# Injecting Knowledge into Sentence Embedding Models for Information Retrieval using Adapters

Final Presentation

23.05.2023

Chair of Software Engineering for Business Information Systems (sebis)
Department of Computer Science
School of Computation, Information and Technology (CIT)
Technical University of Munich (TUM)
wwwmatthes.in.tum.de

# Outline

Motivation

Adapters

Problem Statement

Suitability of Adapters on STS-Tasks

Domain Adaptation on STS-Tasks

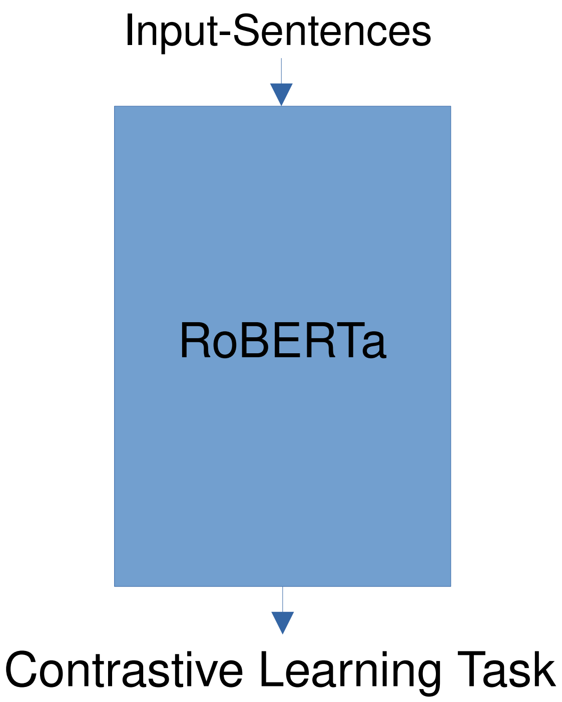
Unsupervised improvement through STS-Adapters

Conclusion

# Motivation – Sentence Embedding Models

- Similarity comparison of sentences/short texts
- Important for multiple downstream tasks
- SBERT fails to capture rich factual knowledge [1]

  → Injecting factual knowledge from Knowledge Graphs



# Motivation – Adapters

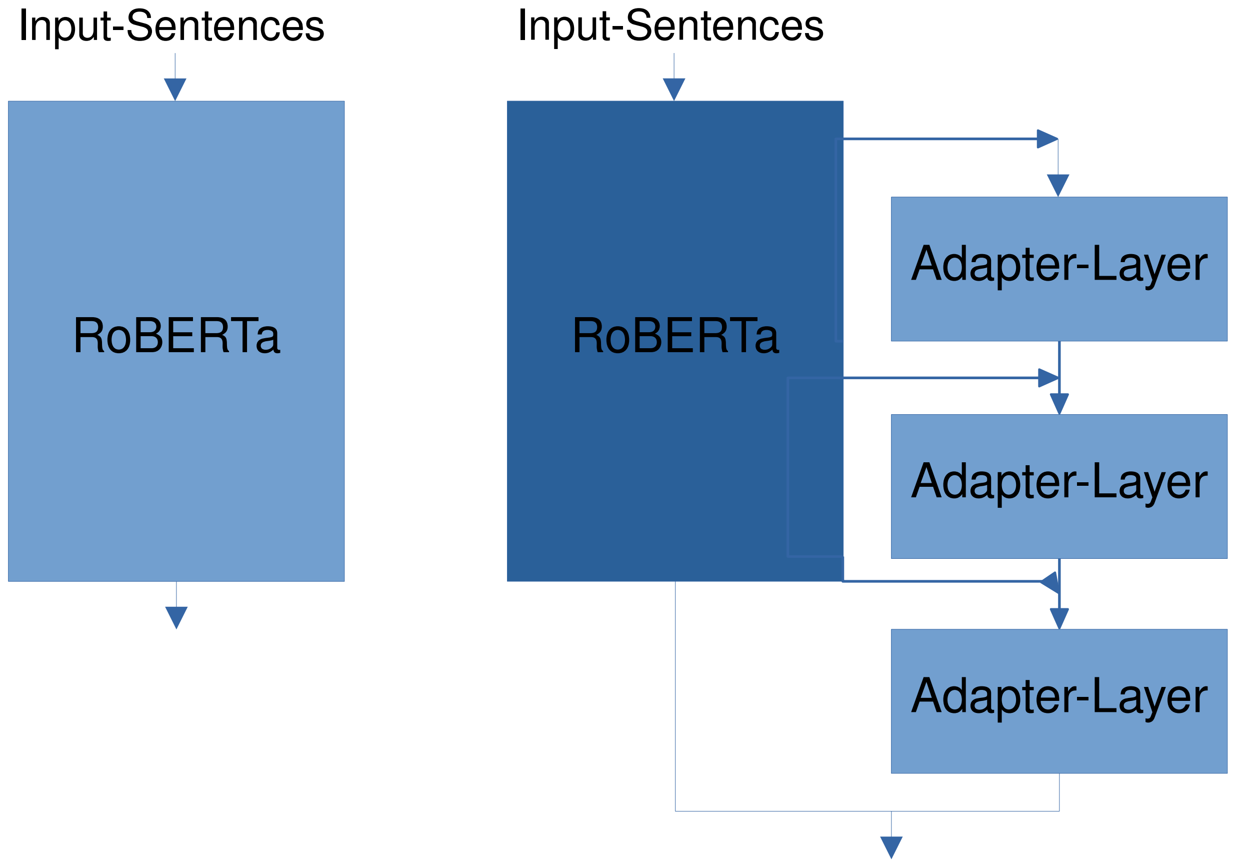
- Training large LMs is resource-intensive

  → Adapters can alleviate this problem

- Pretrained once, „Plug'n'Play" for different use cases

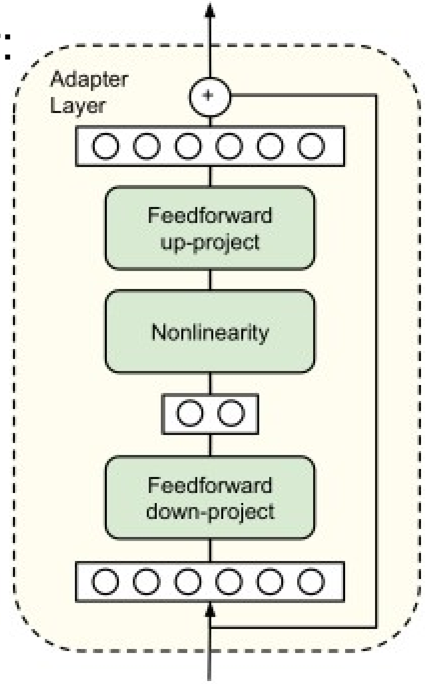
# Adapters

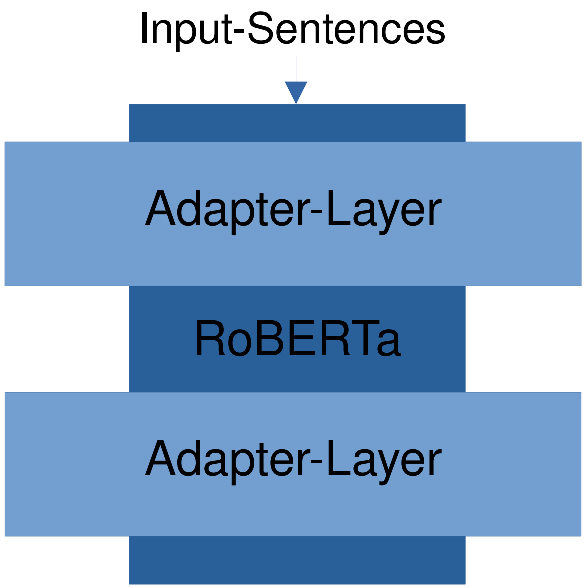
- Freeze large basemodel
- Add trainable Adapter-Layers at multiple layers of the basemodel
  → Leverage learned knowledge from the basemodel
- Train with less parameters (3.6%), finetune basemodel-behaviour on different task



# Adapter-Architectures - Houlsby

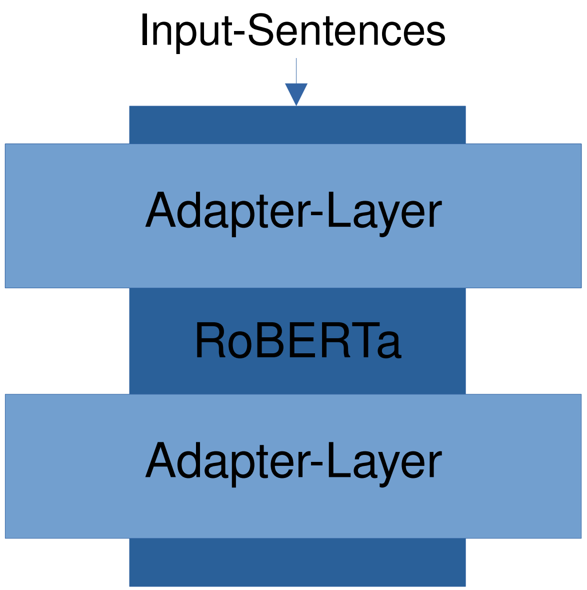
- First Adapter to be developed
- Focus: „Catastrophical Forgetting“
- Initialized to the Identity-Function
- 3.6% of Parameters
- Bottleneck-Adapter:



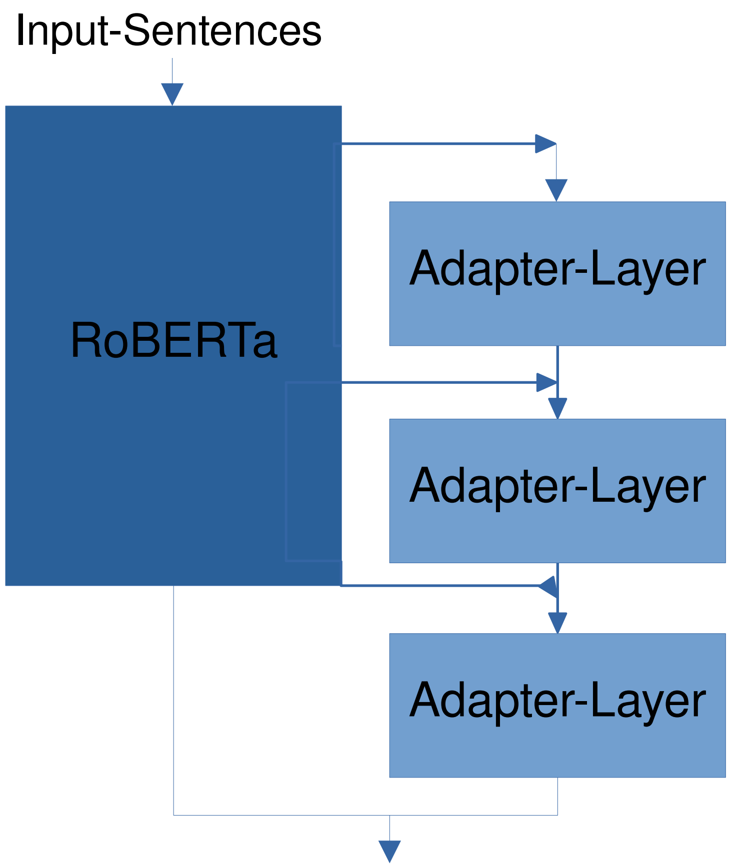
# Adapter-Architectures - Pfeiffer

- Focus: „Merging of Adapters“
- 9% of Parameters



# Adapter-Architectures - K-Adapter

- Very recent paper
- Focus: „Combining knowledge from Adapters“
- Initialized Randomly
- 13-40% of Parameters



# Problem Statement

- Sentence Embedding Models fail to encode factual knowledge [1]
- Enhance Sentence Embedding Models by injecting Knowledge through Adapters
  - Inject structured knowledge from Knowledge Graphs

  → RQ1: How to inject structured knowledge into SE-Models using Adapters?

  → RQ2: Do Knowledge Adapters improve information retrieval tasks of Sentence Embedding Models?
- Domain-specific data is scarce. How can we approach domain-adaptation through Adapters?
  → RQ3: How to combine domain-specific knowledge adapters for the scholarly domain?

# Suitability of Adapters for STS-Tasks

- Basemodel: roberta-large
- Finetuning Adapters on Common-World TREx-rc dataset
- Contrastive Loss
- Evaluated against 7 STS-Tasks (Spearman Correlation)

| Model | STS12 | STS13 | STS14 | STS15 | STS16 | STS-B. | SICK-R. | Avg. |
|---|---|---|---|---|---|---|---|---|
| Houlsby-Adapter | 76.69 | 86.87 | 82.18 | 86.30 | 84.14 | **86.87** | 79.59 | 83.23 |
| Pfeiffer-Adapter | **77.93** | 87.00 | **82.60** | **87.31** | 83.50 | 86.74 | **80.92** | **83.71** |
| K-Adapter | 76.00 | 86.93 | 81.28 | 86.50 | 83.76 | 86.23 | 80.08 | 82.97 |
| Finetuned roberta-large | 77.87 | **87.24** | 82.56 | 87.17 | **84.62** | 86.26 | 79.93 | 83.68 |

# Domain-Adaptation in STS-Tasks

- Enhance generic STS-basemodel with domain-specific Adapter
- Basemodel: princeton-nlp/sup_simcse_bert_base (Pretrained on a common-world NLI-dataset)
- Training on 2 domain-specific datasets: AskUbuntu, SciDocs
- Usage of 2 loss-functions

$$\ell_1 = \max\{(d(h_i, h_i^+) - d(h_i, h_i^-) + m), 0\}$$

$$\ell_2 = -\log \frac{e^{sim(h_i, h_i^+)/\tau}}{\sum_{j=1}^{N}(e^{sim(h_i, h_j^+)/\tau} + e^{sim(h_i, h_j^-)/\tau})}$$

# Domain-Adaptation on STS-Tasks

- Evaluation on 90%/10% training sets
- Houlsby Adapter performs the best

| Datasets → Models ↓ | | AskUbuntu | SciDocs Cite | CC | CR | CV | Average |
|---|---|---|---|---|---|---|---|
| *Out-of-the-box* SimCSE *(lower bound)* | | 60.3 | 79.3 | 82.10 | 76.87 | 78.36 | 75.39 |
| $\ell_1$ | Houlsby-Adapter | 64.0 | **88.2** | 88.69 | **82.42** | 83.99 | 81.46 |
| | Pfeiffer-Adapter | 63.8 | 87.8 | 88.73 | 81.65 | 83.27 | 81.05 |
| | K-Adapter | 62.5 | 85.6 | 87.70 | 80.09 | 82.85 | 79.75 |
| | *In-domain supervised* SimCSE *(upper bound)* | 65.3 | 88.0 | 87.74 | 84.15 | 83.32 | 81.70 |
| $\ell_2$ | Houlsby-Adapter | **64.5** | 87.3 | **89.01** | 82.41 | **84.42** | **81.53** |
| | Pfeiffer-Adapter | 64.2 | 87.0 | 88.63 | 81.98 | 84.41 | 81.24 |
| | K-Adapter | 62.8 | 85.3 | 87.92 | 80.05 | 83.29 | 79.87 |
| | *In-domain supervised* SimCSE *(upper bound)* | 65.2 | 88.3 | 88.11 | 84.46 | 83.63 | 81.94 |

# Unsupervised improvement through STS-Adapters

- Unsupervised Learning on TREx-rc
- Initialization using frozen supervised pretrained Adapters

| Model | STS12 | STS13 | STS14 | STS15 | STS16 | STS-B. | SICK-R. | Avg. |
|---|---|---|---|---|---|---|---|---|
| Unsupervised | 71.27 | 83.75 | 75.26 | 85.04 | 81.17 | 81.69 | 70.84 | 78.43 |
| Houlsby-Adapter | **73.37** | **84.89** | **76.21** | **87.24** | **83.17** | **81.75** | **72.63** | **79.90** |
| Pfeiffer-Adapter | 72.49 | 83.62 | 76.14 | 85.54 | 82.63 | 81.55 | 72.43 | 79.20 |
| K-Adapter | 72.47 | 83.55 | 75.20 | 85.09 | 82.42 | 81.64 | 71.80 | 78.88 |

# Conclusion

- RQ1: How to inject structured knowledge into SE-Models with Adapters?
    - Knowledge from KGs has successfully been learned.
    - Data-triples have been learned by a constrastive approach.
    - Very similar performance to finetuning the entire model.
    - Results obtained with two different loss-functions.

# Conclusion

- RQ2: Do Knowledge Adapters improve information retrieval tasks of Sentence Embedding Models?
  - Similar performance compared to finetuning the entire basemodel.
    - Less Parameters used.
  - Improvement of unsupervised methods by initializing with supervised Adapters.

    → Improvement by introducing a cheap way to finetune a SE-model to new data and new domains.
  - No architecture was found that consistently outperforms the others.

# Conclusion

- RQ3: How to combine domain-specific knowledge adapters for the scholarly domain?
  - Combination has not been looked into.
  - Domain-Adaptation has been assessed over 2 datasets.
  - Successful domain-adaptation.
  - Very similar performance to the finetuning of the entire basemodel. But slightly worse!

    → No evidence of combining knowledge of the basemodel and the Adapter.
  - Houlsby- and Pfeiffer-Adapter performed very similarly.

# Thank you for your attention

- Any questions?

# References


- [1] „K-Adapter: Infusing Knowledge into Pre-Trained Models with Adapters“. R. Wang, D. Tang, N. Duan, X. Huang, J. Ji, G. Cao, D. Jiang, M. Zhou, 2021
- [2] „Parameter-Efficient Transfer Learning for NLP“. N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. de Laroussilhe, A. Gesmundo, M. Attariyan, S. Gelly, 2019
- [3] „AdapterFusion: Non-Destristive Task Composition for Transfer Learning“. J. Pfeiffer, A. Kamath, A. Rueckle, K. Cho, I. Gurevych, 2020
- [4] „Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks“. N. Reimers, I. Gurevych, 2019
- [5] „SimCSE: Simple Contrastive Learning of Sentence Embeddings“. T. Gao, X. Yao, D.Chen, 2021
- [6] „SPECTER: Document-level Representation Learning using Citation-informed Transformers“. A. Cohan, S. Feldman, I.Beltagy, D. Downey, D. S. Weld, 2020
- [7] „T-REx: A Large Scale Alginment of Natural Language with Knowledge Base Triples“. H. El Sahar, P. Vougiouklis, A. Remaci, C. Gravier, J. S. Hare, F. Laforest, E. Simperl, 2018