HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

LM and NMT systems are different, but why?

Raúl Vázquez

FOTRAN
Found in Translation

May 2022

Outline

1. Encoder Spaces
2. Training Dynamics
3. Next Steps
4. Conclusions

Outline

0 Motivation

1 Encoder Spaces

2 Training Dynamics

3 Next Steps

4 Conclusions

Motivation

- Better understand SOTA results (& enhance design process - better than trial and error)
- Optimize resource usage
  - Why pruning works or how to design pruning strategies
  - More efficient transfer of knowledge
- Insights on what info is processed where in the network
  - May trigger new insights
- Get away of the “magical thinking”

1

Differences between MLM and MT encoder spaces

Vázquez, R., H. Celikkanat, M. Creutz, and J. Tiedemann (2021) On the differences between BERT and MT encoder spaces and how to address them in translation tasks. In *Proceedings of the 59th Annual Meeting of the ACL and the 11th IJCNLP: SRW*, pages 337–347, Online. ACL.

Tools used for the analysis

Isotropy

Average-similarity

Contextuality

Self-similarity
&
Intra-sentence similarity

Representational Similarity

RSA
&
PWCCA

Measuring Isotropy (Ethayarajh, 2019)

Isotropy = directional uniformity

It plays an important role when using cosine similarity

AvgSim: average cossim between the representations randomly sampled words from different contexts

$$\mathbb{E}_{x,y\sim U(\mathcal{O})}\left[\cos(f_\ell(x), f_\ell(y))\right]$$

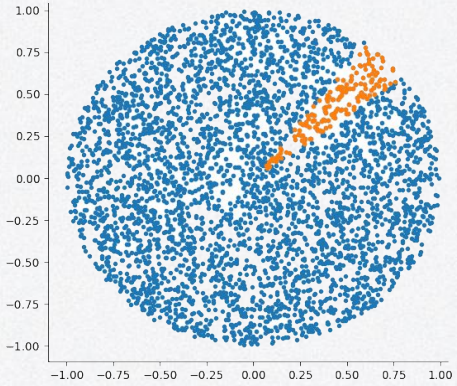


Figure 1: (blue) points sampled uniformly over the unitary sphere. (orange) points sampled over a portion of the unitary sphere such that cossim((x,y) , (1,1)) > 0.99

Measuring Contextuality (Ethayarajh, 2019)

SelfSim — average cossim between the contextualized representations of a word across its occurrences in the corpus

IntraSim — average cossim between the word representations in a sentence and their mean vector.

$$SelfSim_\ell(w) = \frac{1}{n^2 - n} \sum_j \sum_{k \neq j} \cos(f_\ell(s_j, i_j), f_\ell(s_k, i_k))$$

$$IntraSim_\ell(s) = \frac{1}{n} \sum_i \cos(\vec{s}_\ell, f_\ell(s, i))$$

$$\text{where } \vec{s}_\ell = \frac{1}{n} \sum_i f_\ell(s, i)$$

Pretrained models used

HUGGING FACE

AvgSim

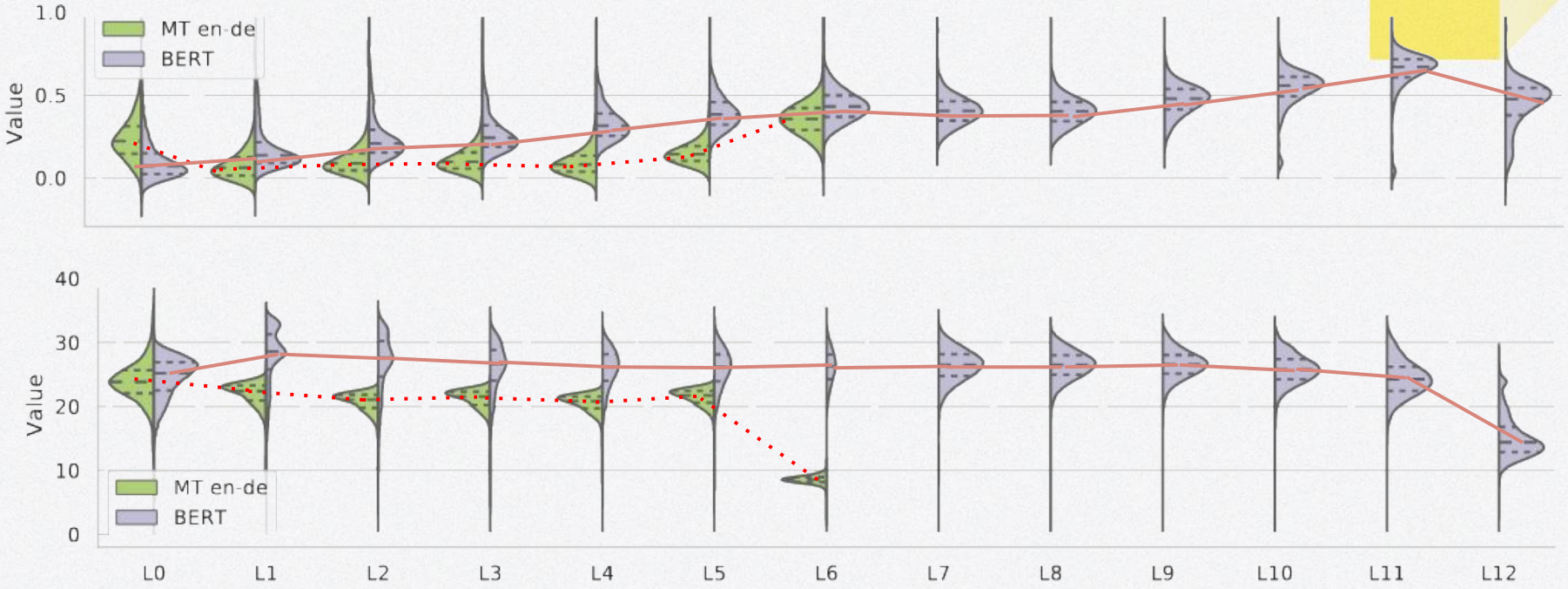


Figure 2: Cosine similarity *(top)* and Euclidean distance *(bottom)* distributions between randomly sampled words.

AvgSim

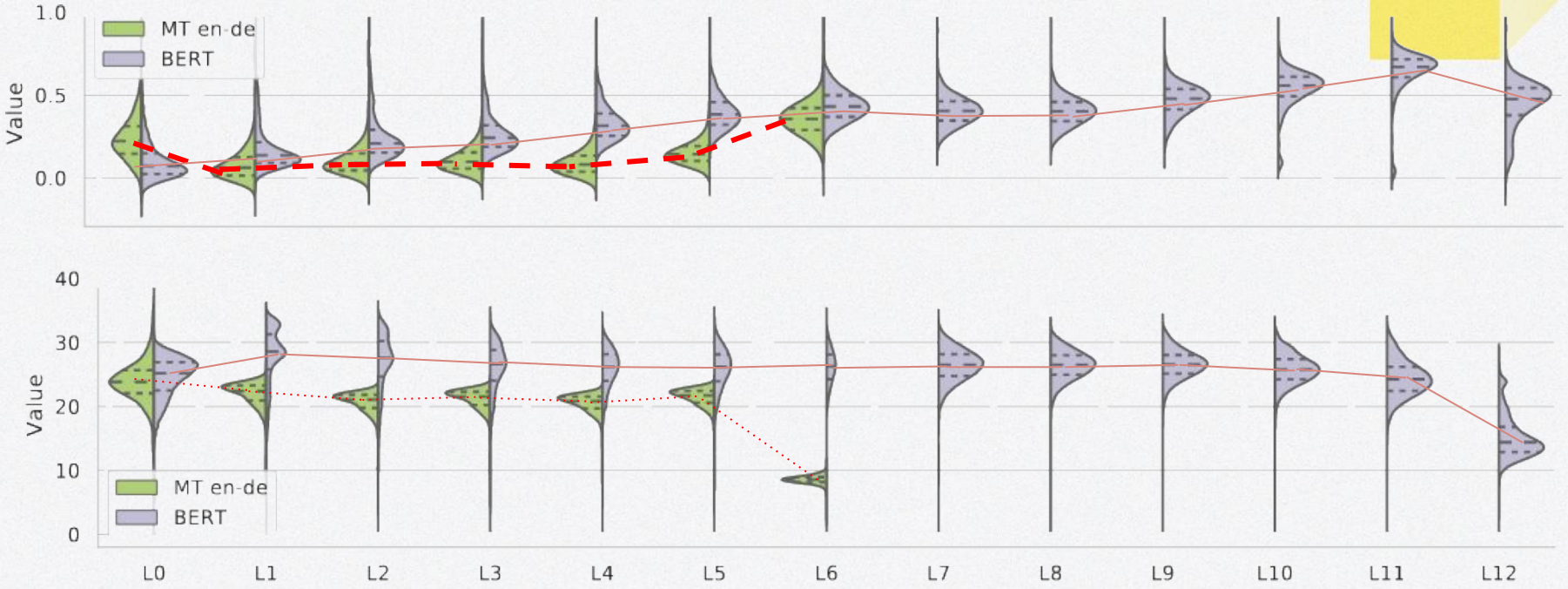
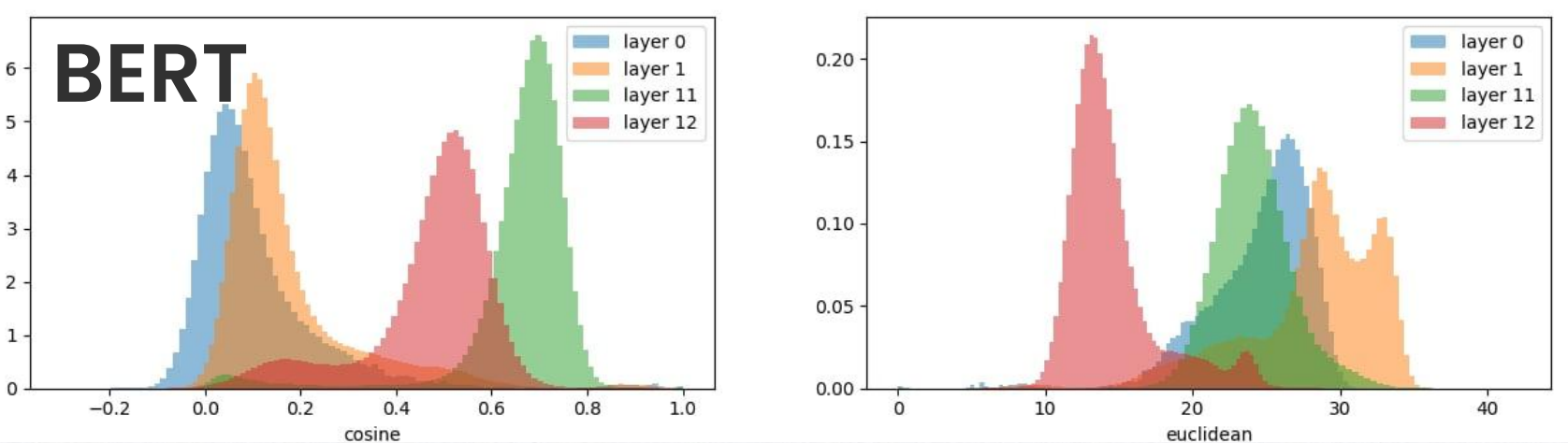
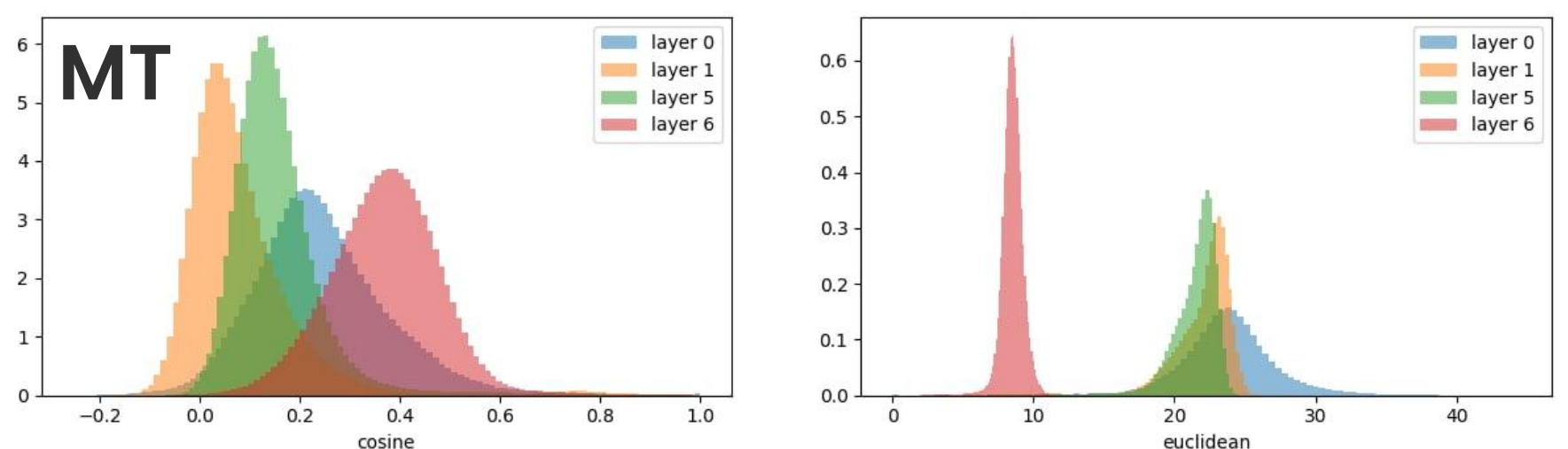


Figure 2: Cosine similarity *(top)* and Euclidean distance *(bottom)* distributions between randomly sampled words.


BERT
layer 0
layer 1
layer 11
layer 12
cosine
euclidean


MT
layer 0
layer 1
layer 5
layer 6
cosine
euclidean

A high average similarity of the MT embeddings layer

In layer 0 the representations are not yet that “contextualized” this early in the model

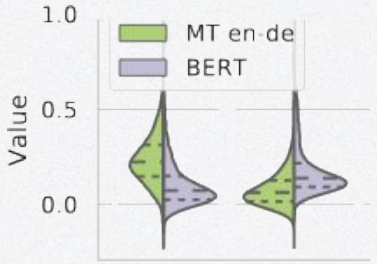
Contrast in the AvgSim behaviour: BERT vs. the MT encoder

It gradually increases in BERT but there is a steep increase at the last layer of MT model

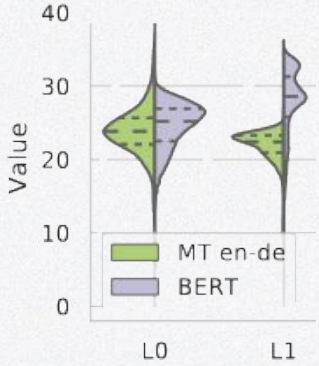
A high average similarity of the MT embeddings layer

In layer 0 the representations are not yet that “contextualized” this early in the model

cosine-based



L2norm-based



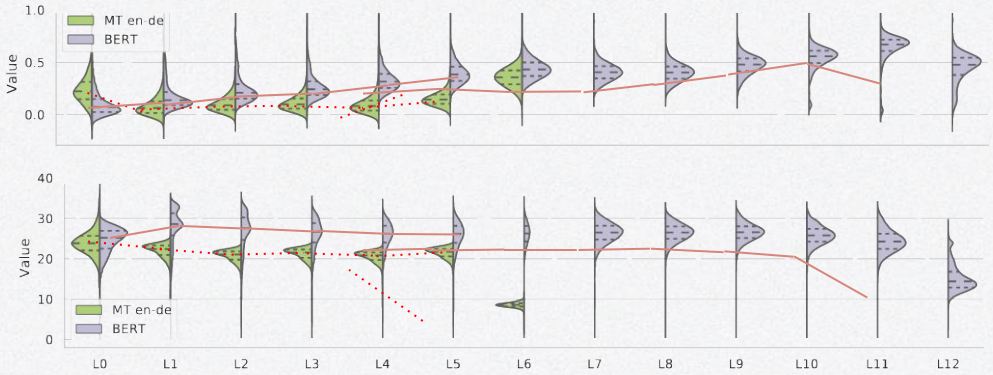


Figure 2: Cosine similarity *(top)* and Euclidean distance *(bottom)* distributions between randomly sampled words.

Contrast in the AvgSim behaviour: BERT vs. the MT encoder

It gradually increases in BERT. but there is a steep increase at the last layer of MT model

Self similarity & Intra-sentence sim

Figure 3: Comparison of contextualization for BERT and MT spaces using SelfSim and IntraSim. We also present the raw SelfSim before anisotropy correction

Representational similarity

4

RSA
Kriegeskorte et al. (2008)

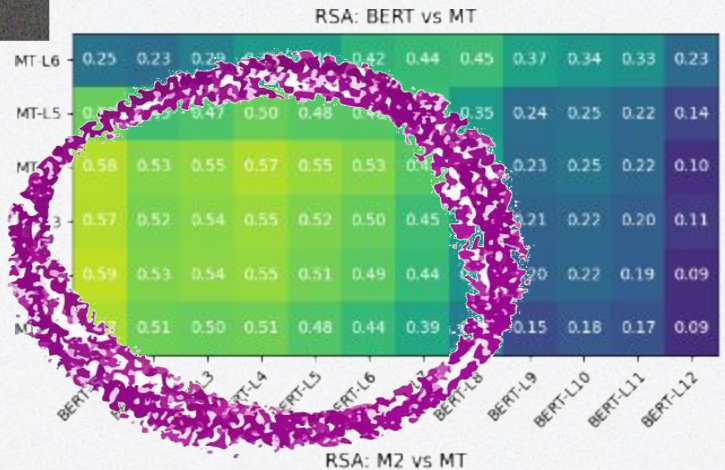
finds connections between data from two representation spaces by contrasting the adjacency matrices via a similarity measure.

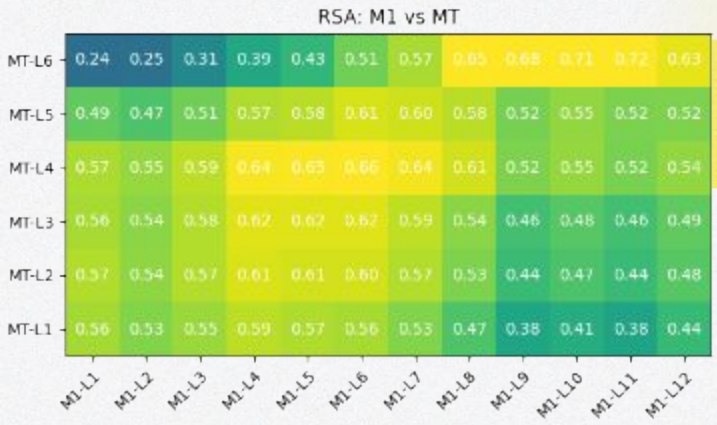
5

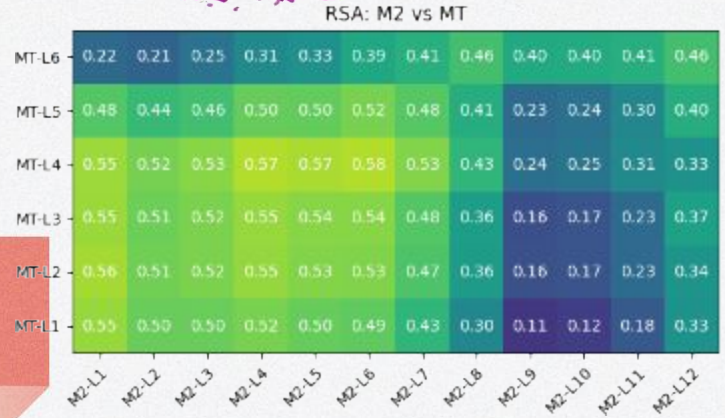
PWCCA
Raghu et al. (2017)

Extends SVCCA by performing

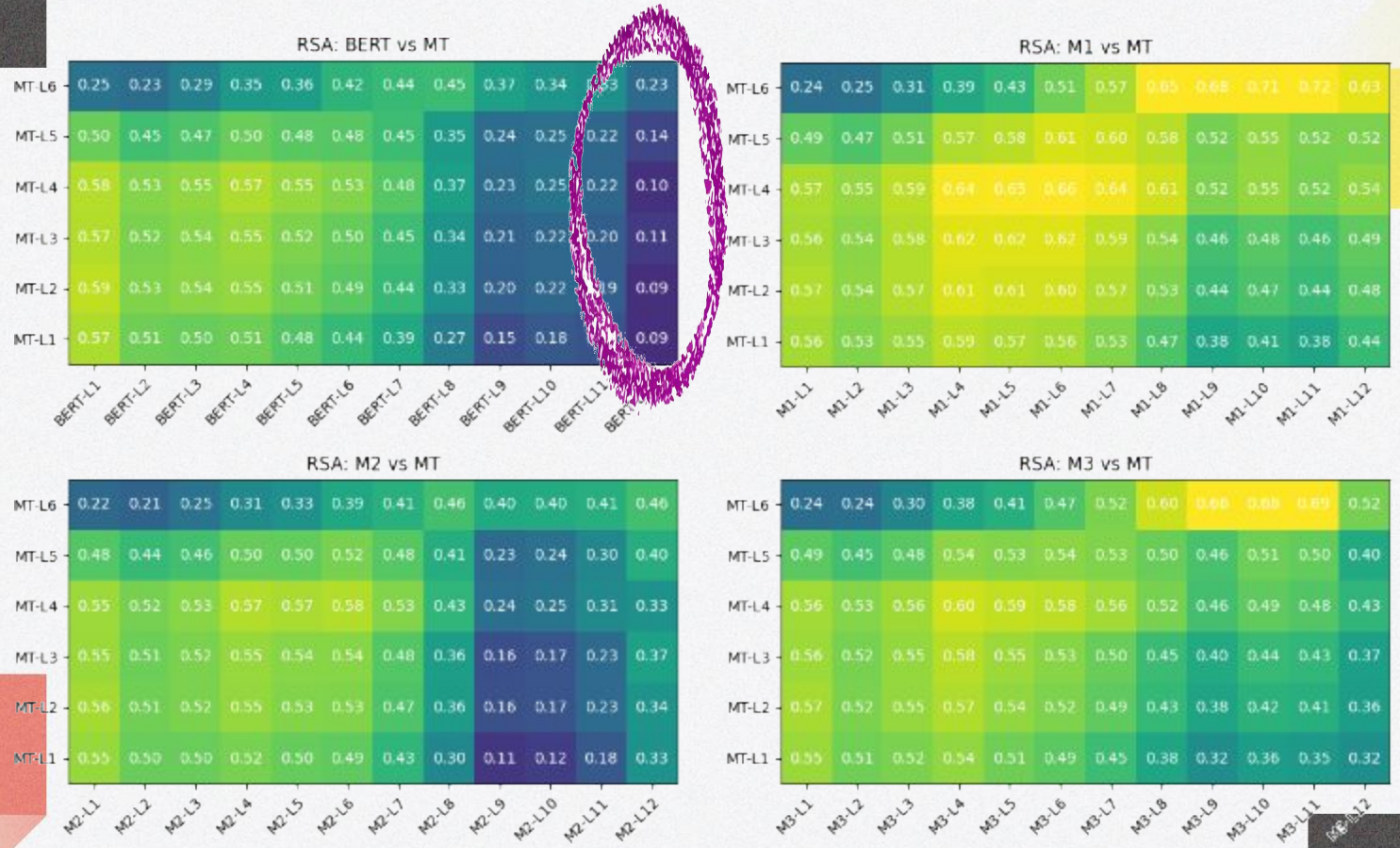
1. SVD over the dimension space,
2. CCA to find max- correlated transformations
3. a weighted average of the correlation coefficients


RSA: BERT vs MT
MT-L6 0.25 0.23 0.42 0.44 0.45 0.37 0.34 0.33 0.23
MT-L5 0.47 0.50 0.48 0.35 0.24 0.25 0.22 0.14
0.58 0.53 0.55 0.57 0.55 0.53 0.23 0.25 0.22 0.10
0.57 0.52 0.54 0.55 0.52 0.50 0.45 0.21 0.22 0.20 0.11
0.59 0.53 0.54 0.55 0.51 0.49 0.44 0.20 0.22 0.19 0.09
0.51 0.50 0.51 0.48 0.44 0.39 0.15 0.18 0.17 0.09
BERT-L9 BERT-L10 BERT-L11 BERT-L12


RSA: M1 vs MT
MT-L6 0.24 0.25 0.31 0.39 0.43 0.51 0.57 0.65 0.68 0.71 0.72 0.63
MT-L5 0.49 0.47 0.51 0.57 0.58 0.61 0.60 0.58 0.52 0.55 0.52 0.52
MT-L4 0.57 0.55 0.59 0.64 0.65 0.66 0.64 0.61 0.52 0.55 0.52 0.54
MT-L3 0.56 0.54 0.58 0.62 0.62 0.62 0.59 0.54 0.46 0.48 0.46 0.49
MT-L2 0.57 0.54 0.57 0.61 0.61 0.60 0.57 0.53 0.44 0.47 0.44 0.48
MT-L1 0.56 0.53 0.55 0.59 0.57 0.56 0.53 0.47 0.38 0.41 0.38 0.44
M1-L1 M1-L2 M1-L3 M1-L4 M1-L5 M1-L6 M1-L7 M1-L8 M1-L9 M1-L10 M1-L11 M1-L12


RSA: M2 vs MT
MT-L6 0.22 0.21 0.25 0.31 0.33 0.39 0.41 0.46 0.40 0.40 0.41 0.46
MT-L5 0.48 0.44 0.46 0.50 0.50 0.52 0.48 0.41 0.23 0.24 0.30 0.40
MT-L4 0.55 0.52 0.53 0.57 0.57 0.58 0.53 0.43 0.24 0.25 0.31 0.33
MT-L3 0.55 0.51 0.52 0.55 0.54 0.54 0.48 0.36 0.16 0.17 0.23 0.37
MT-L2 0.56 0.51 0.52 0.55 0.53 0.53 0.47 0.36 0.16 0.17 0.23 0.34
MT-L1 0.55 0.50 0.50 0.52 0.50 0.49 0.43 0.30 0.11 0.12 0.18 0.33
M2-L1 M2-L2 M2-L3 M2-L4 M2-L5 M2-L6 M2-L7 M2-L8 M2-L9 M2-L10 M2-L11 M2-L12


RSA: M3 vs MT
MT-L6 0.24 0.24 0.30 0.38 0.41 0.47 0.52 0.60 0.60 0.60 0.60 0.52
MT-L5 0.49 0.45 0.48 0.54 0.53 0.54 0.53 0.50 0.46 0.51 0.50 0.40
MT-L4 0.56 0.53 0.56 0.60 0.59 0.58 0.56 0.52 0.46 0.49 0.48 0.43
MT-L3 0.56 0.52 0.55 0.58 0.55 0.53 0.50 0.45 0.40 0.44 0.43 0.37
MT-L2 0.57 0.52 0.55 0.57 0.54 0.52 0.49 0.43 0.38 0.42 0.41 0.36
MT-L1 0.55 0.51 0.52 0.54 0.51 0.49 0.45 0.38 0.32 0.36 0.35 0.32
M3-L1 M3-L2 M3-L3 M3-L4 M3-L5 M3-L6 M3-L7 M3-L8 M3-L9 M3-L10 M3-L11 M3-L12


RSA: BERT vs MT
MT-L6 0.25 0.23 0.29 0.35 0.36 0.42 0.44 0.45 0.37 0.34 0.33 0.23
MT-L5 0.50 0.45 0.47 0.50 0.48 0.48 0.45 0.35 0.24 0.25 0.22 0.14
MT-L4 0.58 0.53 0.55 0.57 0.55 0.53 0.48 0.37 0.23 0.25 0.22 0.10
MT-L3 0.57 0.52 0.54 0.55 0.52 0.50 0.45 0.34 0.21 0.23 0.20 0.11
MT-L2 0.59 0.53 0.54 0.55 0.51 0.49 0.44 0.33 0.20 0.22 0.19 0.09
MT-L1 0.57 0.51 0.50 0.51 0.48 0.44 0.39 0.27 0.15 0.18 0.16 0.09
BERT-L1 BERT-L2 BERT-L3 BERT-L4 BERT-L5 BERT-L6 BERT-L7 BERT-L8 BERT-L9 BERT-L10 BERT-L11 BERT-L12
RSA: M1 vs MT
MT-L6 0.24 0.25 0.31 0.39 0.43 0.51 0.57 0.65 0.68 0.71 0.72 0.63
MT-L5 0.49 0.47 0.51 0.57 0.58 0.61 0.60 0.58 0.52 0.55 0.52 0.52
MT-L4 0.57 0.55 0.59 0.64 0.65 0.66 0.64 0.61 0.52 0.55 0.52 0.54
MT-L3 0.56 0.54 0.58 0.62 0.62 0.62 0.59 0.54 0.46 0.48 0.46 0.49
MT-L2 0.57 0.54 0.57 0.61 0.61 0.60 0.57 0.53 0.44 0.47 0.44 0.48
MT-L1 0.56 0.53 0.55 0.59 0.57 0.56 0.53 0.47 0.38 0.41 0.38 0.44
M1-L1 M1-L2 M1-L3 M1-L4 M1-L5 M1-L6 M1-L7 M1-L8 M1-L9 M1-L10 M1-L11 M1-L12
RSA: M2 vs MT
MT-L6 0.22 0.21 0.25 0.31 0.33 0.39 0.41 0.46 0.40 0.40 0.41 0.46
MT-L5 0.48 0.44 0.46 0.50 0.50 0.52 0.48 0.41 0.23 0.24 0.30 0.40
MT-L4 0.55 0.52 0.53 0.57 0.57 0.58 0.53 0.43 0.24 0.25 0.31 0.33
MT-L3 0.55 0.51 0.52 0.55 0.54 0.54 0.48 0.36 0.16 0.17 0.23 0.37
MT-L2 0.56 0.51 0.52 0.55 0.53 0.53 0.47 0.36 0.16 0.17 0.23 0.34
MT-L1 0.55 0.50 0.50 0.52 0.50 0.49 0.43 0.30 0.11 0.12 0.18 0.33
M2-L1 M2-L2 M2-L3 M2-L4 M2-L5 M2-L6 M2-L7 M2-L8 M2-L9 M2-L10 M2-L11 M2-L12
RSA: M3 vs MT
MT-L6 0.24 0.24 0.30 0.38 0.41 0.47 0.52 0.60 0.66 0.68 0.69 0.52
MT-L5 0.49 0.45 0.48 0.54 0.53 0.54 0.53 0.50 0.46 0.51 0.50 0.40
MT-L4 0.56 0.53 0.56 0.60 0.59 0.58 0.56 0.52 0.46 0.49 0.48 0.43
MT-L3 0.56 0.52 0.55 0.58 0.55 0.53 0.50 0.45 0.40 0.44 0.43 0.37
MT-L2 0.57 0.52 0.55 0.57 0.54 0.52 0.49 0.43 0.38 0.42 0.41 0.36
MT-L1 0.55 0.51 0.52 0.54 0.51 0.49 0.45 0.38 0.32 0.36 0.35 0.32
M3-L1 M3-L2 M3-L3 M3-L4 M3-L5 M3-L6 M3-L7 M3-L8 M3-L9 M3-L10 M3-L11 M3-L12

Remark

The results presented so far were used to motivate the use of an alignment method between both embedding spaces.

Aligning them enabled BERT for performing translation.

Outline

0 Motivation

1 Encoder Spaces

2 Training Dynamics

3 Next Steps

4 Conclusions

2

The learning dynamics of neural LM and MT models

Work under review.

Learning objectives

MLM

$$X = (x_1, \ldots, x_{i-1}, [\text{MASK}], x_{i+1}, ..., x_T)$$

$$P(x_i|X, \theta)$$

CLM

$$X = (x_1, ..., x_{i-1})$$

$$P(x_i|x_1, ..., x_{i-1}, \theta)$$

MT

$$X = (x_1, \ldots, x_{T_x})$$

$$Y = (y_1, \ldots, y_{T_y})$$

$$P(y_i|X, y_1, \ldots y_{i-1}, \theta)$$

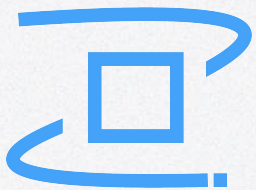
We enable a systematic comparison: standard architecture (tf-base), consistent experimental setup, train systems from scratch, same toolkit, same hardware and training time

Learning objectives

MLM **CLM** **MT**

We enable a systematic comparison: standard architecture (tf-base), consistent experimental setup, train systems from scratch, same toolkit, same hardware and training time

FAIRSEQ



1 V100 GPU
48hrs

Parameter	Value
num. layers	12 [6]
att. heads	8
embeddings dim.	512
ffwd hidden dim.	2048
update-freq	16
precision	fp16
total-num-update	125000
warmup-updates	8000
lr-scheduler	polynomial decay
lr	0.0005
optimizer	adam (2015)
adam-betas	(0.9, 0.98)
adam-eps	1e-06
weight-decay	0.01
clip-norm	0.0
dropout	0.1
attention dropout	0.0
activation-fn	relu
tokens-per-sample	512
max-tokens-per-sample	2048

Table: Set of hyper-parameters shared across all our models

Aims

- Identify the network components that contribute to the optimization of the loss function
- Reveal the effects of learning objectives and network components on training dynamics
- Detect the impact of training data size and distribution on parameter optimization

Loss Change Allocation (Lan et al., 2019)

It allows for a fine-grained analysis:

- measures the contribution of each parameter to the change in the loss at each gradient update
- decomposes the approximate path integral along the training trajectory

$$L(\theta_T) - L(\theta_0) = \int_C \langle \nabla_\theta L(\theta), d\theta \rangle$$

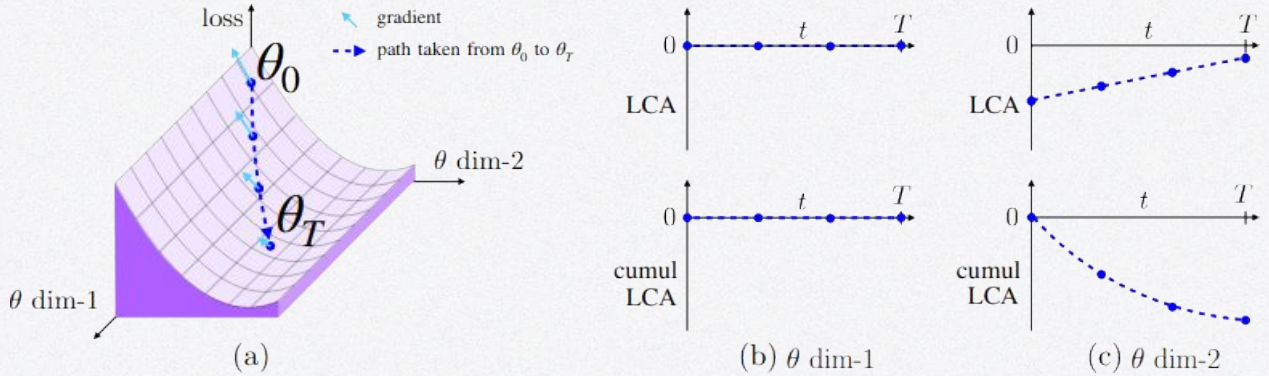
$$L(\theta_{t+1}) - L(\theta_t) \approx \langle \nabla_\theta L(\theta_t),\ \theta_{t+1} - \theta_t \rangle$$

$$= \sum_{i=0}^{K-1} (\nabla_\theta L(\theta_t))^{(i)} (\theta_{t+1}^{(i)} - \theta_t^{(i)})$$

Loss Change Allocation (Lan et al., 2019)

$$L(\theta_T) - L(\theta_0) = \int_C \langle \nabla_\theta L(\theta), d\theta \rangle$$

$$L(\theta_{t+1}) - L(\theta_t) \approx \langle \nabla_\theta L(\theta_t),\ \theta_{t+1} - \theta_t \rangle$$
$$= \sum_{i=0}^{K-1} (\nabla_\theta L(\theta_t))^{(i)} (\theta_{t+1}^{(i)} - \theta_t^{(i)})$$



Loss Change Allocation (Lan et al., 2019)

cumulative

The sum of all individual components equals the total change in loss

interpretable

At a given update a parameter having

- negative LCA => that parameter is beneficial
- positive LCA => that parameter is “hurting”

Loss Change Allocation (Lan et al., 2019)

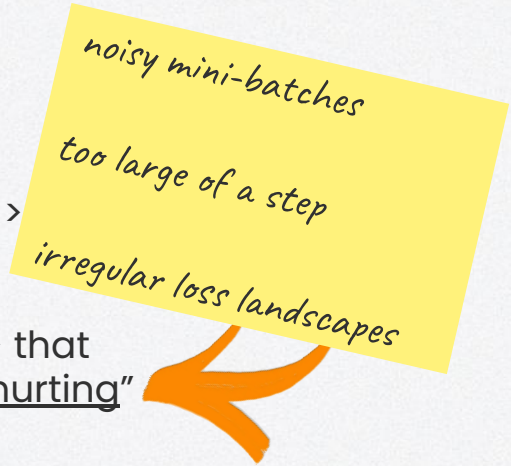
cumulative

The sum of all individual components equals the total change in loss

interpretable

At a given update a parameter having

- negative LCA => parameter is beneficial
- positive LCA => that parameter is "hurting"



Analysis

1

Layer-wise contributions

Aggregating over time, dissecting over transformer sub-layers

2

Dynamics over time

Aggregating sub-layers, decomposed over time

3

Attention-heads-wise

Aggregating over time

4

Parameters that hurt training

decomposed by layer over time

Crash recap: Trf-layers

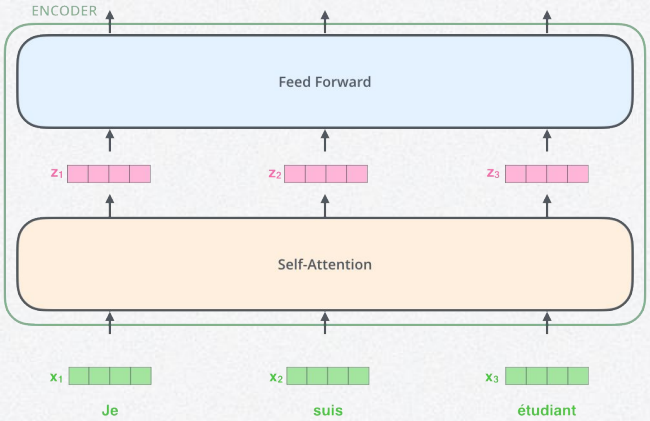


Figure: Transformer Layer

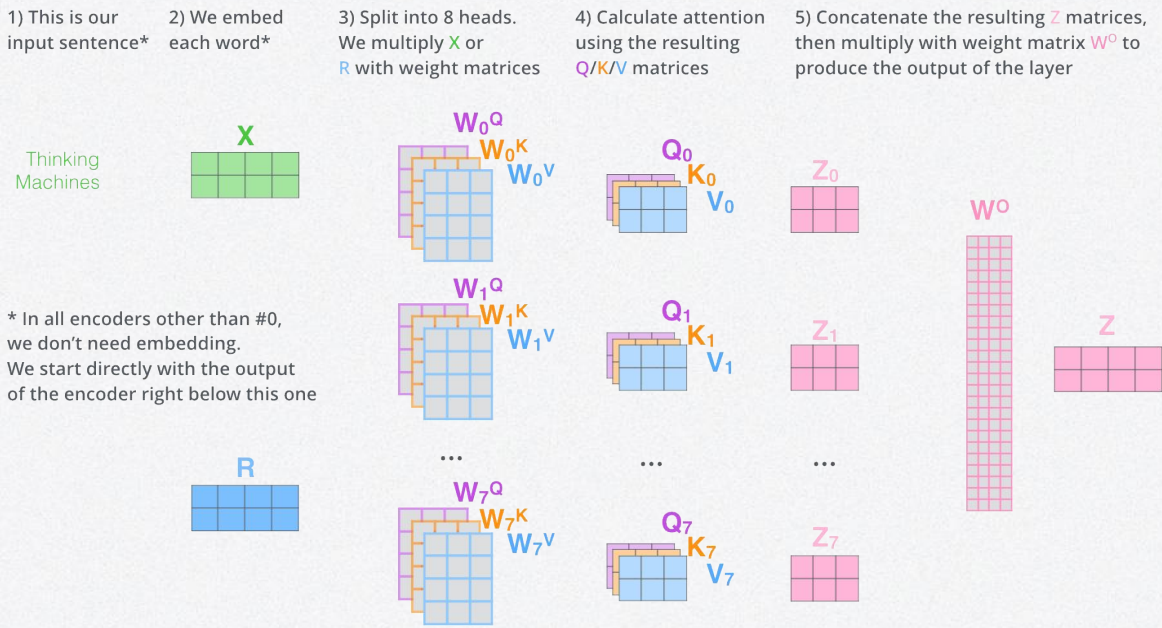


Figure: Overview of self-attention

**Illustrations taken from: Alammar, J (2018). The Illustrated Transformer [Blog post].*

Layer-wise contributions

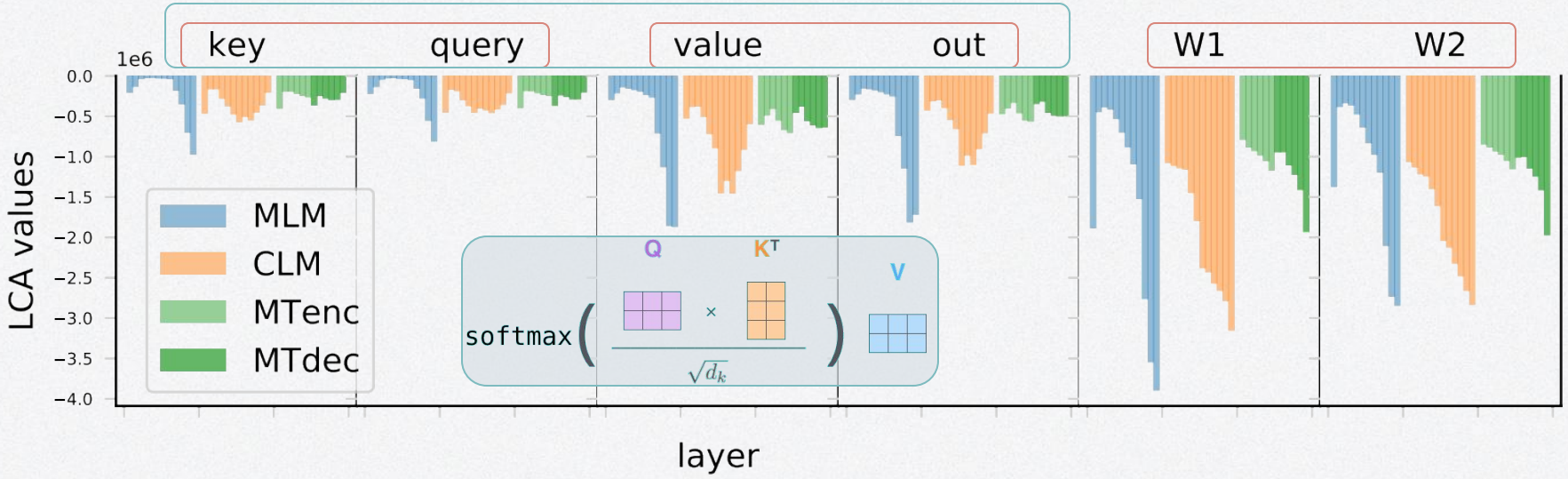
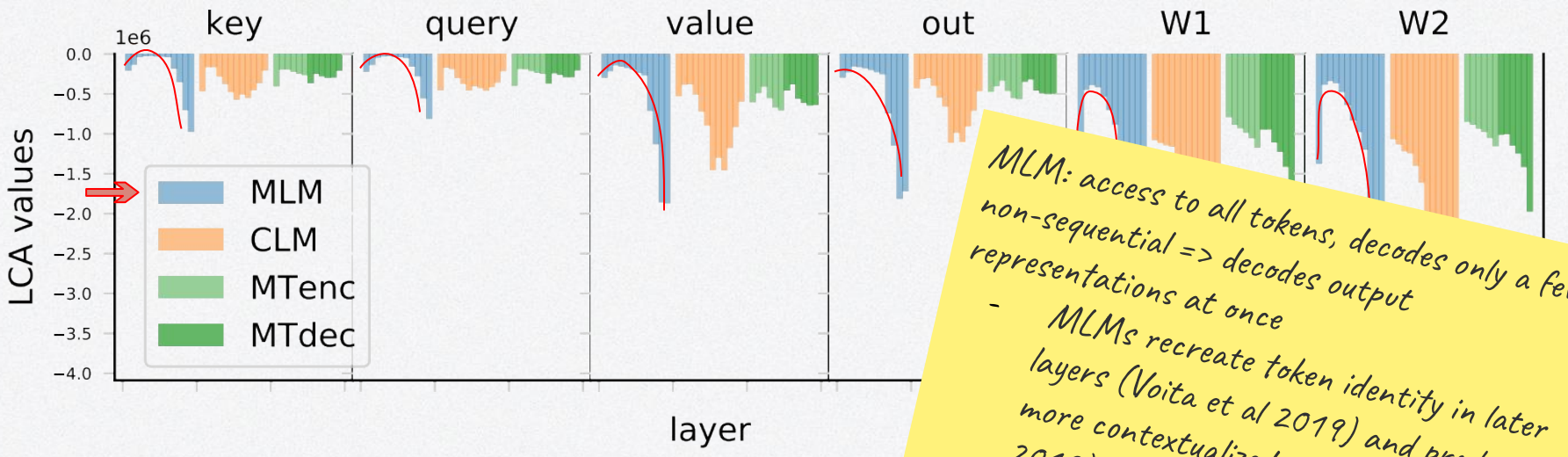


Figure: LCA aggregated over all training steps results for individual components of different layers. Each bar location in the x-axis denotes a separate layer.

Layer-wise contributions



MLM: access to all tokens, decodes only a few, non-sequential => decodes output representations at once

- MLMs recreate token identity in later layers (Voita et al 2019) and produce more contextualized ones (Ethayarajh 2019)

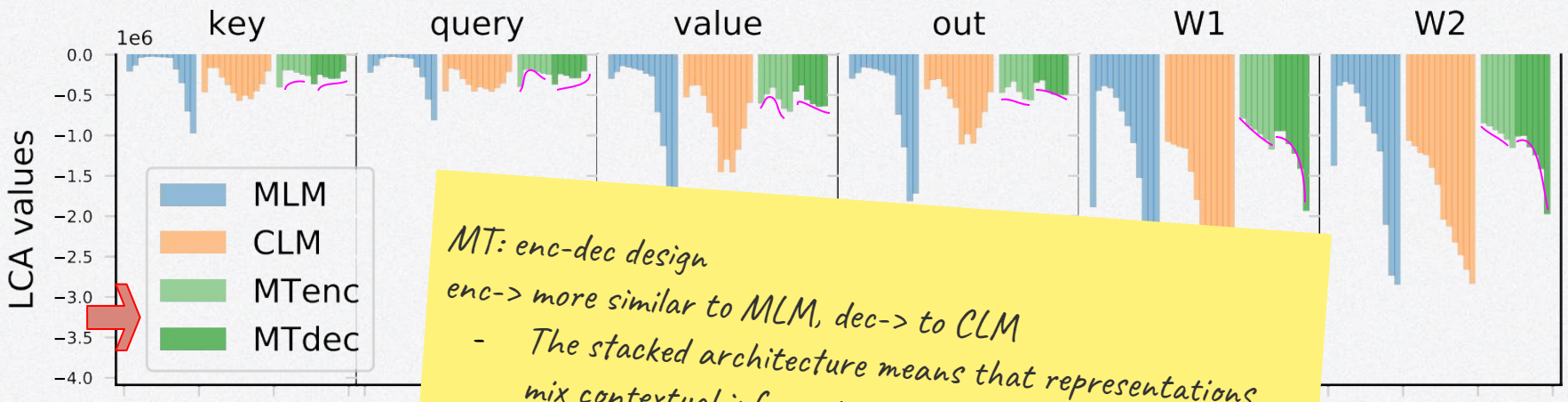
Figure: LCA aggregated over all training steps results for individual compon
Each bar location in the x-axis denotes a separate layer.

Layer-wise contributions

CLM: access to decoded tokens, decodes 1-by-1
=> incrementally build representations over layers
- Middle layers potentially help learning fine-grained syntactic structures (Tenney et al 2019), hence the attn-block behavior

Figure: LCA aggrega… …mponents of different layers. Each bar location in the x-axis…

Layer-wise contributions



MT: enc-dec design

enc-> more similar to MLM, dec-> to CLM

- The stacked architecture means that representations mix contextual information => FFWD peak at upper layers may come from this.
- MT is a composition of several sub-tasks (Voita et al. 2019)

Figure: LCA aggrega… …f different layers. Each bar location in t…

Analysis

1

Layer-wise contributions

Aggregating over time, dissecting over transformer sub-layers

2

Dynamics over time

Aggregating sub-layers, decomposed over time

3

Attention-heads-wise

Aggregating over time

4

Parameters that hurt training

decomposed by layer over time

Learning dynamics over time

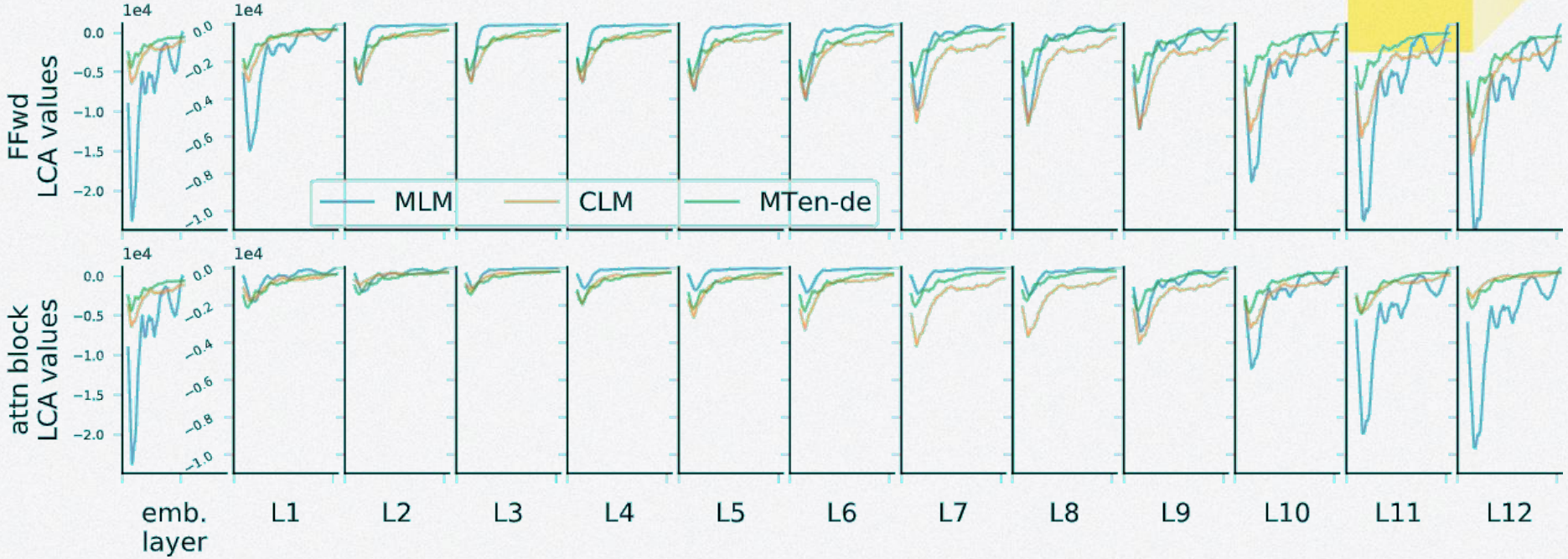


Figure: Training dynamics trends for the three learning objectives. The embeddings layer plot is the same on both rows, and it shows a different scale for visualization purposes.

Analysis

1

Layer-wise contributions

Aggregating over time, dissecting over transformer sub-layers

2

Dynamics over time

Aggregating sub-layers, decomposed over time

3

Attention-heads-wise

Aggregating over time

4

Parameters that hurt training

decomposed by layer over time

Attention heads training dynamics

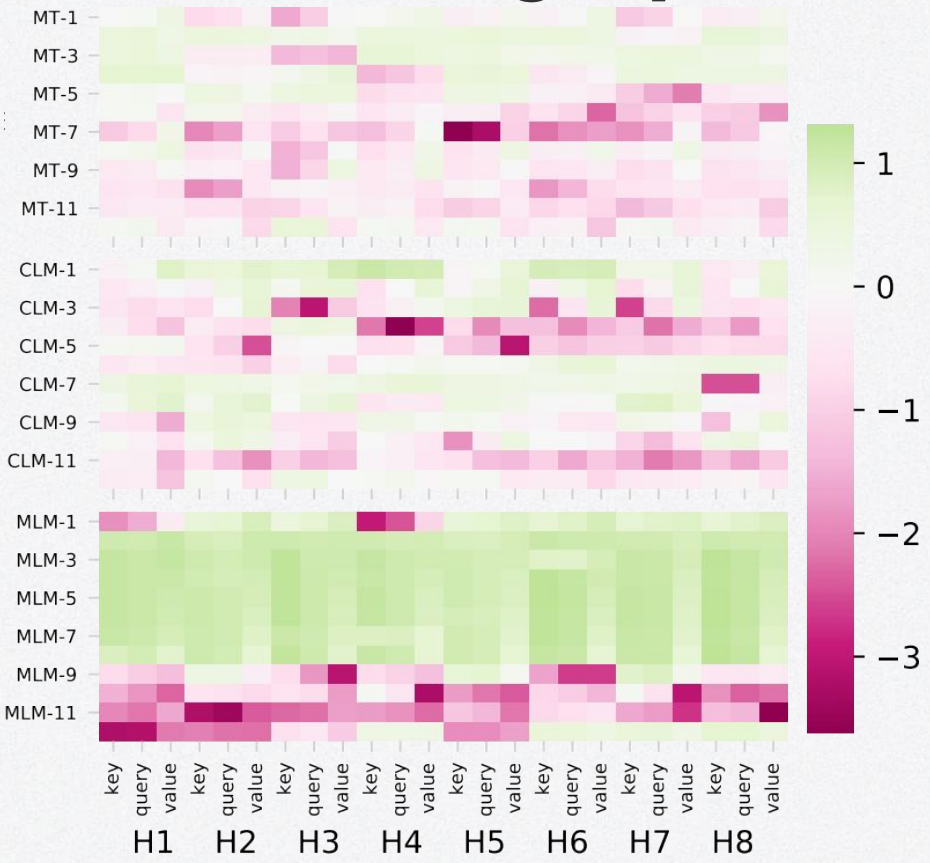
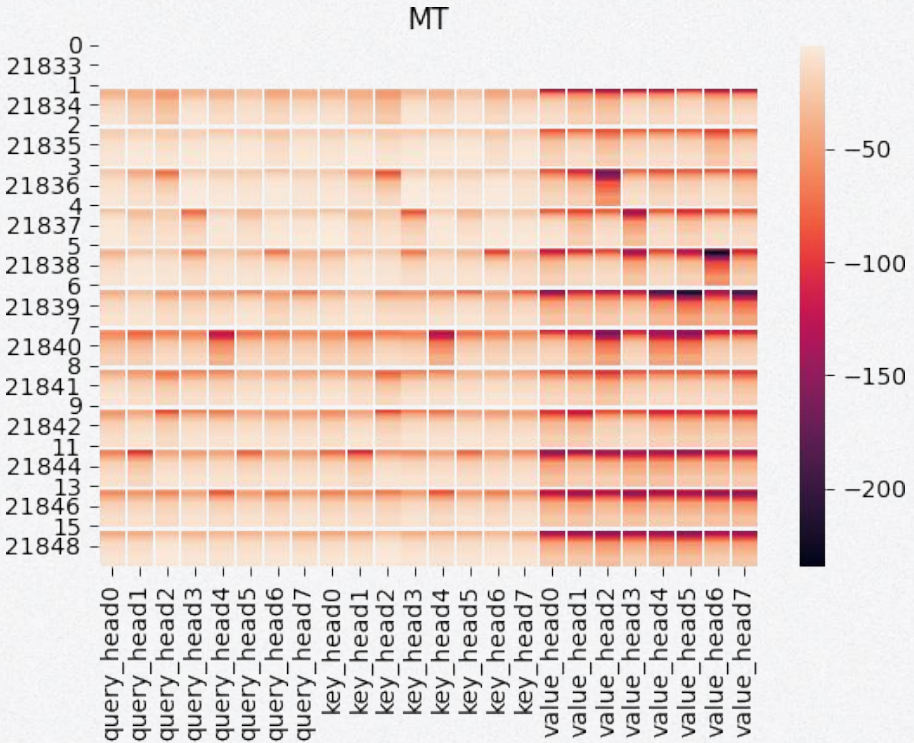


Figure: Contribution from individual attention heads, presented as normalized LCA values. Normalization is performed over the entire heatmap.

Attention heads training dynamics



Attention heads training dynamics

This “non-finding”:

- is in (apparent) contradiction with the idea of heads specialization (Voita et al., 2019)
- highlights the similarity in training dynamics across all attention heads
- supports the idea effective head pruning (Michel et al., 2019; Zhang et al., 2021)
- the apparent redundancy of AHs is the result of dropout (Clark et al., 2019)

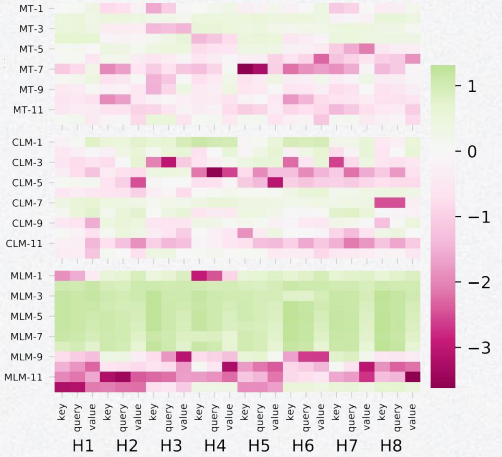


Figure: Contribution from individual attention heads, presented as normalized LCA values.

Analysis

1

Layer-wise contributions

Aggregating over time, dissecting over transformer sub-layers

2

Dynamics over time

Aggregating sub-layers, decomposed over time

3

Attention-heads-wise

Aggregating over time

4

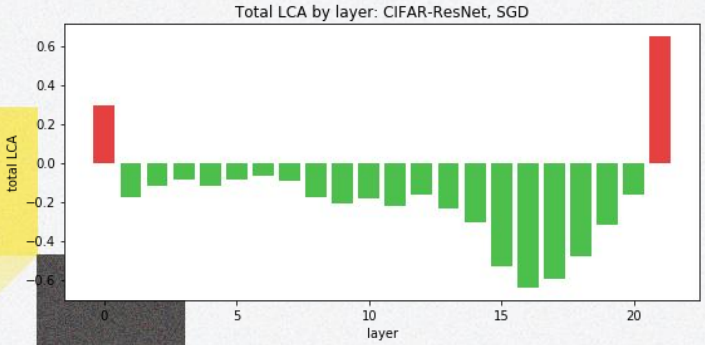
Parameters that hurt training

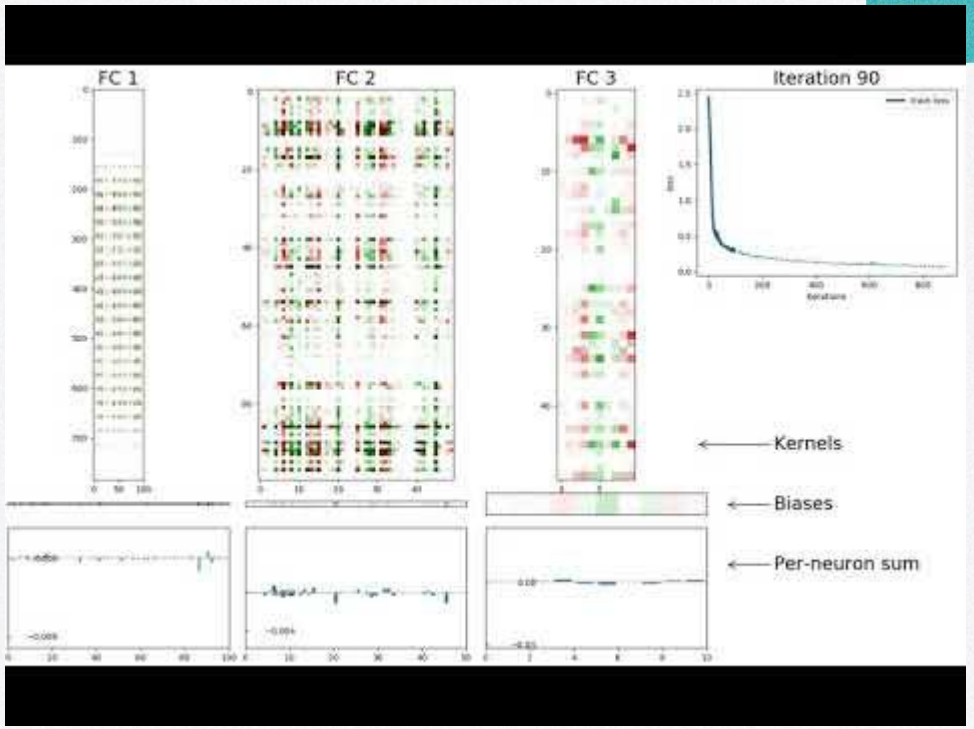
decomposed by layer over time

Parameters that hurt training

Training is fuzzy

Lan et al. (2019)





Parameters that hurt training

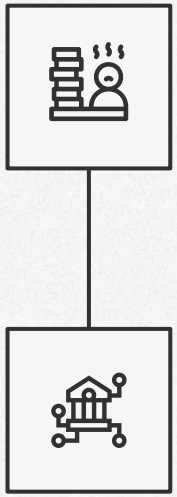
Training is fuzzy

Lan et al. (2019)

Parameters that hurt training

Training is fuzzy

Lan et al. (2019)

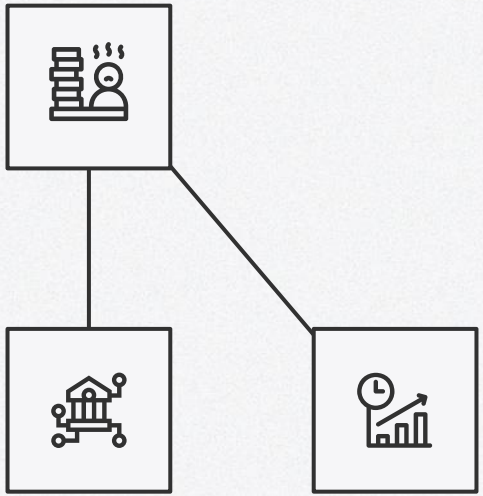


Is this also true for Trf-based models?

Parameters that hurt training

Training is fuzzy

Lan et al. (2019)



Is this also true for Trf-based models?

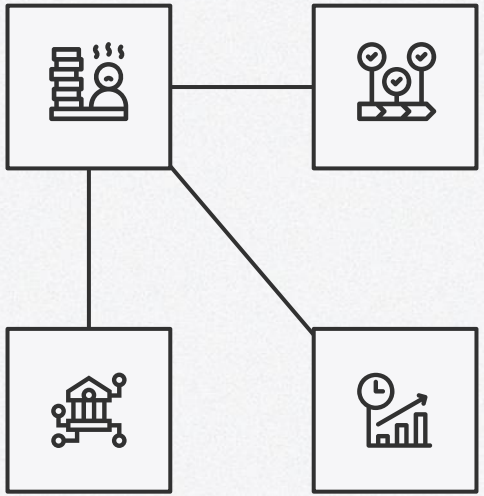
How does this change during the training process?

Parameters that hurt training

Training is fuzzy

Lan et al. (2019)

Is this also true for Trf-based models?



Consistency across learning objectives?

How does this change during the training process?

Parameters that hurt training

Is this also true for Trf-based models?

Consistency across learning objectives?

How does this change during the training process?

	CLM	MLM	MT
emb.	2	0	2
key	29	1565	64
query	24	2055	64
value	24	24	130
out	24	6	113
W1	24	0	59
W2	24	0	131
Last step w/positive	1072	59727	2673
Overall max.	5.58 (key)	68.18 (value)	95 (W2)

Table: Summary of parameters that hurt the training process.
- number of times a block of parameters has positive LCA
- the last training step where this happens
- the magnitude of the param. with the highest positive LCA

Parameters that hurt training

How does this change during the training process?

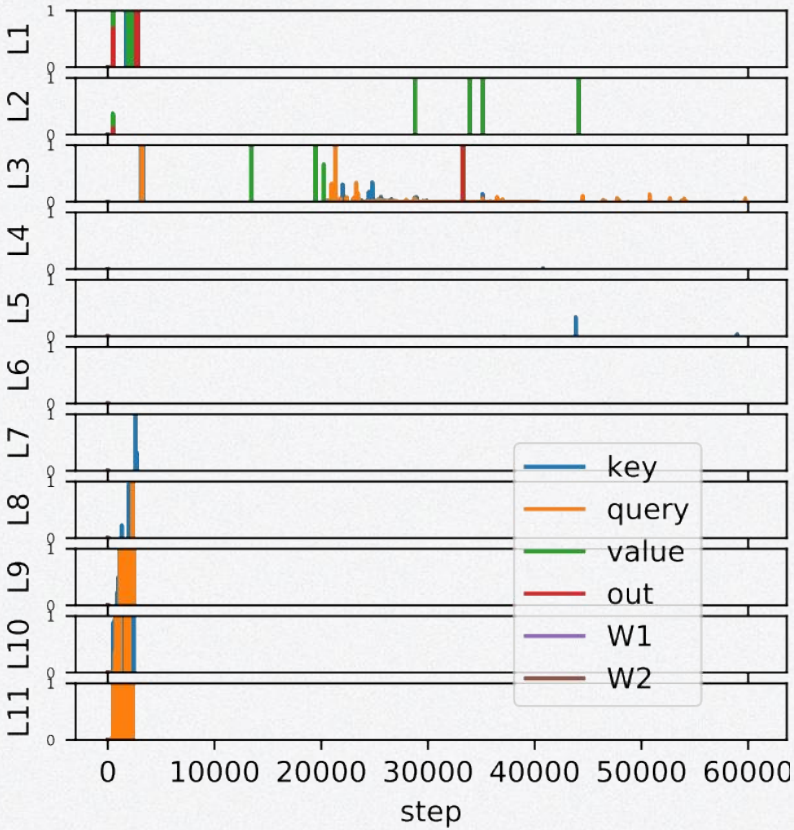


Figure: An overview of the **MLM** training process. To keep watch on the positive LCA we only show the interval [0,1].

Analysis

1

Layer-wise contributions

Are all these valid across domains/languages/ varying sizes?

4

Parameters that hurt training

decomposed by layer over time

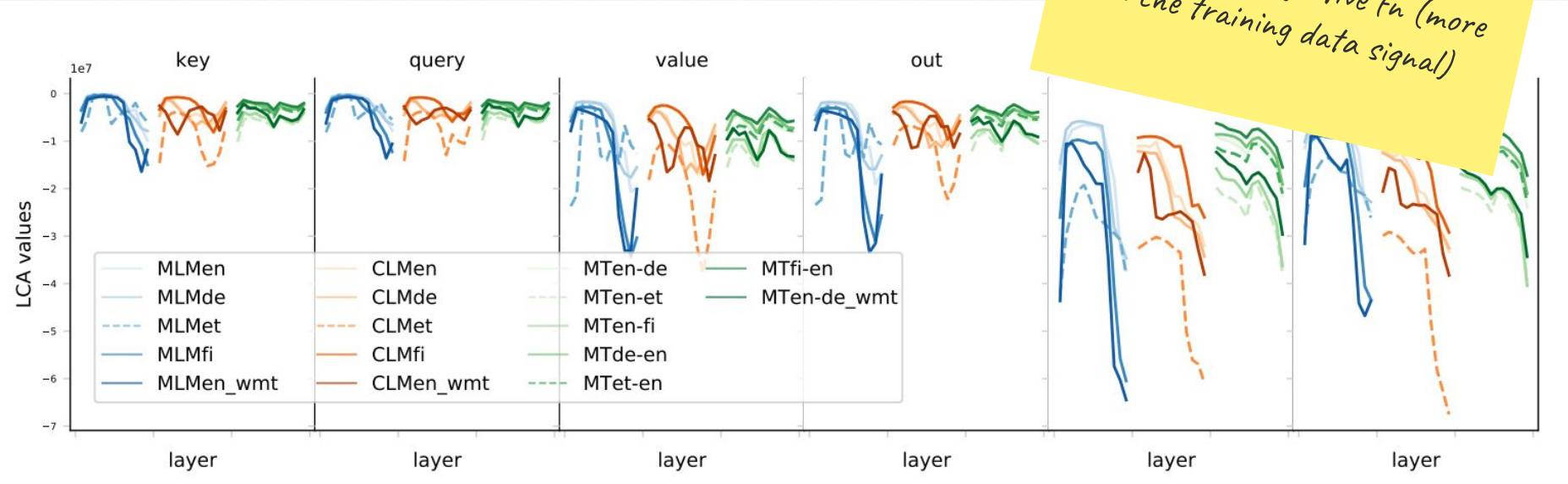
Effects of data size and distribution

Dataset	Languages	# Sentences		
		Train	Val.	Test
Europarl	En-De	1.92M	2.5K	2.5K
	En-Fi	1.91M	2.5K	2.5K
	En-Et	0.7M	2.5K	2.5K
WMT-2018	En-De	5M	3K	-
newstest2018	En-De	-	-	2.9K
	En-Fi	-	-	3.0K
	En-Et	-	-	2.0K

Table: We replicated the experiments using training data of different sizes (700k, 2M and 5M), domains (Europarl and WMT) and languages (English, German, Finnish and Estonian). Hyperparameters are kept constant.

Effects of data size and distribution

Observed stability => contributions to optimization determined by objective fn (more than the training data signal)



Outline

0 Motivation

1 Encoder Spaces

2 Training Dynamics

3 **Next Steps**

4 Conclusions

Next Steps

Conclusions

Analyzing models => better use resources

Despite model similarities ∃ remarkable discrepancies, stemming from the learning objectives

The learning objectives seem to guide the learning process more than signals in training data

Out-of-the-box BERT performs poorly in MT due to the differences in embeddings spaces

HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

Thanks!

Do you have any questions?

raul.vazquez@helsinki.fi

FOTRAN
Found in Translation