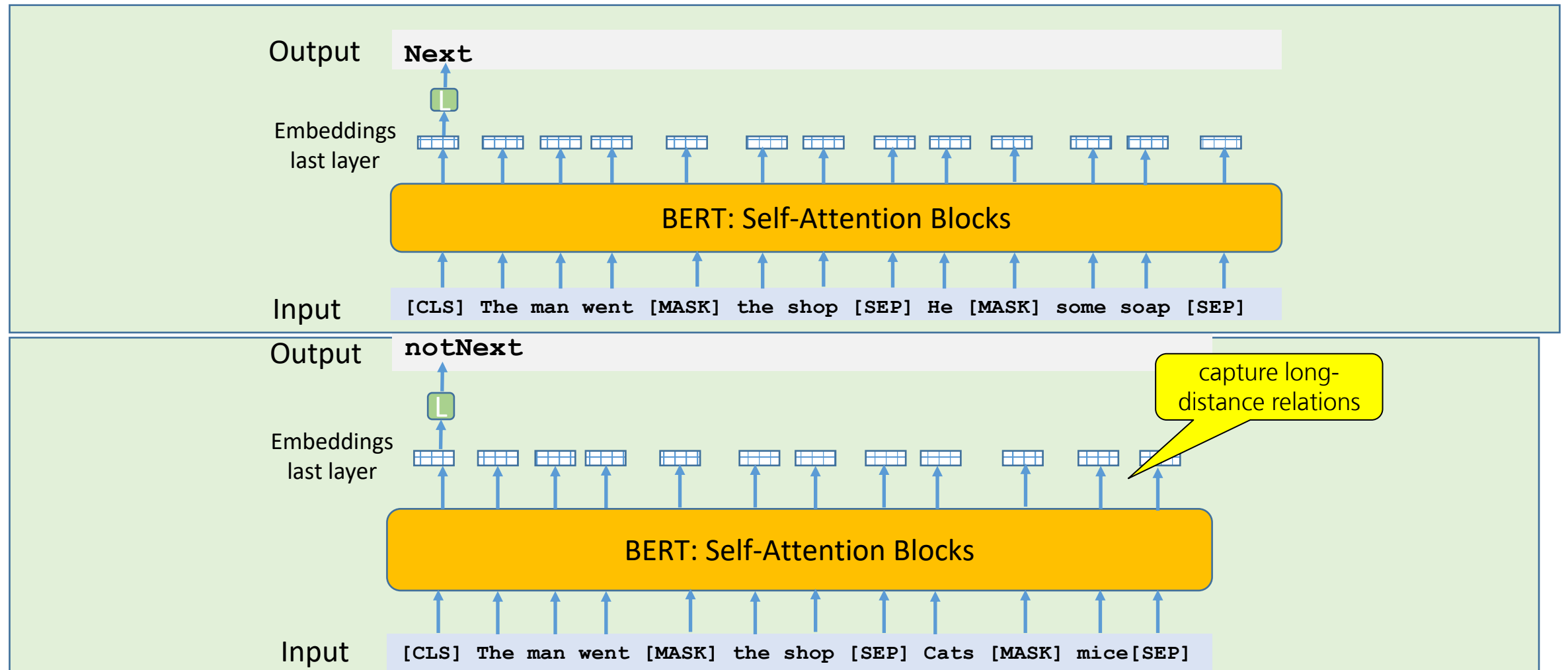


Image rights are held by: Sven Giesselbach, Fraunhofer IAIS

# BERT: Pretrained models

- need a large corpus for pretraining
  - BooksCorpus 800M words
  - English Wikipedia 2500M words

	# layers	hidden size	# param	training time
BERT base	12	768	110M	4 days on 4 TPUs
BERT large	24	1024	340M	4 days on 16 TPUs

- many pretrained models available
  - tensorflow <https://github.com/google-research/bert>
  - pytorch: by huggingface <https://github.com/huggingface/transformers>
  - keras: <https://github.com/Separius/BERT-keras>
  - multilingual BERT for 100 languages: <https://github.com/google-research/bert/blob/master/multilingual.md>

# BERT - Characteristics

---

- Consider Polysemy: Different embeddings of the same word
- Contextual: The representation for each word depends on the entire context in which it is used.
- Deep: combine all layers of a deep pre-trained neural network.
- Character based submodel of words: form robust representations for out-of-vocabulary tokens unseen in training.
- Available from huggingface: <https://huggingface.co/>

# BERT: Fine tuning

- Pretrained models knows the structure of language
  - syntax: how words may form a sentence
  - semantics: how to express a fact or relation
- Adapt this model to a new task
  - Example: classify opinion expressed in a sentence:  
positive, negative, neutral

Yet the act is still charming here.

**positive**

This isn't a new idea.

**negative**

Image rights are held by: Sven Giesselbach, Fraunhofer IAIS

# Application I: Relation Extraction

- Detect Relations between entities
  - Link them if they belong together
  - Detect the class of the relationship between them

Anmerkungen auswählen

Steve Jobs , damals CEO von Apple , trug wie immer einen Rollkragenpullover.

ENTITY1 ENTITY2

Löschen

Ausführen

entity1 -.-OrganizationLeadership- -> entity2

Steve Jobs, damals CEO von Apple, trug wie immer einen Rollkragenpullover.

Source: Fraunhofer IAIS – NLU Showroom

# Application II: Aspect-Based Opinion Mining

- Detect Aspects and Opinions
  - Link them if they belong together
  - Detect the sentiment of an opinion
  - Uses KnowBERT and the SenticNet ontology

Der Kaffee war schön warm, der Kuchen lecker.

ASPECT #0 OPINION #0 ASPECT #1 OPINION #1

Das Essen war gut, der Service war aber unfreundlich.

ASPECT #0 OPINION #0 ASPECT #1 OPINION #1

Source: Fraunhofer IAIS – NLU Showroom

# Healthcare Analytics: Evaluation of patient records

## Intelligent support in the evaluation of patient files

### Customer challenge

- MDK Reform Act 2020:  
Support of hospitals in the billing of patients by semi-automatic evaluation of their records

### Our solution

- Automated weighted assignment of accounting codes to the respective files
- Self-learning through user feedback
- Highlights of relevant text passages that led to a decision by the AI

### Customer benefits

- Work of the coders is facilitated, improved and reduced
- Obligation of proof is complied with
- Customers benefit from additional users (swarm intelligence)



# SmartSentencing

## Intelligent Mining of Legal Sentences

### Customer challenge

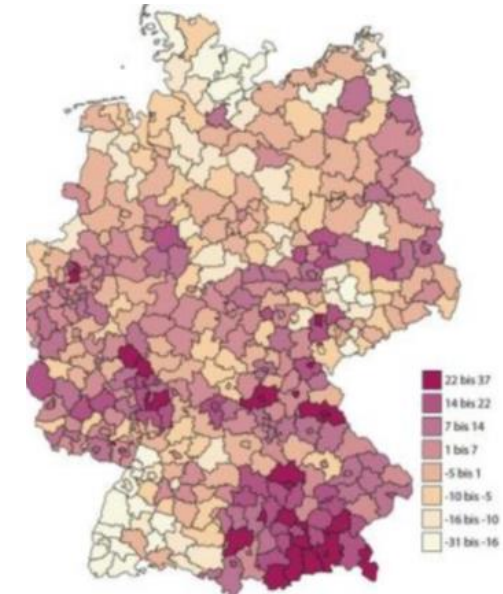
- There is no central database in Germany containing all legal sentences which make them easily comparable
- Scientists assume that there is a gap between legal sentences issued in south and north of Germany, how can we assess this?

### Our solution

- Automated extraction of structured information from legal documents based on German BERT models and probabilistic logic
- Dashboard for visualisation and analysis

### Customer benefits

- Reduce manual effort to a minimum
- Make sentencing more transparent and comparable



Source: <https://rotary.de/gesellschaft/was-ist-eine-gerechte-strafe-a-13312.html>

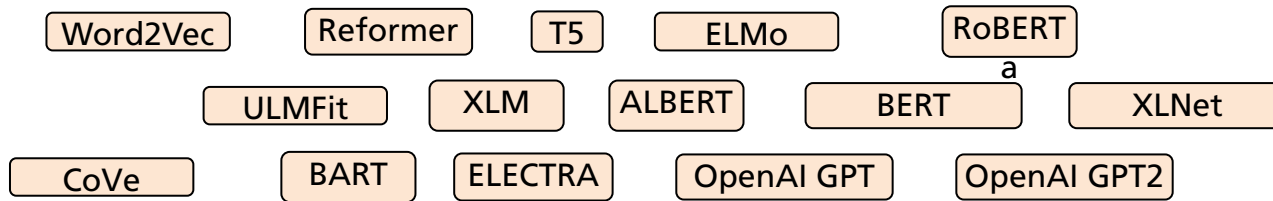
© Martin Küsting

# New Paradigm for Natural Language Processing

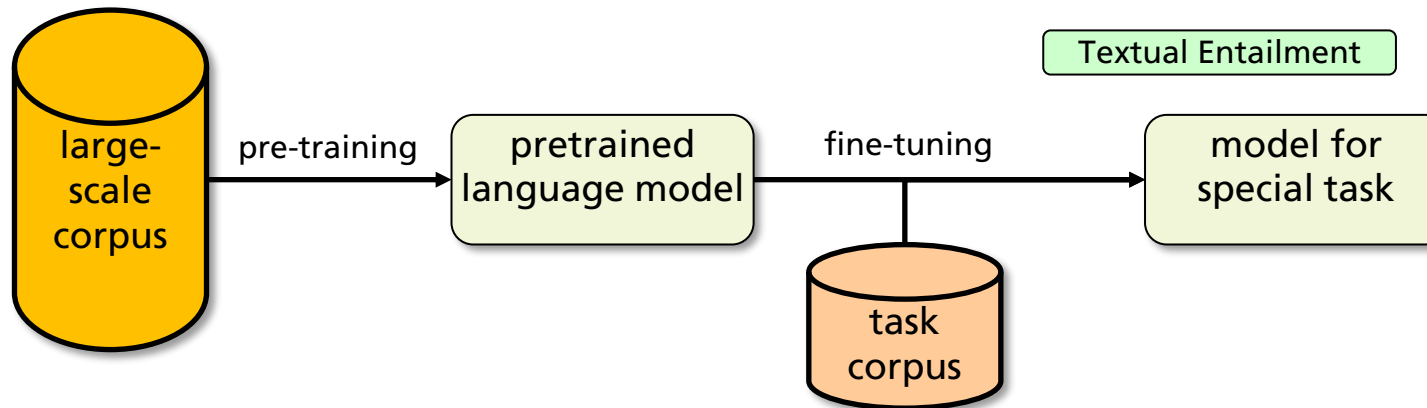
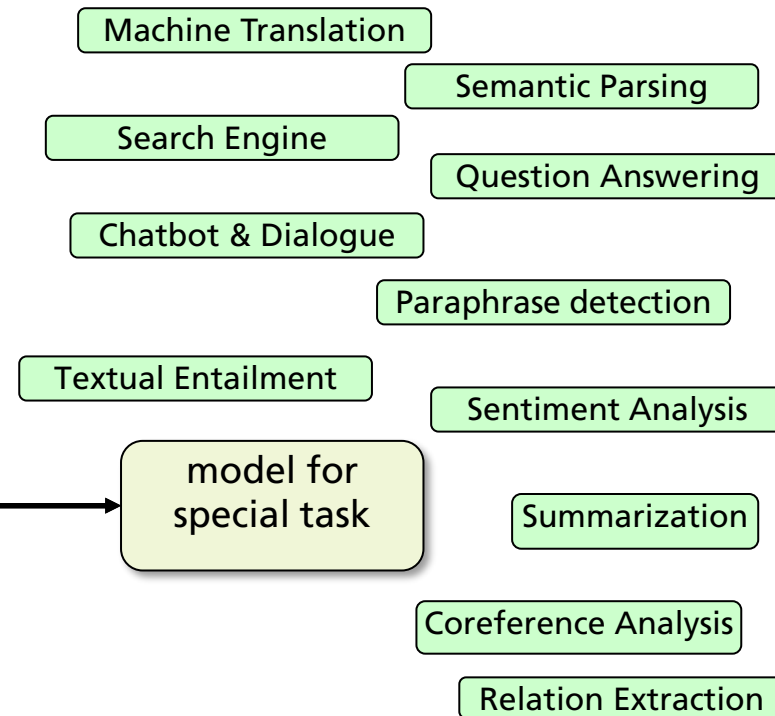
- Train a language model on a large corpus without annotations
- Get context sensitive embeddings: know the structure of language
- Add a small model on top: train for some special task  
→ Transfer learning

Source: <https://github.com/huggingface/transformers>

## Embedding & Language Models

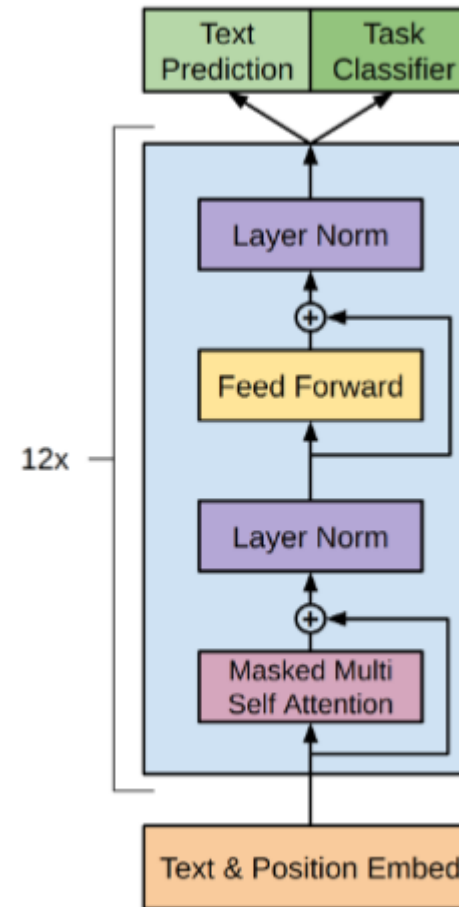


## NLP Tasks



# A new Language model: GPT2

- OpenAI released a new language model <https://blog.openai.com/better-language-models/>
- Based on decoder architecture of the Transformer [Vaswani et al. 2017][Radford et al. 2018]
- GPT2: 1.5B parameters  
BERT: 340M parameters
- Trained on 40 GB of text  
8M docs, linked from Reddit with at least 3karma.
- Byte-Pair Encoding: char. sequences for rare words



Zero-shot learning

Image rights are held by: Sven Giesselbach, Fraunhofer IAIS

# Language model GPT2

- state-of-the-art scores on 7 of 8 specific language modeling tasks.
- not trained on any of the data specific to any of these tasks

Zero-shot learning

Data		SOTA	GPT2
LAMBADA	acc	59.2	63.2
ChildBT-CN	acc	85.7	93.3
ChildBT-NE	acc	82.3	89.1
WikiText2	perplex	39.1	18.3
PTB	perplex	45.5	38.8
enwik8	bpc	0.99	0.93
text8	bpc	1.08	0.98
WikiText103	perplex	18.3	17.5
1BWords	perplex	21.8	42.2

shuffled  
sents

# SmartHospital.NRW

## Generation of discharge summaries

### Customer request

- Automate the process of discharge summary generation, for which one needs to take into account the complete history of a patient (structured and unstructured data)

### Our approach

- Automated creation of text snippets
- Integration of structured information about the patient in the text
- Intuitive editor for final adjustments

### Technologies

- Generative language models (GPT-like)
- Pre-Created text snippets



KiNRW

♀ **Petra Meier (56)**

Diagnosen

Therapien

Medikationen

Source: SmartHospital.NRW

# The size of Pretrained Language Models

## GPT-2

Training data: 40 GB of text

Parameters	Layers	$d_{model}$
117M	12	768
345M	24	1024
762M	36	1280
1542M	48	1600

## GPT-3

Training data: 570 GB of text

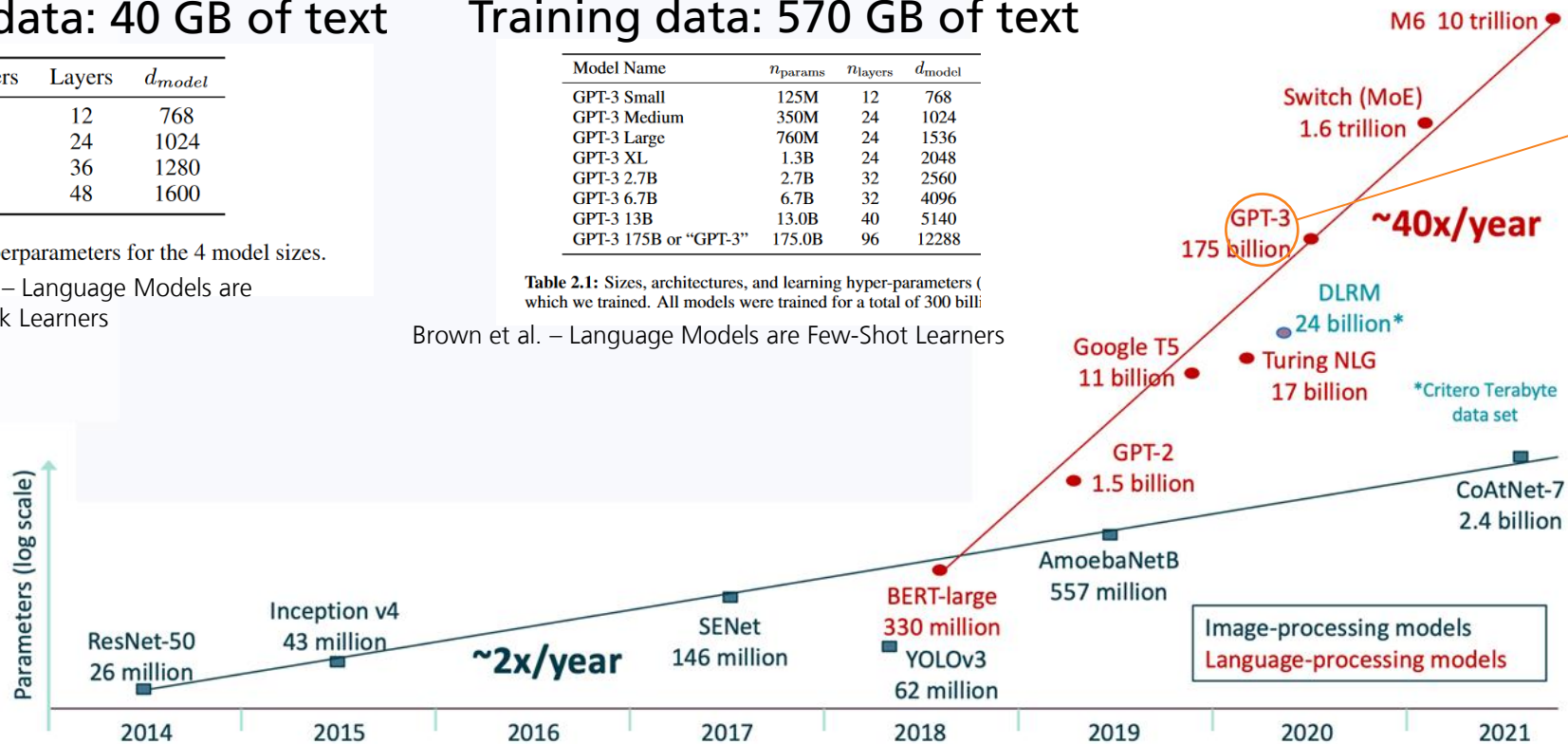
Model Name	$n_{params}$	$n_{layers}$	$d_{model}$
GPT-3 Small	125M	12	768
GPT-3 Medium	350M	24	1024
GPT-3 Large	760M	24	1536
GPT-3 XL	1.3B	24	2048
GPT-3 2.7B	2.7B	32	2560
GPT-3 6.7B	6.7B	32	4096
GPT-3 13B	13.0B	40	5140
GPT-3 175B or "GPT-3"	175.0B	96	12288

Table 2. Architecture hyperparameters for the 4 model sizes.

Source: Radford et al. – Language Models are Unsupervised Multitask Learners

Table 2.1: Sizes, architectures, and learning hyper-parameters (which we trained. All models were trained for a total of 300 billi

Brown et al. – Language Models are Few-Shot Learners



OpenAI GPT3  
175B parameters

Source: <https://leogao.dev/2020/05/29/GPT-3-A-Brief-Summary/>

# GPT-3

## Language Models are Few-Shot Learners

- Trained as a language model, evaluated on NLP/NLU tasks
- No task specific fine-tuning
- In-Context Learning: During inference language model gets a written description of the task and zero, one or few examples as text

The three settings we explore for in-context learning

### Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

```
1 Translate English to French: ← task description
2 cheese => ..... ← prompt
```

### One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

```
1 Translate English to French: ← task description
2 sea otter => loutre de mer ← example
3 cheese => ..... ← prompt
```

### Few-shot

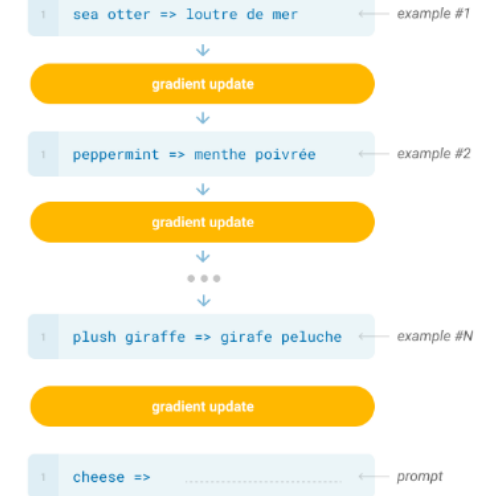
In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

```
1 Translate English to French: ← task description
2 sea otter => loutre de mer ← examples
3 peppermint => menthe poivrée ←
4 plush girafe => girafe peluche ←
5 cheese => ..... ← prompt
```

Traditional fine-tuning (not used for GPT-3)

### Fine-tuning

The model is trained via repeated gradient updates using a large corpus of example tasks.



Source: Brown et al. – Language Models are Few-Shot Learners

# GPT-3

## Language Models are Few-Shot Learners

### Across almost all tasks

- Largest model is best (often by far)
- Few-Shot Setting works best
- In-context learning seems to work
- Other few-shot models are beaten
- Sometimes SOTA-finetuned models are beaten
- Model can deal surprisingly well with 2-3 digit calculations
- More Examples of what GPT-3 can do:  
<https://machinelearningknowledge.ai/openai-gpt-3-demos-to-convince-you-gpt-3-can-steal-jobs-from-software-engineers/>

A "whatpu" is a small, furry animal native to Tanzania. An example of a sentence that uses the word whatpu is:  
We were traveling in Africa and we saw these very cute whatpus.

To do a "farduddle" means to jump up and down really fast. An example of a sentence that uses the word farduddle is:

**One day when I was playing tag with my little sister, she got really excited and she started doing these crazy farduddles.**

A "yalubalu" is a type of vegetable that looks like a big pumpkin. An example of a sentence that uses the word yalubalu is:

**I was on a trip to Africa and I tried this yalubalu vegetable that was grown in a garden there. It was delicious.**

A "Burringo" is a car with very fast acceleration. An example of a sentence that uses the word Burringo is:

**In our garage we have a Burringo that my father drives to work every day.**

A "Gigamuru" is a type of Japanese musical instrument. An example of a sentence that uses the word Gigamuru is:

**I have a Gigamuru that my uncle gave me as a gift. I love to play it at home.**

To "screeg" something is to swing a sword at it. An example of a sentence that uses the word screeg is:

**We screeghed at each other for several minutes and then we went outside and ate ice cream.**

**Figure 3.16:** Representative GPT-3 completions for the few-shot task of using a new word in a sentence. Boldface is GPT-3's completions, plain text is human prompts. In the first example both the prompt and the completion are provided by a human; this then serves as conditioning for subsequent examples where GPT-3 receives successive additional prompts and provides the completions. Nothing task-specific is provided to GPT-3 other than the conditioning shown here.

Source: Brown et al. – Language Models are Few-Shot Learners



# ChatGPT

SV Schreibe und erkläre Code zur Bestimmung der Tiefe eines Baums, erkläre auch die Speicher- und Laufzeitkomplexität.

Die Tiefe eines Baums ist definiert als die Länge des längsten Pfades von der Wurzel zu einem Blattknoten. Ein einfacher Weg, die Tiefe eines Baums zu bestimmen, besteht darin, eine rekursive Funktion zu schreiben, die den größten Wert von links und rechts für jeden Knoten zurückgibt und dann den größten dieser Werte um 1 erhöht, um die Tiefe des Elternknotens zu bestimmen. Hier ist ein Beispiel in Python:

```
def depth(node):  
    if node is None:  
        return 0  
    return 1 + max(depth(node.left), depth(node.right))
```

Die Laufzeitkomplexität dieser Funktion ist  $O(n)$ , da jeder Knoten im Baum genau einmal besucht wird. Die Speicherkomplexität ist ebenfalls  $O(n)$ , da für jeden Knoten im Baum ein Aktivierungsrahmen erstellt wird, der in der Funktion aufgerufen wird.

Alternativ könnte man auch eine iterative Lösung verwenden, indem man eine Queue verwendet, um die Knoten in der Reihenfolge ihrer Tiefe zu durchlaufen. Die Laufzeitkomplexität dieser Lösung ist immer noch  $O(n)$ , aber die Speicherkomplexität ist  $O(w)$ , wobei  $w$  die Breite des Baums ist (d.h. die maximale Anzahl von Knoten auf einer gegebenen Ebene).

```
def depth(root):  
    if root is None:  
        return 0  
  
    depth = 0  
    queue = [(root, 1)]  
    while queue:  
        node, level = queue.pop(0)  
        depth = max(depth, level)  
        if node.left:  
            queue.append((node.left, level+1))  
        if node.right:  
            queue.append((node.right, level+1))  
    return depth
```

Source: <https://chat.openai.com/chat>

SV Schreibe die Outline für eine Präsentation vor Bachelor-Studenten der Informatik die für das Thema "große Sprachmodelle" begeistert werden sollen.

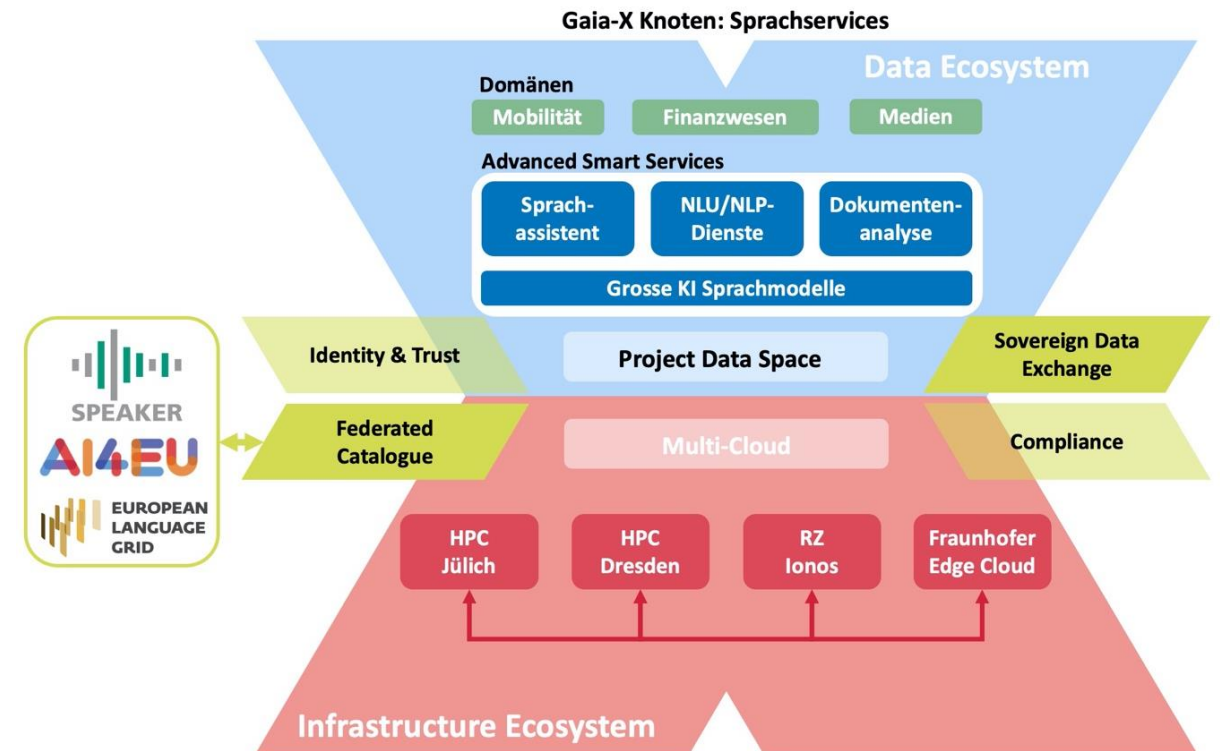
1. Einführung in große Sprachmodelle: Erklären Sie, was große Sprachmodelle sind und warum sie wichtig sind. Erwähnen Sie, dass sie in verschiedenen Anwendungen wie Maschinenübersetzung, Textgenerierung und Dialogsystemen eingesetzt werden.
2. Geschichte von Sprachmodellen: Zeigen Sie auf, wie sich die Leistung von Sprachmodellen im Laufe der Jahre verbessert hat und welche Fortschritte dazu beigetragen haben. Erwähnen Sie auch wichtige Meilensteine wie das Aufkommen von Wortvektoren und neuronale Netze.
3. Funktionsweise von großen Sprachmodellen: Erklären Sie, wie große Sprachmodelle funktionieren und welche Arten von Architekturen verwendet werden (z.B. Transformer, BERT). Erklären Sie auch, wie sie trainiert werden und wie sie ihre Leistung verbessern.
4. Anwendungen von großen Sprachmodellen: Zeigen Sie auf, wie große Sprachmodelle in verschiedenen Anwendungen eingesetzt werden und welche Vorteile sie bieten. Erwähnen Sie auch mögliche Einschränkungen und Herausforderungen bei der Verwendung von großen Sprachmodellen.
5. Zukunft von großen Sprachmodellen: Diskutieren Sie, welche Fortschritte in der Zukunft erwartet werden und welche Auswirkungen diese auf die Anwendungen von großen Sprachmodellen haben werden. Erwähnen Sie auch mögliche ethische Herausforderungen und die Notwendigkeit von verantwortungsvollem Einsatz.
6. Schlussfolgerung: Zusammenfassen Sie die wichtigsten Punkte und zeigen Sie auf, warum große Sprachmodelle ein wichtiges und vielversprechendes Gebiet der Informatik sind. Ermutigen Sie die Studenten, sich weiter mit dem Thema zu beschäftigen und ihr eigenes Verständnis zu vertiefen.

Source: <https://chat.openai.com/chat>

# OpenGPT-X: Large Language Models made in Europe

## Creation of large language models with data privacy and security in mind

- Open and accessible language models
- Assuring digital sovereignty
- Partners from research and industry collaborate together
- First 13bn parameter has been trained on about 600 GPUs
- Next Year: 150bn parameter model on 6.000 GPUs
- Project Volume: 19 Mio. €



Source: Fraunhofer IAIS

# Cool Ressources

---

## Good online courses on machine learning and NLU:

- <https://www.udacity.com/courses/all?field=school-of-data-science&field=school-of-ai&price=Free>
- <https://www.coursera.org/specializations/machine-learning-introduction>
- Stanford Course: <https://www.youtube.com/watch?v=rmVRLeJRkl4>

## Playgrounds:

- <https://huggingface.co/>
- <https://openai.com/blog/chatgpt/>
- <https://openai.com/dall-e-2/>

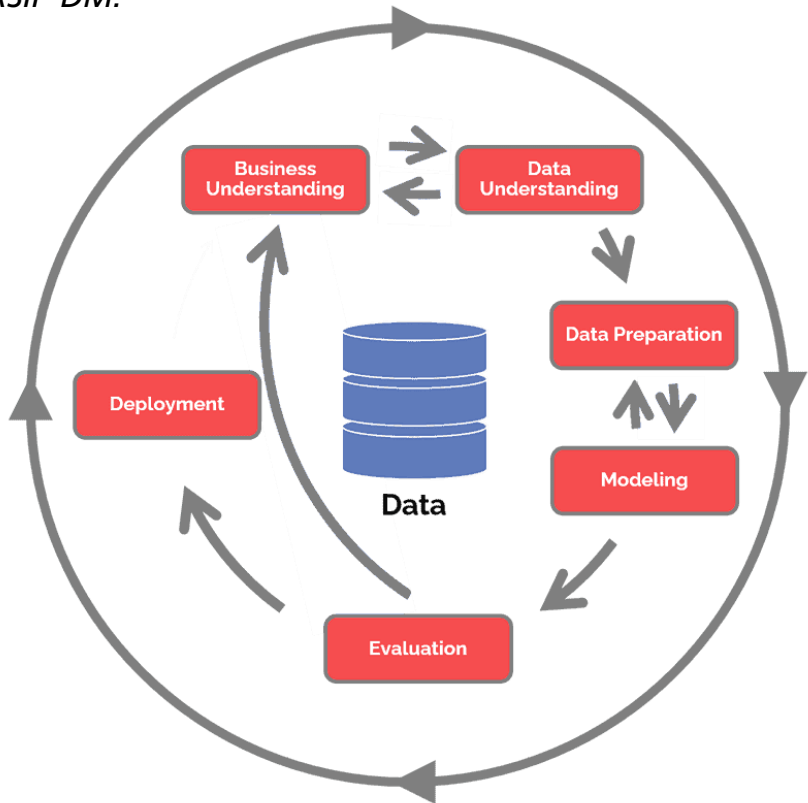
## Challenges:

- <https://www.kaggle.com/competitions>

# Teaser: MLOps

## Tomorrow 6pm c.t.

CRISP-DM:



Source: <https://www.datascience-pm.com/crisp-dm-2/>

MLOps:



Image rights are held by: Dennis Wegener, Fraunhofer IAIS

# Summary

---

- Artificial Intelligence (AI) and Natural Language Understanding (NLU) are becoming more and more present in our every day life
- Natural Language Understanding enables the automated processing of text/speech data
- Sample Use Cases:
  - Automatic Topic Discovery in large text collections
  - Automated Fake News Detection
  - Automated processing of patient histories
  - Automated generation of discharge summaries
- Large language models and pre-training + fine-tuning has become the standard procedure in NLU
- Models such as ChatGPT can perform tasks without being explicitly trained on them just by interpreting natural language prompts

# Become a student assistant at IAIS.NLU Team

# Student Assistant Job

Become a student assistant at IAIS.NLU Team

---

## Initiativbewerbung / Speculative application

- **QR-Code or Link**
  - <https://jobs.fraunhofer.de/job/Sankt-Augustin-Initiativbewerbung-studentische-Hilfskraft-53757/765891401/>
- Apply for one of our Teams „NLU“ or „MLOps“
  
- Or visit this link to get an overview
  - <https://www.iais.fraunhofer.de/de/karriere.html>
  - <https://www.iais.fraunhofer.de/en/career.html>

