UNIVERSITY OF HELSINKI
FACULTY OF ARTS

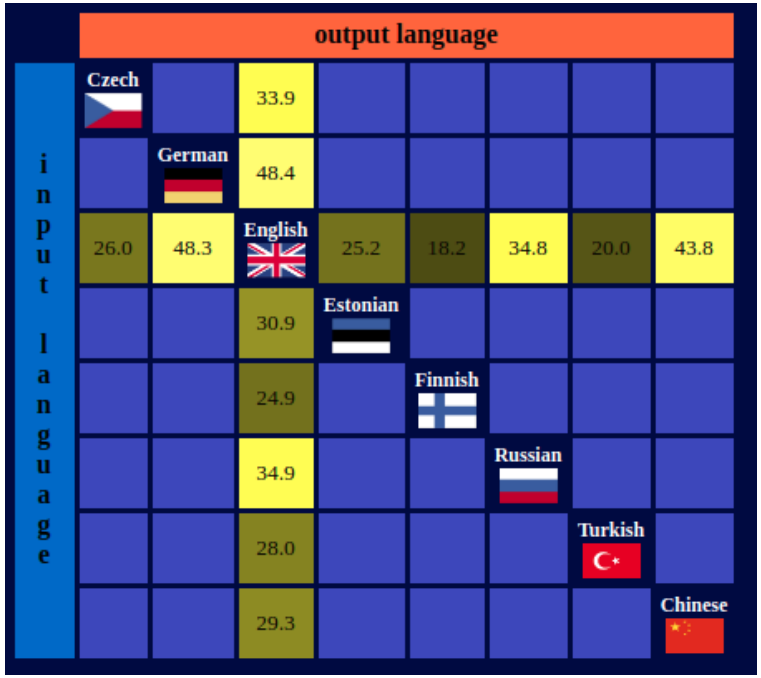
An Analysis of Encoder Representations in Transformer-Based MT

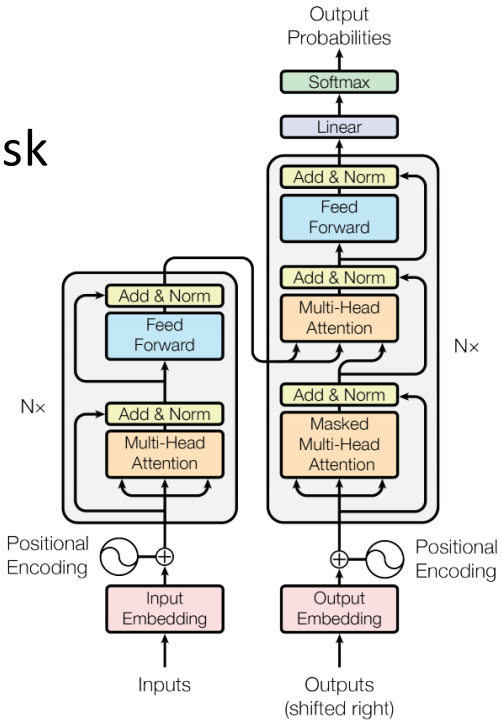
Alessandro Raganato *and* ***Jörg Tiedemann***

HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

Neural Machine Translation

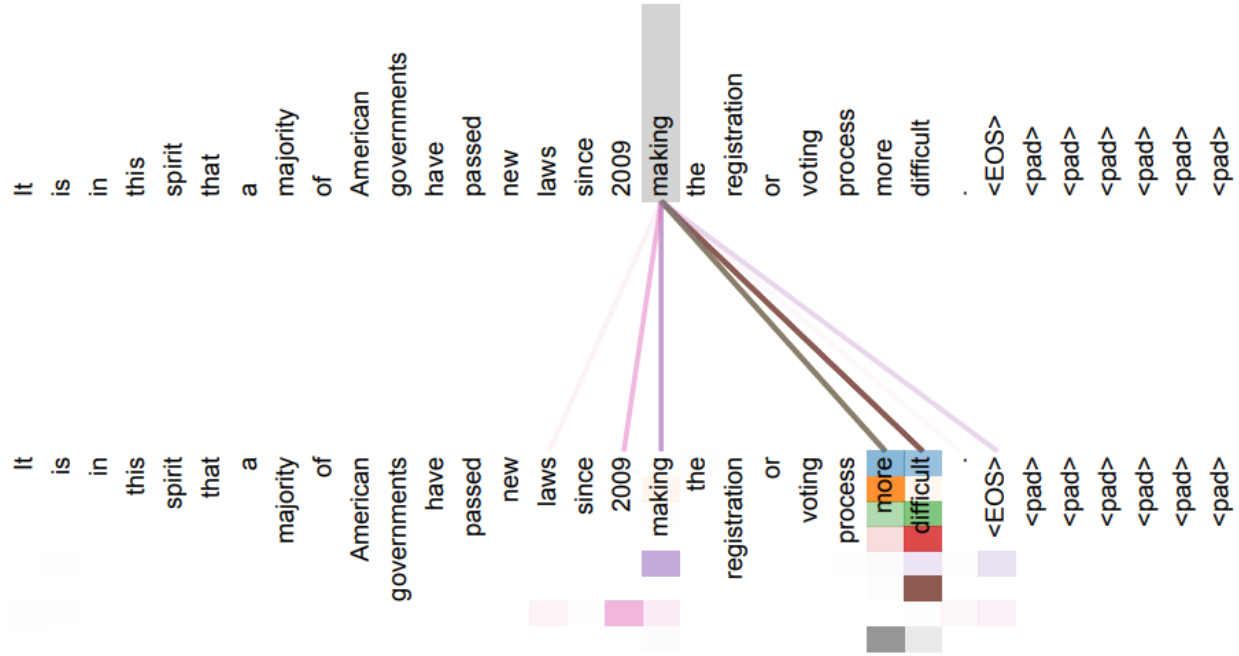
- Self-attention (Transformer model, Attention Is All You Need [Vaswani et al., 2017])
- Most submissions for the WMT18 News shared task use the Transformer architecture





NMT - Transformer model

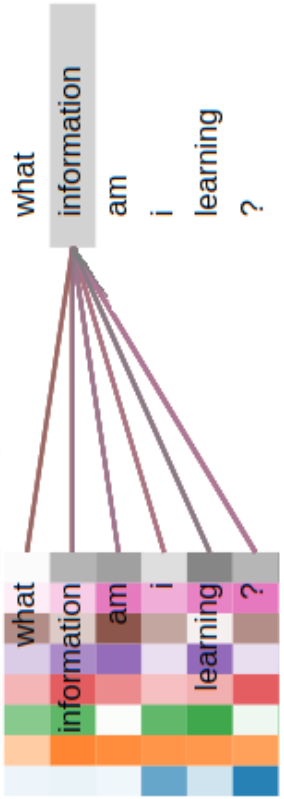
Attention visualizations:
encoder self-attention



picture from Vaswani et al., 2017

NMT - Transformer model

Our goal:



Encoder Evaluation

- **How?**
  1. Inducing tree structure:
     - use the attention weights in each layer to extract trees from the input sentences and inspect whether they reflect dependency trees

Encoder Evaluation

- **How?**
  1. Inducing tree structure:
     - use the attention weights in each layer to extract trees from the input sentences and inspect whether they reflect dependency trees

  2. Probing the encoder weights of the trained models to address different sequence labeling tasks:
     - Part-of-Speech tagging (PoS)
     - Chunking (CHUNK)
     - Named Entity Recognition (NER)
     - Semantic tagging (SEM)

Encoder Evaluation

- **How?**
  1. Inducing tree structure:
     - use the attention weights in each layer to extract trees from the input sentences and inspect whether they reflect dependency trees

  2. Probing the encoder weights of the trained models to address different sequence labeling tasks:
     - Part-of-Speech tagging (PoS)
     - Chunking (CHUNK)
     - Named Entity Recognition (NER)
     - Semantic tagging (SEM)

  3. Transfer learning:
     - we use the encoder weights of a high-resource language pair to initialize a low-resource language pair

Model setup

- Transformer architecture:
  - *base* version, 6 layers, 8 attention heads, etc.
- Training data:
  - WMT18 News Task
  - vocabulary 100K full word forms
- Testing data:
  - newstest 2017
  - newstest 2018

	#Training sentences
English → Czech	51.391.404
English → German	25.746.259
English → Estonian	1.064.658
English → Finnish	2.986.131
English → Russian	9.140.469
English → Turkish	205.579
English → Chinese	23.861.542

	newstest 2017	newstest 2018
English → Czech	18.11	17.36
English → German	23.37	34.46
English → Estonian	–	13.05
English → Finnish	15.06	10.32
English → Russian	21.30	18.96
English → Turkish	6.93	6.22
English → Chinese	23.10	23.75

Encoder Evaluation: inducing tree structure

Encoder Evaluation: inducing tree structure

Find the highest scoring tree (Chu-Liu-Edmonds algorithm)

Dataset:

- English PUD treebank from the CoNLL 2017 Shared Task (1000 sentences), Unlabeled Attachment Score.

ACADEMY OF FINLAND

Encoder Evaluation: inducing tree structure

	en->cs	en->de	en->et	en->fi	en->ru	en->tr	en->zh
Layer 0	30.26	32.90	31.38	31.63	17.13	**31.81**	33.26
	9.94	10.62	7.76	8.31	11.08	9.78	9.69
Layer 1	35.08	**35.94**	**35.07**	**35.30**	36.05	26.20	35.77
	10.72	10.65	10.85	10.10	10.27	11.03	9.62
Layer 2	35.46	33.76	33.16	29.43	**36.08**	22.56	35.80
	10.17	9.02	7.40	9.00	7.63	9.52	10.71
Layer 3	35.20	35.59	22.62	27.24	35.03	21.53	**38.87**
	8.28	9.97	7.99	7.78	9.27	7.18	9.02
Layer 4	29.66	27.88	32.87	24.00	27.68	25.40	35.40
	10.37	10.69	9.95	9.52	10.72	11.06	11.56
Layer 5	**36.02**	35.32	33.68	31.87	35.56	28.23	29.73
	11.86	8.30	14.83	11.01	9.77	7.98	13.45

Encoder Evaluation: inducing tree structure

	en->cs	en->de	en->et	en->fi	en->ru	en->tr	en->zh
Layer 0	30.26	32.90	31.38	31.63	17.13	**31.81**	33.26
	9.94	10.62	7.76	8.31	11.08	9.78	9.69
Layer 1	35.08	**35.94**	**35.07**	**35.30**	36.05	26.20	35.77
	10.72	10.65	10.85	10.10	10.27	11.03	9.62
Layer 2	35.46	33.76	33.16	29.43	**36.08**	22.56	35.80
	10.17	9.02	7.40	9.00	7.63	9.52	10.71
Layer 3	35.20	35.59	22.62	27.24	35.03	21.53	**38.87**
	8.28	9.97	7.99	7.78	9.27	7.18	9.02
Layer 4	29.66	27.88	32.87	24.00	27.68	25.40	35.40
	10.37	10.69	9.95	9.52	10.72	11.06	11.56
Layer 5	**36.02**	35.32	33.68	31.87	35.56	28.23	29.73
	11.86	8.30	14.83	11.01	9.77	7.98	13.45

Supervised Approach	88.22
Random baseline	10.1
Left-branch baseline	10.39
Right-branch baseline	35.08

Encoder Evaluation: inducing tree structure

	en->cs	en->de	en->et	en->fi	en->ru	en->tr	en->zh
Layer 0	30.26	32.90	31.38	31.63	17.13	**31.81**	33.26
	9.94	10.62	7.76	8.31	11.08	9.78	9.69
Layer 1	35.08	**35.94**	**35.07**	**35.30**	36.05	26.20	35.77
	10.72	10.65	10.85	10.10	10.27	11.03	9.62
Layer 2	35.46	33.76	33.16	29.43	**36.08**	22.56	35.80
	10.17	9.02	7.40	9.00	7.63	9.52	10.71
Laye	35.20	35.59	22.62	27.24	35.03	21.53	**38.87**
Laye							
Laye							

Supervised Approach	88.22
Random baseline	10.1
Left-branch baseline	10.39
	08

Encoder evaluation: Probing Sequence labeling tasks

We evaluate the quality of the decoder on a given task to assess how discriminative the encoder representation is for that task.

- one decoder layer using one attention head and one feed-forward layer.
- assess the quality of the encoder representation across stacked layers.

Datasets:

- POS: Universal Dependencies English Web Treebank v2.0
- CHUNK: CoNLL2000 Chunking shared task
- NER: CoNLL2003 NER shared task
- SEM: Parallel Meaning Bank for Semantic tagging

Encoder evaluation: Probing Sequence labeling tasks

		en → cs	en → de	en → et	en → fi	en → ru	en → tr	en → zh
POS	layer 0	91.13 / 7.70	91.06 / 8.20	84.49 / 18.20	86.88 / 25.00	89.47 / 6.00	**68.47** / 52.10	90.81 / 12.20
	layer 1	92.79 / **2.90**	93.12 / 4.60	**87.11** / 18.40	**87.58** / **12.40**	90.67 / 10.60	67.53 / 47.00	**92.60** / **7.90**
	layer 2	**93.20** / 5.40	**93.18** / **4.50**	84.99 / **14.70**	86.41 / 15.20	91.86 / **3.90**	68.13 / **45.40**	91.68 / 13.30
	layer 3	92.24 / 9.50	92.31 / 8.60	84.51 / 16.60	85.16 / 18.70	91.46 / 6.00	66.50 / 53.20	89.52 / 19.00
	layer 4	91.66 / 10.80	90.85 / 13.70	82.65 / 23.70	83.46 / 24.40	**91.98** / 12.00	65.66 / 53.90	86.47 / 22.10
	layer 5	87.14 / 19.10	87.83 / 24.10	82.11 / 23.60	80.41 / 33.30	89.47 / 16.30	62.80 / 54.80	82.95 / 31.30
CHUNK	layer 0	90.28 / 4.37	89.78 / 9.49	86.98 / 13.47	87.75 / **8.90**	88.12 / 6.61	**72.64** / **31.21**	90.37 / 5.42
	layer 1	92.98 / **4.32**	92.91 / 3.58	**88.00** / **11.78**	**88.92** / 10.19	91.16 / **4.03**	71.59 / 40.81	92.76 / **6.71**
	layer 2	**93.56** / 6.56	**93.92** / **3.53**	**88.00** / 12.28	88.65 / 13.22	91.60 / 5.82	70.25 / 37.38	**93.40** / 11.18
	layer 3	93.46 / 12.33	**93.92** / 10.14	87.56 / 14.36	87.41 / 19.93	**92.78** / 5.91	69.20 / 46.17	90.83 / 16.90
	layer 4	92.68 / 14.66	92.83 / 12.77	85.80 / 22.81	86.60 / 20.13	92.73 / 12.72	68.54 / 51.04	89.30 / 19.09
	layer 5	90.87 / 14.46	89.92 / 16.60	85.34 / 19.88	84.04 / 27.14	90.95 / 15.11	65.01 / 53.33	82.82 / 31.71
NER	layer 0	91.18 / 23.75	92.71 / 12.02	87.21 / 33.03	89.38 / 29.53	91.29 / 14.58	86.49 / 39.47	91.72 / **11.05**
	layer 1	93.29 / 9.80	93.36 / 7.27	88.65 / **15.99**	90.14 / **20.77**	92.22 / 10.07	85.66 / 38.14	92.93 / 11.13
	layer 2	**93.83** / **7.11**	94.13 / **11.13**	87.46 / 37.30	90.20 / 26.47	**93.20** / **8.12**	86.52 / 43.05	**93.72** / 12.35
	layer 3	93.23 / 16.53	**94.32** / 14.85	**88.95** / 33.31	**90.22** / 26.57	93.14 / 9.42	86.82 / **37.68**	93.07 / 18.32
	layer 4	93.72 / 11.81	93.93 / 12.51	88.57 / 40.55	89.14 / 34.28	92.02 / 12.65	**87.21** / 53.99	91.93 / 26.95
	layer 5	92.62 / 21.63	94.11 / 17.35	87.64 / 30.13	89.40 / 31.49	92.33 / 13.98	86.06 / 44.25	92.35 / 30.08
SEM	layer 0	83.99 / 13.56	84.05 / 13.35	81.87 / 14.73	81.99 / 14.69	83.36 / 14.07	79.04 / **16.87**	84.08 / 13.63
	layer 1	84.84 / 12.48	85.27 / 12.16	82.25 / **14.11**	82.70 / 13.97	84.12 / 13.26	78.80 / 17.10	84.93 / 11.88
	layer 2	85.17 / 11.95	85.11 / 12.16	82.28 / 14.25	82.76 / 14.85	84.09 / 13.03	78.26 / 18.09	85.40 / 11.74
	layer 3	85.34 / 12.02	84.77 / 11.45	82.17 / 14.41	82.82 / 14.00	**85.21** / 12.32	**79.22** / 17.28	84.79 / 11.91
	layer 4	85.29 / **11.38**	**85.91** / **9.93**	**82.44** / 14.50	**83.19** / **13.77**	84.26 / 12.50	78.36 / 19.26	85.38 / 11.42
	layer 5	**86.27** / 11.68	85.71 / 10.78	82.27 / 14.55	82.96 / 13.84	84.56 / **11.79**	78.67 / 18.78	**85.98** / **10.62**

precision

error rate on the sentence length

Encoder evaluation: Transfer learning

We used the encoder weights from one high resource language, i.e., English-German, to train a Transformer model for a low resource language pair, English-Turkish.

Experiments:

- initializing and fine tuning the encoder weights **(TL1)**
- initializing and keeping the encoder weights fixed **(TL2)**

Encoder evaluation: Transfer learning

We used the encoder weights from one high resource language, i.e., English-German, to train a Transformer model for a low resource language pair, English-Turkish.

Experiments:

- initializing and fine tuning the encoder weights **(TL1)**
- initializing and keeping the encoder weights fixed **(TL2)**

	newstest 2017	newstest 2018
English → Turkish	6.93	6.22
English TL1 → Turkish	8.72	7.93
English TL2 → Turkish	7.82	6.91

Conclusion

- We find that each layer has at least one attention head that encodes a significant amount of syntactic dependencies.
- Consistent with previous findings on the sequence-to-sequence paradigm, probing the encoder to four different sequence labeling tasks reveals that lower layers tend to encode more syntactic information, whereas upper layers move towards semantic tasks.
- The information about the length of the input sentence starts to vanish after the third layer.
- The study corroborates that attention can be used to transfer knowledge between high- and low-resource languages.

ACADEMY
OF FINLAND

FOTRAN
Found in Translation

Thank You!

HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI