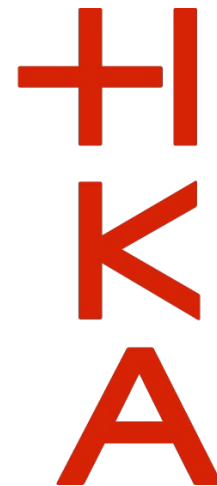




KI Labor - Wintersemester 2021

NLP - Sprintwechsel &
Vorstellung Assignment



Darjan Salaj, Sven Müller, **Maximilian Blanck**,
Sebastian Blank, Frederik Martin, **Pascal Fecht**

Karlsruhe, 19. Nov. 2021

Schedule

Datum	Thema	Inhalt	Präsenz
01.10.21	Allg.	Organisation, Teamfindung	Nein
08.10.21	CV	Vorstellung CV	Nein
15.10.21	CV	Q&A Sessions	Nein
22.10.21	CV	Sprintwechsel, Vorstellung Assignment	Ja
29.10.21	CV	Q&A Sessions	Nein
05.11.21	CV / NLP	Abgabe CV, Vorstellung NLP	Ja
12.11.21	NLP	Q&A Sessions	Nein
19.11.21	NLP	Sprintwechsel, Vorstellung Assignment	Ja
26.11.21	NLP	Q&A Sessions	Nein
03.12.21	NLP	Keine Veranstaltung Q&A Sessions	Nein
10.12.21	NLP / RL	Abgabe NLP, Vorstellung RL	Ja
17.12.21	RL	Q&A Sessions	Nein
14.01.22	RL	Sprintwechsel, Vorstellung Assignment	Ja
21.01.22	RL	Q&A Sessions	Nein
28.01.22	RL	Abgabe RL, Abschluss KI Labor	Ja

Agenda

› **Besprechung Übungsaufgaben**

- Word Embeddings Alice im Wunderland (Aufgabe 1)
- Sentiment Analyse für Twitter Posts (Aufgabe 2)

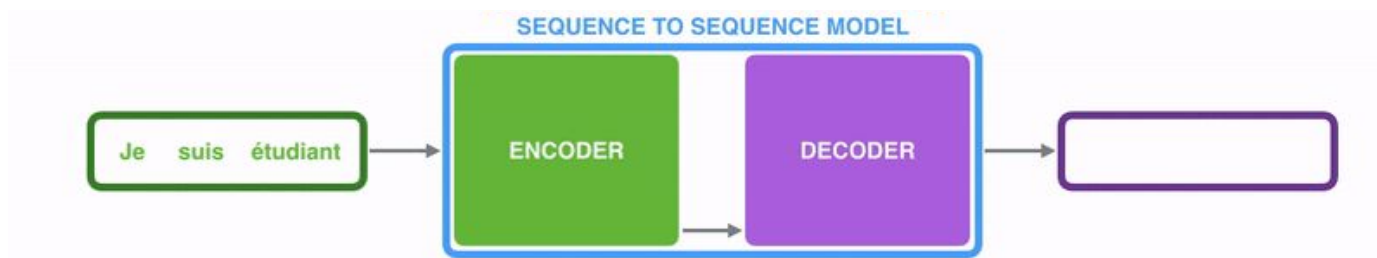
› **Vorstellung Assignment**

- Sequence-To-Sequence Modelle
- Implementierung eines Seq2Seq Modells, *oder*
- Sequential Transfer Learning mit Transformern

Sequence-**To**-Sequence

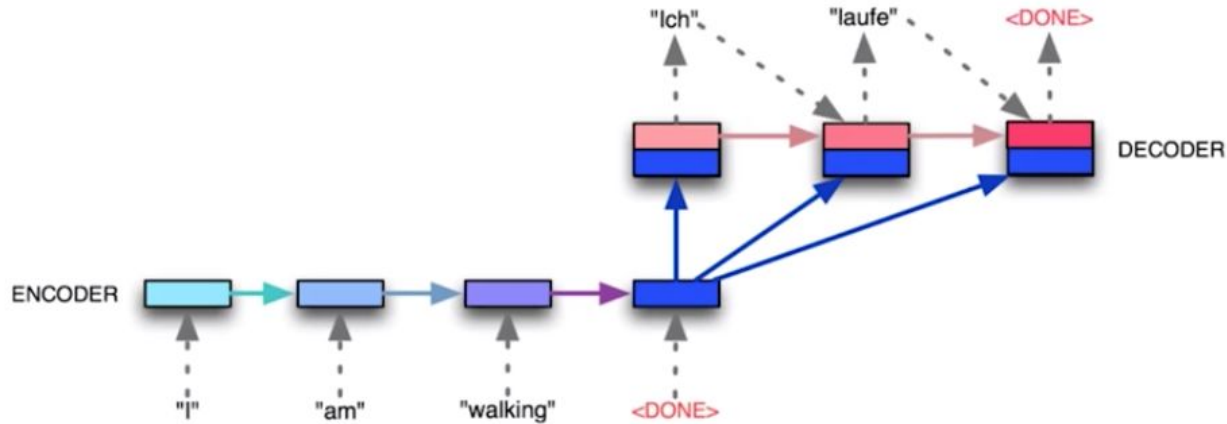
Sequence2Sequence Model

UseCase: Maschinelles Übersetzen (Encoder/Decoder)

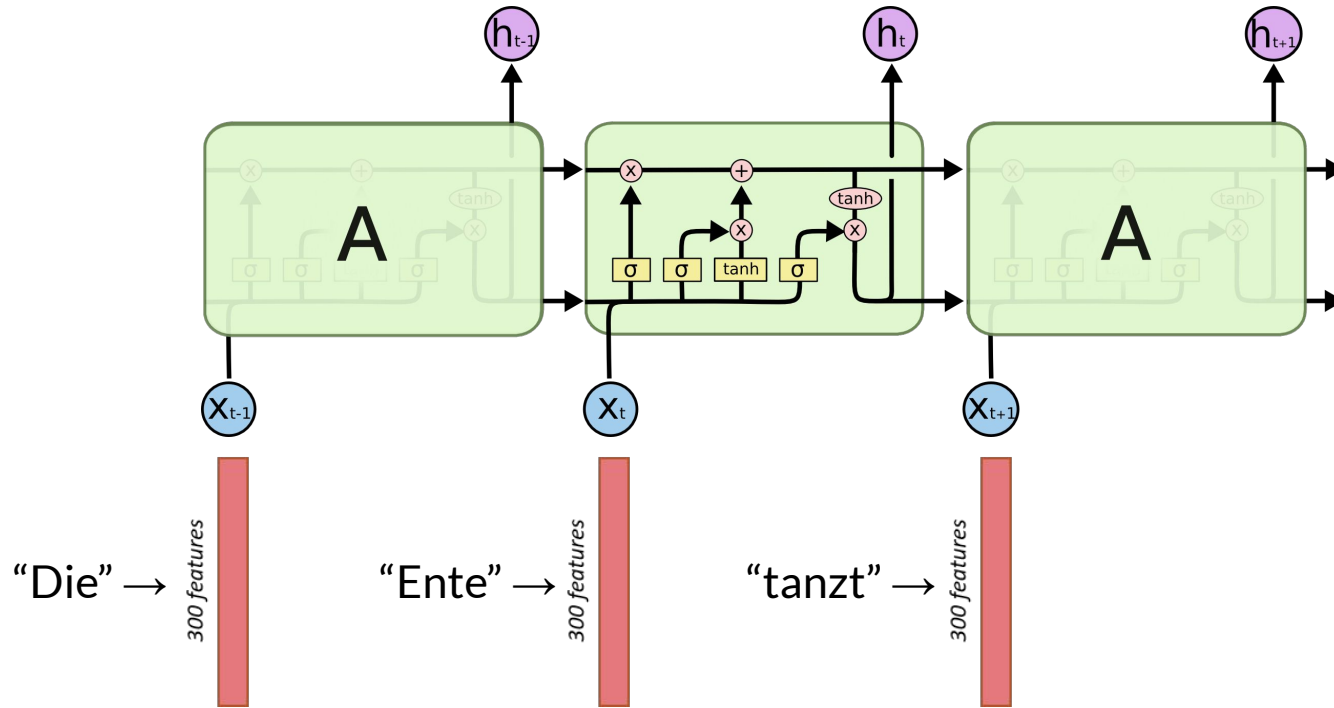


Encoder-Decoder Architektur

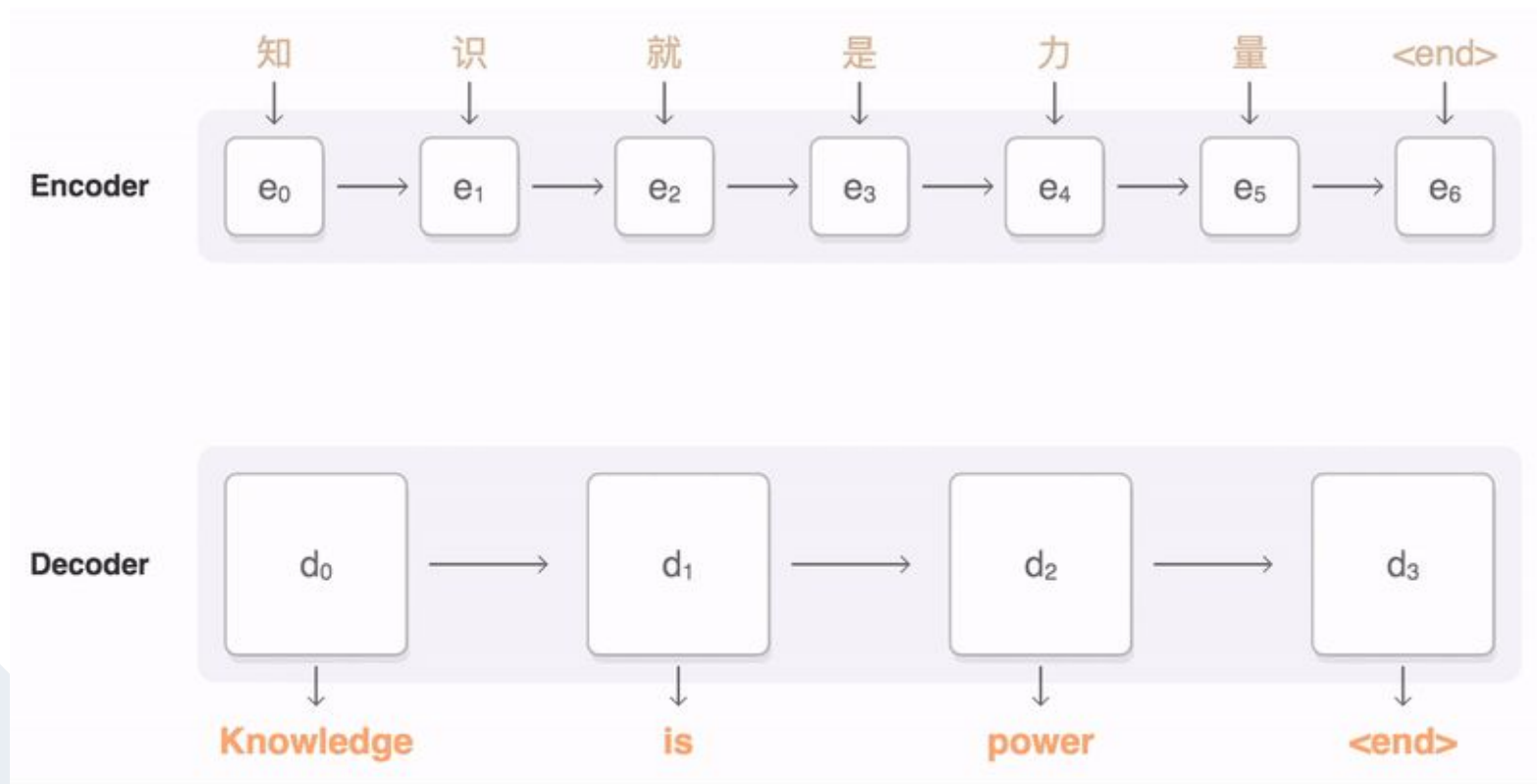
Maschinelles Übersetzen



LSTM - Long Short Term Memory

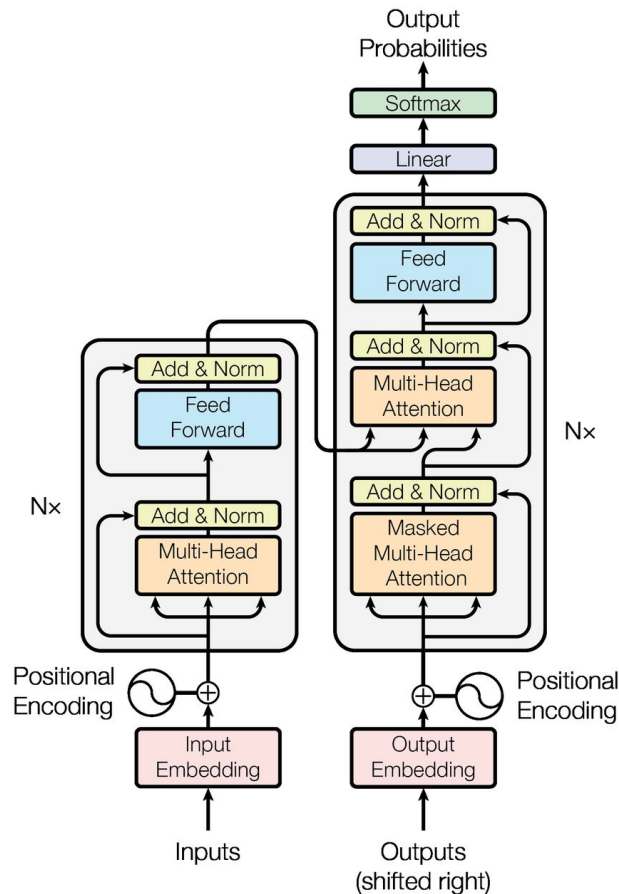


Attention



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.



Implementierung eines Seq2Seq Modells

Implementierung Encoder- Decoder

Encoder

- › Auswahl von RNN / Attention / Transformer
- › Bei RNNs: Uni- oder Bidirektional
- › Embeddings (Chars, Wörter)
- › **Padding**

Decoder

- › **Welche Inputs werden bei Training/Inferenz benötigt**
- › Was ist genau der Output

Padding

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

- › **Beispiel**

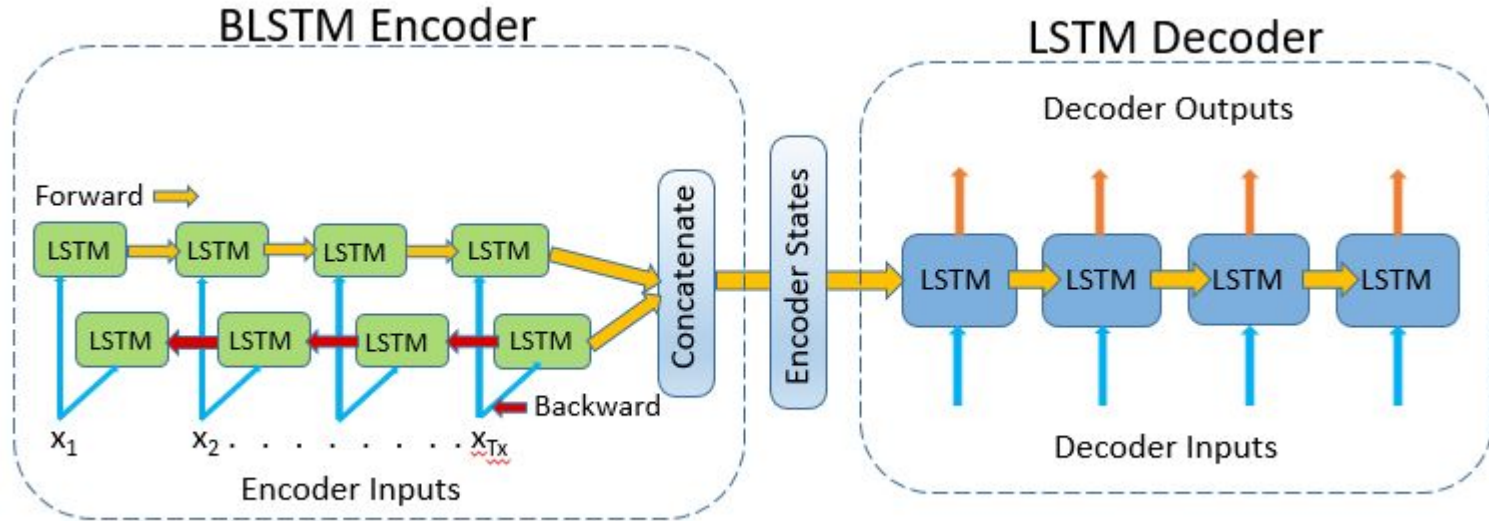
- Text_1: ["Die", "Ente", "tanzt", "und", "quakt"]
- Text_2: ["Die", "Ente", "schwimmt"]

- › 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]]

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

Sequential Transfer Learning mit **Transformern**

Language Model

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

› the weather was <target> ⇒ target = hot

⇒ Semi-Supervised Task

⇒ Große LMs sind Grundlage für Transfer Lernen in NLP

⇒ Sprachmodelle können bidirektional sein.

Transfer Lernen in NLP

Sequential Transfer Learning

1. Pre-Training

Typically: *Language modeling*

Source task



2. Fine-Tuning

Classification

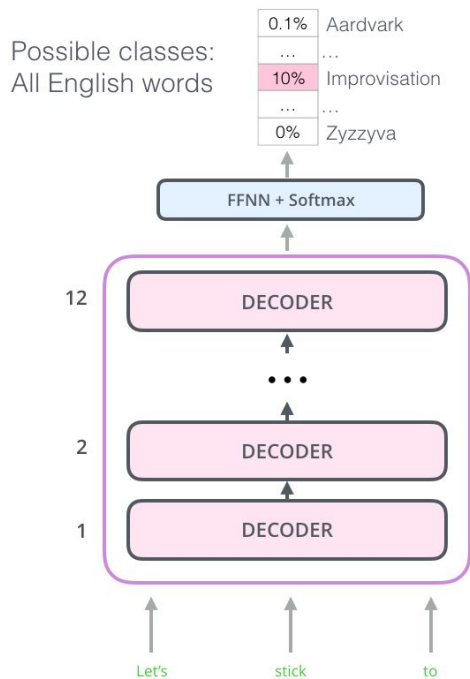
Seq2Seq (Translation, ..)

...

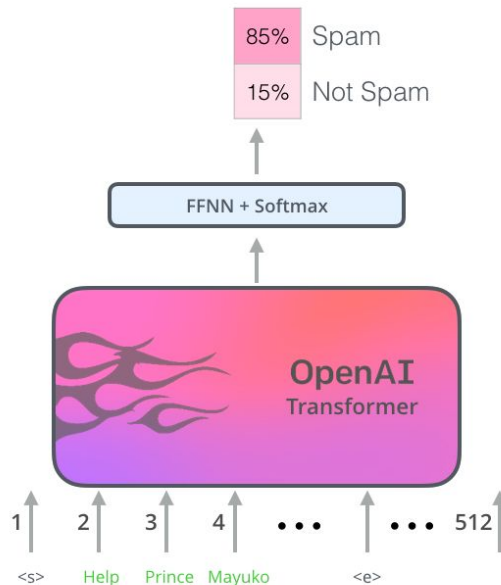
Target task(s)

OpenAI Transformer

Pre-Training



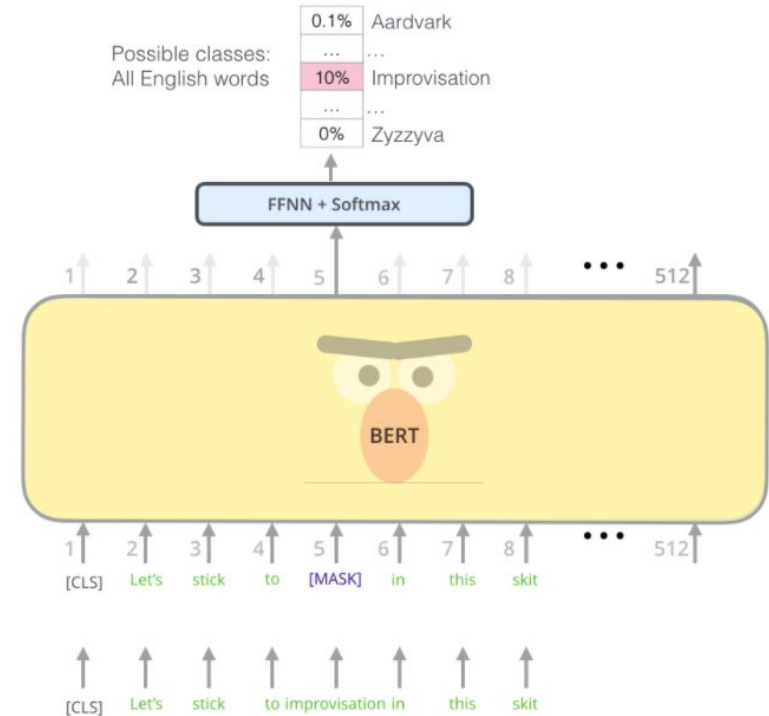
Fine-Tuning



BERT

- › Basiert auf Transformer Encodern
- › Masked-Language Model

Aber: Transfer Learning Idee bleibt.



BERT

Adaptionen und Weiterentwicklungen

- › Robustere BERT Modelle (z. B. [ROBERTa](#))
- › Verschiedene Sprachen (z.B. [CamemBERT](#))
- › Verkleinerung des Modells (z.B. [DistilBERT](#))
- › Text-To-Text Transfer Transformer ([T5](#))
- › Viele weitere (siehe [tomohideshibata/BERT-related-papers](#))

Huggingface Transformers



- › Python Library für Training, Fine-Tuning, Deployment, ...
- › Model Hub: Tausende Modelle für unterschiedliche
 - › Tasks
 - › Datensätze
 - › Sprachen
 - › <https://huggingface.co/models> für Text2Text

Assignment

Assignment

Open-Ended Assignment

Minimale Anforderungen

- › Wähle einen passenden Seq2Seq-Task (Machine Translation, Text summarization, Text generation, ...) und Datensatz.
 - › Datensatz und Task verstehen und erklären.
 - › Netzwerk und Ergebnisse verstehen und erklären.
- › Entweder (nur 1 aussuchen und machen!):
 - › Implementierung eines Seq2Seq Modells und auf Datensatz / Task anwenden, *oder*
 - › Fine-Tuning eines Transformers auf Datensatz / Task

Assignment

› Quellen zur Inspiration

- › <https://machinelearningmastery.com/text-generation-lstm-recurrent-neural-networks-python-keras/>
- › <https://blog.keras.io/a-ten-minute-introduction-to-sequence-to-sequence-learning-in-keras.html>
- › <https://machinelearningmastery.com/the-bahdanau-attention-mechanism/>
- › <https://paperswithcode.com/method/seq2seq>
- › <https://huggingface.co/models>

Vielen Dank

inovex GmbH
Ludwig-Erhard-Allee 6
76131 Karlsruhe

