UNDERSTANDING AND IMPROVING MORPHOLOGICAL LEARNING

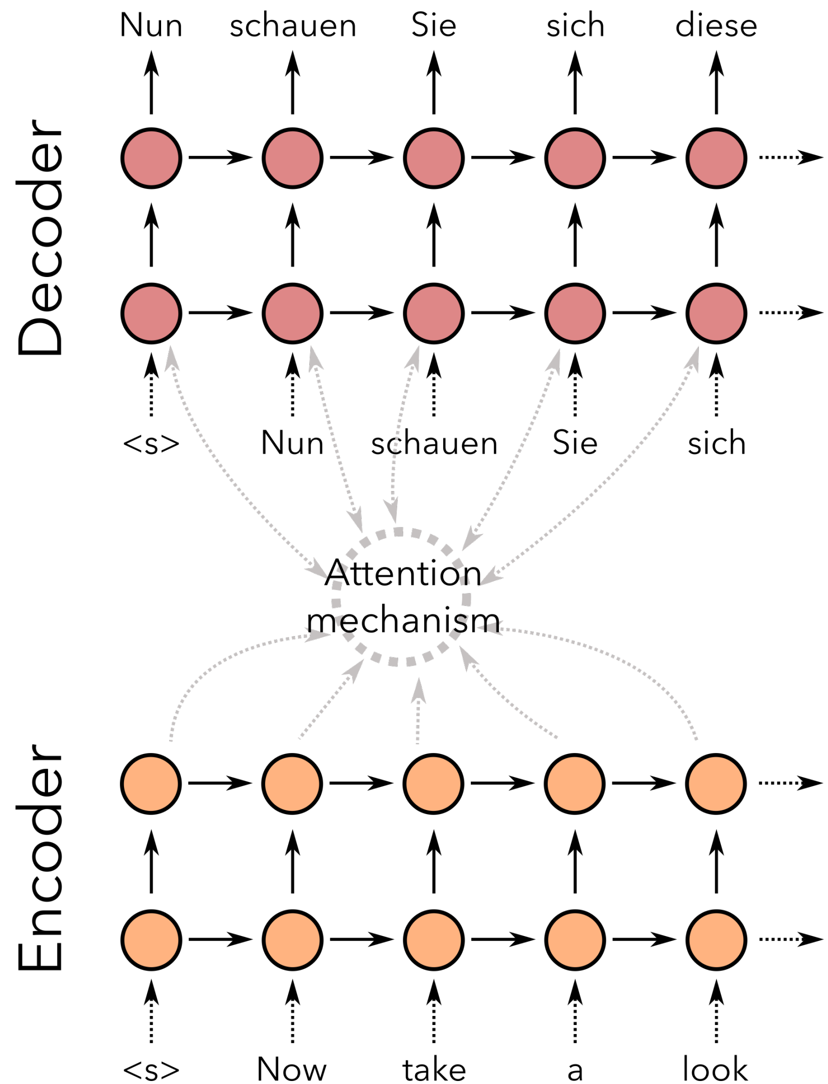
IN THE NEURAL MACHINE TRANSLATION DECODER

Fahim Dalvi Nadir Durrani Hassan Sajjad
Yonatan Belinkov Stephan Vogel
Qatar Computing Research Institute, HBKU
CSAIL, MIT

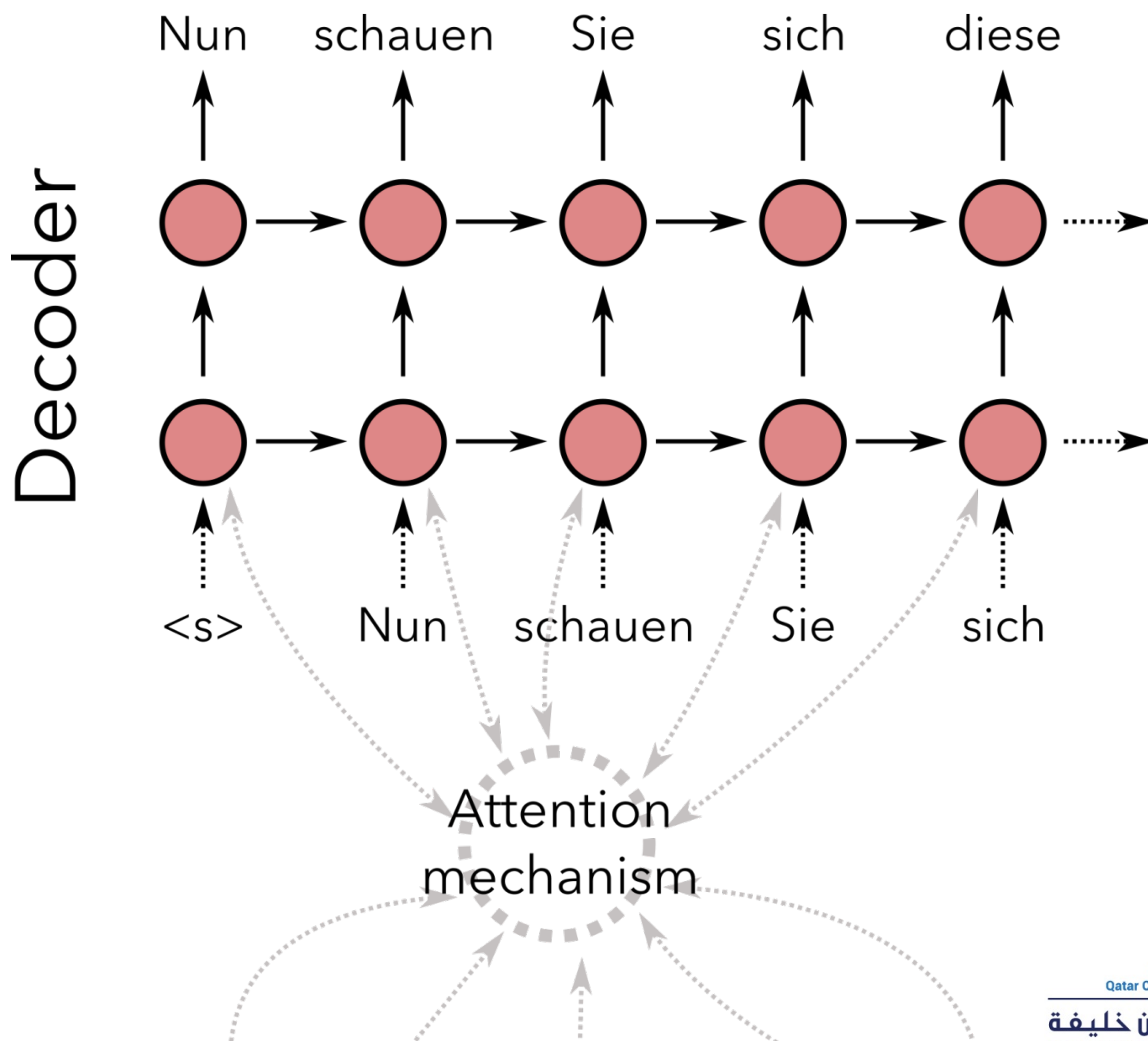
Goal

Improve overall **Neural Machine Translation** performance by providing the system with **explicit morphological knowledge**

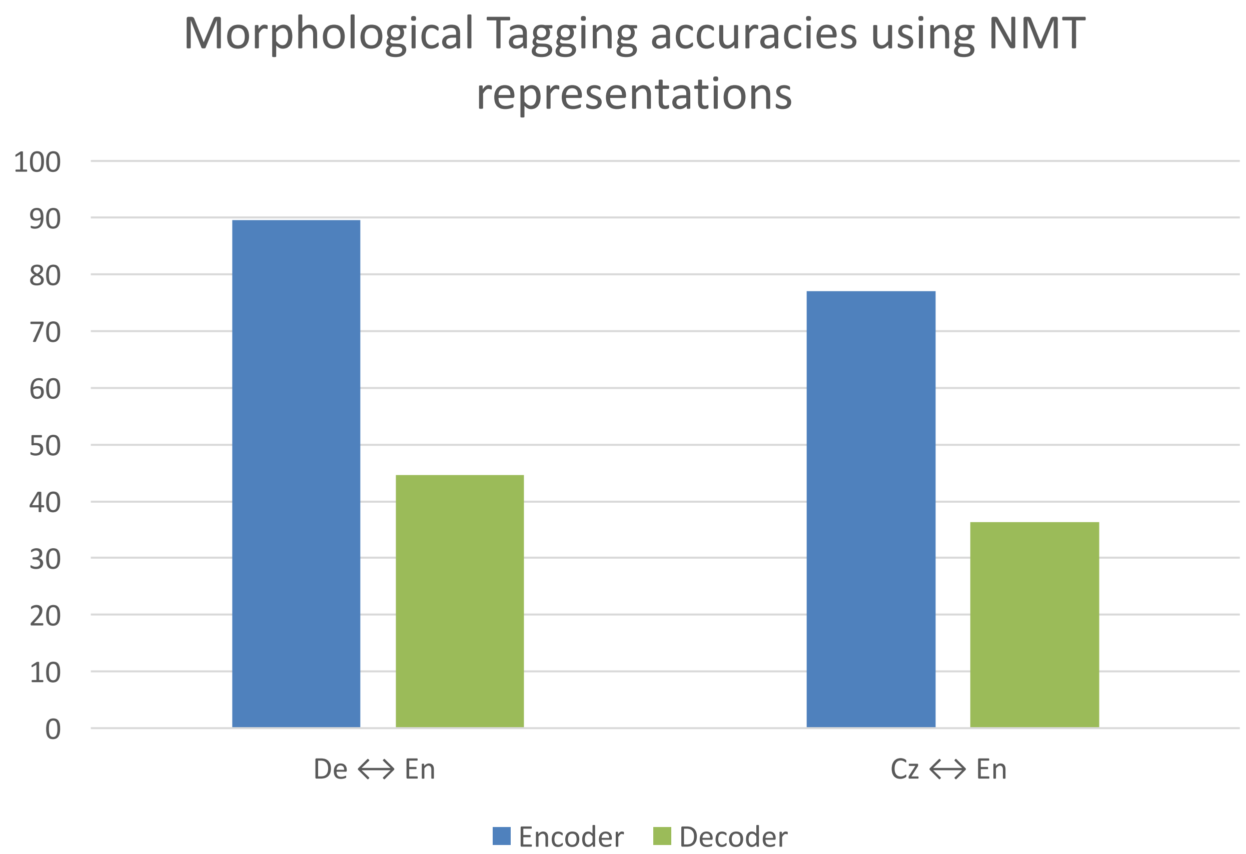
Recap: Neural Machine Translation



Recap: Neural Machine Translation



Motivation



Belinkov et. al. What do Neural Machine Translation Models Learn about Morphology? (ACL 2017)

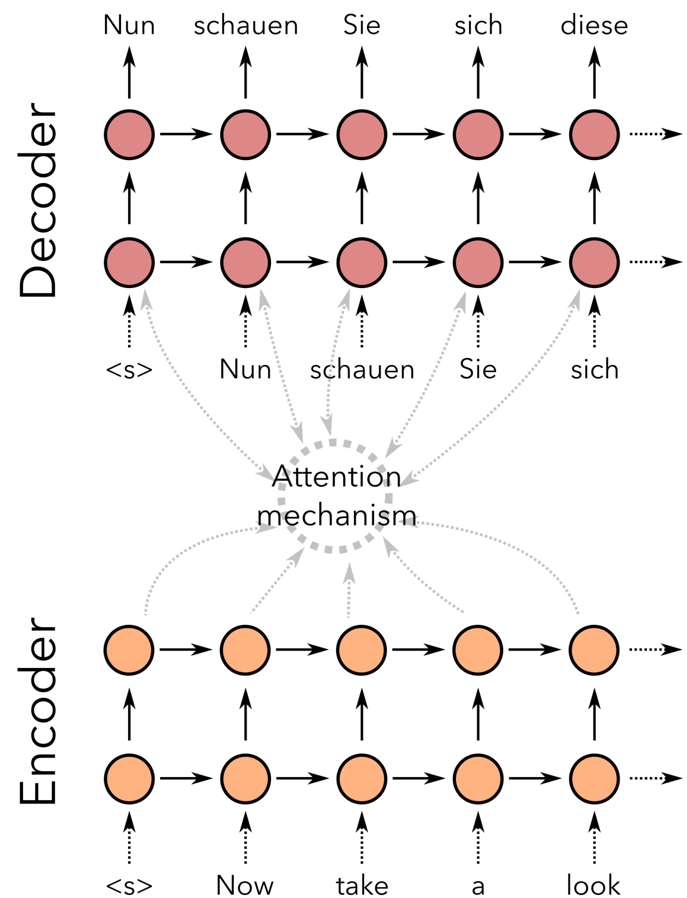
Our Work

I. Analyze why the decoder learns less morphological knowledge compared to the encoder
II. Inject morphological knowledge explicitly into the decoder to improve overall translation performance

Part I: NMT Decoder Analysis

Methodology

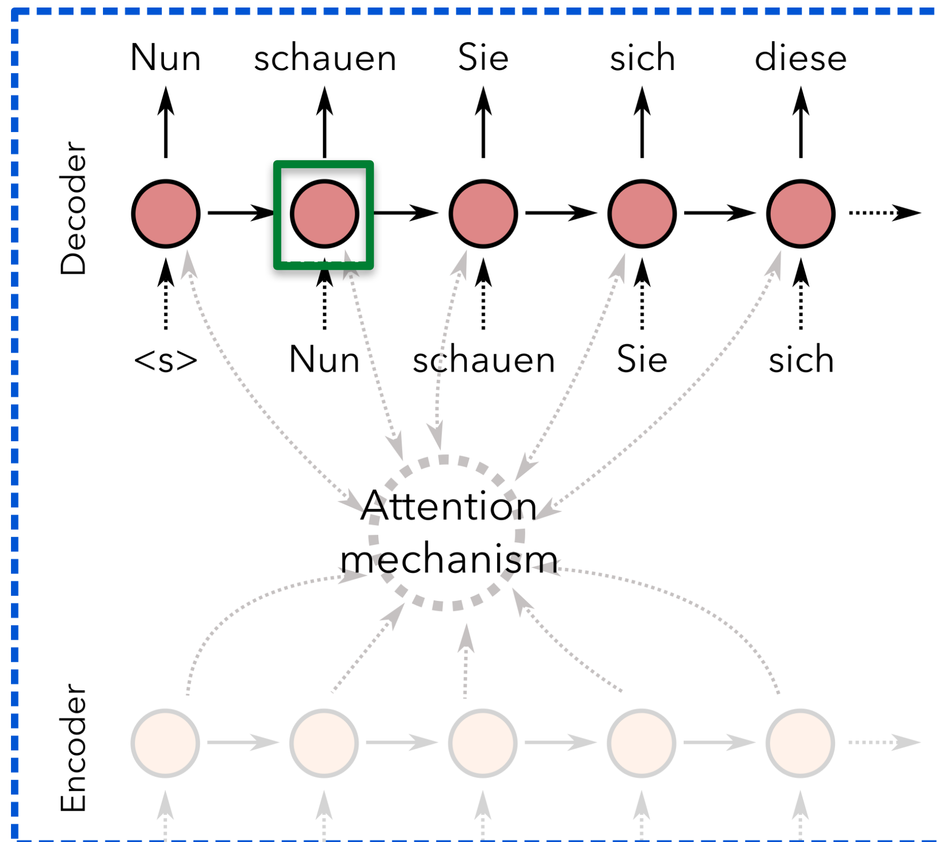
Step I: Train an NMT model



Methodology

Step I: Train an NMT model

Step II: Extract activations from desired layer



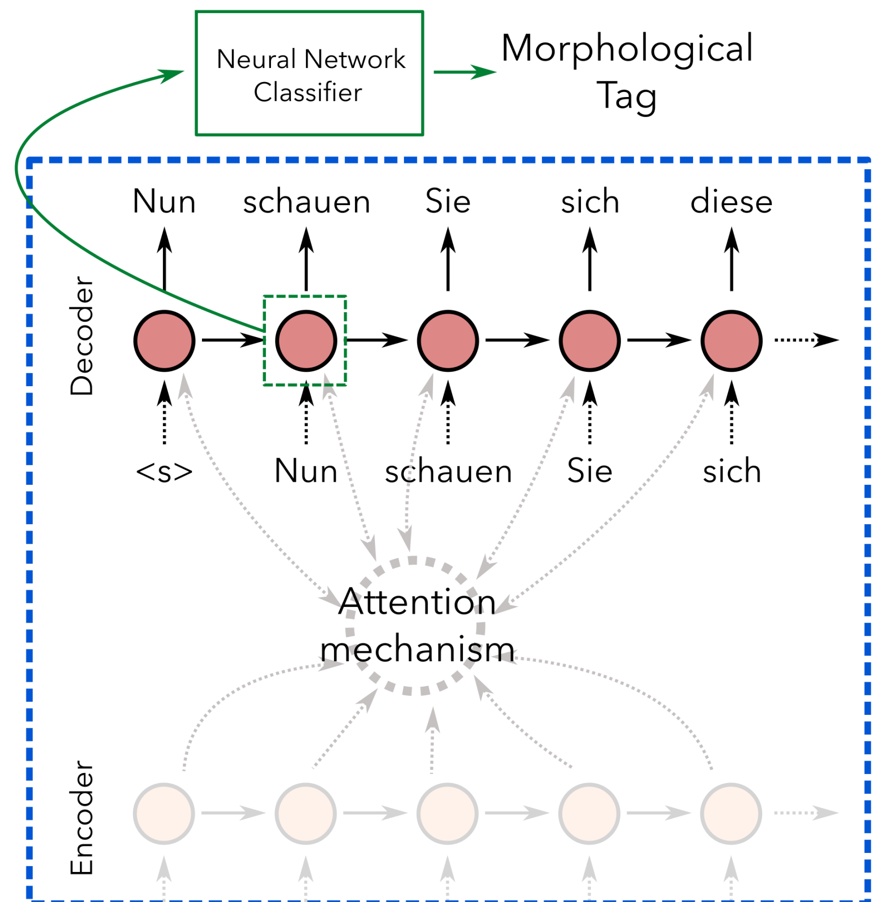
Pretrained Neural Machine Translation Model

Methodology

Step I: Train an NMT model

Step II: Extract activations from desired layer

Step III: Train an external classifier

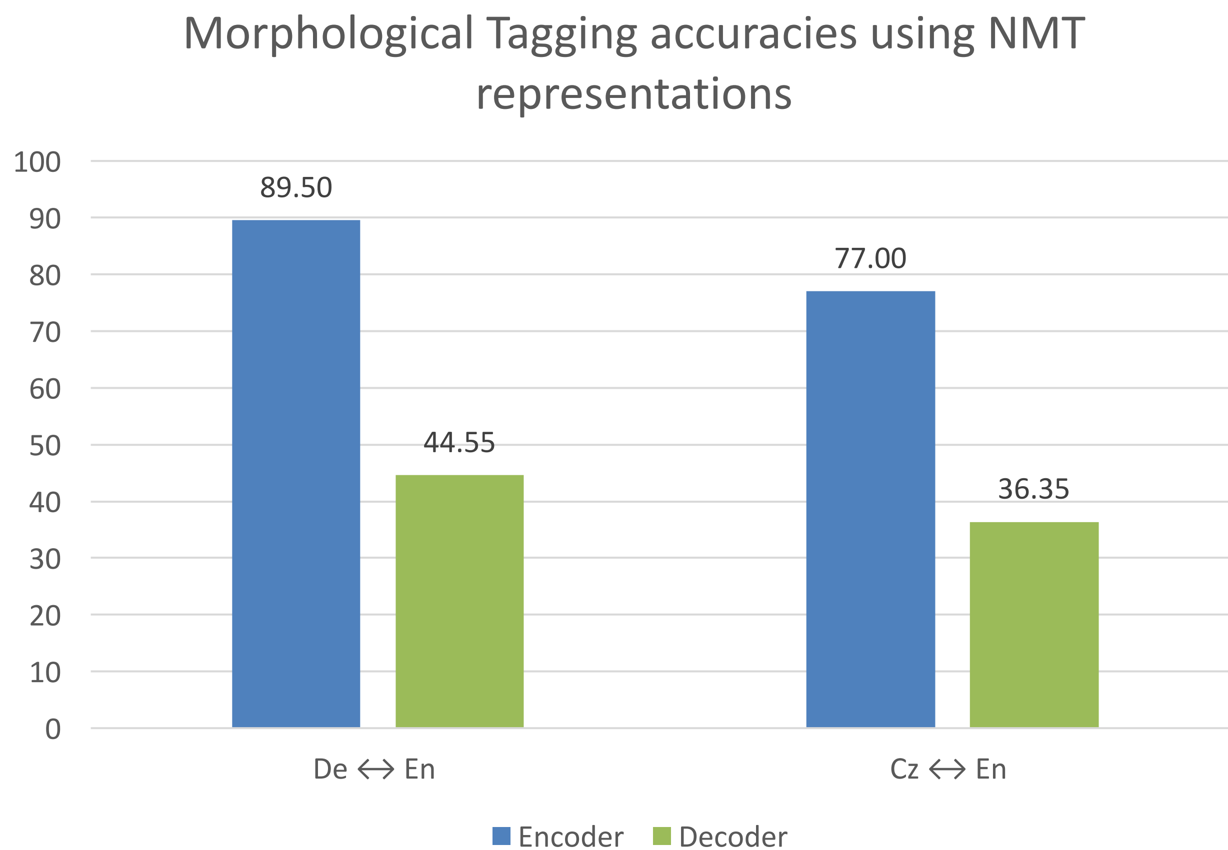


QCRI
معهد قطر لبحوث الحوسبة
Qatar Computing Research Institute
جامعة حمد بن خليفة
HAMAD BIN KHALIFA UNIVERSITY

Methodology

The **accuracy of the classifier** can be used as a proxy for how much **morphological knowledge** NMT has learned

Analysis: Encoder vs Decoder



All morphological tagging is done on German or Czech.
For encoder we use {De,Cz} → En systems
For decoder we use En → {De,Cz} systems

Analysis: Encoder vs Decoder

NMT decoders are able to produce good translations even in morphologically rich languages

Analysis: Encoder vs Decoder

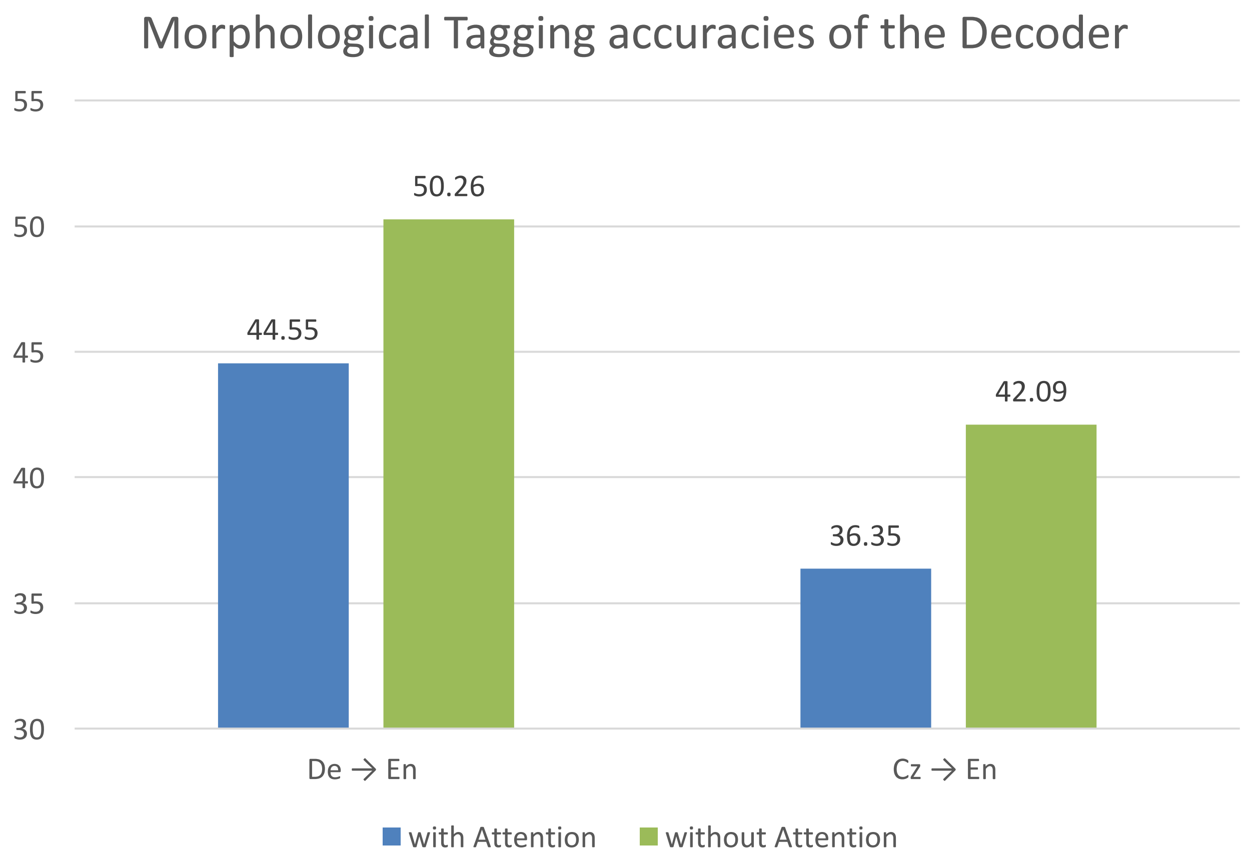
NMT decoders are able to produce good translations even in morphologically rich languages

Is there **another part** in the network that aids the decoder for **target side morphology**?

Does the decoder even **need to learn more** morphology than what is already learned?

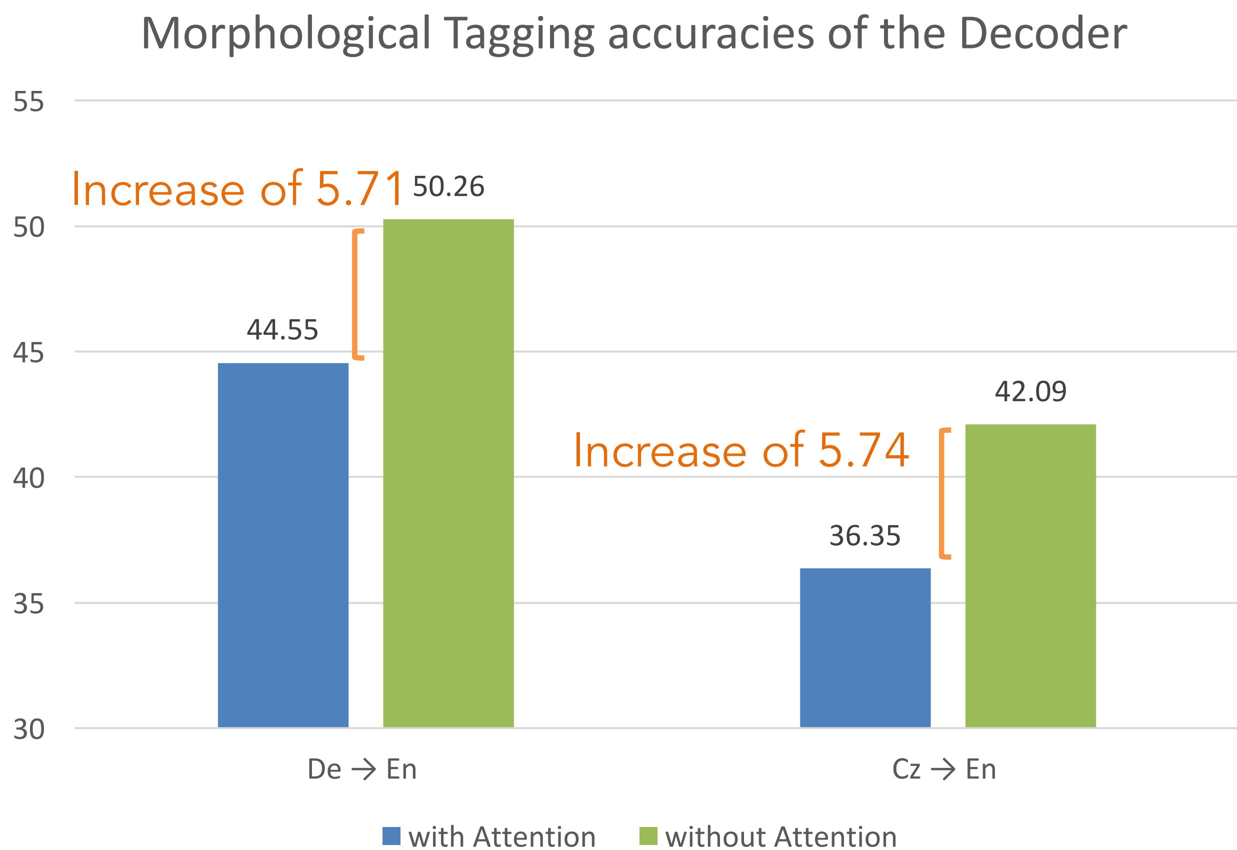
Analysis: Effect of Attention

Analysis: Effect of Attention



QCRI
معهد قطر لبحوث الحوسبة
Qatar Computing Research Institute
جامعة حمد بن خليفة
HAMAD BIN KHALIFA UNIVERSITY

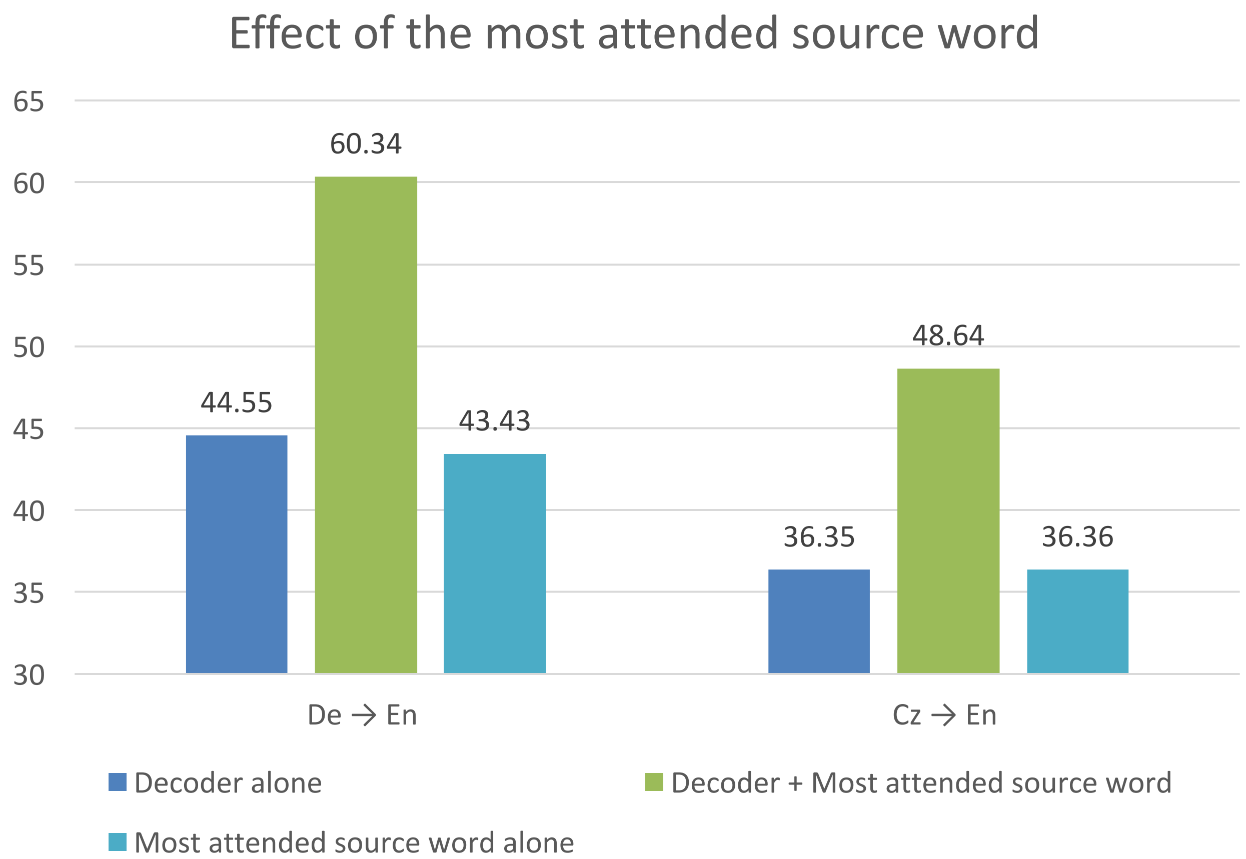
Analysis: Effect of Attention



Analysis: Effect of Attention

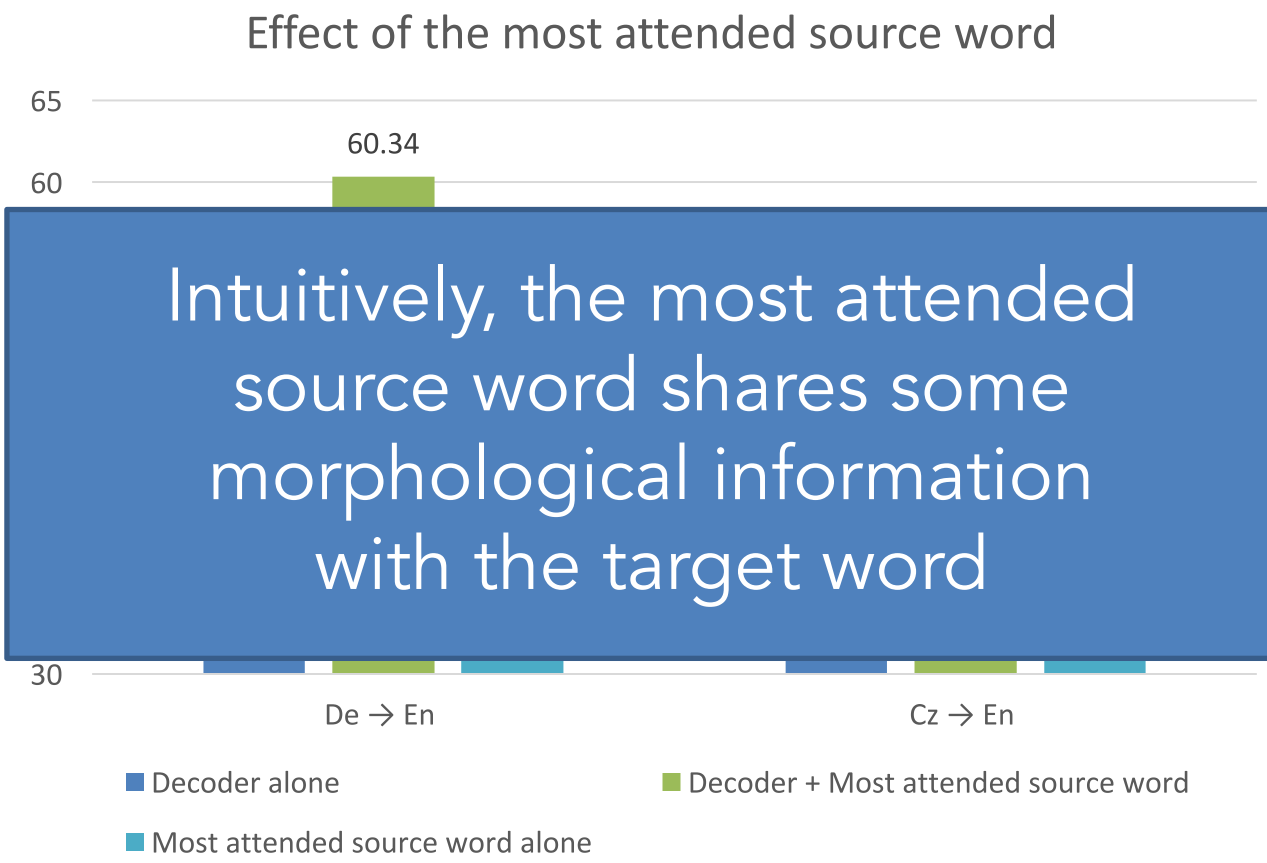
The decoder actually see's more then the **decoder state** – it also sees a **weighted representation** of the source words (through attention)

Analysis: Effect of Attention



QCRI
معهد قطر لبحوث الحوسبة
Qatar Computing Research Institute
جامعة حمد بن خليفة
HAMAD BIN KHALIFA UNIVERSITY

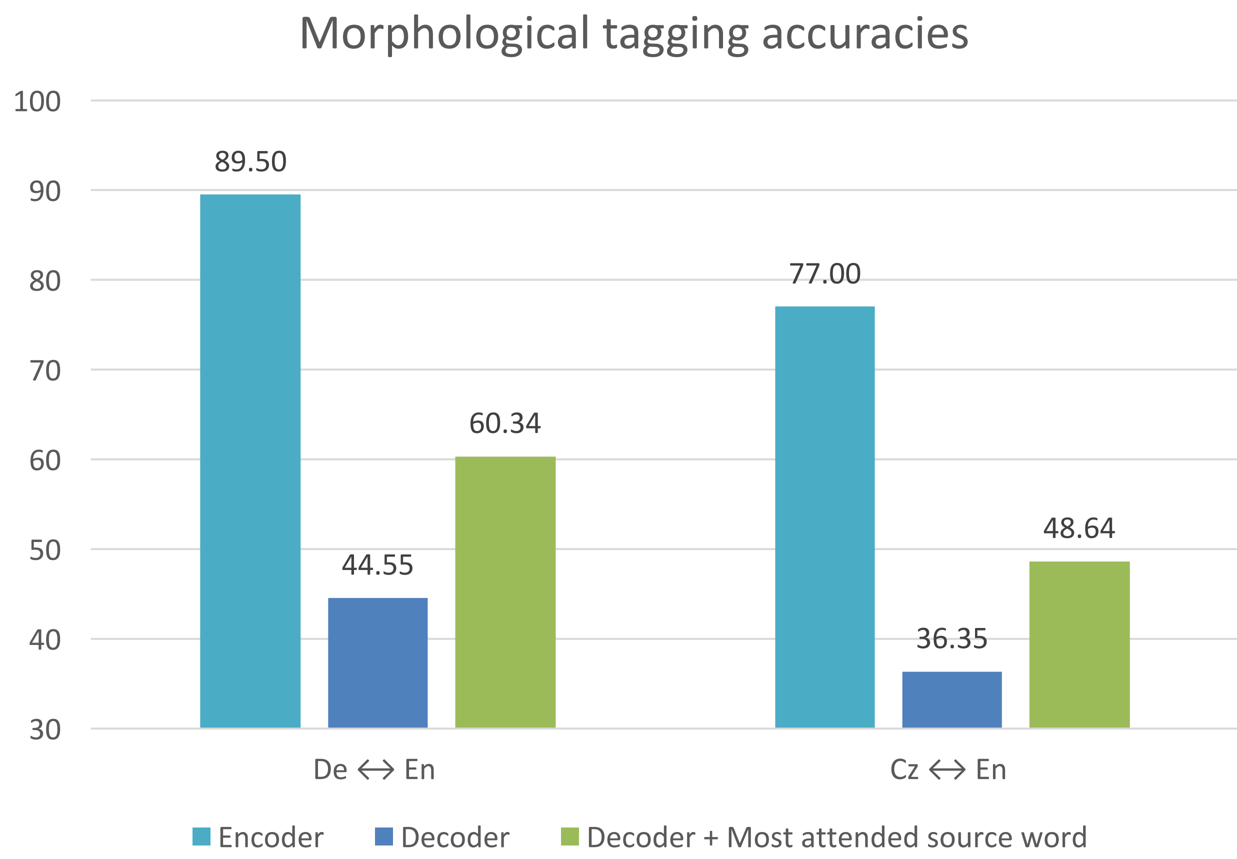
Analysis: Effect of Attention



Intuitively, the most attended source word shares some morphological information with the target word

QCRI
معهد قطر لبحوث الحوسبة
Qatar Computing Research Institute
جامعة حمد بن خليفة
HAMAD BIN KHALIFA UNIVERSITY

Analysis: Summary



All morphological tagging is done on German or Czech.
For encoder we use {De,Cz} → En systems
For decoder we use En → {De,Cz} systems

QCRI
معهد قطر لبحوث الحوسبة
Qatar Computing Research Institute
جامعة حمد بن خليفة
HAMAD BIN KHALIFA UNIVERSITY

Analysis: Conclusion

1) Overall, the decoder **does not perform as well** as the encoder on morphological tagging
2) The source-side representations and the attention mechanism **aid the decoder** even with regards to target morphology
3) Even with this aid, decoder accuracies **are not as high** as the encoder

Part II: Morphology Injection

Morphology Injection

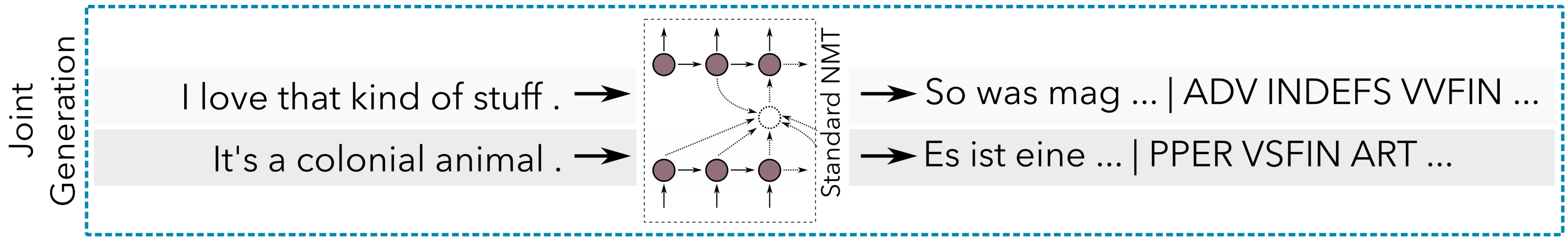
We have seen that there is room for improvement in the decoder's morphological tagging performance

Morphology Injection

We propose three techniques to explicitly inject morphology into the decoder:

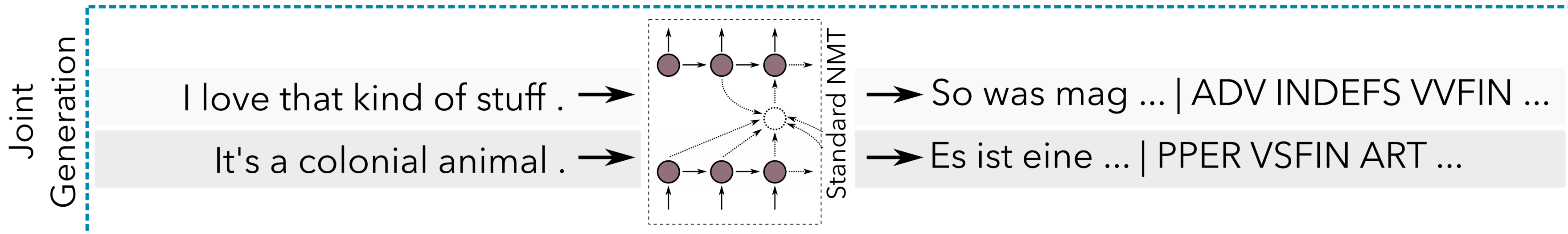
1) Joint generation
2) Joint-data learning
3) Multi-task learning

Joint generation



Force the decoder to produce the **POS sequence alongside the usual translation** sequence

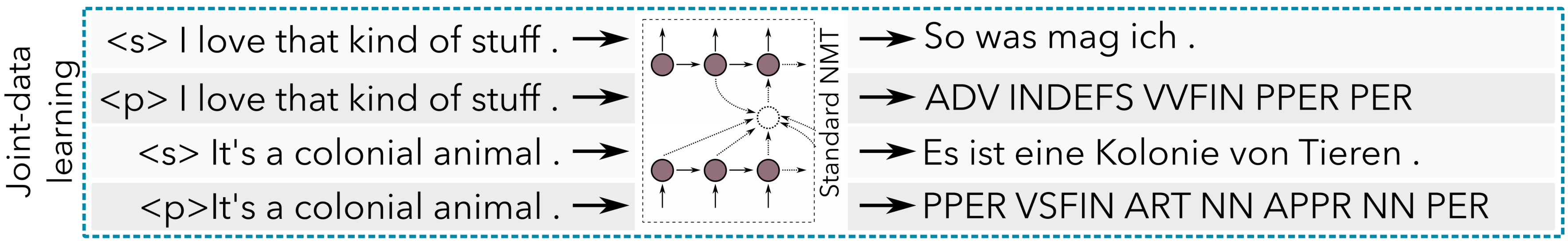
Joint generation



Pro: No changes in existing NMT architecture

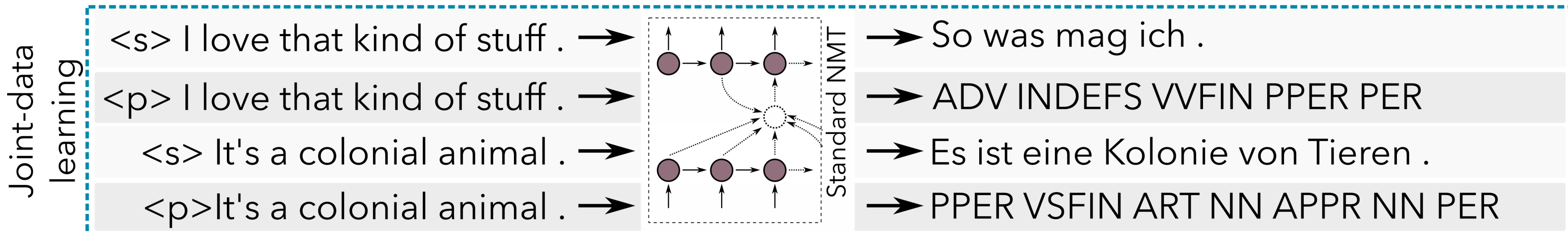
Con: Word and POS bases are far from each other, will require attention to attend to each source word twice

Joint-data learning



Make the decoder predict **translation or POS** sequence. Output type is defined by <s>/<p> tags in source sentence

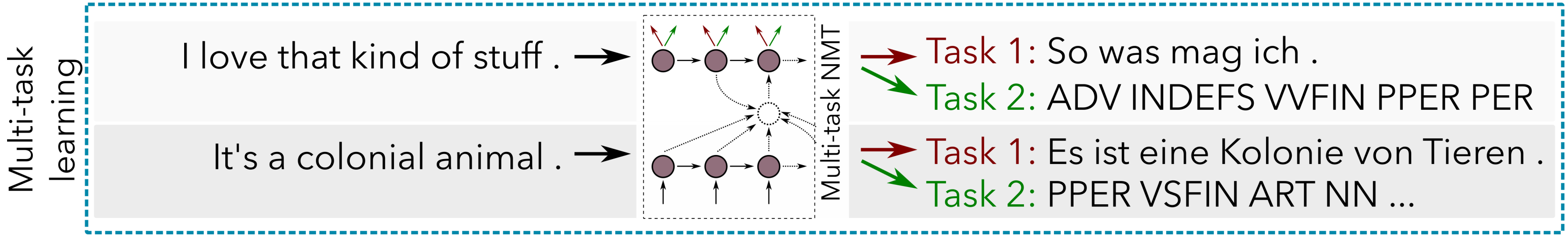
Joint-data learning



Pro: No changes in existing NMT architecture

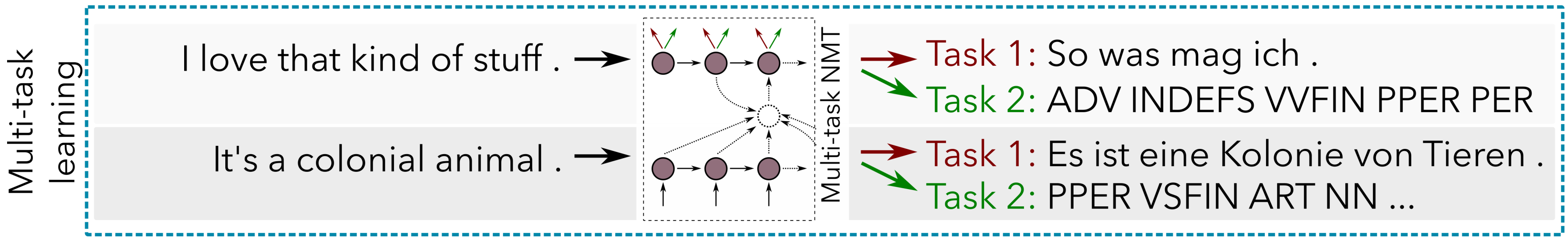
Con: Data is explicitly doubled, so training takes longer

Multi-task learning



Make the decoder predict both the **translation and POS sequence simultaneously**

Multi-task learning

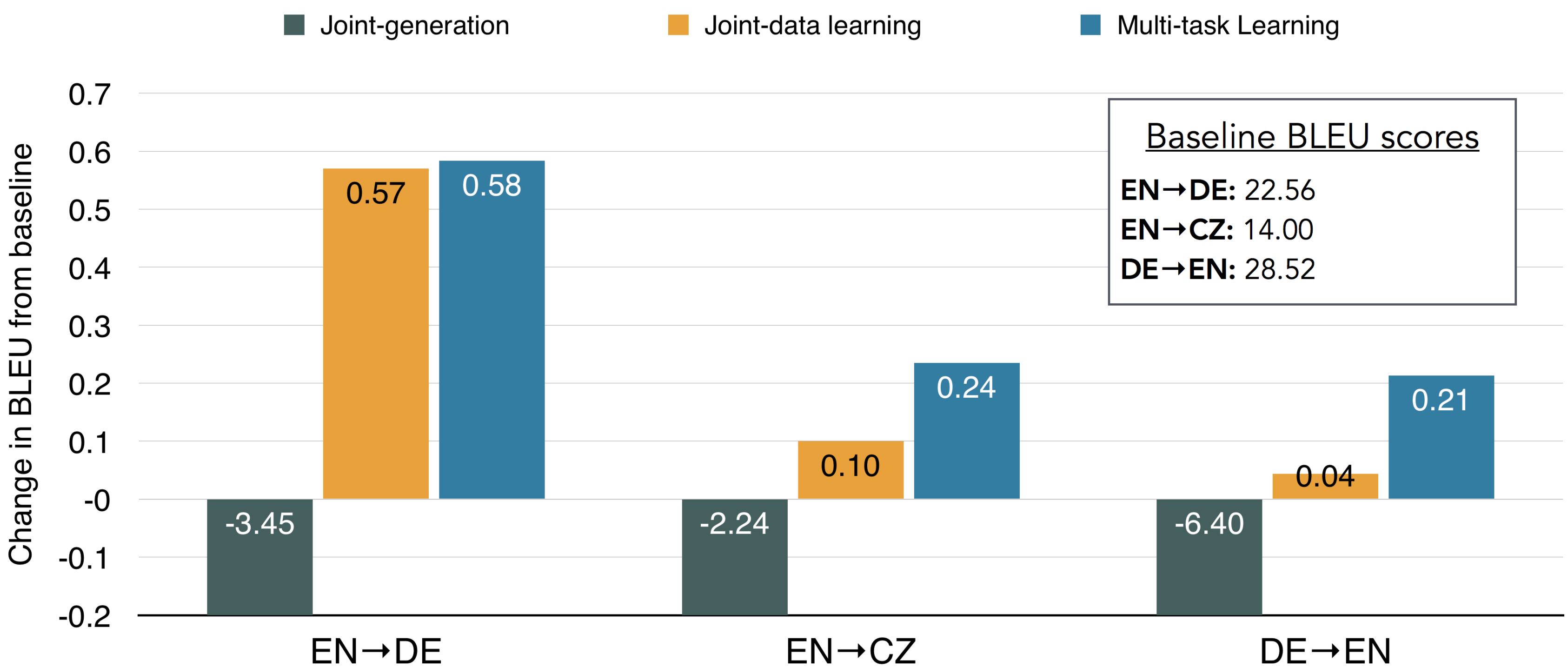


Pro: Principled approach, avoids issues of previous methods

Con: Requires modification to standard sequence-to-sequence to perform multiple tasks

QCRI
معهد قطر لبحوث الحوسبة
Qatar Computing Research Institute
جامعة حمد بن خليفة
HAMAD BIN KHALIFA UNIVERSITY

Results



QCRI
معهد قطر لبحوث الحوسبة
Qatar Computing Research Institute
جامعة حمد بن خليفة
HAMAD BIN KHALIFA UNIVERSITY

Conclusion

1) Explicit morphological knowledge injection leads to improved translation performance
2) Code is available at:
   https://github.com/fdalvi/seq2seq-attn-multitask

Thank you!

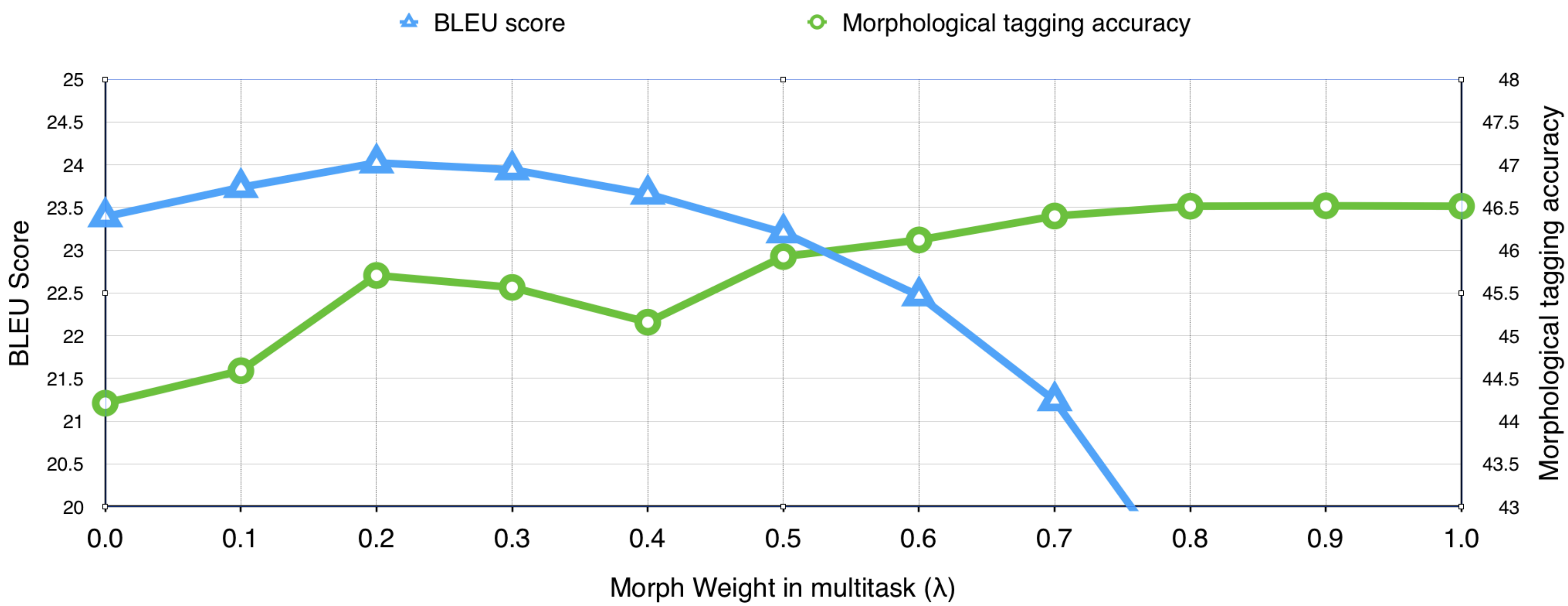
Questions?

Backup

Results

Multi-task learning has **two objective functions** in our case – one for translation and one for POS tagging. We can introduce a **hyper parameter to weigh** the importance of these objective functions

Results



Hyper parameter tuning results for
En → De model

Results

Intuitively, translation is a **much more important task**, and hence this weighing **should not be equal**

The other methods (Joint generation and Joint-data learning) do not allow us to weigh these two different tasks easily, which is an advantage of Multi-task learning!