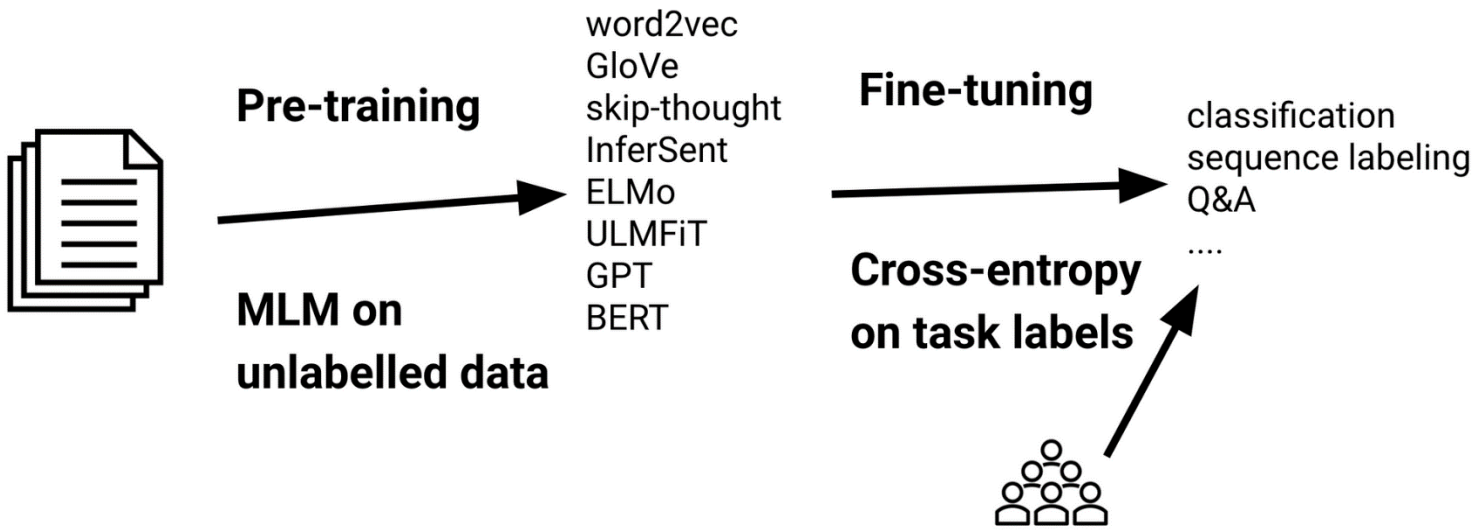
# Recent Advances in Language Model Fine-tuning

Author: SEBASTIAN RUDER

# Overview

- Fine-tuning a pretrained language model (e.g., BERT, mBERT, XLM-R) has become a new norm for training a model on an NLP downstream task.




*Source: https://ruder.io/recent-advances-lm-fine-tuning/*

# Overview

- Methods discussed in this presentation:

| Category | Methods | Motivation |
|---|---|---|
| **Adaptive Fine-tuning** | Domain/task/ language adaptive fine-tuning | Specialise to target domain |
| **Behavioural Fine-tuning** | Intermediate-task training, self-supervised, frame as MLM | Specialise to target task |
| **Parameter-efficient Fine-tuning** | Adapters, sparse parameter permutations, pruning | Reduce space of fine-tuned models |
| **Text-to-text Fine-tuning** | Frame as text-to-text, prompt engineering, controllable NLG | Effectively use large autoregressive pre-trained LMs |
| **Mitigating Fine-tuning Instabilities** | Stop runs early, use a small lr, regularisation, avoid random init | Reduce variance of fine-tuning runs |

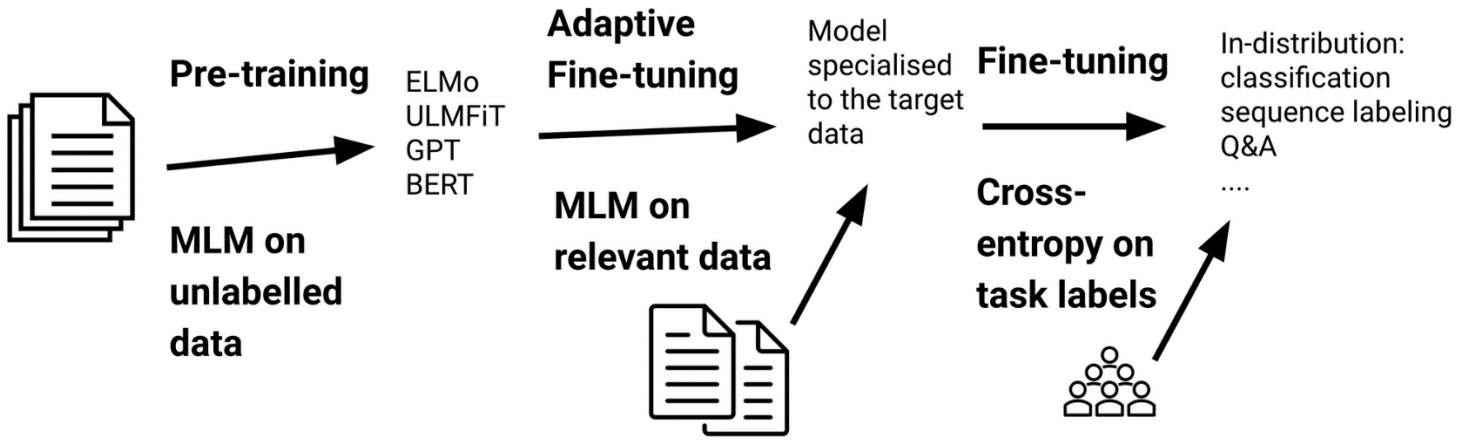
*Source: https://ruder.io/recent-advances-lm-fine-tuning/*

# Adaptive fine-tuning

- Pretrained language models can get in trouble with downstream data that is very different from the pretraining data in terms of .
- Adaptive fine-tuning is a way to bridge such a shift in distribution by fine-tuning the model on relevant unlabeled data:

  - Domain-adaptive: Unlabeled data is close to the training data of the downstream task.

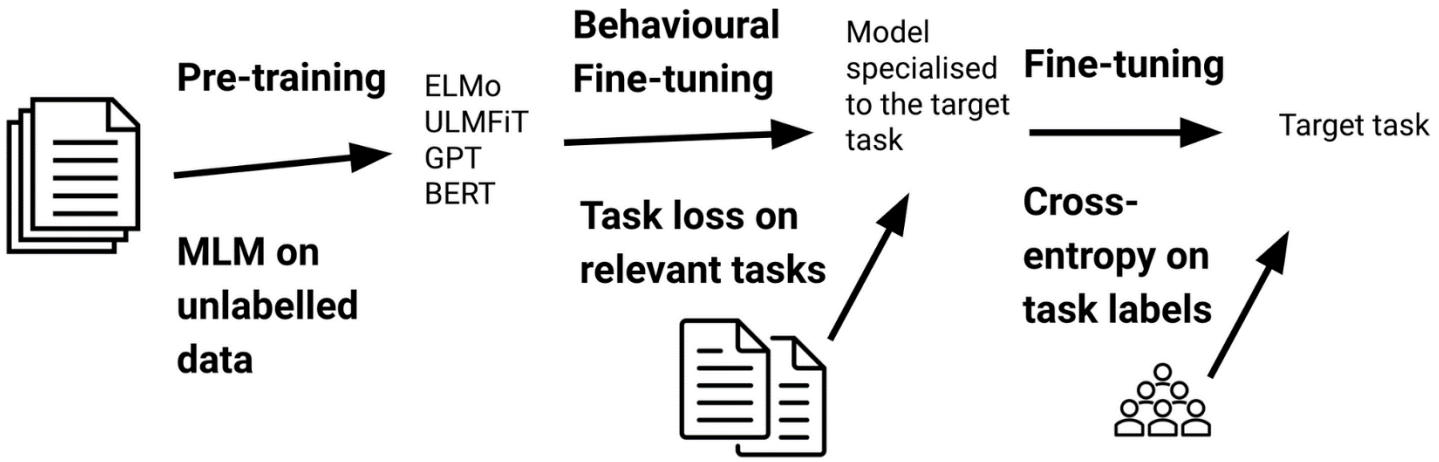
  - Task-adaptive: Unlabeled data is the label-removed version of the downstream training data.




*Source: https://ruder.io/recent-advances-lm-fine-tuning/*

# Behavioural fine-tuning

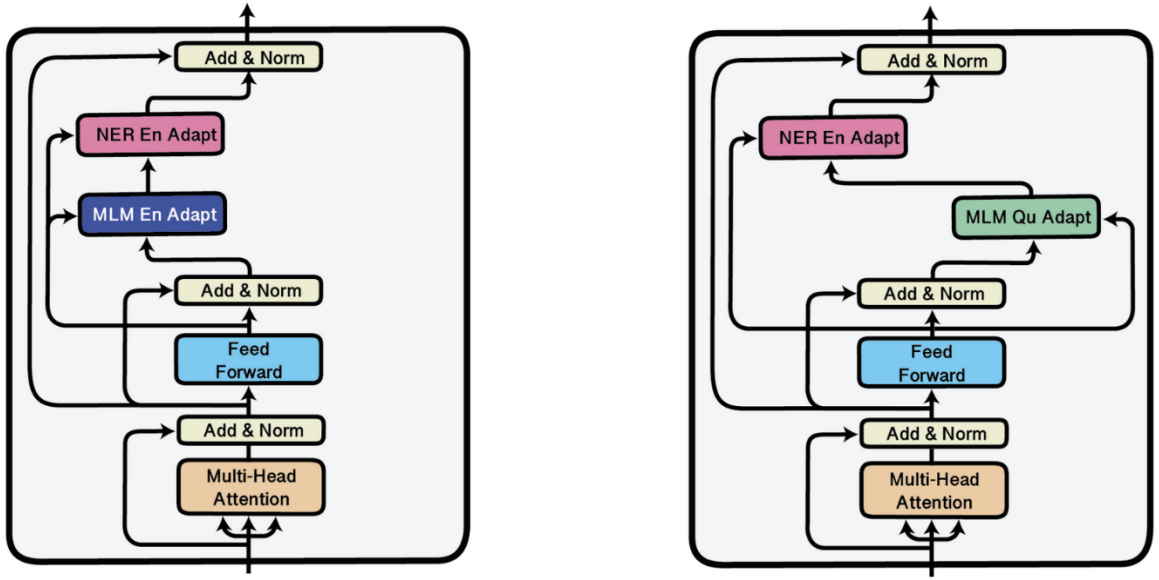
- Behavioural fine-tuning focuses on learning behaviours that are useful for the target task by fine-tuning the model on **labeled data** of the relevant tasks.
- E.g.: This paper (https://arxiv.org/abs/2101.11038) pre-finetunes the model on 50 labelled datasets (classification, summarization, common sense, MRC) in a massively multi-task setting and obtains significant improvement for Natural Language Inference, Question Answering, …




*Source: https://ruder.io/recent-advances-lm-fine-tuning/*

# Parameter-efficient fine-tuning

- Using adapters which are small networks injected between transformer layers. During fine-tuning, original weights of the pretrained language model are fixed while adapter and task-specific weights are updated.

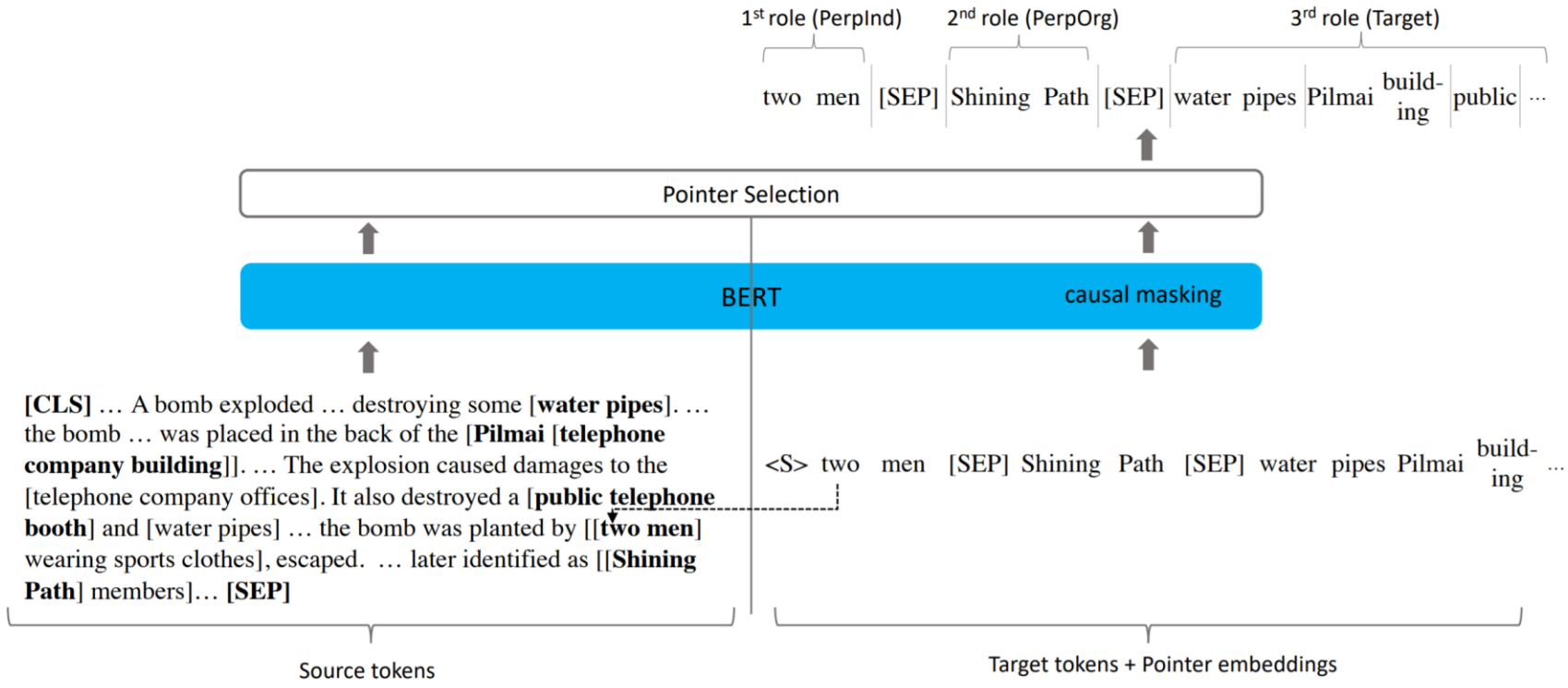



*Source:* MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer

- Removing some last few layers of the pretrained language models and performing the fine-tuning with the remaining ones might still give similar performance.

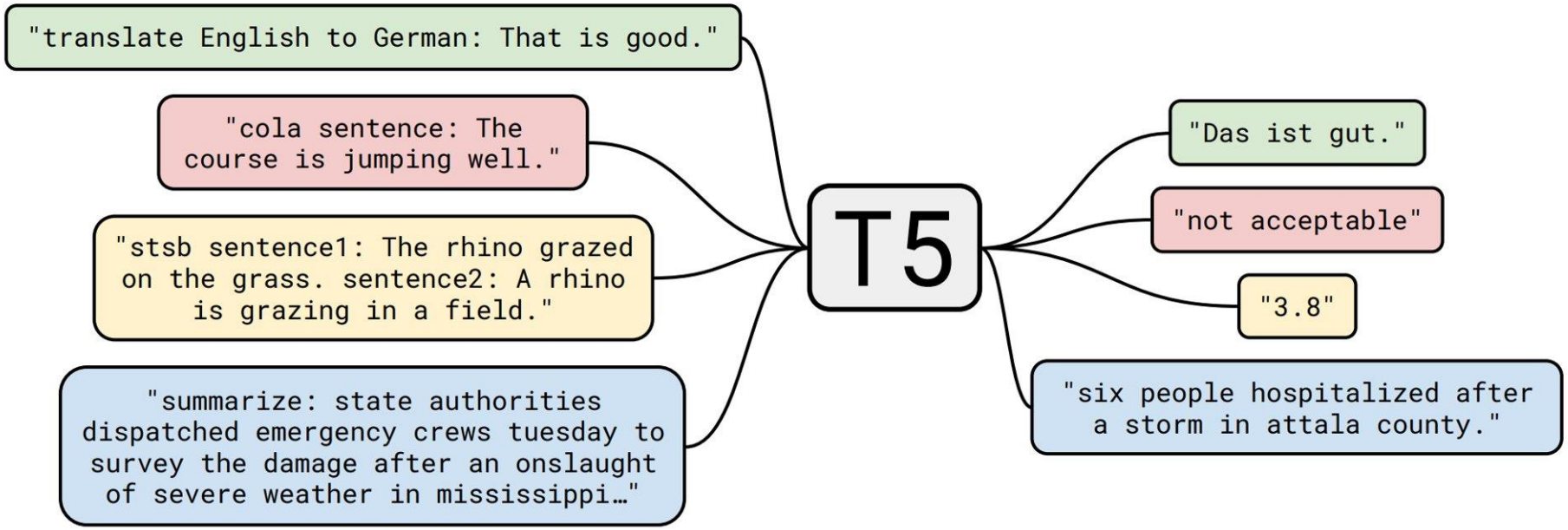
# Text-to-text fine-tuning

- Casting the downstream task to a generative extraction task.



# Text-to-text fine-tuning

- Casting the downstream task to a text generation task.



# Mitigating fine-tuning instabilities

- Performance for fine-tuning on a small downstream dataset can vary drastically between different runs.
- Previous work shows that weight initialization of the output layer and the order of the training data contribute to variation in performance.
- To avoid instabilities, avoid randomly initialized output layers. We can use behavioural or text-to-text fine-tuning.