Aktueller Stand der Forschung in Neuronaler Maschineller Übersetzung

Christopher Brix
christopher.brix@rwth-aachen.de

January 14th, 2020

www.christopher-brix.de

Inhalt

Über Mich

Studium:

- 2014 - 2018: B.Sc. Informatik, RWTH Aachen
- 2018 - 2020: M.Sc. Informatik, RWTH Aachen

Forschung:

- Wissenschaftliche Hilfskraft seit 2016
  - i6: Human Language Technology and Pattern Recognition (Prof. Dr.-Ing. Ney)
  - Betreuer: Parnia Bahar
- Co-Autor von "Empirical Investigation of Optimization Algorithms in Neural Machine Translation", veröffentlicht im PBML
- Co-Autor paper "Towards Two-Dimensional Sequence to Sequence Model in Neural Machine Translation", veröffentlicht in EMNLP

Highlights:

- LxMLS, Teilnehmer & Tutor
- Google NLP Summit 2019
- Google Research Intern 2020

Neural Machine Translation (NMT)

Maschinelle Übersetzung:

- **Übersetzung eines Quellsatzes f_1^J in die Ziel-Hypothese $\hat{e}_1^{\hat{I}}$**
- $\hat{e}_1^{\hat{I}} = \underset{I, e_1^I}{\operatorname{argmax}} \left\{ \Pr(e_1^I | f_1^J) \right\}$

SMT:

- **Zerlegung mittels Bayes Theorem**
- $\hat{e}_1^{\hat{I}} = \underset{I, e_1^I}{\operatorname{argmax}} \left\{ \Pr(f_1^J | e_1^I) \cdot \Pr(e_1^I) \right\}$

NMT:

- **Direkte Modellierung von $\Pr(e_1^I | f_1^J)$**
- **Wörter werden von neuronalem Netz generiert**

Encoder-Decoder

Idee: Erst Zusammenfassen, dann Übersetzuen [Sutskever+ 14]

- **Zusammenfassen des Quell-Satzes in einem Vektor fester Größe**
- **Ausgeben der übersetzten Zusammenfassung**

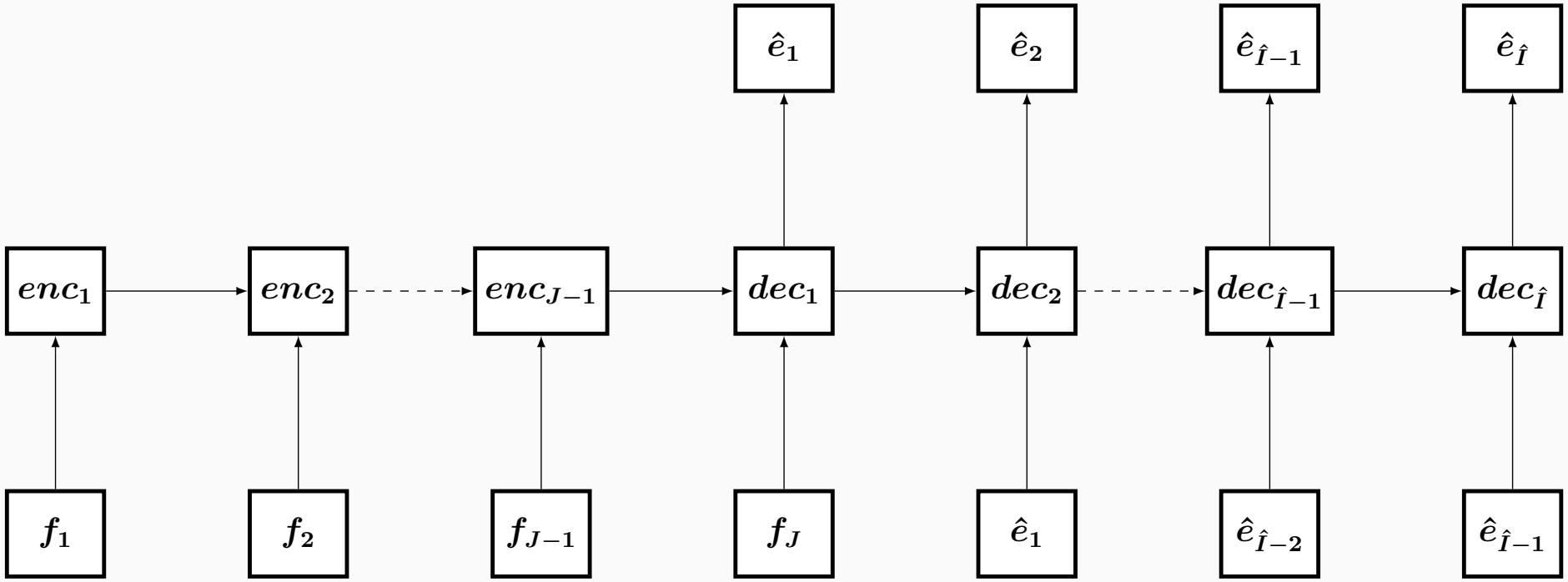


Figure: Architektur eines Encoder-Decoder NMT Systems

Attention

Idee: Fokus auf einzelnen Wörtern [Bahdanau+ 15]:

- **Zusammenfassen eines Satzteils**
- **Ausgeben des nächsten Wortes**
- **Wiederholen, bis Übersetzung komplett ist**

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

Attention

Encoder:

$$f_1^J \rightarrow \overrightarrow{h_j} = LSTM(f_j, \overrightarrow{h}_{j-1})$$

$$f_1^J \rightarrow \overleftarrow{h_j} = LSTM(f_j, \overleftarrow{h}_{j+1})$$

$$h_j = \begin{bmatrix} \overrightarrow{h_j} \\ \overleftarrow{h_j} \end{bmatrix}$$

Attention:

$$\alpha(j|i) = A_j(s_{i-1}, h_1^J)$$

$$c_i = \sum_{j=1}^{J} \alpha(j|i) \cdot h_j$$

Decoder:

$$e_i \leftarrow t_i = Y(e_{i-1}, s_{i-1}, c_i)$$

$$s_i = LSTM([e_i, c_i], s_{i-1})$$

$$p_i(e_i = w|e_1^{i-1}, f_1^J) = softmax(t_i)_w$$

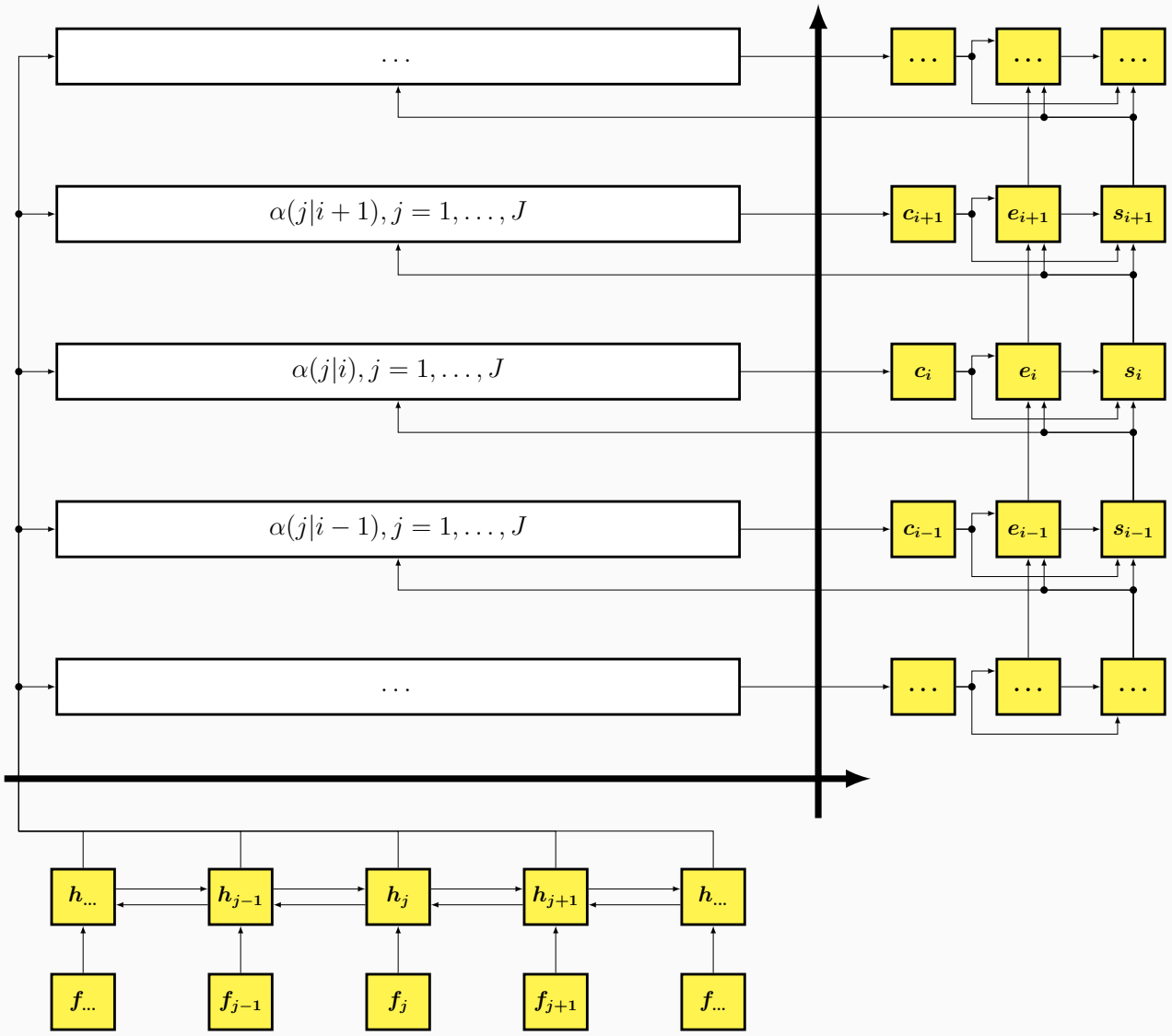


Figure: Architektur eines Attention NMT Systems

Transformer

Idee: Self-Attention für hohe Parallelisierbarkeit nutzen [Vaswani+ 17]:

- **Jedes Wort bestimmt den Einfluss aller anderen Wörter auf sich selbst**
- **Verschiedene Indizes sind unabhängig**
- $\alpha(j|j') = A_j(h_{j'}, h_1^J)$

$$\hat{h}_{j'} = \sum_{j=1}^{J} \alpha(j|j') \cdot h_j$$

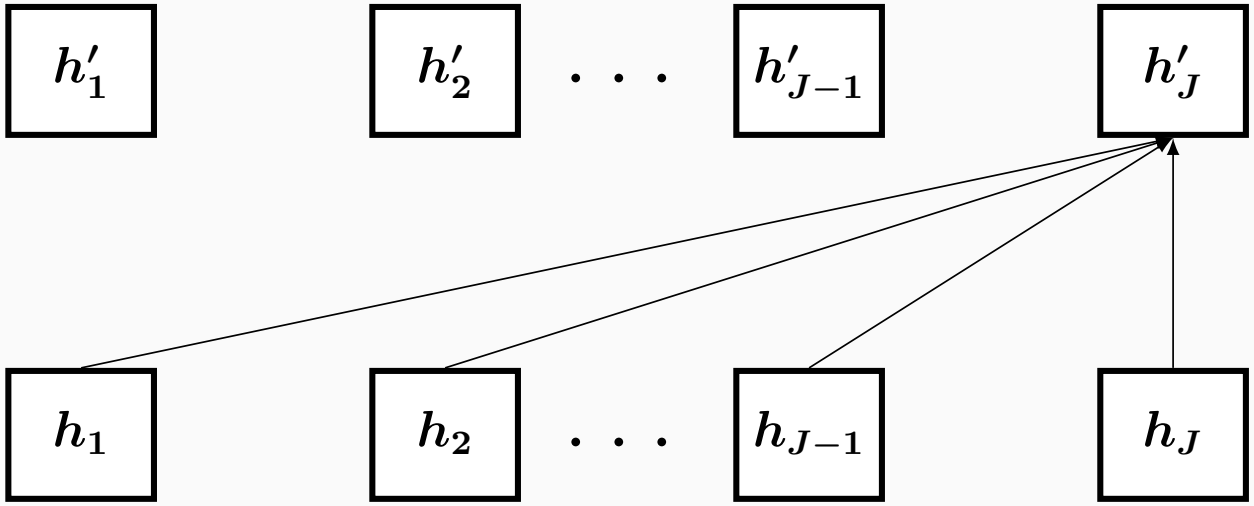


Figure: Self-Attention

Online: http://jalammar.github.io/illustrated-transformer

Transformer

Positional Encoding:

- **Sine/Cosine Kodierung des Wortposition**

6 Encoder Layer:

- **Multi-Head Attention**
- **Feed Forward Layer**

6 Decoding Layer:

- **Masked Multi-Head Attention (über den Ziel-Satz)**
- **Multi-Head Attention (über den letzten Encoder Layer)**
- **Feed Forward Layer**

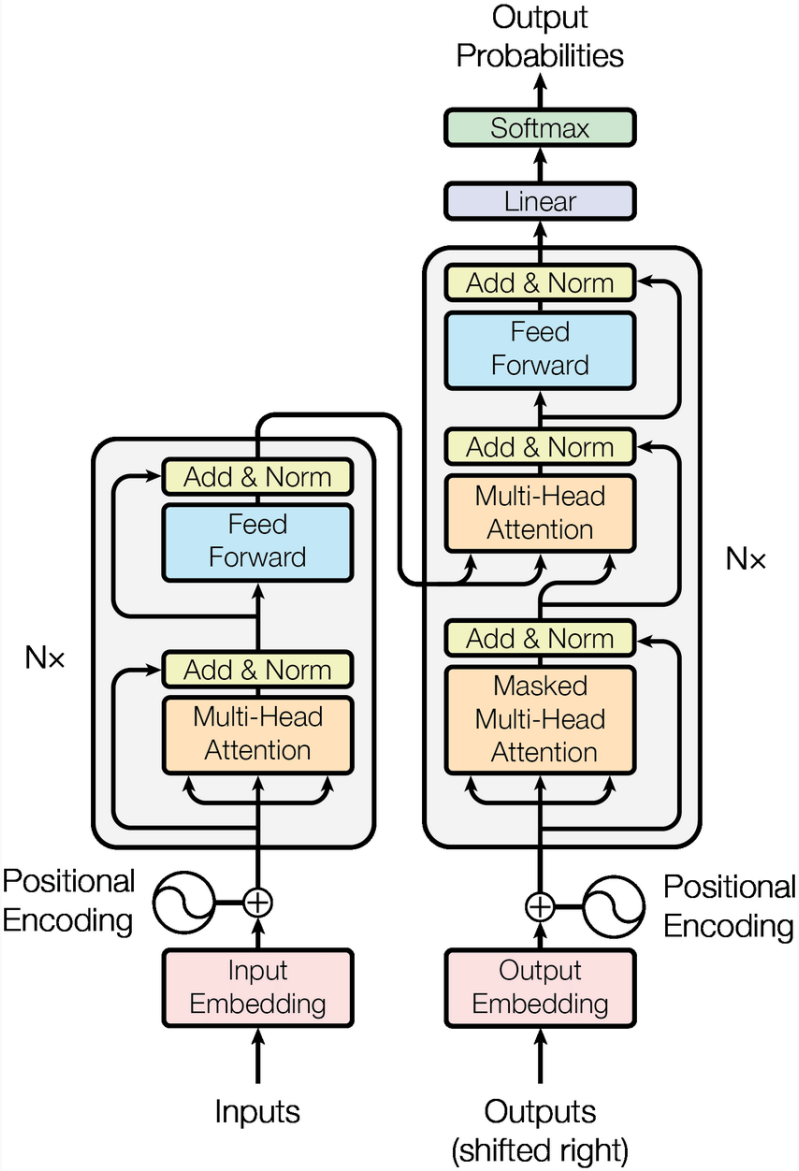


Figure: Architektur eines Transformer NMT Systems

2D LSTM

- 1D LSTM verarbeitet 1D Daten
- Viele Daten haben mehr als 1 Dimension, z.B. Bilder
- LSTMs können entsprechend erweitert werden [Graves+ 07]

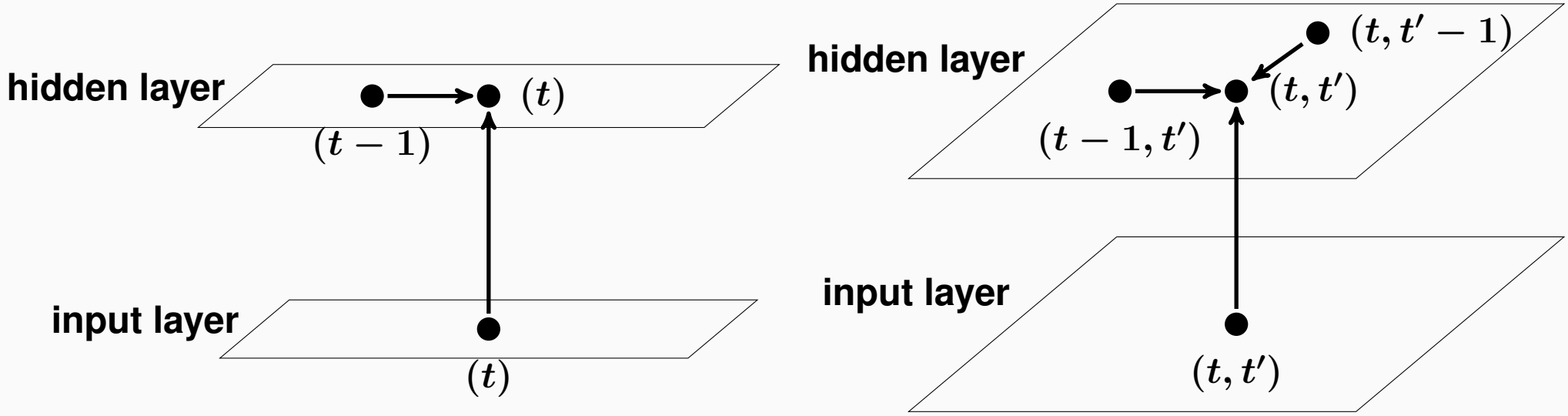


Figure: Erweiterung eines LSTMs für 2D Daten

2D LSTM

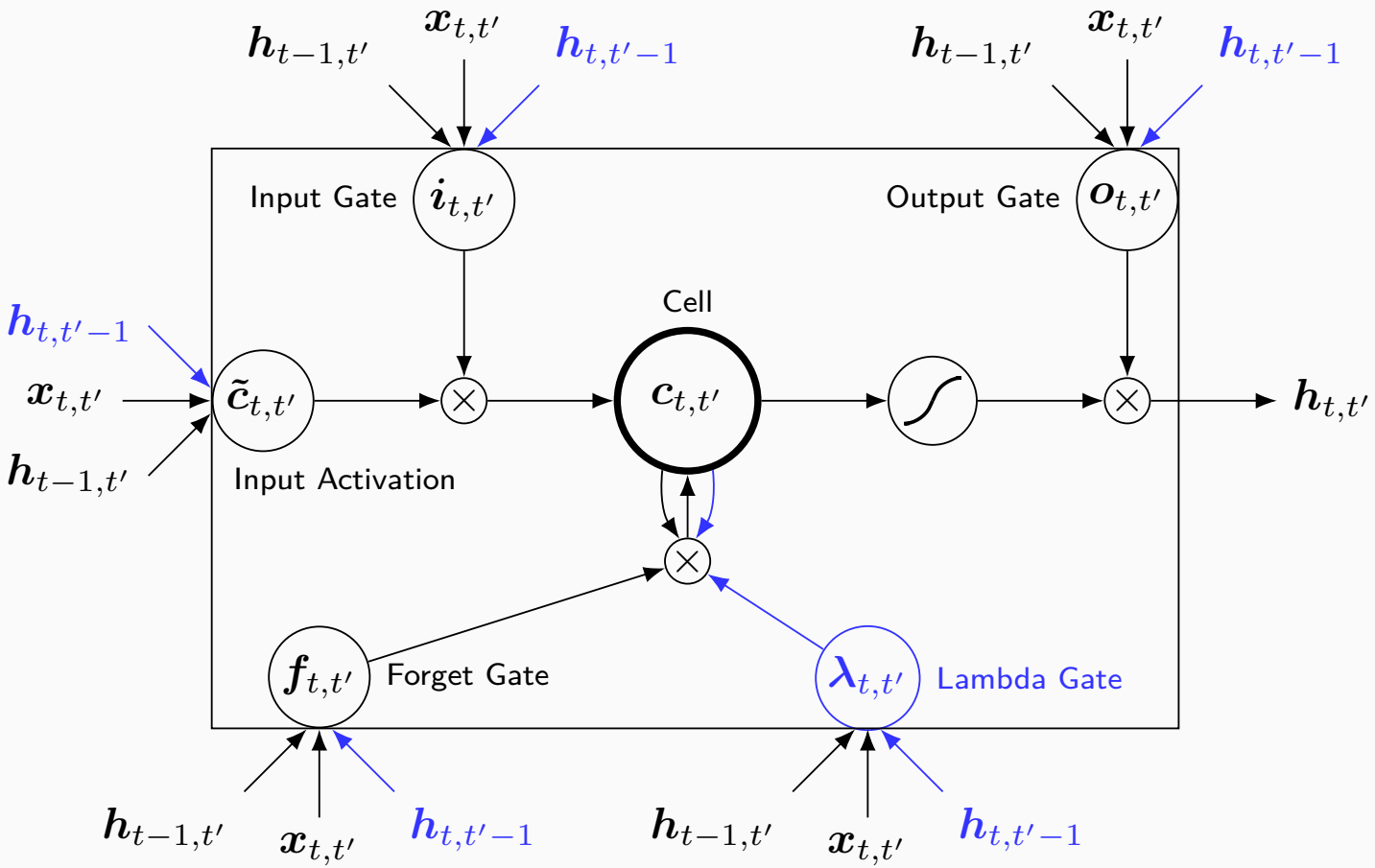
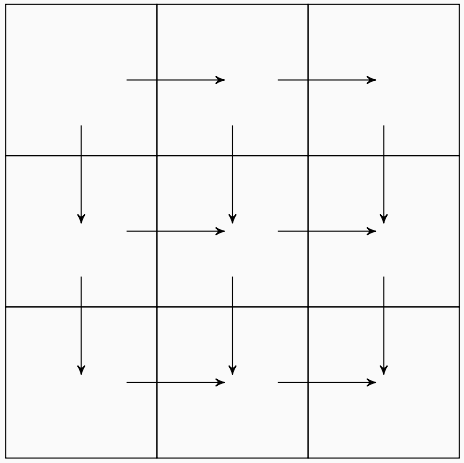


Figure: 2DLSTM cell

Parallele Berechnung

- **1D LSTM für n Daten: $\mathcal{O}(n)$ Operationen**



(a) Dependencies

1	2	3
4	5	6
7	8	9

(b) Ordered processing

1	2	3
2	3	4
3	4	5

(c) Parallel processing

- **2DLSTM kann innerhalb von $\mathcal{O}(n+m)$ Operationen berechnet werden [Voigtlaender+ 16]**

2D Sequence to Sequence (2D seq2seq)

Neue Architektur [Bahar+ 18]:

- **Kein expliziter encoder**
- **Kein expliziter decoder**
- **Komplexität:** $\mathcal{O}(I + J)$

2D LSTM:

$$a_{0,0} = 0$$

$$a_{j,i} = 2DLSTM([f_j, e_{i-1}], a_{j-1,i}, a_{j,i-1})$$

Decoding:

$$\hat{e}_i \leftarrow \text{softmax}(a_{J,i})$$

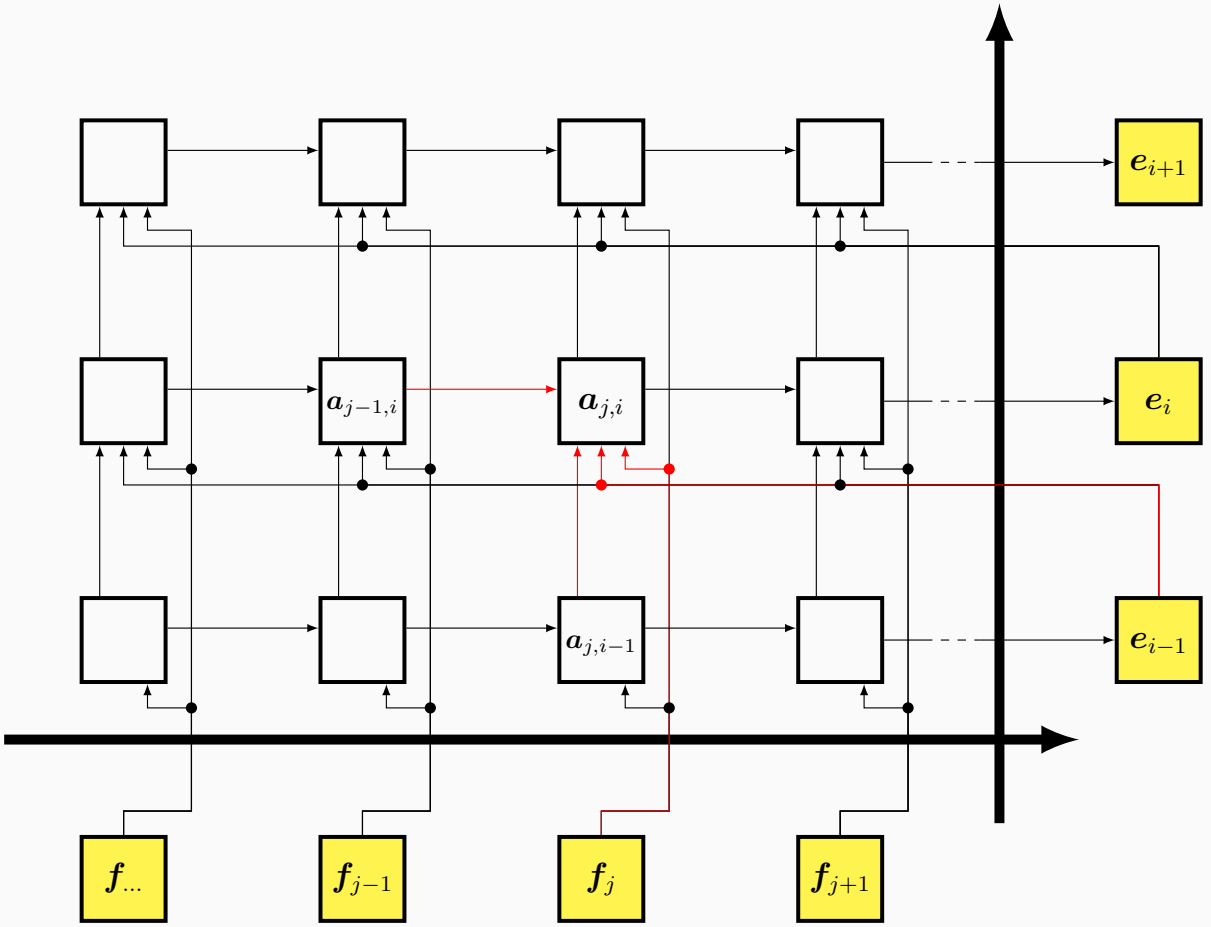


Figure: 2D seq2seq Architektur

2D seq2seq - Ergebnisse

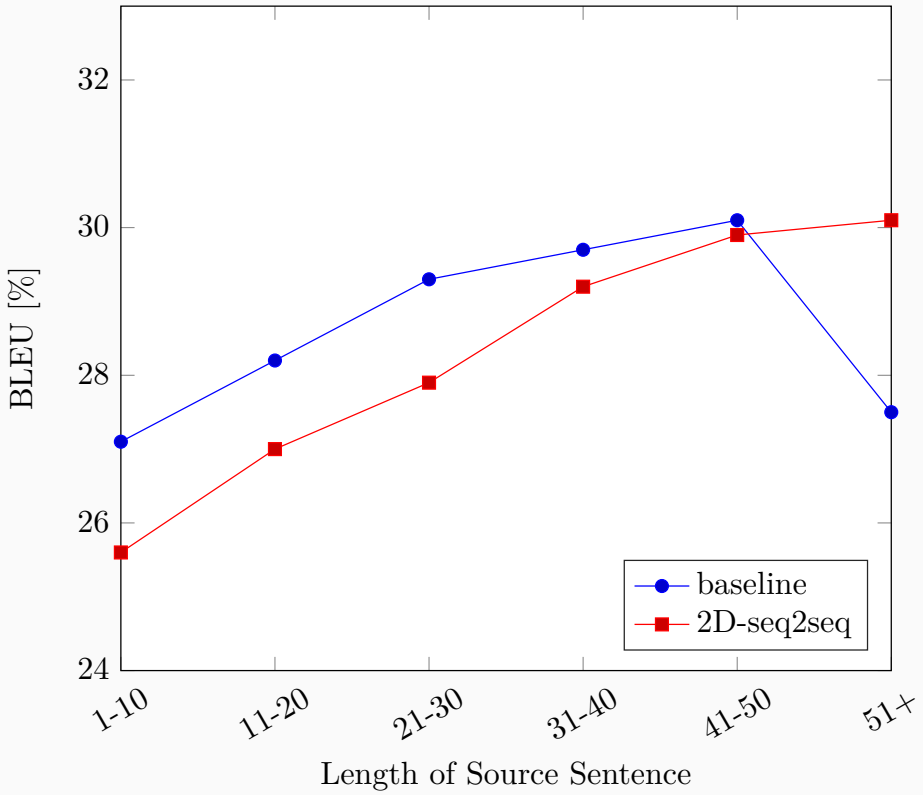
Table: WMT 2016/17, Encoder/Attention/Decoder/2DLSTM Größe von 1000.

	Deutsch→Englisch				Englisch→Deutsch			
	BLEU [%]		TER [%]		BLEU [%]		TER [%]	
	2016	2017	2016	2017	2016	2017	2016	2017
Baseline	33.1	29.0	47.5	51.9	27.4	22.9	53.9	60.2
2D seq2seq	33.7	29.3	46.9	51.9	28.9	23.2	52.6	59.5

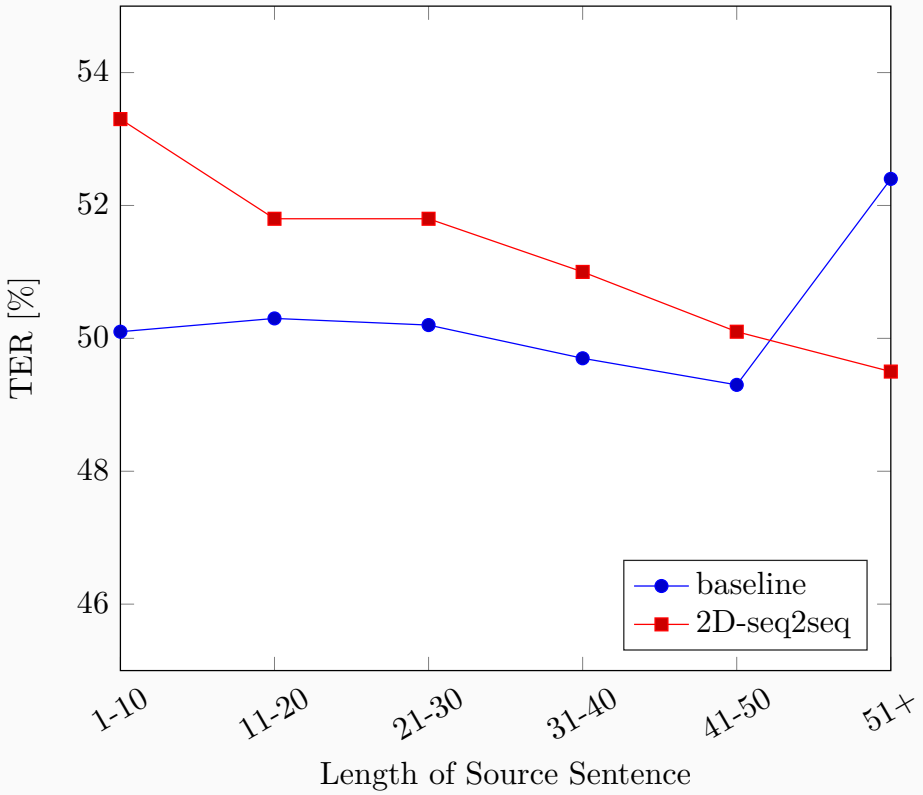
Table: Trainings- und Übersetzungs-Gewschwindigkeit.

	Training [Token/s]	Übersetzung [Token/s]
Baseline	2,944	48
2D seq2seq	791	0.7

2D seq2seq - Performanz bei langen Sätzen



(a) BLEU



(b) TER

Figure: WMT 2017 `newstest2015`, `newstest2016` and `newstest2017` Deutsch→Englisch

- **Gruppengröße: 1455, 3081, 2133, 990, 344 und 169 Satzpaare**
- **2D seq2seq kann längere Sätze besser übersetzen**

Data Cleaning/Augmentation/Fairness

Data Cleaning:

- **Paracrawl Korpus: 5.000.000.000 Deutsch-Englisch Satzpaare**
- **Viele schlechte Paare**

Data Augmentation:

- **Nützlich bei kleinen Korpora**
- **Monolinguale Daten mit Model A übersetzen, um Model B zu trainieren**

Data Fairness:

- **Bias im Datensatz => Bias im Model (Geschlecht, Höflichkeit, etc.)**
- **Schaffung von Datensätzen ohne Bias**
- **Model steuerbar machen**

Document Level Translations

Problem:

- **Satzweise Übersetzung kann inkonsistent sein**
  - **Geschlecht**
  - **Technische Begriffe**
  - **Fehlender Kontext**

Mögliche Ansätze:

- **Attention über vorherigen Satz**
- **Zusätzliche Zusammenfassungen des Dokuments**

Sparsity

Idee:

- **Teile des Netzwerks werden entfernt, um Zeit/Platz zu sparen**

Verschiedene Typen:

- **Structured sparsity**
  - **Ganze Layer/Module**
  - **Einzelne Neuronen**
  - **Blöcke von Verbindungen**
- **Unstructured sparsity**
  - **Einzelne Verbindungen**

Vielen Dank für Ihre Aufmerksamkeit!

Christopher Brix

Christopher.Brix@rwth-aachen.de

www.christopher-brix.de

References