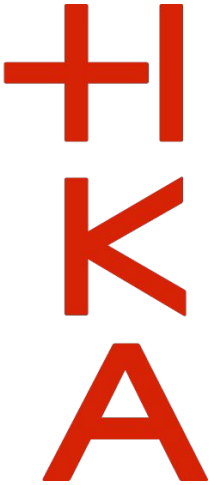
inovex



KI Labor - Sommersemester 2022

NLP - Sprintwechsel & Vorstellung Assignment

Robin Baumann, Sven Müller, **Maximilian Blanck, Pascal Fecht,** Tim Bossenmaier, Matthias Richter

Karlsruhe, 06. Mai 2022

Schedule

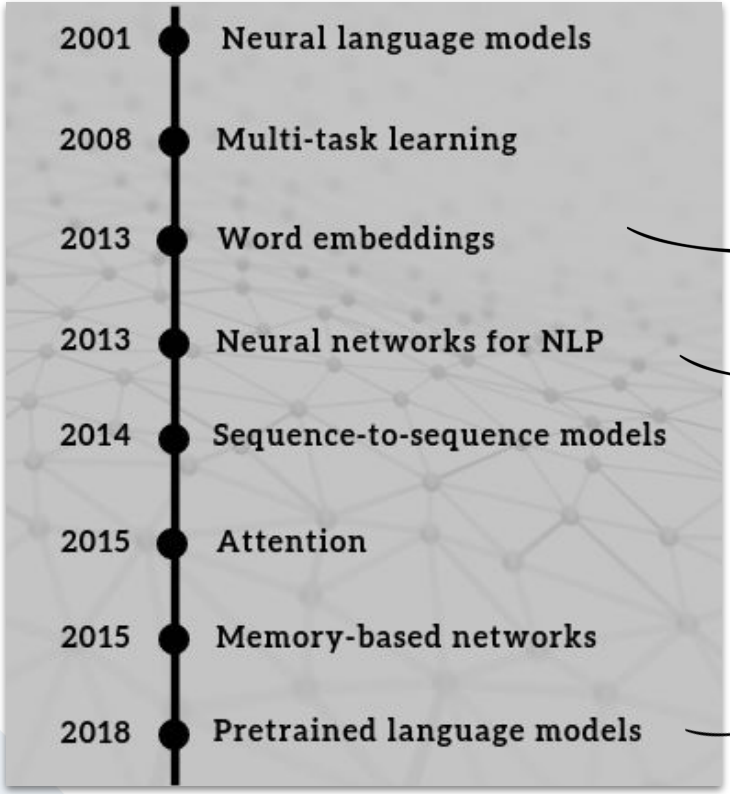
Datum	Thema	Inhalt	Präsenz
18.03.22	**Allg.**	**Organisation, Teamfindung, Vorstellung CV**	Ja
25.03.22	CV	Q&A Sessions	Nein
01.04.22	CV	Sprintwechsel, Vorstellung Assignment	Ja
08.04.22	CV	Q&A Sessions	Nein
15.04.22	Ostern		
22.04.22	CV / NLP	Abgabe CV, Vorstellung NLP	Ja
29.04.22	NLP	Q&A Sessions	Nein
06.05.22	**NLP**	**Sprintwechsel, Vorstellung Assignment**	**Ja**
13.05.22	NLP	Q&A Sessions	Nein
20.05.22	**NLP / RL**	**Abgabe NLP, Vorstellung RL**	**Ja**
27.05.22	RL	Sprintwechsel, Vorstellung Assignment	Nein
03.06.22	Ausfall (Sommerplenum)		
10.06.22	Pfingsten (H-KA zu)		
17.06.22	RL	Q&A Sessions (Brückentag)	Nein
24.06.22	**RL**	**Abgabe RL, Abschluss KI Labor**	**Ja**
01.07.22		Puffer	

Agenda

- **Besprechung Übungsaufgaben**
  - Word Embeddings Alice im Wunderland (Aufgabe 1)
  - Sentiment Analyse für Twitter Posts (Aufgabe 2)
- **Vorstellung Assignment**
  - Transfer Learning *bzw.* Zero-/Few-Shot Learning mit Transformern

Deep Learning in NLP

Die letzten 20+ Jahre in NLP



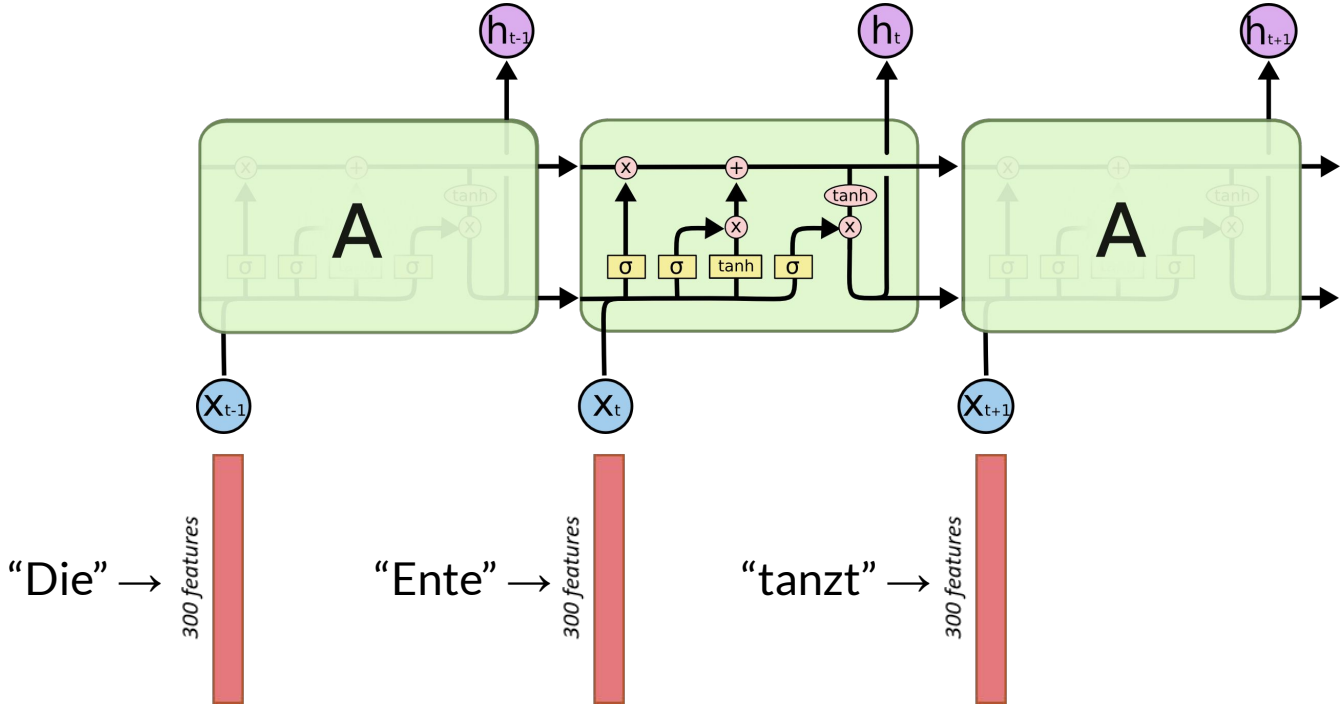
1. Übungsaufgabe

2. Übungsaufgabe

3. Assignment

https://ruder.io/a-review-of-the-recent-history-of-nlp/

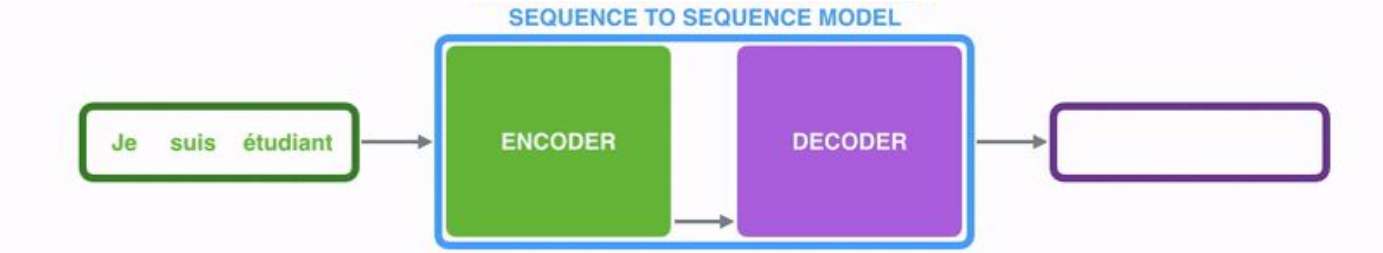
Example: LSTM - Long Short Term Memory



https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Sequence2Sequence Model

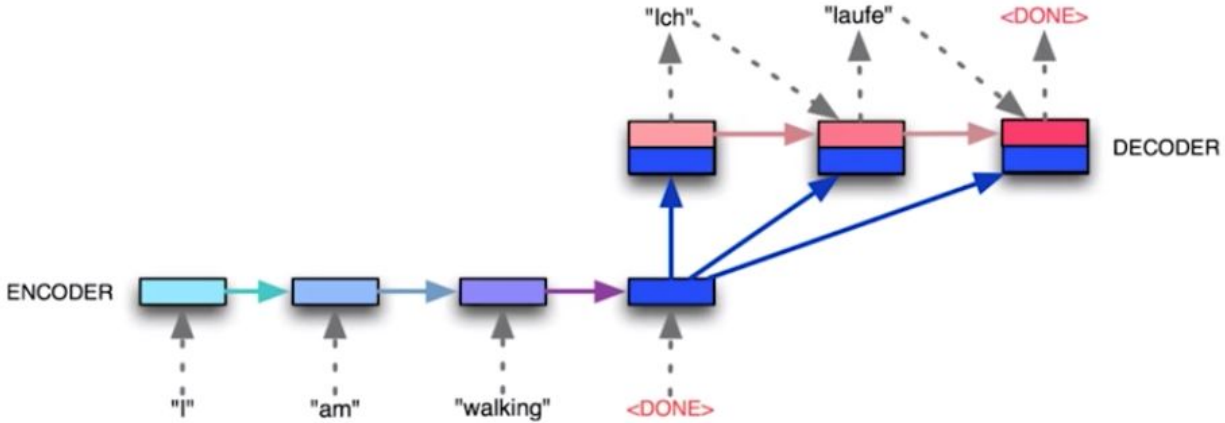
UseCase: Maschinelles Übersetzen (Encoder/Decoder)



https://jalammar.github.io/visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-attention/

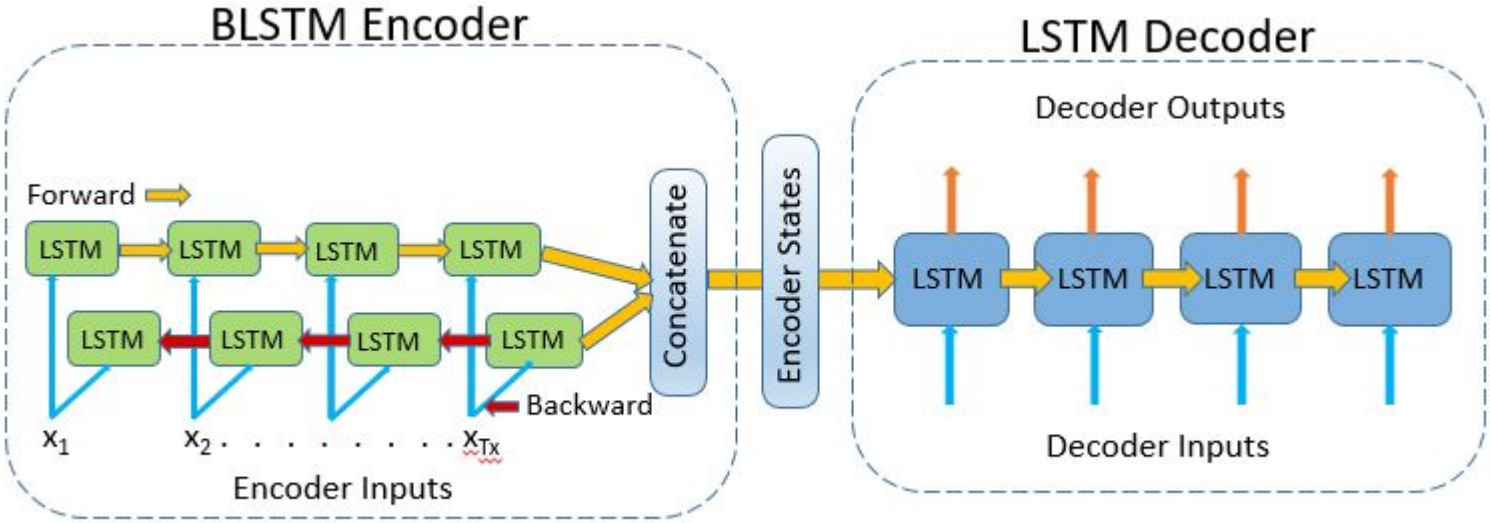
Encoder-Decoder Architektur

Maschinelles Übersetzen



inovex

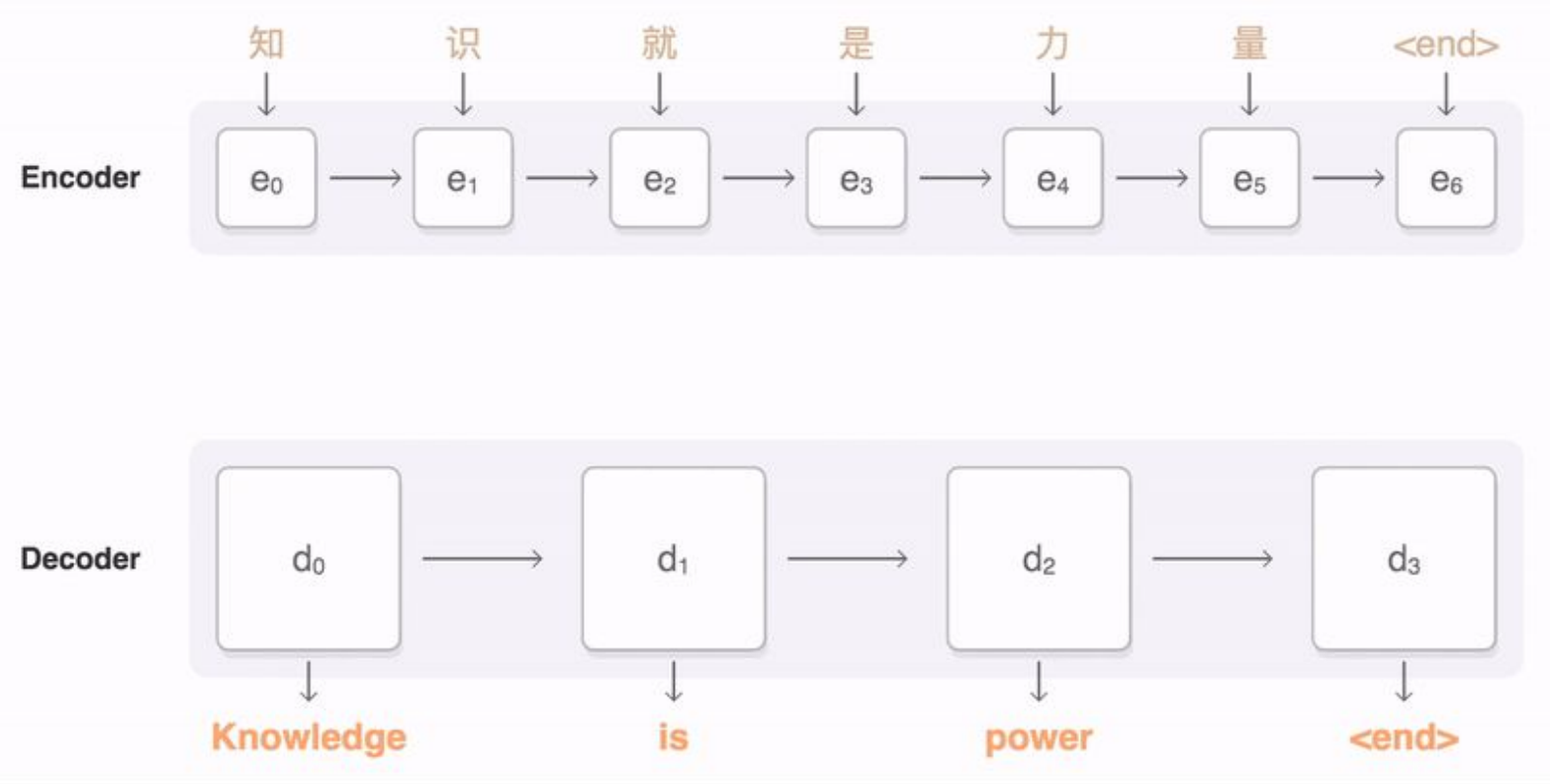
Decoder I/O während Training / Inferenz



- **Beispiel Machine Translation**
- **Training: Der Decoder wird mit dem Übersetzungstext der Zielsprache trainiert**
- **Inferenz: Start Token von Zielsprache und Encoder State wird übergeben**

inovex

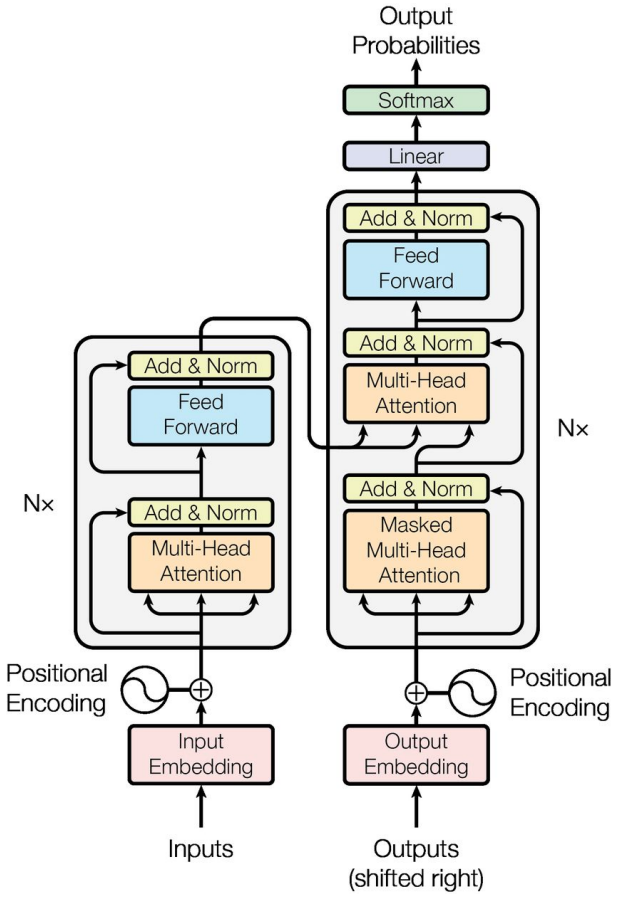
Attention



inovex

Transformer

- **Keine Rekurrenz**: Statt Wort-für-Wort werden Sequenzen (Wörter, Sätze) parallel verarbeitet
- **Self-Attention**: Wichtigkeit von Wörtern im Satz wird gelernt
- **Positional Encodings** kombiniert mit Embeddings um Position im Satz zu behalten.



inovex

Language Models

Language Models

Language Model (LM): Gegeben eines Kontexts, was ist das nächste Wort?

- the weather was *<target>* ⇒ target = hot
- Grundlage für viele NLP tasks (in abgewandelter Form auch für word embeddings)

⇒ Semi-supervised Task (keine Labels benötigt)

⇒ Kann auf großen Datensätzen, gecrawlt im Netz, trainiert werden.

Typen von Language Modellen

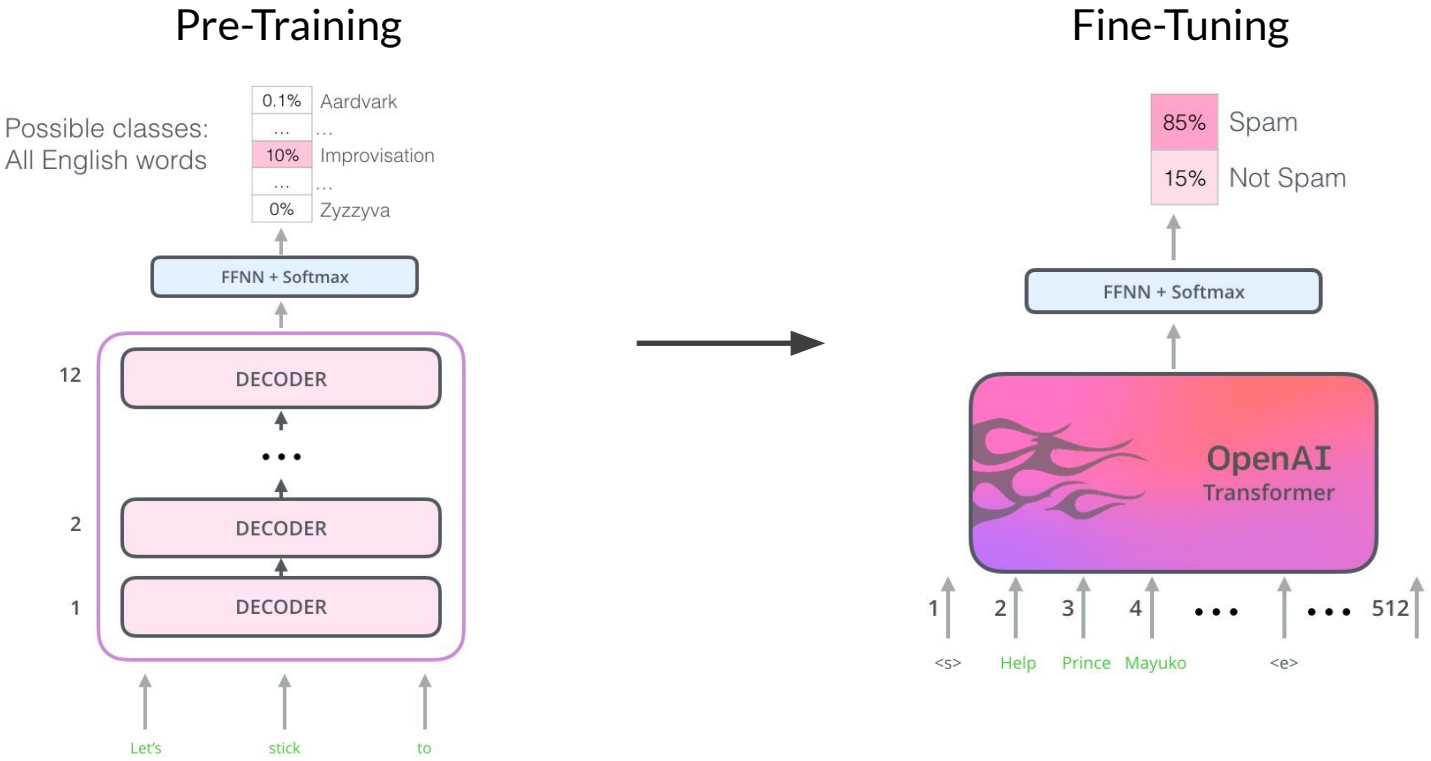
Modelltyp	Beschreibung	Beispiele
Autoregressiv	• Sage das nächste Wort auf Basis der vorherigen vor. • Basierend auf Transformer-**Decoder** • Unidirektional • Häufig für text generative Tasks.	GPT-1 GPT-2 XL-NET
Auto-Encoding	• Sage ein fehlendes Wort in einem Text voraus (Lückentext) • Basierend auf Transformer-**Encoder** • Bidirektional • Häufig für Token- und Sentence-Klassifikation	BERT ALBERT ROBERTA
Seq-2-Seq	• Encoder und Decoder aus Original Paper • Für Anwendungen wie Translation, Summarization und QA	Transformer BART T5

https://medium.com/swlh/understanding-gpt-3-openais-latest-language-model-a3ef89cffac2

Transfer Lernen in NLP

Sequential Transfer Learning

OpenAI Transformer

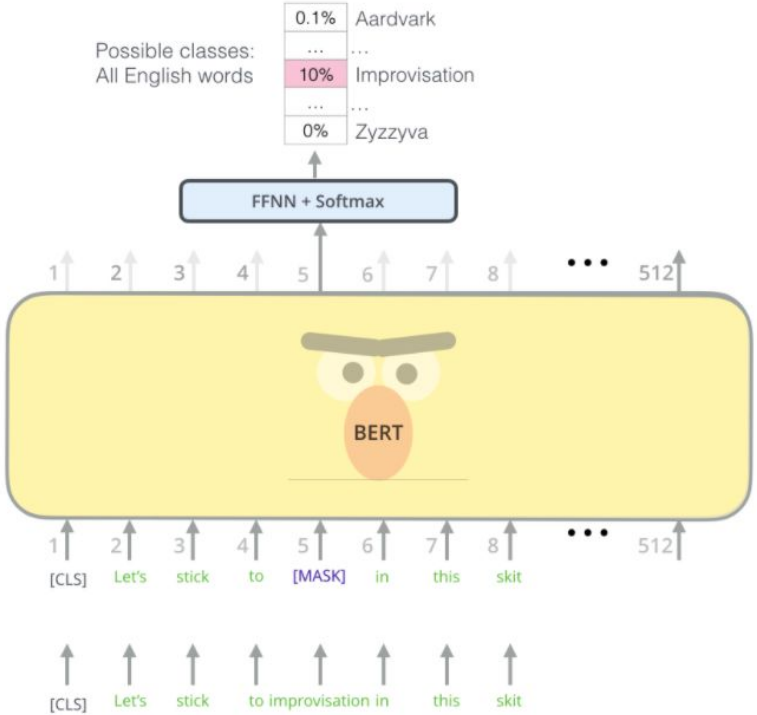


https://jalammar.github.io/illustrated-transformer/

BERT

- Basiert auf Transformer Encodern
- Masked-Language Model

Aber: Transfer Learning Idee bleibt.



https://jalammar.github.io/illustrated-bert/

Zero- / Few-Shot-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
```

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
```

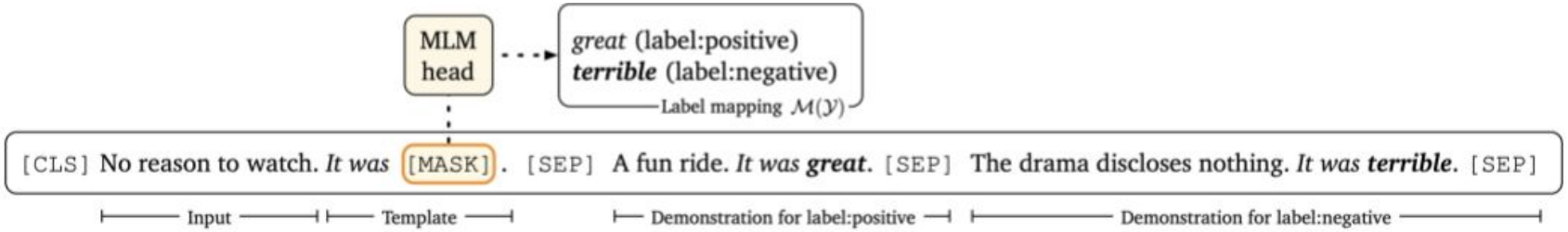
Beispiele: https://beta.openai.com/examples

Language Models are Few Shot Learners, https://arxiv.org/pdf/2005.14165.pdf

Zero- / Few-Shot-Learning

- **Text Generation**
  - Output kann direkt verwendet werden
  - Häufig Prefix Prompts (meistens Autoregressive Modelle)

- **Text Classification**
  - Häufig Lückentext-Prompts (meistens Auto-Encoding Prompts)
  - Label Mapping (Answer Engineering) notwendig



inovex

Prompt Engineering

Forschungsfeld rund um die Frage: Wie finde ich den besten Prompt für meinen Task und Datensatz?

- Viele Techniken (z. B. Nicht direkt fragen, sondern Kontext mitgeben)
- Prompts suchen lassen (z. B. autoprompt für Auto-Encoding-Models)
- …

https://generative.ink/posts/methods-of-prompt-programming/

Assignment

Assignment

Open-Ended Assignment

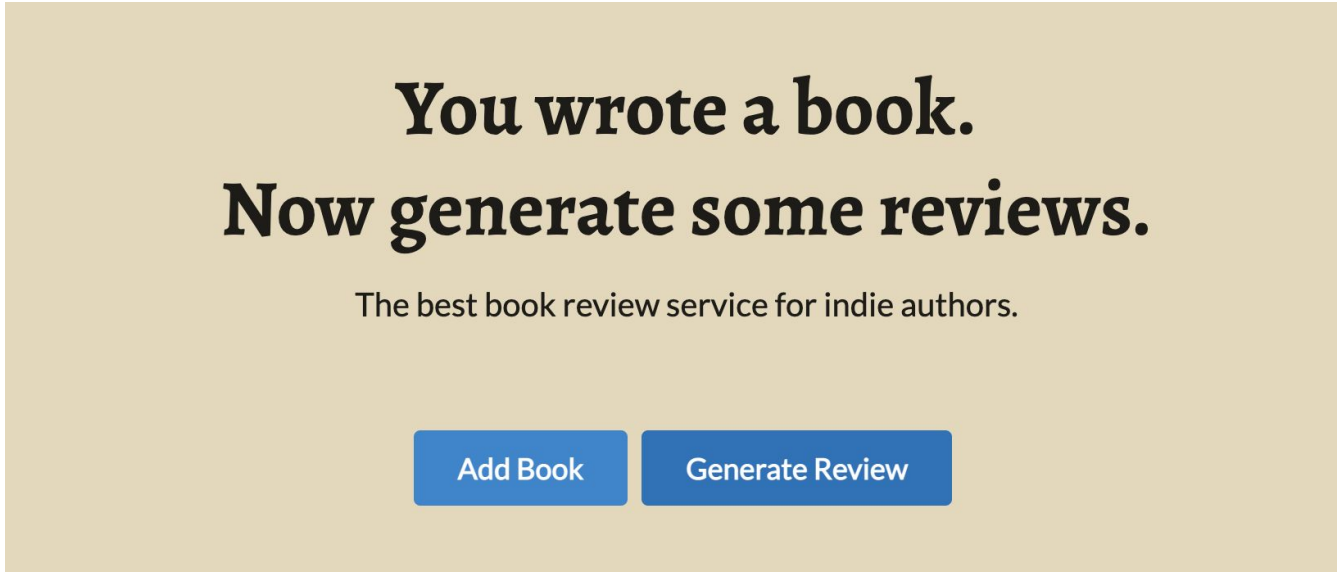
Anforderungen

- Wähle einen NLP-Task (Translation, Text Generation, Classification, QA, …) und Datensatz.
- Datensatz und Task verstehen und erklären.
- Transformer-basiertes Modell auf Datensatz und Task anwenden
  - **Minimale Anforderung:** Transfer Learning *oder* Zero-/ Few-Shot Learning (nur ein Ansatz)
  - Gewählten Ansatz verstehen und Vorgehen erklären.
  - Ergebnisse evaluieren und erklären .

Huggingface Transformers

- De facto Standard für Transformer in NLP
- Python Library für Training, Fine-Tuning, Deployment, ...
- Model Hub: Tausende Modelle für unterschiedliche
  - Tasks
  - Datensätze
  - Sprachen

Beispiel: Generierung von Book Reviews



https://www.bookreview.io/

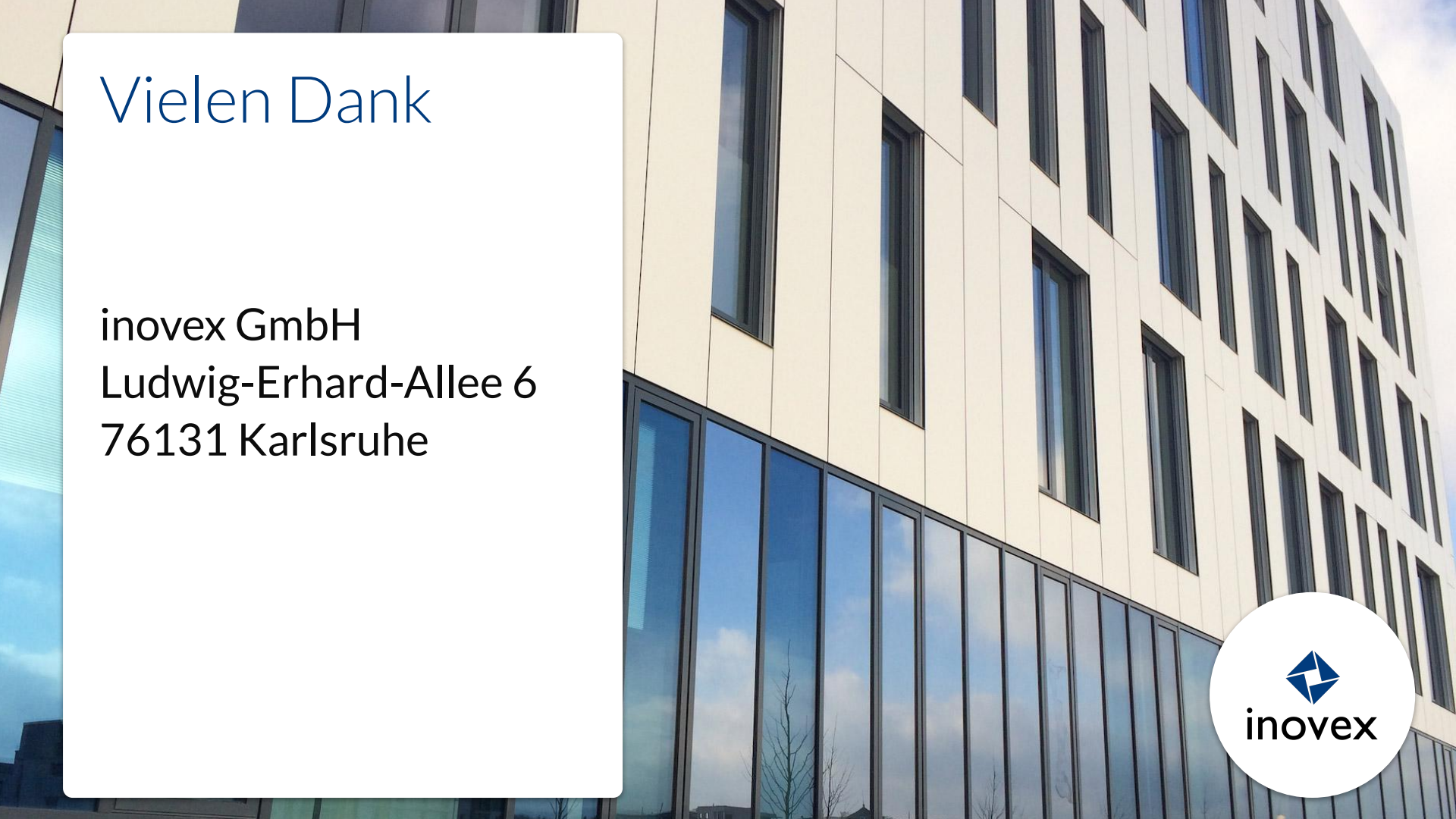
inovex

Assignment

- Quellen zur Inspiration
  - https://beta.openai.com/examples
  - https://www.buildgpt3.com/
  - https://paperswithcode.com/methods/area/natural-language-processing
  - https://huggingface.co/models

Vielen Dank

inovex GmbH
Ludwig-Erhard-Allee 6
76131 Karlsruhe



Padding

Problem: Nicht alle Sequenzen haben die gleiche Länge...

- **Beispiel**
  - Text_1: [“Die”, “Ente”, “tanzt”, “und” ,”quakt”]
  - Text_2: [“Die”, “Ente”, “schwimmt”]
- Die beiden Sätze sind in einem Batch und sollen an das NN gefüttert werden.
- Problem: Wir müssen mit Tensoren arbeiten, die die gleichen Dimensionen haben.
- Lösung Padding:
  - Text_1: [“Die”, “Ente”, “tanzt”, “und” ,”quakt”]
  - Text_2: [“Die”, “Ente”, “schwimmt”, ”PADDING”, ”PADDING”]
  - Batch : [[1,2,3,4,5],[1,2,6,0,0]]