Neural Hidden Markov Model for Machine Translation

Weiyue Wang, Derui Zhu, Tamer Alkhouli, Zixuan Gan and Hermann Ney

`{surname}@i6.informatik.rwth-aachen.de`

July 17th, 2018

Human Language Technology and Pattern Recognition
Lehrstuhl für Informatik 6
Computer Science Department
RWTH Aachen University, Germany

Introduction

- **Attention-based neural translation models**
  - **attend to specific positions on the source side to generate translation**
  - **improvements over pure encoder-decoder sequence-to-sequence approach**

- **Neural HMM has been successfully applied on top of SMT systems [Wang & Alkhouli^{+} 17]**
- **This work explores its application in standalone decoding**
  - **end-to-end, only with neural networks $\rightarrow$ NMT**
  - **LSTM structures outperform FFNN variants in [Wang & Alkhouli^{+} 17]**

Neural Hidden Markov Model

- **Translation**
  - source sentence $f_1^J = f_1 ... f_j ... f_J$
  - target sentence $e_1^I = e_1 ... e_i ... e_I$
  - alignment $i \rightarrow j = b_i$
- **Model translation using an alignment model and a lexicon model:**

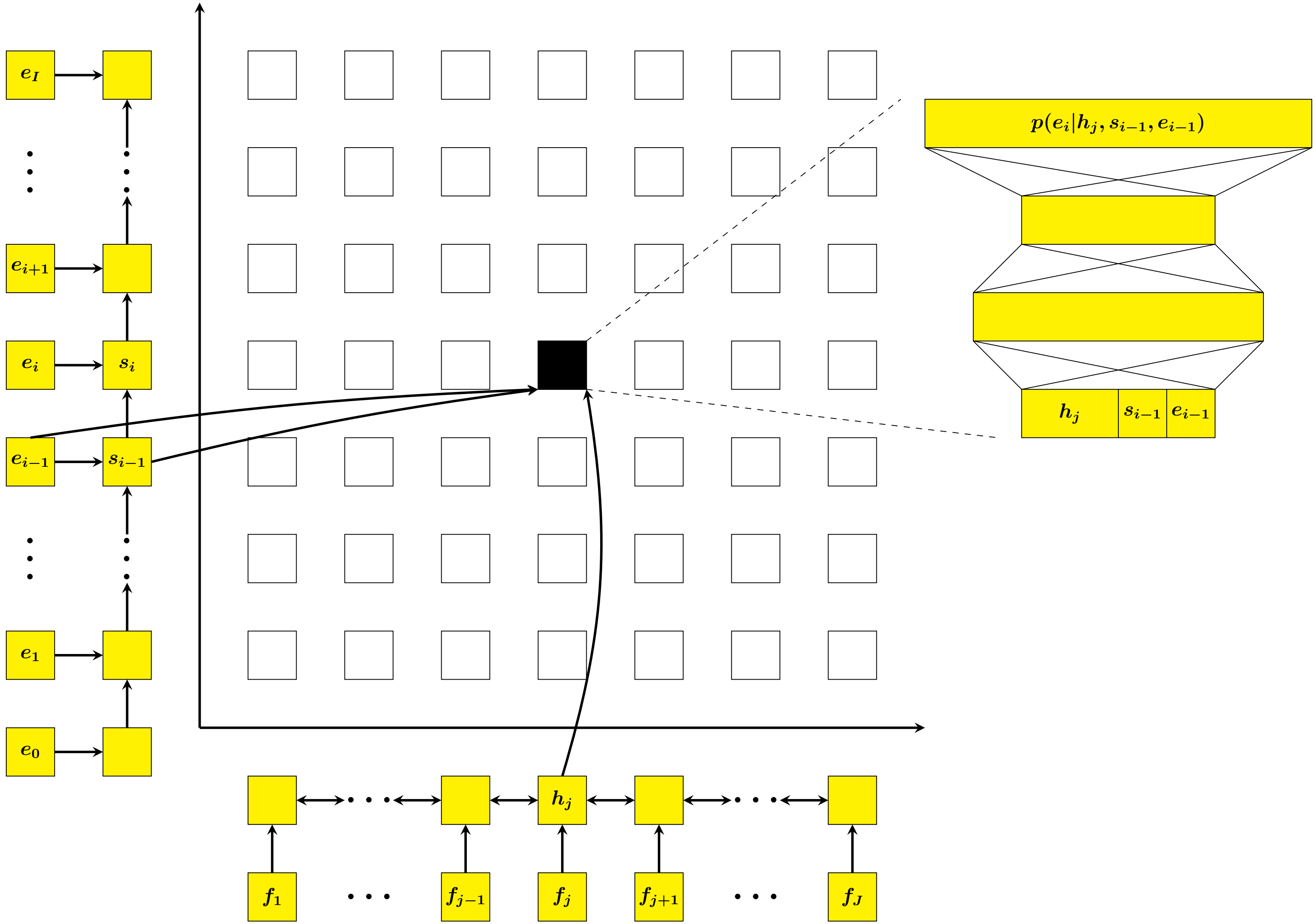
$$p(e_1^I|f_1^J) = \sum_{b_1^I} p(e_1^I, b_1^I|f_1^J) \tag{1}$$

$$:= \sum_{b_1^I} \prod_{i=1}^{I} \underbrace{p(e_i|b_1^i, e_0^{i-1}, f_1^J)}_{\text{lexicon model}} \cdot \underbrace{p(b_i|b_1^{i-1}, e_0^{i-1}, f_1^J)}_{\text{alignment model}} \tag{2}$$

with $p(b_i|b_1^{i-1}, e_0^{i-1}, f_1^J) := p(\Delta_i|b_1^{i-1}, e_0^{i-1}, f_1^J)$

- predicts the jump $\Delta_i = b_i - b_{i-1}$

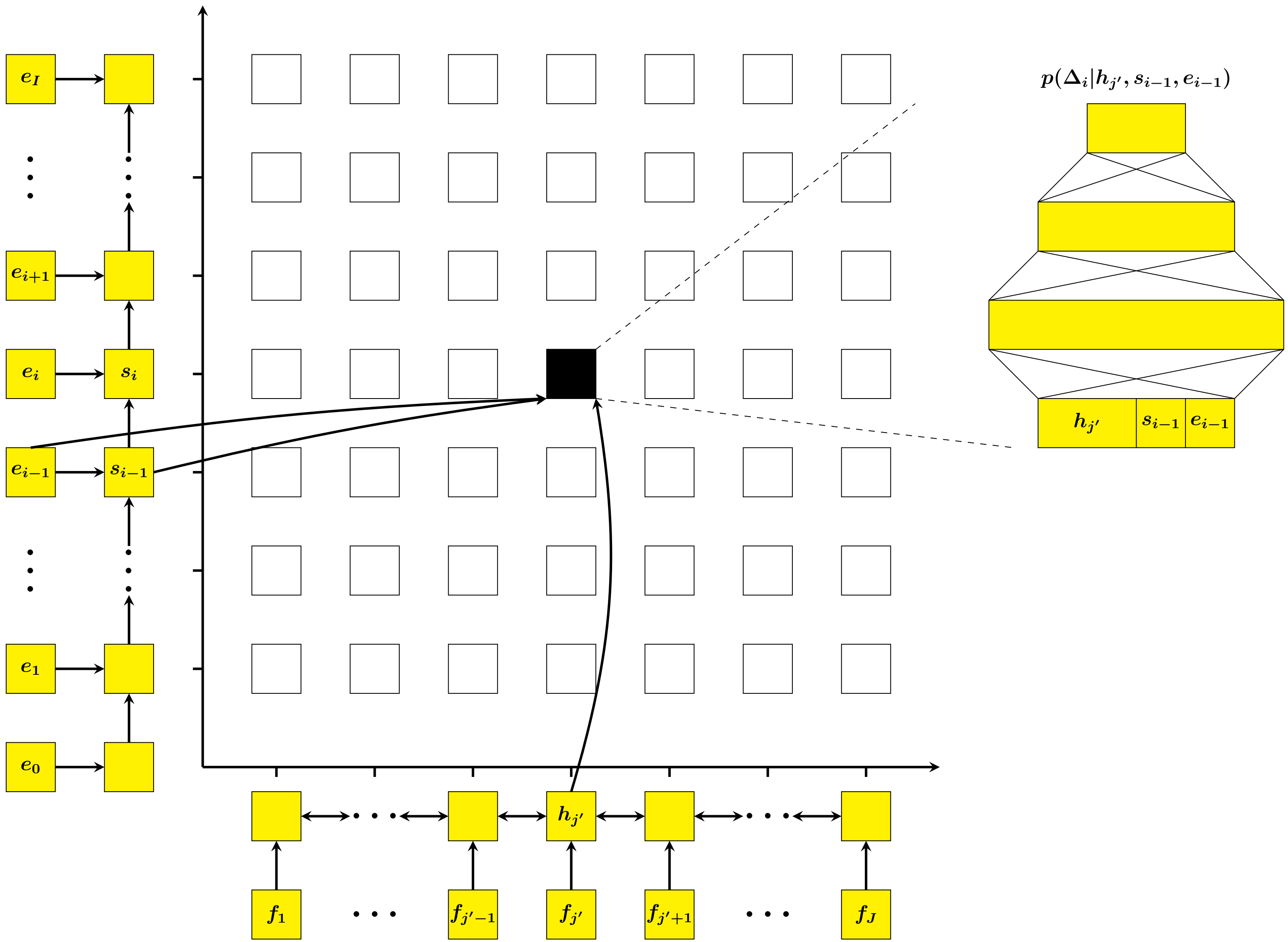
Neural Hidden Markov Model



- **Neural network based lexicon model**

RWTH AACHEN UNIVERSITY

Neural Hidden Markov Model



- **Neural network based alignment model ($j' = b_{i-1}$)**

Training

- **Training criterion for sentence pairs $(F_r, E_r), r = 1, ..., R$:**

$$\operatorname*{argmax}_{\theta} \left\{ \sum_{r} \log p_\theta(E_r|F_r) \right\} \tag{3}$$

- **Derivative for a single sentence pair $(F, E) = (f_1^J, e_1^I)$:**

$$\frac{\partial}{\partial\theta} \log p_\theta(E|F) = \sum_{j',j} \sum_{i} \underbrace{p_i(j', j|f_1^J, e_1^I; \theta)}_{\text{HMM posterior weights}} \cdot \frac{\partial}{\partial\theta} \log p(j, e_i|j', e_0^{i-1}, f_1^J; \theta) \tag{4}$$

- **Entire training procedure: backpropagation in an EM framework**
  1. **compute:**
     - **the HMM posterior weights**
     - **the local gradients (backpropagation)**
  2. **update neural network weights**

Decoding

- **Search over all possible target strings**

$$\max_{e_1^I} p(e_1^I|f_1^J) = \max_{e_1^I} \left\{ \sum_{b_1^I} \prod_i p(b_i, e_i|b_{i-1}, e_0^{i-1}, f_1^J) \right\}$$

- **Extending partial hypothesis from e_0^{i-1} to e_0^i**

$$Q(i, j; e_0^i) = \sum_{j'} \left[p(j, e_i|j', e_0^{i-1}, f_1^J) \cdot Q(i-1, j'; e_0^{i-1}) \right] \tag{5}$$

- **Pruning:**

$$\begin{aligned} Q(i; e_0^i) &= \sum_j Q(i, j; e_0^i) \\ \operatorname*{argmax}_{e_i} Q(i; e_0^i) &\leftarrow \textbf{select several candidates} \end{aligned} \tag{6}$$

Decoding

- **No explicit coverage constraints**
  - **one-to-many alignment cases and unaligned source words**
- **Search space in decoding**
  - **neural HMM: consists of both alignment and translation decisions**
  - **attention model: consists only of translation decisions**
- **Decoding complexity (J = source sentence length, I = target sentence length)**
  - **neural HMM: $\mathcal{O}(J^2 \cdot I)$**
  - **attention model: $\mathcal{O}(J \cdot I)$**
  - **in practice, neural HMM 3 times slower than attention model**

Experimental Setup

- WMT 2017 German↔English and Chinese→English translation tasks
- Quality measured with case sensitive BLEU and TER on `newstests2017`
- *Moses* tokenizer and truecasing scripts [Koehn & Hoang+ 07]
- *Jieba*[1] segmenter for Chinese data
- $20K$ byte pair encoding (BPE) operations [Sennrich & Haddow+ 16]
  - joint for German↔English and separate for Chinese→English
- Attention-based system are trained with *Sockeye* [Hieber & Domhan+ 17]
  - encoder and decoder embedding layer size 620
  - a bidirectional encoder layer with 1000 LSTMs with peephole connections
  - *Adam* [Kingma & Ba 15] as optimizer with a learning rate of 0.001
  - batch size 50, 30% dropout
  - beam search with beam size 12
  - model weights averaging

[1] https://github.com/fxsjy/jieba

Experimental Setup

- Neural hidden markov model implemented in *TensorFlow* [Abadi & Agarwal+ 16]
  - encoder and decoder embedding layer size 350
  - projection layer size 800 (400+200+200)
  - three hidden layers of sizes 1000, 1000 and 500 respectively
  - normal softmax layer
    - lexicon model: large output layer with roughly 25K nodes
    - alignment model: small output layer with 201 nodes
  - *Adam* as optimizer with a learning rate of 0.001
  - batch size 20, 30% dropout
  - beam search with beam size 12
  - model weights averaging

Experimental Results

WMT 2017	# free parameters	German→English BLEU[%]	German→English TER[%]	English→German BLEU[%]	English→German TER[%]	Chinese→English BLEU[%]	Chinese→English TER[%]
FFNN-based neural HMM	33M	28.3	51.4	23.4	58.8	19.3	64.8
LSTM-based neural HMM	52M	29.6	50.5	24.6	57.0	20.2	63.7
Attention-based neural network	77M	29.5	50.8	24.7	57.4	20.2	63.8

- **FFNN-based neural HMM: [Wang & Alkhouli$^+$ 17] applied in decoding**
- **LSTM-based neural HMM: this work**
- **Attention-based neural network: [Bahdanau & Cho$^+$ 15]**
- **All models trained without synthetic data**
- **Single model used for decoding**

- **LSTM models improve FFNN-based system by up to 1.3% BLEU and 1.8% TER**
- **Comparable performance with attention-based system**

Summary

- **Apply NNs to conventional HMM for MT**
- **End-to-end with a stand-alone decoder**
- **Comparable performance with the standard attention-based system**
  - **significantly outperforms the feed-forward variant**

- **Future work**
  - **Speed up training and decoding**
  - **Application in automatic post editing**
  - **Combination with attention or *transformer* [Vaswani & Shazeer^{+} 17] model**

Thank you for your attention

Weiyue Wang

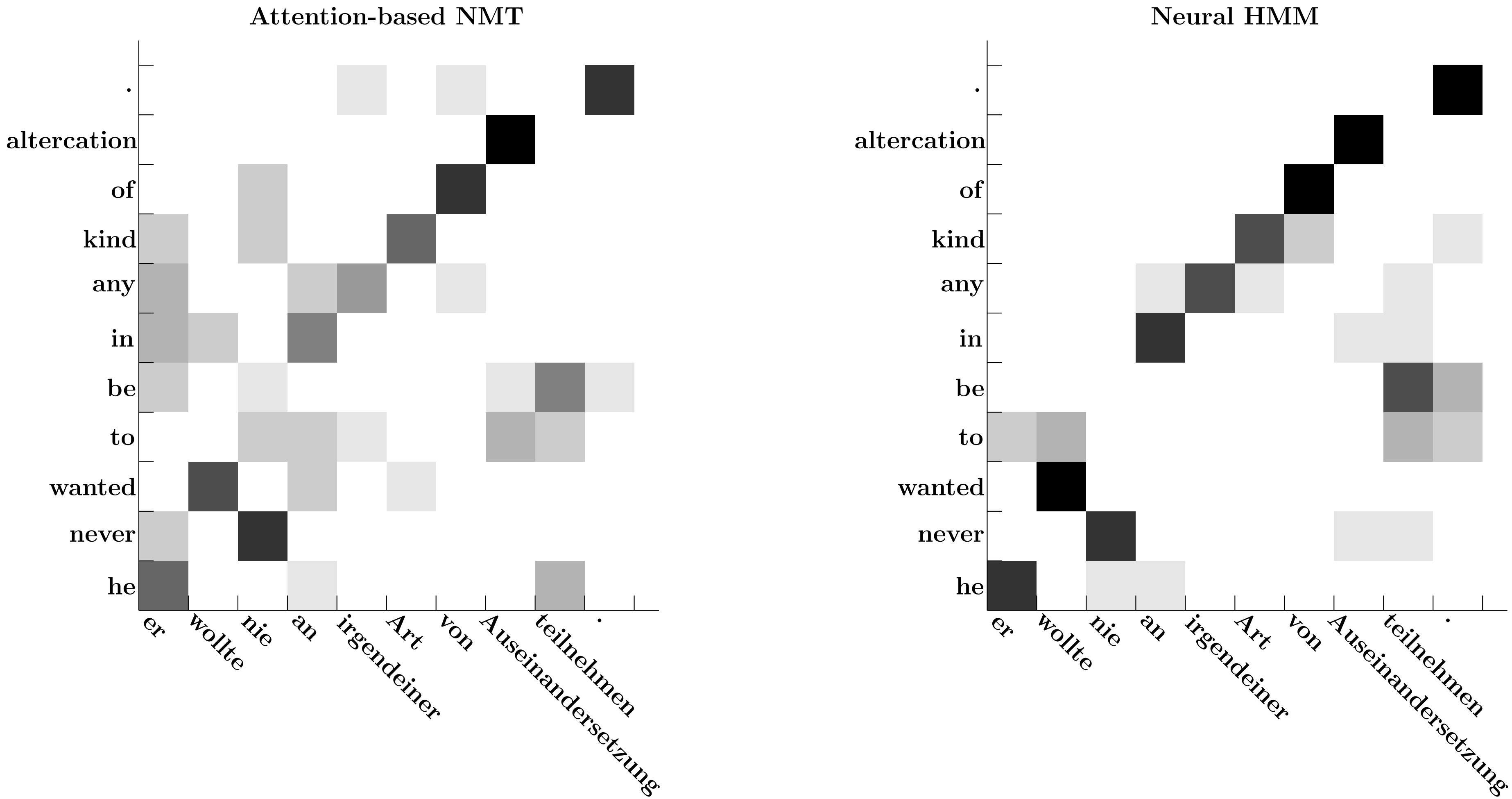
wwang@cs.rwth-aachen.de

http://www-i6.informatik.rwth-aachen.de/

Appendix: Motivation

- **Neural HMM compared to attention-based systems**
  - **recurrent encoder and decoder without attention component**
  - **replacing attention mechanism by a first-order HMM alignment model**
    - **attention levels: deterministic normalized similarity scores**
    - **HMM alignments: discrete random variables and must be marginalized**
  - **separating the alignment model from the lexicon model**
    - **more flexibility in modeling and training**
    - **avoids propagating errors from one model to another**
    - **implies an extended degree of interpretability and control over the model**

Appendix: Analysis



- Attention weight and alignment matrices visualized in heat map form
- Generated by attention NMT baseline and neural HMM

Appendix: Analysis

1	source	28-jähriger Koch in San Francisco Mall tot aufgefunden
	reference	28-Year-Old Chef Found Dead at San Francisco Mall
	attention NMT	28-year-old cook in San Francisco Mall found dead
	neural HMM	28-year-old cook found dead in San Francisco Mall
2	source	Frankie hat in GB bereits fast 30 Jahre Gewinner geritten , was toll ist .
	reference	Frankie 's been *riding winners* in the UK for the best part of 30 years which is great to see .
	attention NMT	Frankie has been a **winner** in the UK for almost 30 years , which is great .
	neural HMM	Frankie has *ridden winners* in the UK for almost 30 years , which is great .
3	source	Wer baut Braunschweigs günstige Wohnungen ?
	reference	Who is going to *build Braunschweig 's* low-cost housing ?
	attention NMT	Who **does Braunschweig build** cheap apartments ?
	neural HMM	Who *builds Braunschweig 's* cheap apartments ?

- **Sample translations from the WMT German→English `newstest2017` set**
  - **underline source words of interest**
  - **italicize *correct* translations**
  - **bold-face for incorrect translations**

References

[Abadi & Agarwal+ 16] M. Abadi, A. Agarwal, P. Barham et al.: TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. *CoRR*, Vol. abs/1603.04467, 2016. 9

[Bahdanau & Cho+ 15] D. Bahdanau, K. Cho, Y. Bengio: Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceedings of the 3rd International Conference on Learning Representations*, San Diego, CA, USA, May 2015. 10

[Hieber & Domhan+ 17] F. Hieber, T. Domhan, M. Denkowski, D. Vilar, A. Sokolov, A. Clifton, M. Post: Sockeye: A Toolkit for Neural Machine Translation. *ArXiv e-prints*, Vol. abs/1712.05690, December 2017. 8

[Kingma & Ba 15] D.P. Kingma, J.L. Ba: Adam: A method for stochastic optimization. In *Proceedings of the Third International Conference on Learning Representations*, San Diego, CA, USA, May 2015. 8

[Koehn & Hoang+ 07] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, E. Herbst: Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the Association for*

Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions, pp. 177–180, Prague, Czech Republic, June 2007. 8

[Sennrich & Haddow+ 16] R. Sennrich, B. Haddow, A. Birch: Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 1715–1725, Berlin, Germany, August 2016. 8

[Vaswani & Shazeer+ 17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin: Attention Is All You Need. In *31st Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, USA, December 2017. 11

[Wang & Alkhouli+ 17] W. Wang, T. Alkhouli, D. Zhu, H. Ney: Hybrid Neural Network Alignment and Lexicon Model in Direct HMM for Statistical Machine Translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pp. 125–131, Vancouver, Canada, August 2017. 2, 10