LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

# Attention-Driven Dropout

## A Simple Method to Improve Self-supervised Contrastive Sentence Embeddings

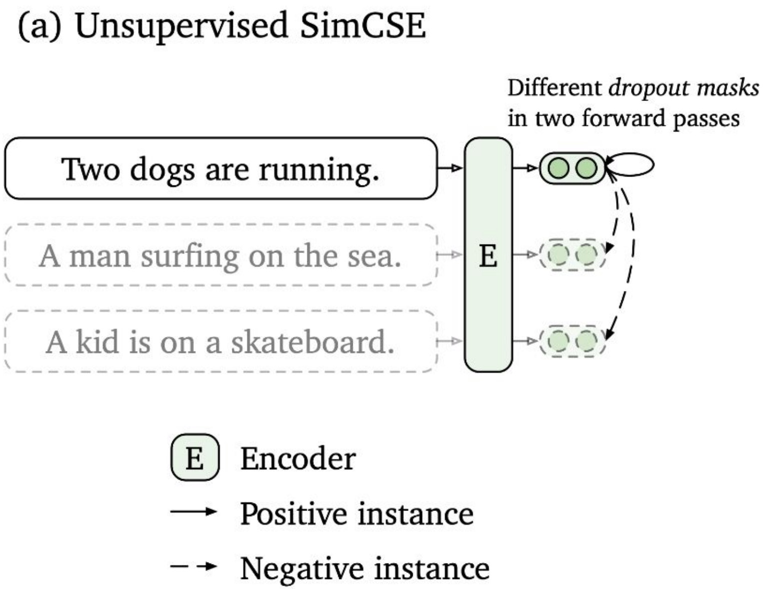

**Fabian Stermann[1,2], Ilias Chalkidis[4], Amihossein Vahidi[1,3], Bernd Bischl[1,3], Mina Rezaei[1,3]**

1. Department of Statistics, LMU Munich, Munich, Germany
2. Convalid Analytics, Munich, Germany
3. Munich Center for Machine Learning, Munich, Germany
4. Department of Computer Science, University of Copenhagen, Denmark

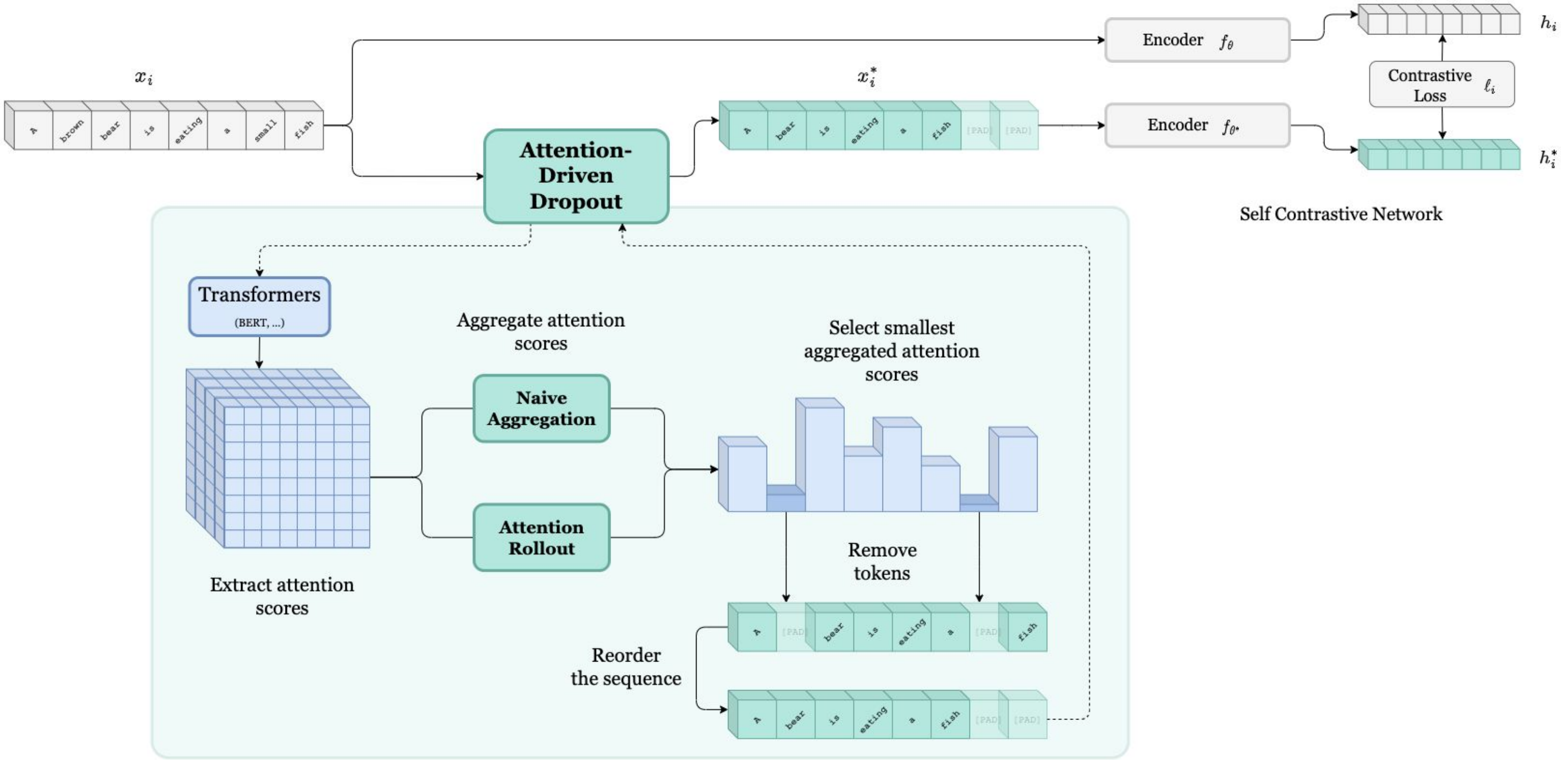
# Background Contrastive Learning / SimCSE

- **Contrastive Learning**
  - **Positive** examples: similar sentences
  - **Negative** examples: dissimilar sentences
  - Objective
    - Minimize distance of positive examples
    - Maximize distance of negative examples
- ***Unsupervised SimCSE*** (Gao et al. 2021)
  - Positive instances simply have different dropout masks
  - Uses in-batch samples as negatives

$$\ell_{(x_i,x_j)} = -log \frac{\exp(\mathrm{sim}(\boldsymbol{h}_i, \boldsymbol{h}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{I}_{[k \neq i]} \exp(\mathrm{sim}(\boldsymbol{h}_i, \boldsymbol{h}_k)/\tau)}$$



# Method


$x_i$
A
brown
bear
is
eating
a
small
fish
Attention-Driven Dropout
$x_i^*$
[PAD]
Encoder $f_\theta$
Encoder $f_{\theta^*}$
$h_i$
$h_i^*$
Contrastive Loss $\ell_i$
Self Contrastive Network
Transformers
(BERT, ...)
Extract attention scores
Aggregate attention scores
Naive Aggregation
Attention Rollout
Select smallest aggregated attention scores
Remove tokens
Reorder the sequence

# Aggregation Methods

- **Naive Aggregation**
  - Summation across layers and heads
  - “Attentiveness” of tokens to specific tokens

$$a_i = \sum_{l=1}^{L} \sum_{h=1}^{H} \sum_{s=1}^{S_1} A_{lihs}$$

- **Rollout Aggregation**
  - *Abnar and Zuidema (2020)*
  - Approximate Attention to input tokens

$$\widetilde{A}(l) = \begin{cases} A(l)\widetilde{A}(l^-) & \text{if } l > l^- \\ A(l) & \text{if } l = l^- \end{cases}$$

$$a_i = \sum_{l=1}^{L} \sum_{s=1}^{S_1} \widetilde{A}_{lis}$$

# Naive Aggregation Example

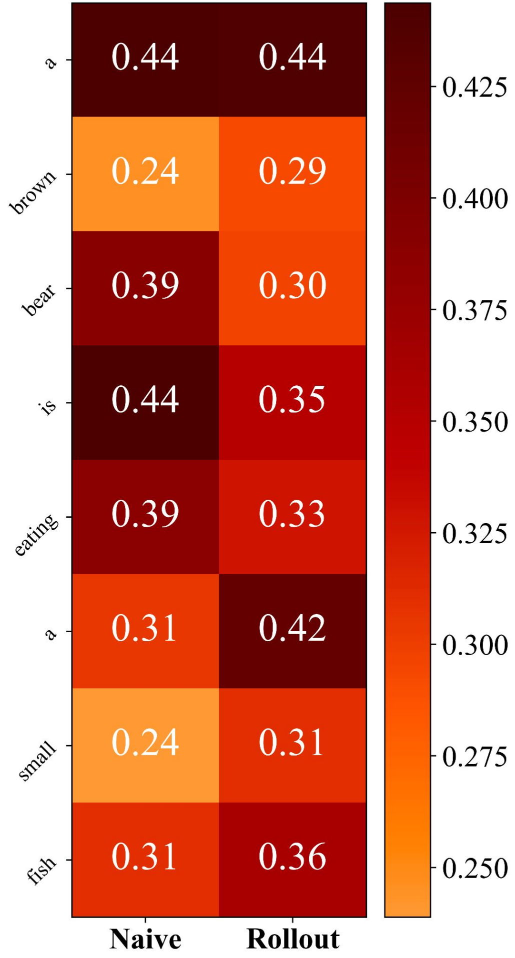
0: A brown bear is eating a small fish.

1: A brown bear is eating a ~~small~~ fish.

2: A ~~brown~~ bear is eating a ~~small~~ fish.

3: A ~~brown~~ bear is eating ~~a~~ ~~small~~ fish.

4: A ~~brown~~ bear is eating ~~a~~ ~~small~~ ~~fish~~.



# Dropout Rates

## Static

- Use a predefined constant $k$ as the dropout rate.
- Fixed token removal across sequences, regardless of length.

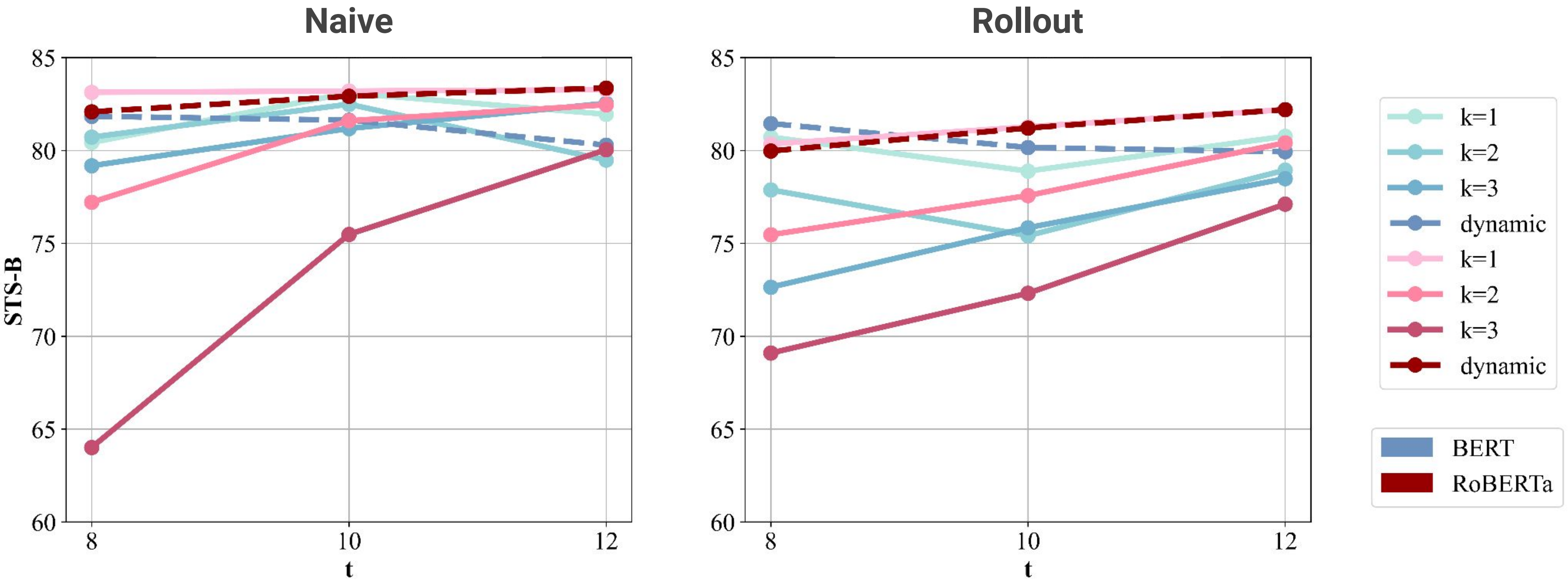
## Dynamic

- Remove more redundant information in long sequence.
- $k$ is calculated dynamically based on non-padding tokens.
- For sequence $x_i^+$ with $t_s$ tokens:

$$k = \left\lfloor \frac{t_s}{t} \right\rfloor$$

- Select $g_i = imin(a_i, k) \in \mathbb{R}^k$, the indices of $k$ lowest attention scores.
- Replace $x_{ij}^+$ with padding for $j \in g_i$, forming $x_i^*$.
- Reorder $x_i^*$ with padding tokens aligned to the right.

# Hyperparameters




**k** Number of tokens to drop

**t** Minimum amount of tokens in order to drop tokens

**Dynamic** Set k based on sequence length

# Hyperparameters

- Base models: BERT, RoBERTa
- 1 epoch

- Learning rate: {3e-5, 1e-5}
- Batch size: {64, 128, 256, 512}

- **k** {1, 2, 3}
- **t** {8, 10, 12}

*static*

- *dynamic*

| | | | | STS | | Transfer | |
|---|---|---|---|---|---|---|---|
| **PLM** | **AA** | **k** | **t** | **BS** | **LR** | **BS** | **LR** |
| BERT | naive | 1 | 10 | 64 | 3e-5 | 64 | 1e-5 |
| | rollout | dynamic | 8 | 64 | 3e-5 | 512 | 1e-5 |
| RoBERTa | naive | dynamic | 12 | 64 | 1e-5 | 256 | 1e-5 |
| | rollout | dynamic | 12 | 64 | 1e-5 | 128 | 1e-5 |

# Results Semantic Textual Similarity

| Model | STS12 | STS13 | STS14 | STS15 | STS16 | STS-B | SICK-R | Avg. |
|---|---|---|---|---|---|---|---|---|
| SimCSE-$\text{BERT}_{\texttt{base}}$ ♡ | *68.40* | **82.41** | *74.38* | *80.91* | *78.56* | *76.85* | **72.23** | *76.25* |
| + $\text{ADD}_{\texttt{naive}}$ | **71.00** | *82.24* | **75.10** | **82.73** | **79.03** | **78.51** | *72.12* | **77.25** |
| + $\text{ADD}_{\texttt{rollout}}$ | 65.20 | 77.98 | 71.26 | 80.62 | 77.27 | 76.26 | 69.68 | 74.04 |
| SimCSE-$\text{RoBERTa}_{\texttt{base}}$ ♡ | **70.16** | *81.77* | *73.24* | *81.36* | *80.65* | *80.22* | 68.56 | *76.57* |
| + $\text{ADD}_{\texttt{naive}}$ | *67.45* | **83.43** | **74.67** | **82.48** | **81.69** | **82.00** | **70.43** | **77.45** |
| + $\text{ADD}_{\texttt{rollout}}$ | 65.34 | 80.97 | 71.29 | 81.08 | 80.34 | 79.83 | *69.54* | 75.48 |

STS task performance for sentence embeddings (Spearman's correlation, "all" setting).
The best performance for the corresponding task is marked in bold, the second best is in italics.
♡: results from Gao et al. (2021); other results are evaluated by us.

# Results Transfer Learning

| Model | MR | CR | SUBJ | MPQA | SST | TREC | MRPC | Avg. |
|---|---|---|---|---|---|---|---|---|
| SimCSE-BERT$_{base}$ ♡ | 81.18 | 86.46 | 94.45 | 88.88 | 85.50 | 89.80 | 74.43 | 85.81 |
| + MLM ♡ | **82.92** | 87.23 | **95.71** | 88.73 | *86.81* | 87.01 | **78.07** | *86.64* |
| **+ ADD$_{naive}$** | 81.82 | 86.89 | 94.83 | **89.43** | 85.28 | 89.40 | 75.25 | 86.13 |
| + MLM | 82.18 | *87.74* | *95.66* | 88.16 | 86.55 | *91.00* | 75.07 | 86.62 |
| **+ ADD$_{rollout}$** | 81.41 | 85.72 | 94.79 | 89.32 | 84.84 | 88.60 | 75.07 | 85.68 |
| + MLM | *82.40* | **87.97** | 95.62 | *89.38* | **86.93** | **91.20** | *75.59* | **87.01** |
| SimCSE-RoBERTa$_{base}$ ♡ | 81.04 | 87.74 | 93.28 | 86.94 | 86.60 | 84.60 | 73.68 | 84.84 |
| + MLM ♡ | 83.37 | 87.76 | **95.05** | 87.16 | 89.02 | *90.80* | 75.13 | 86.90 |
| **+ ADD$_{naive}$** | 82.30 | 88.05 | 93.70 | 87.50 | 88.25 | 84.60 | 74.84 | 85.61 |
| + MLM | *83.86* | *89.06* | 94.65 | *88.27* | *89.51* | 90.60 | *76.75* | *87.53* |
| **+ ADD$_{rollout}$** | 82.08 | 88.40 | 93.13 | 87.54 | 87.97 | 87.00 | 75.88 | 86.00 |
| + MLM | **84.68** | **89.91** | *94.97* | **88.37** | **90.61** | **92.20** | **78.43** | **88.45** |

Transfer task performance for sentence embeddings, measures represent accuracy.
The best performance for the corresponding task is marked in bold, the second best is in italics.
♡: results from Gao et al. (2021); other results are evaluated by us.
MLM: MLM is added as an auxiliary task with λ = 0.1.

# Conclusion

- Quantification of token-relevance by simple Attention aggregation
- Overall improvement of performance
- Applicable to any self-contrastive network

## Future Directions

- Explainability in Language Models
  - Identify important words/tokens in a sentence
  - Reducing sentences to essential tokens

# Outro

Thank you for your $\text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$ !

## Find the paper on

github.com/fstermann/attention-driven-dropout

## Find me on

*github.com/fstermann*

*linkedin.com/in/fstermann*

*fabian.stermann@convalid-ai.de*