CreativeCommons Attribution-NonCommercial-ShareAlike 4.0 International

Multilingual NLP

7. Modularization & Language Adaptation

(+ Parameter-Efficient Fine-Tuning)

Prof. Dr. Goran Glavaš
Center for AI and Data Science (CAIDAS), Uni Würzburg

Image: Alexander Mikhalchyk

After this lecture, you'll...

- Understand what „curse of multilinguality" is
- Know some common strategies for language adaptation of MMTs
- Be familiar with parameter-efficient fine-tuning (PEFT) methods
- Understand how PEFT can be leveraged to improve CL transfer

Content

- **Curse of Multilinguality**
- Modularity & Parameter-Efficient Fine-Tuning (PEFT)
- PEFT-Based CL Transfer

Poor CL Transfer with MMTs

- MMTs (mBERT, XLM-R) exhibit huge performance drops in CL transfer to low-resource languages, especially if they are distant from English

- Even for large and closely-related languages (e.g., DE, ES, IT) we see drop in performance compared to English.
  - Q: Why?

- For English, we get better results by fine-tuning monolingual English BERT/RoBERTa than by fine-tuning mBERT or XLM-R.
  - Q: Why?

Curse of Multilinguality

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., ... & Stoyanov, V. (2020, July). Unsupervised Cross-lingual Representation Learning at Scale. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (pp. 8440-8451).

- Performance better for „big languages"
  - Vocabulary more tailored to languages with more data
  - More data for a language → better performance
- But also: for any single language (even English)
  - Performance of an MMT pretrained on 10 languages better than performance of an MMT pretrained on 100 languages
  - Performance of monolingual models for large languages (e.g., English BERT/RoBERTa) better than that of MMT (e.g., XLM-R) for that language

Curse of Multilinguality

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., ... & Stoyanov, V. (2020, July). Unsupervised Cross-lingual Representation Learning at Scale. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (pp. 8440-8451).

- **Curse of multilinguality**: problem that occurs due to „cramming" too many languages into a model of insufficient capacity
  - Insufficient capacity to precisely represent all languages
- For any model of fixed capacity (i.e., fixed no. parameters), the performance of the model (monolingual and in CL transfer):
  - Improves with increasing the number of pretraining languages up until some threshold number of languages N_L
  - After N_L, performance decreases with adding more pretraining languages

Curse of Multilinguality

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., ... & Stoyanov, V. (2020, July). Unsupervised Cross-lingual Representation Learning at Scale. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (pp. 8440-8451).

- At N_L languages the capacity of model becomes „fully used"
  - Adding more languages for the same capacity – same number of parameters – means (more) sharing of parameters across
  - This means loss of information for any concrete language
- Tradeoff between generality and per-language performance
  - MMTs, in principle, support 100+ languages (CL transfer between them)
  - But even for the most-resource langs, MMTs will be worse than dedicated monolingual models for those languages

Curse of Multilinguality

- Q: Why not train monolingual BERT for each language?
  - Independently trained → repres. spaces not semantically aligned
  - Q: But can't we post-hoc align the monolingual BERTs (like we did monolingual word embedding spaces for CLWEs)?
  - To some extent, but only for high-resource languages
    - Lots of parallel data needed
    - Word-level supervision not enough for alignment

Curse of Multilinguality

- Q: Why not simply train monolingual BERT for each language?
  - Independently trained → representation spaces not semantically aligned
  - Q: Good monolingual LMs for low-resource languages (e.g., Quechua-BERT)?
    - Impossible to obtain, too little training data
  - Alignment of monolingual encoders becomes <u>more difficult</u> the more distant and less-resourced the two languages are
    - Monolingual representation spaces very very far from isomorphic
    - Good alignment requires more parallel data
    - "Catch 22": less parallel data available for low-resource languages

Curse of Multilinguality

- Q: Solutions?
  - Problem: we need to increase the quality of MMTs' representations for individual languages, especially low-resource
- Q: Just take the pretrained MMT and continue LM training only on texts of one (or few) language(s) you want improvements for?
  - Will improve the performance for that language
    - But: MMT parameters shared across languages
  - „Curse" → improving one language means <u>deteriorating others</u>
    - Trading multiling. generality for language-specific performance
  - Updates of all parameters of the MMT → for very large models, computationally infeasible

Curse of Multilinguality

- Q: What is the source of the problem?
  - All MMT's parameters are shared across all of the languages
- Solution: **modularity**
  - Make some parameters of the MMT language-specific, that is, not shared between languages
  - Such „private" parameters cannot suffer from the „curse"
  - When to enforce modularity:
    - Post-hoc, after the MMT was pretrained
      - Remedying for the „curse" after it occurred
    - Enforced already in multilingual pretraining
      - Preventing the „curse" from occurring

Content

- Curse of Multilinguality
- **Modularity & Parameter-Efficient Fine-Tuning (PEFT)**
- PEFT-Based CL Transfer

Why Modularity?

- What is meant by **modularity** in the context of neural LMs?
- **Module**: any subset of model's parameters that are trained/updated together with a particular aim
  - Can be a layer, sublayer, a particular parameter matrix in some layer, …
- Q: Why modularity (in general)?
  - Because neural LMs are becoming too large for **full fine-tuning**
    - I.e., updating all LMs parameters in task-specific fine-tuning

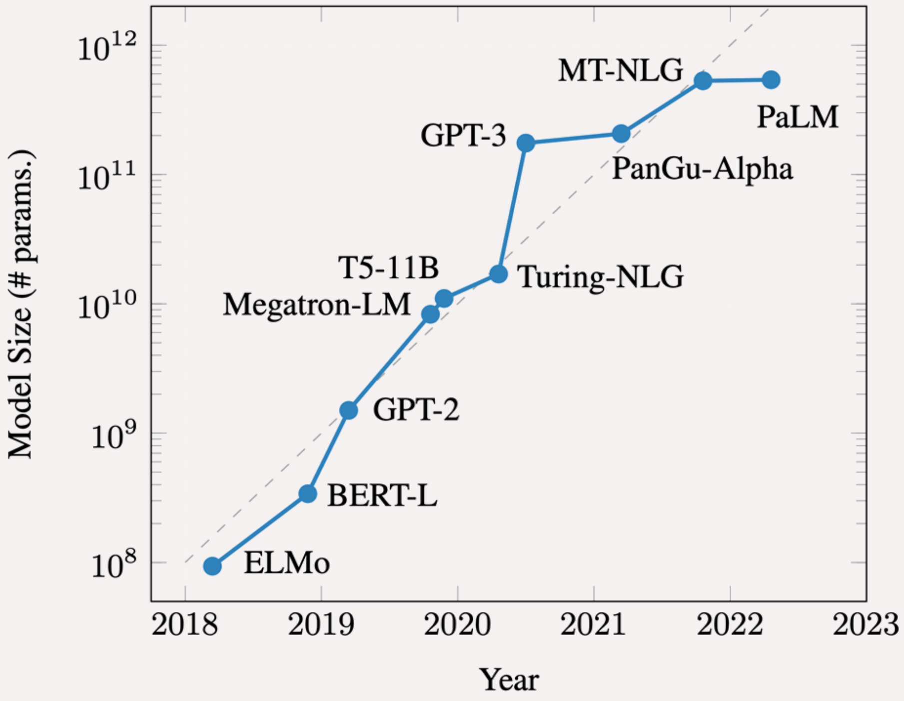
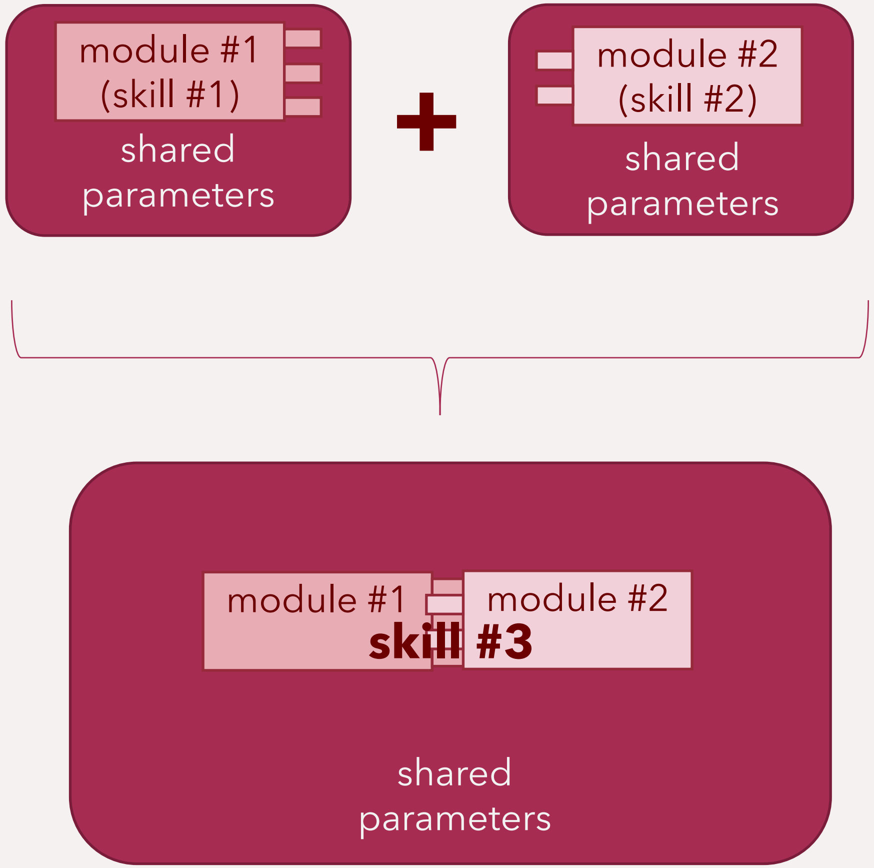


Image from: Treviso, M., Ji, T., Lee, J. U., van Aken, B., Cao, Q., Ciosici, M. R., ... & Schwartz, R. (2022). Efficient methods for natural language processing: a survey. *arXiv preprint arXiv:2209.00099*.

Why Modularity?

- Q: Why modularity (in general)?
  - Modular representations
  - Can be combined for unseen cases
    - Compositionality → combining existing modules, we can solve new tasks
  - For example:
    - module #1: trained for POS-tagging across many languages but not Quechua (no data)
    - module #2: trained for Quechua (via LM-ing or some other task data)
    - Combining module #1 and module #2:
      - Can do POS-tagging for Quechua



Modularity in NLP

- We're going to examine three popular modular architectures
  - Modularity enables parameter-efficient fine-tuning (PEFT)
  - In literature, you'll fine these methods commonly as „PEFT approaches"

1. Adapters
2. Prefix tuning
3. Low-rank adaptation (LoRA)

Adapters

- **Adapter** is (any kind of) module inserted into a pretrained neural LM (Transformer)
- Additional parameters, not a subset of the original parameters
- **Adapter-based fine-tuning**
  - „Freeze" original Transformer parameters
  - Update only the adapter parameters
- Typically one adapter added to each Transformer layer

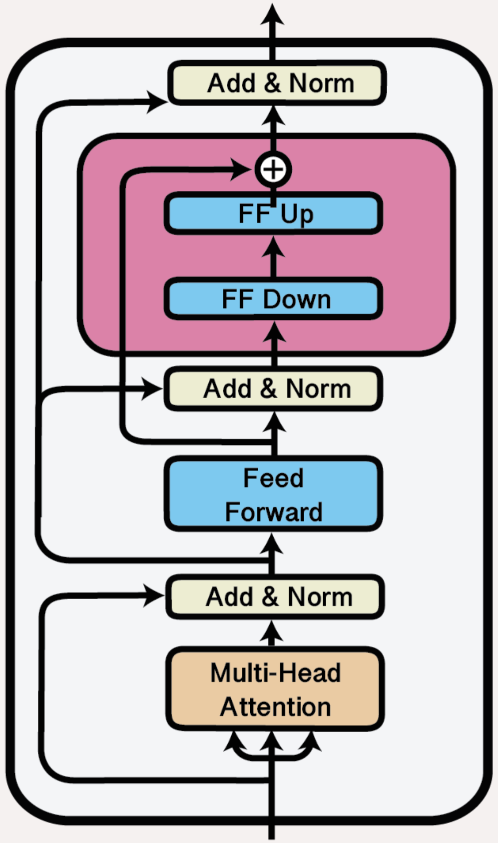


Image from: Pfeiffer, J., Vulić, I., Gurevych, I., & Ruder, S. (2020, November). MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In *Proceedings of EMNLP* (pp. 7654-7673).

Adapters

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., ... & Gelly, S. (2019, May). Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning* (pp. 2790-2799). PMLR.

- Adapter added as an additional sublayer of the Transformer layer
  - After the feed-forward layer
- **x**: the final output (residual+layer normalized) output of the FF sublayer
- **r**: the raw output of the FF layer (prior to residual summation and layer normalization)

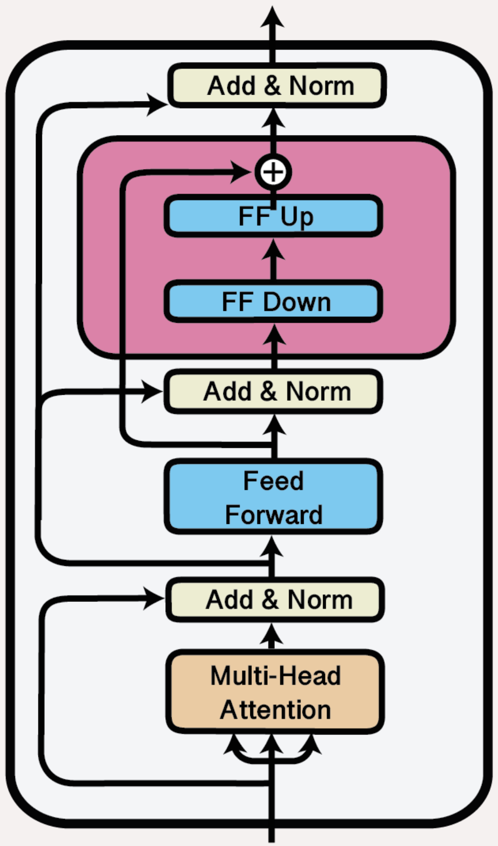


Image from: Pfeiffer, J., Vulić, I., Gurevych, I., & Ruder, S. (2020, November). MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In *Proceedings of EMNLP* (pp. 7654-7673).

Adapters

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., ... & Gelly, S. (2019, May). Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning* (pp. 2790-2799). PMLR.

- $\mathbf{x} \in \mathbb{R}^h$: the final output (residual + LN)
- $\mathbf{r} \in \mathbb{R}^h$: the raw output (prior to residual + LN)
- The most widely used type of the adapter is the so-called **bottleneck adapter**

$$\mathit{Adapter}(\mathbf{x}, \mathbf{r}) = \mathbf{W_U}(g(\mathbf{x}\mathbf{W_D})) + \mathbf{r}$$

  - $\mathbf{W_U} \in \mathbb{R}^{h \times m}$ (down-projection) and $\mathbf{W_D} \in \mathbb{R}^{m \times h}$ (up-projection): adapter's trainable parameters
  - Q: Why „bottleneck"? Because $m < h$
  - g is a non-linearity: tanh or ReLU

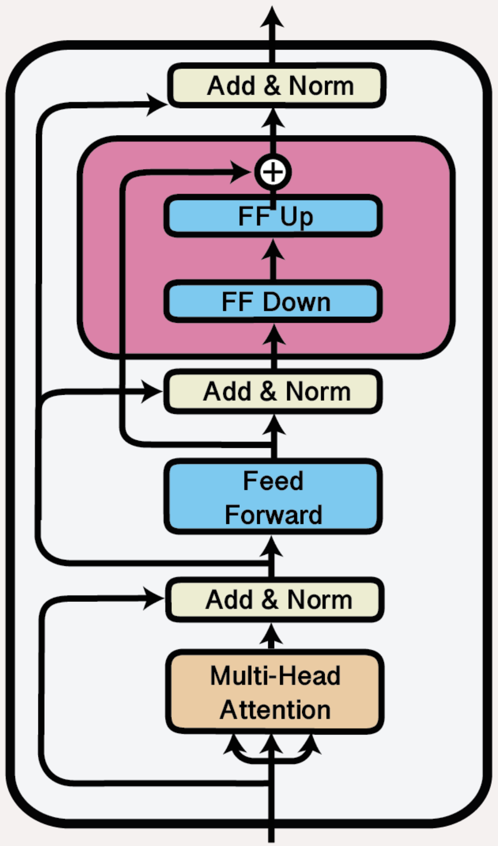


Image from: Pfeiffer, J., Vulić, I., Gurevych, I., & Ruder, S. (2020, November). MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In *Proceedings of EMNLP* (pp. 7654-7673).

Adapters

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., ... & Gelly, S. (2019, May). Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning* (pp. 2790-2799). PMLR.

- The most widely used type of the adapter is the so-called **bottleneck adapter**

$$\mathit{Adapter}(\mathbf{x}, \mathbf{r}) = \mathbf{W_U}(g(\mathbf{x}\mathbf{W_D})) + \mathbf{r}$$

- Initialization of adapter parameters is critical
  - Inserted inside of a pretrained layer
  - Initially needs to behave as an identity function
  - $\mathbf{W_U}(g(\mathbf{x}\mathbf{W_D}))$ needs to be a near-zero matrix
    - Easy to achieve by initializing $\mathbf{W_U}$ to a near-zero matrix

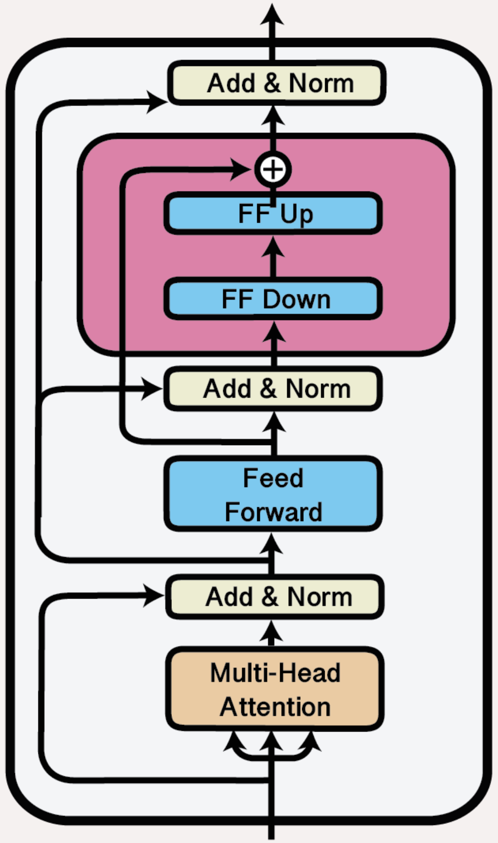


Image from: Pfeiffer, J., Vulić, I., Gurevych, I., & Ruder, S. (2020, November). MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In *Proceedings of EMNLP* (pp. 7654-7673).

Adapters

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., ... & Gelly, S. (2019, May). Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning* (pp. 2790-2799). PMLR.

- Q: Why is adapter-based fine-tuning parameter-efficient?
- We're updating only the adapter parameters: **2*m*h** parameters
  - With m << h, fewer than in the Transformer layer
  - Bottleneck size m is a hyperparameter – the smaller m is, the more parameter-efficient FT becomes
  - No free lunch: smaller m → lower performance

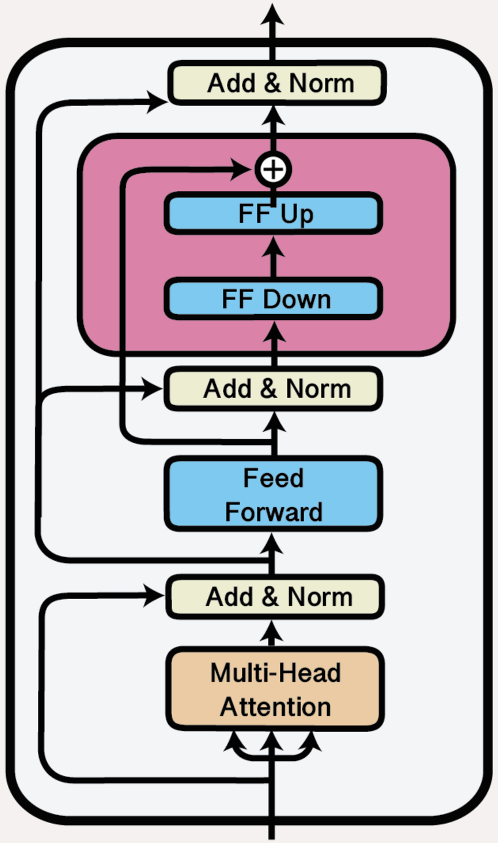
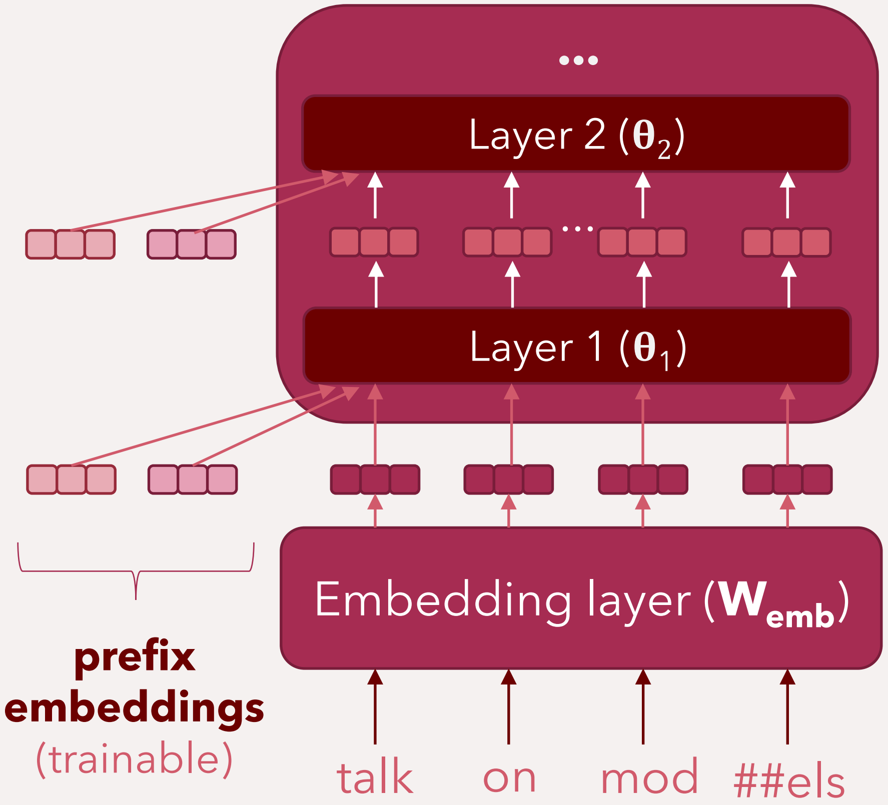


Image from: Pfeiffer, J., Vulić, I., Gurevych, I., & Ruder, S. (2020, November). MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In *Proceedings of EMNLP* (pp. 7654-7673).

Prefix Tuning

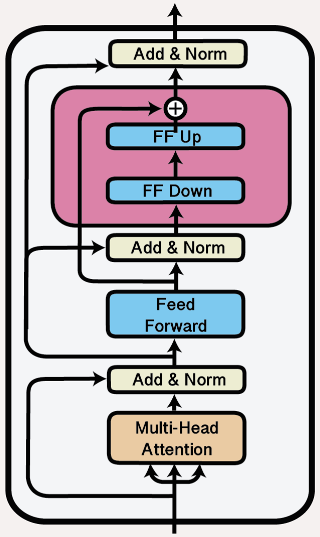
Li, X. L., & Liang, P. (2021). Prefix-Tuning: Optimizing Continuous Prompts for Generation. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers) (pp. 4582-4597).

- A special type of adapters
  - „modules" that are not inserted into Transformer layers but between them
  - At the input of each Transformer layer, we insert k trainable embeddings before the embeddings of real tokens
  - Q: number of prefix parameters?
    - k (pref. tokens) * N (layers) * h

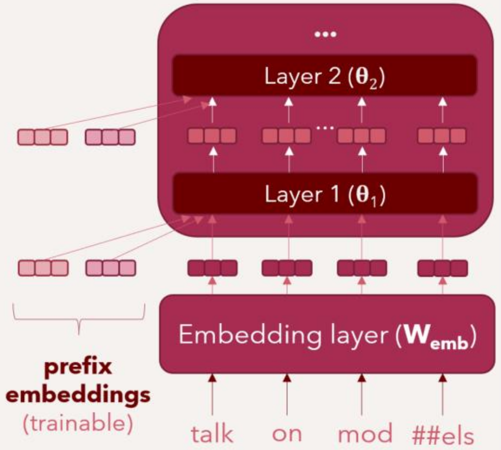


Shortcomings of Adapter-Based Models

- Performance
  - Both bottleneck adapters and prefix tuning typically underperform full fine-tuning
  - Performance good with more adapter parameters
    - Larger bottleneck size (large m)
    - Or many prefix „tokens" (large k)



- Inference speed
  - Adapters make training more efficient, but not inference (i.e., making predictions with the model)
  - Bottleneck adapters: model deeper
  - Prefix tuning: model wider (remember self-attention)



Low-Rank Adaptation (LoRA)

Hu, E. J., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. LoRA: Low-Rank Adaptation of Large Language Models. In International Conference on Learning Representations.

- A parameter-efficient fine-tuning approach that does not increase the total number of parameters of the whole Transformer-based model
- Instead of training new (adapter) parameters, LoRA learns a low-rank approximation of updates $\Delta\mathbf{W}$ to the existing parameters matrices $\mathbf{W}$
- Parameter update in standard fine-tuning:

$$\mathbf{W}^{(i+1)} = \mathbf{W}^{(i)} + \Delta\mathbf{W}$$

- LoRA „aggregates" the updates „on the side"

$$\mathbf{Wx} = \mathbf{W}_{pt}\mathbf{x} + \Delta\mathbf{Wx} = \mathbf{W}_{pt}\mathbf{x} + (\mathbf{BA})\mathbf{x}$$

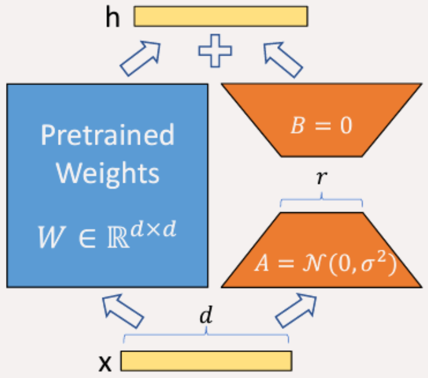


Image from the paper.

Low-Rank Adaptation (LoRA)

Hu, E. J., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. LoRA: Low-Rank Adaptation of Large Language Models. In International Conference on Learning Representations.

- Instead of training new (adapter) parameters, LoRA learns a low-rank approximation of updates $\Delta \mathbf{W}$ to the existing parameters matrices $\mathbf{W}$
- LoRA „aggregates" the updates „on the side"

$$\mathbf{W}\mathbf{x} = \mathbf{W}_{pt}\mathbf{x} + \Delta\mathbf{W}\mathbf{x} = \mathbf{W}_{pt}\mathbf{x} + (\mathbf{BA})\mathbf{x}$$

- $A \in \mathbb{R}^{h \times r}$ and $B \in \mathbb{R}^{r \times h}$ are trainable parameter matrices of LoRA
  - Rank r (w.r.t. h) determines the param. efficiency

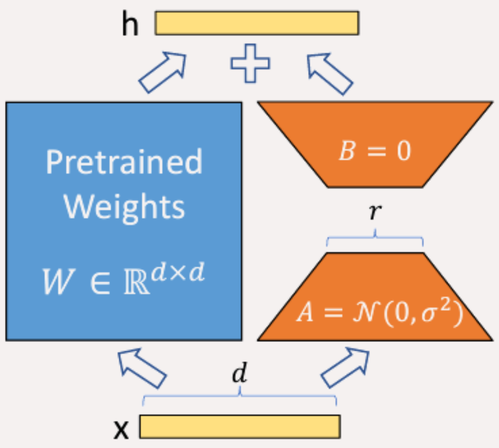


Image from the paper.

Low-Rank Adaptation (LoRA)

Hu, E. J., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. LoRA: Low-Rank Adaptation of Large Language Models. In International Conference on Learning Representations.

- LoRA „aggregates" the updates „on the side"

$$\mathbf{Wx} = \mathbf{W}_{pt}\mathbf{x} + \Delta\mathbf{Wx} = \mathbf{W}_{pt}\mathbf{x} + (\mathbf{BA})\mathbf{x}$$

- $A \in \mathbb{R}^{h \times r}$ and $B \in \mathbb{R}^{r \times h}$ trainable parameters
- Similar to adapters, LoRA parameters initialized to result in an identity function, i.e., $\Delta\mathbf{W} = 0$
  - **B** initialized to a zero matrix
  - **A** is initialized by sampling from a zero-mean Gaussian

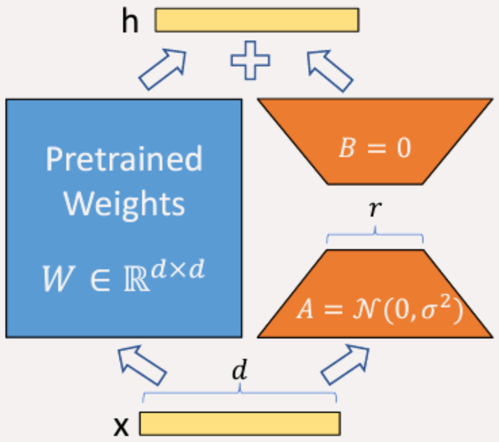


Image from the paper.

Low-Rank Adaptation (LoRA)

Hu, E. J., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. LoRA: Low-Rank Adaptation of Large Language Models. In International Conference on Learning Representations.

- LoRA „aggregates" the updates „on the side"

$$\mathbf{Wx} = \mathbf{W}_{\mathrm{pt}}\mathbf{x} + \Delta\mathbf{Wx} = \mathbf{W}_{\mathrm{pt}}\mathbf{x} + (\mathbf{BA})\mathbf{x}$$

- At the end of fine-tuning, final parameters:
  - $\mathbf{W} = \mathbf{W}_{\mathrm{pt}} + \mathbf{BA}$
- The resulting fine-tuned model has exactly the same number of parameters as the starting one
  - No inference latency!

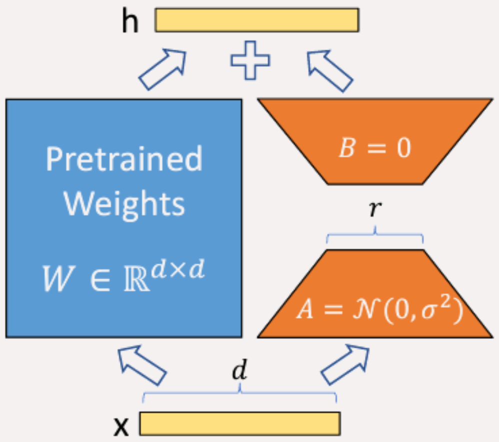


Image from the paper.

Low-Rank Adaptation (LoRA)

Hu, E. J., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. LoRA: Low-Rank Adaptation of Large Language Models. In International Conference on Learning Representations.

- At the end of training, final parameters:
  - $\mathbf{W} = \mathbf{W}_{pt} + \mathbf{BA}$
- Theoretically, any parameter matrix of the original model (Transformer) can be LoRA fine-tuned
- The original paper applies LoRA only to the matrices in the multi-head attention (of each layer)
  - Best tradeoff if only $\mathbf{W}_Q$ and $\mathbf{W}_V$ of each self-attention mechanism are LoRA fine-tuned

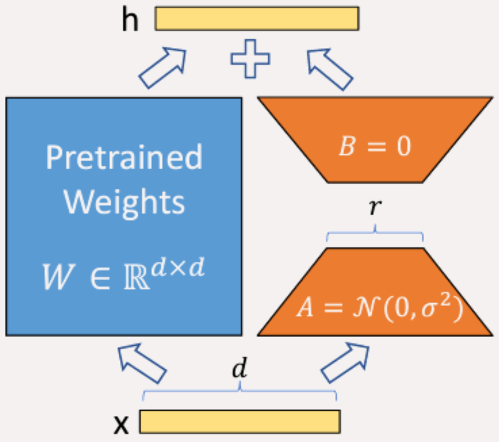


Image from the paper.

Content

- Curse of Multilinguality
- Modularity & Parameter-Efficient Fine-Tuning (PEFT)
- **PEFT-Based CL Transfer**

Modular CL Transfer

- Q: How to leverage modularity and PEFT to:

  (1) Improve the representations for under-resourced languages and
  (2) Consequently, cross-lingual transfer for downstream tasks?

- There are different strategies, but all based on the idea of providing and additional capacity for each language
  - Additional LM-ing training of language-specific modules

Adapter-Based CL Transfer

Pfeiffer, J., Vulić, I., Gurevych, I. & Ruder, S. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. EMNLP 2020 (pp. 7654–7673).

- MAD-X: starts from a pretrained MMT
  - mBERT or XLM-R

1. Training a set of monolingual adapters
   - Independently: English adapter, Quechuan adapter, …
   - Trained via (M)LM-ing on the monolingual corpus of the language

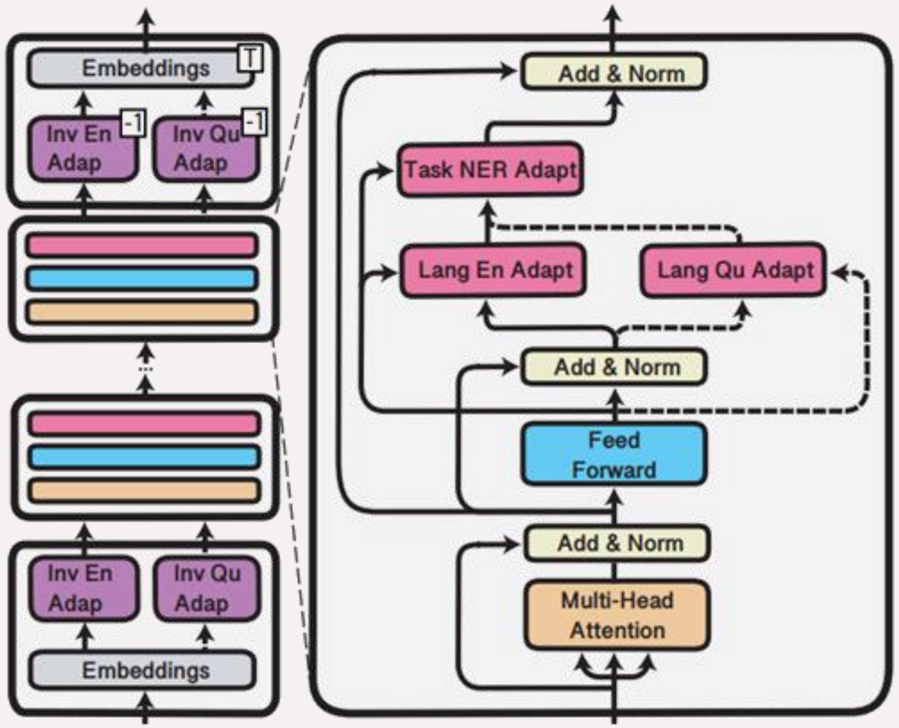


Image from the paper.

Adapter-Based CL Transfer

Pfeiffer, J., Vulić, I., Gurevych, I. & Ruder, S. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. EMNLP 2020 (pp. 7654–7673).

- MAD-X: starts from a pretrained MMT
  - mBERT or XLM-R

1. Training a set of monolingual language adapters (LAs)
   - Each language adapter trained independently on top of the same (pretrained) Transformer backbone
   - Training of an LA: (M)LM-ing on the monolingual corpus

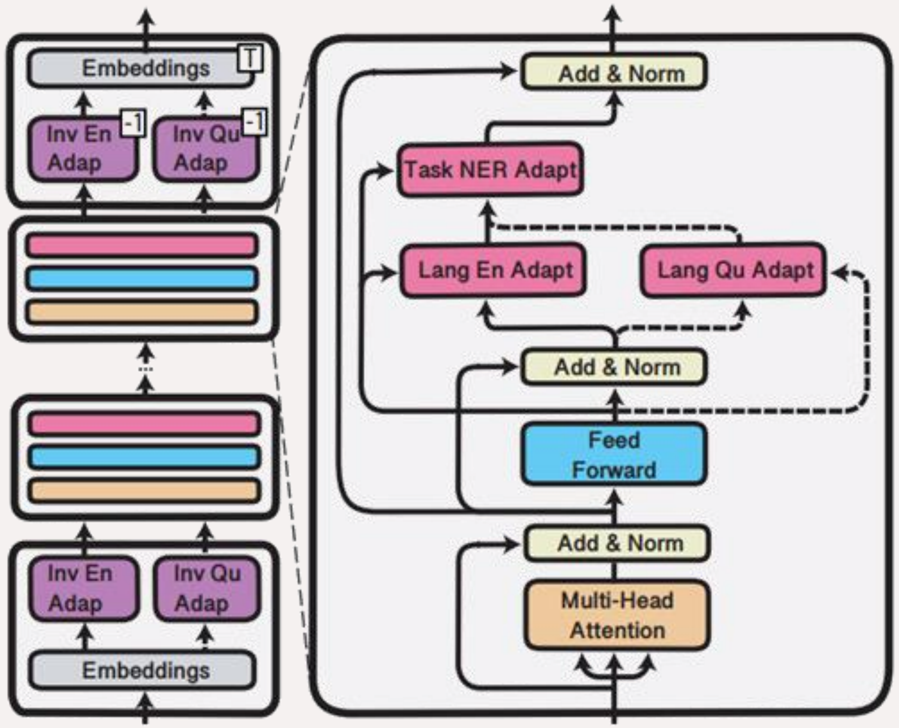


Image from the paper.

Adapter-Based CL Transfer

Pfeiffer, J., Vulić, I., Gurevych, I. & Ruder, S. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. EMNLP 2020 (pp. 7654–7673).

- MAD-X: starts from a pretrained MMT
  - mBERT or XLM-R

2. Task-specific training
   - Training data in the source lang. L_S
   - Insert the LA of L_S into the MMT
   - Insert and initialize the new task adapter (TA) on top of the LA of L_S
   - Train the parameters of TA, while keeping the MMT and TA parameters frozen

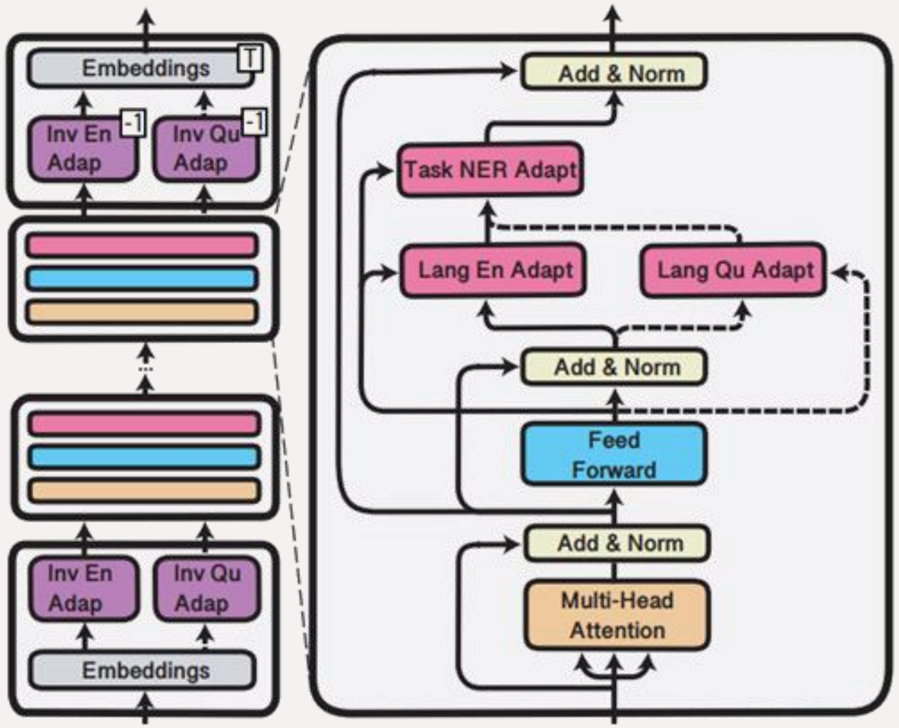


Image from the paper.

Adapter-Based CL Transfer

Pfeiffer, J., Vulić, I., Gurevych, I. & Ruder, S. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. EMNLP 2020 (pp. 7654–7673).

- MAD-X: starts from a pretrained MMT
  - mBERT or XLM-R

3. Inference

   - Make predictions for data in the target language L_T
   - Replace the LA of L_S (used in training) with the LA of the target language L_T
   - In other words, place the TA on top of target language LA

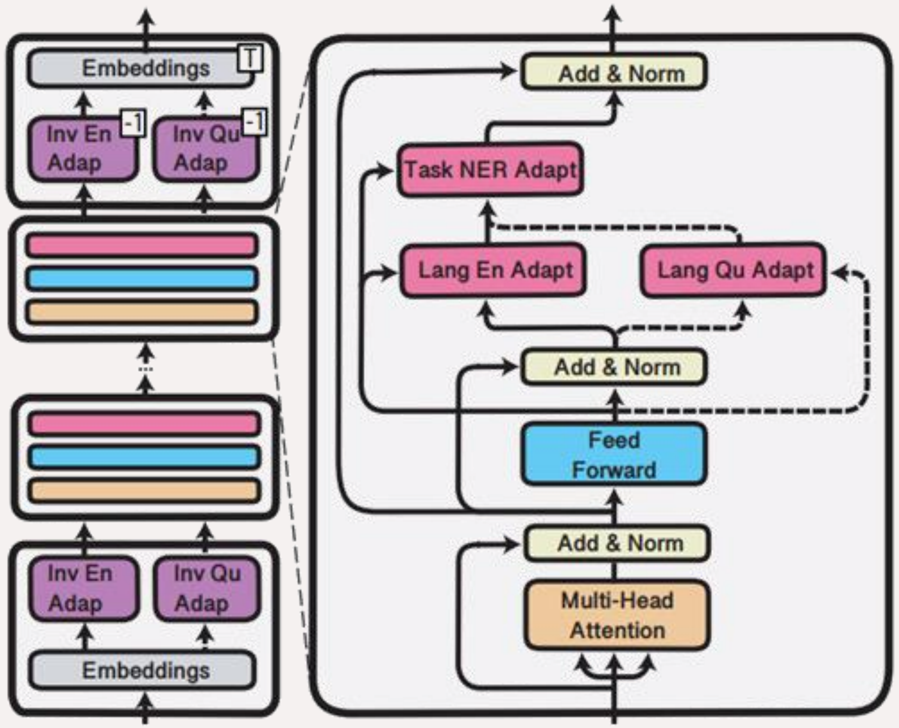


Image from the paper.

Adapter-Based CL Transfer

Pfeiffer, J., Vulić, I., Gurevych, I. & Ruder, S. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. EMNLP 2020 (pp. 7654–7673).

- MAD-X: some limitations
  - Transferring between L_S and L_T, but respective LAs trained independently
    - No positive interaction between the two languages
  - LA is trained on (relatively) large corpus of the language – what about languages with very small monolingual corpora?
    - Not possible to train a reliable LA for those languages

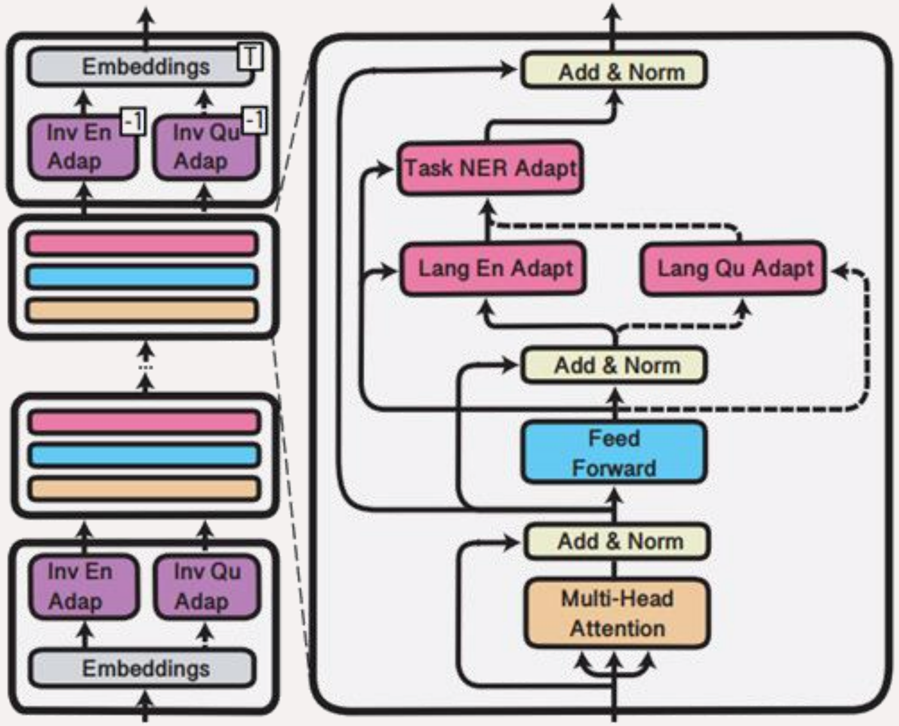


Image from the paper.

Adapter-Based CL Transfer

Parović, M., Glavaš, G., Vulić, I., & Korhonen, A. (2022). BAD-X: Bilingual Adapters Improve Zero-Shot Cross-Lingual Transfer. In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics (pp. 1791-1799).

- BAD-X: bilingual adapters
  - Trains bilingual adapters for pairs of languages L_S and L_T: via (M)LM-ing on the bilingual corpus, concatenation of monolingual corpora of L_S and L_T
  - Enables direct interaction between the two languages through shared parameters of the bilingual adapter

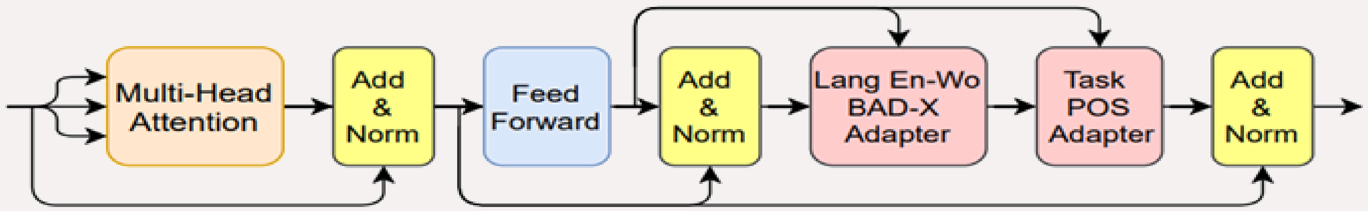


Image from the paper.

Adapter-Based CL Transfer

Parović, M., Glavaš, G., Vulić, I., & Korhonen, A. (2022). BAD-X: Bilingual Adapters Improve Zero-Shot Cross-Lingual Transfer. In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics (pp. 1791-1799).

- BAD-X vs. MAD-X = Performance vs. Generality
  - MAD-X trains N monolingual LAs, with which it supports any of the N^2 possible transfer directions (more general)
  - BAD-X gives better CL transfer performance for any transfer direction ($L_S \rightarrow L_T$), but to support all N^2 of them, we need to train N^2 bilingual adapters

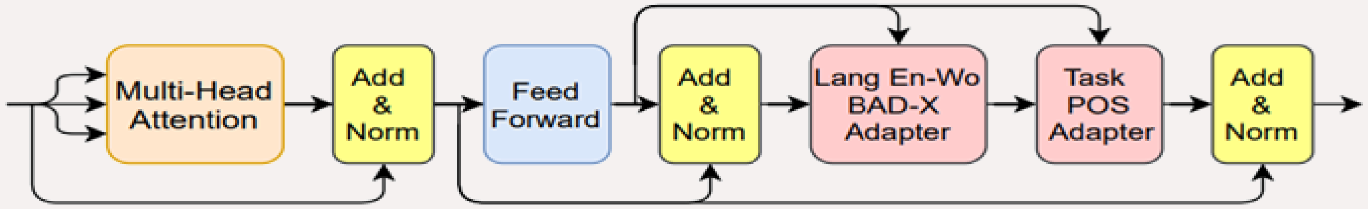


Image from the paper.

The End

Image: Alexander Mikhalchyk