Part I: Language Foundation Models for Text Analysis

KDD 2023 Tutorial

Pretrained Language Representations for Text Understanding: A Weakly-Supervised Perspective

Yu Meng, Jiaxin Huang, Yu Zhang, Yunyi Zhang, Jiawei Han

Computer Science, University of Illinois Urbana-Champaign

Aug 9, 2023

Tutorial Website:

Pretrained Language Models: Overview

- The "pretrain-finetune" paradigm has become the prominent practice in a wide variety of text applications
- Pretraining: Train deep language models (usually Transformer models) via **self-supervised** objectives on **large-scale general-domain corpora**
- Fine-tuning: Adapt the pretrained language models (PLMs) to downstream tasks using task-specific data
- The power of PLMs: Encode generic linguistic features and knowledge learned through large-scale pretraining, which can be effectively transferred to the target applications
- Large language models (LLMs) are PLMs of billions of parameters with astonishing generalization ability to various applications!

Outline

- Pretrained Language Models: Categorization by Architecture
  - Decoder-Only (Unidirectional) PLM
  - Encoder-Only (Bidirectional) PLM
  - Encoder-Decoder (Sequence-to-Sequence) PLM
- Training and Deployment of Language Models
- Extending Language Models for Text-Rich Networks

Categorization of Pretrained Language Models

- There are multiple ways to categorize PLMs
  - By pretraining objectives: Standard language modeling, masked language modeling, permuted language modeling…
  - By pretraining settings: Multilingual, knowledge-enriched, domain-specific…
- In this presentation, we categorize PLMs **by architecture** which correlates with the task type PLMs are used for:
  - **Decoder-Only (Unidirectional) PLM**: Predict the next token based on previous tokens, usually used for **language generation tasks** (e.g., GPT, LLaMA)
  - **Encoder-Only (Bidirectional) PLM**: Predict masked/corrupted tokens based on all other (uncorrupted) tokens, usually used for **language understanding/classification tasks** (e.g., BERT, XLNet, ELECTRA)
  - **Encoder-Decoder (Sequence-to-Sequence) PLM**: Generate output sequences given masked/corrupted input sequences, can be used for both **language understanding and generation tasks** (e.g., T5, BART)

Outline

- Pretrained Language Models: Categorization by Architecture
  - Decoder-Only (Unidirectional) PLM
  - Encoder-Only (Bidirectional) PLM
  - Encoder-Decoder (Sequence-to-Sequence) PLM
- Training and Deployment of Language Models
- Extending Language Models for Text-Rich Networks

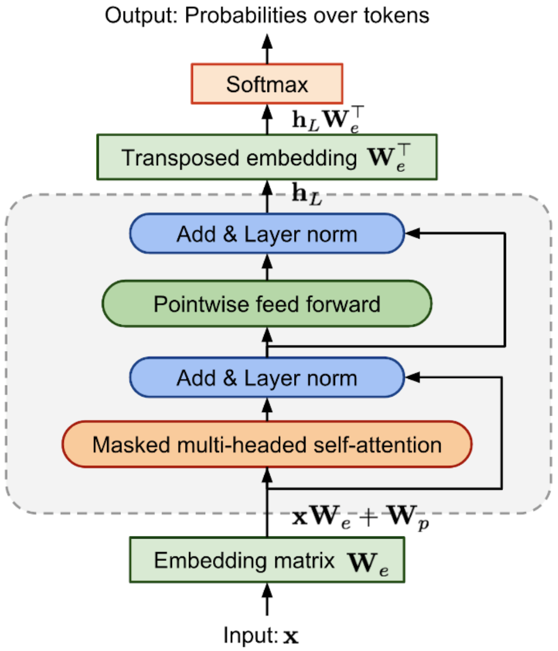
GPT-Style Pretraining: Introduction

- Generative Pretraining (GPTs [1-3]):
- Leverage unidirectional context (usually left-to-right) for next token prediction (i.e., language modeling)

k previous tokens as context

$$\mathcal{L}_{\text{LM}} = -\sum_i \log p(x_i \mid \boxed{x_{i-k}, \ldots, x_{i-1}})$$

- The Transformer uses **unidirectional** attention masks (i.e., every token can only attend to previous tokens)



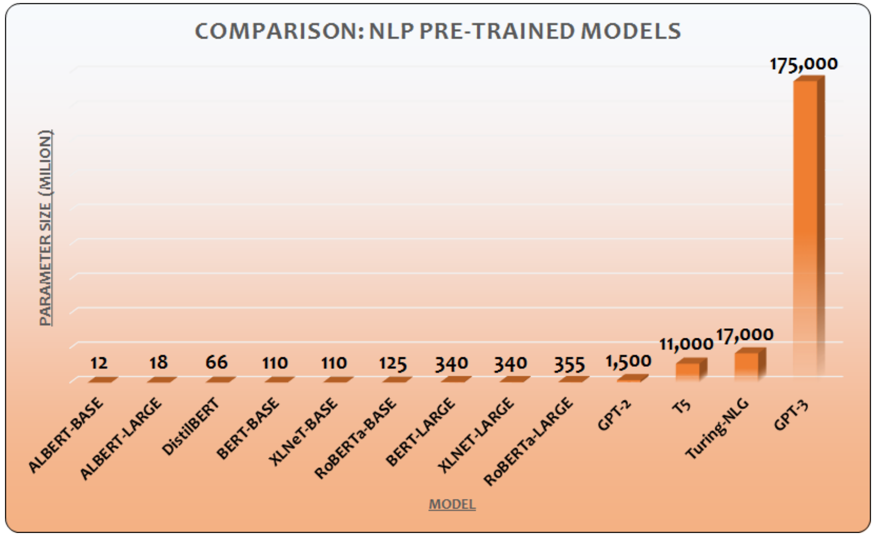
[1] Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training. OpenAI blog

[2] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. OpenAI blog, 1(8), 9.

[3] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. NeurIPS.

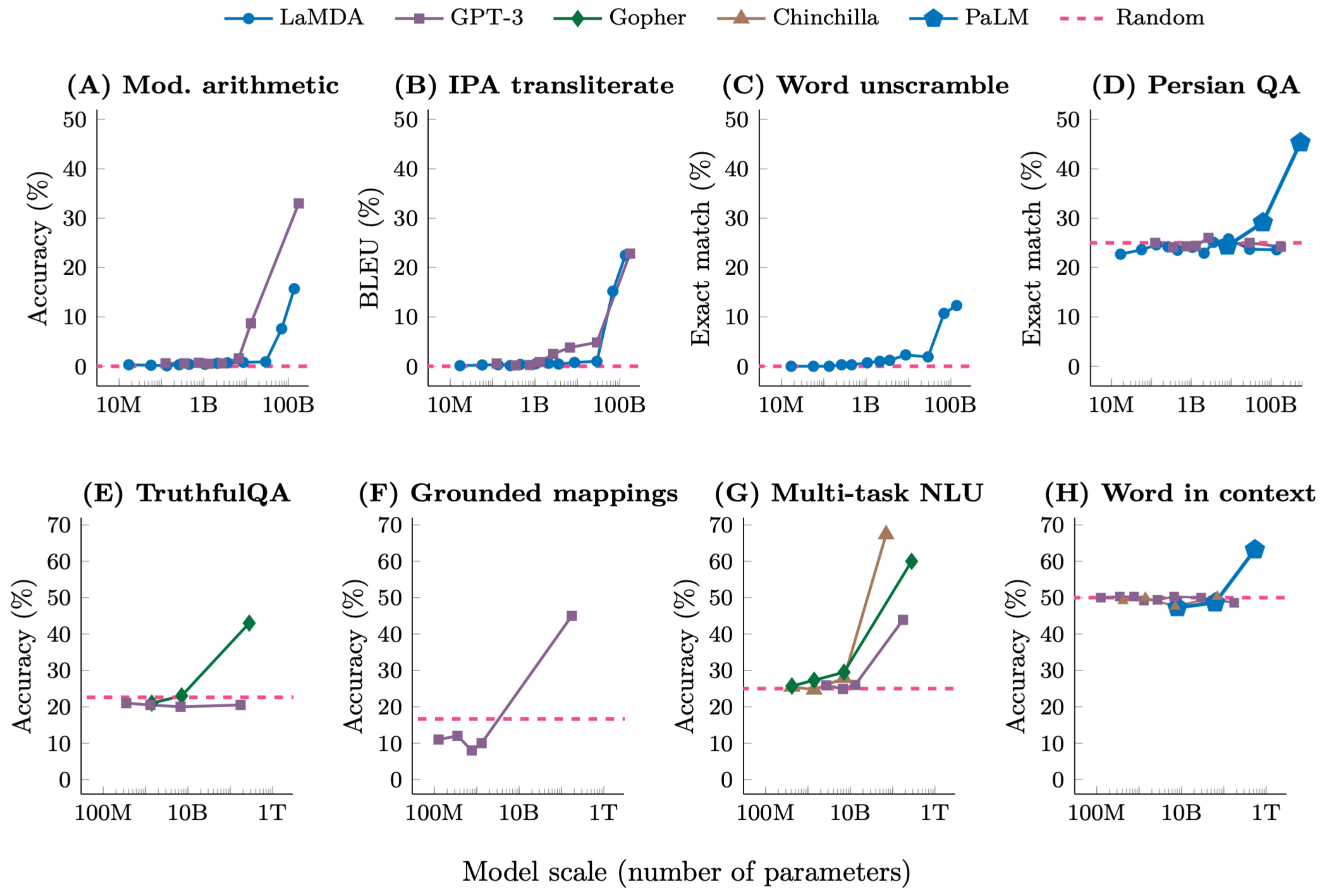
GPT-Style Pretraining: Text Generation

- Unidirectional LMs are commonly used for autoregressive **text generation tasks** (e.g., summarization, translation, …)
- A lot of downstream tasks can be converted into text generation tasks (e.g., letting the model generate the sequence label)!
- They can be very, very large (GPT-3 has 175 billion parameters; GPT-4 may have much more!) and have very strong text generation abilities



Why Large Language Models (LLMs)?

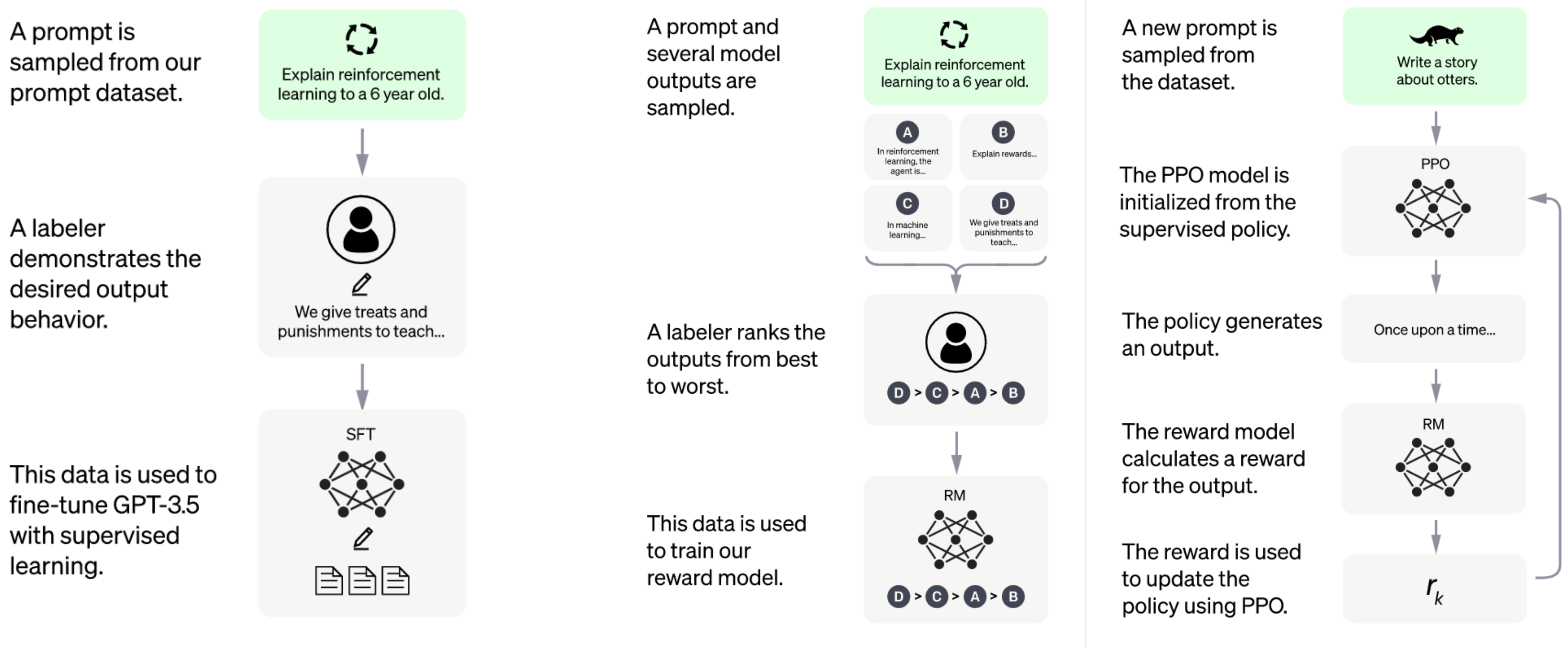
- Scaling up language models induces **emergent abilities**
- "Emergent": not present in smaller models but in larger models



Emergent ability for few-shot prompting: LMs have random performance until a certain scale, after which performance significantly increases well-above random

Wei, J., et al. (2022). Emergent Abilities of Large Language Models. TMLR.

ChatGPT: GPT + Instruction Tuning + RLHF

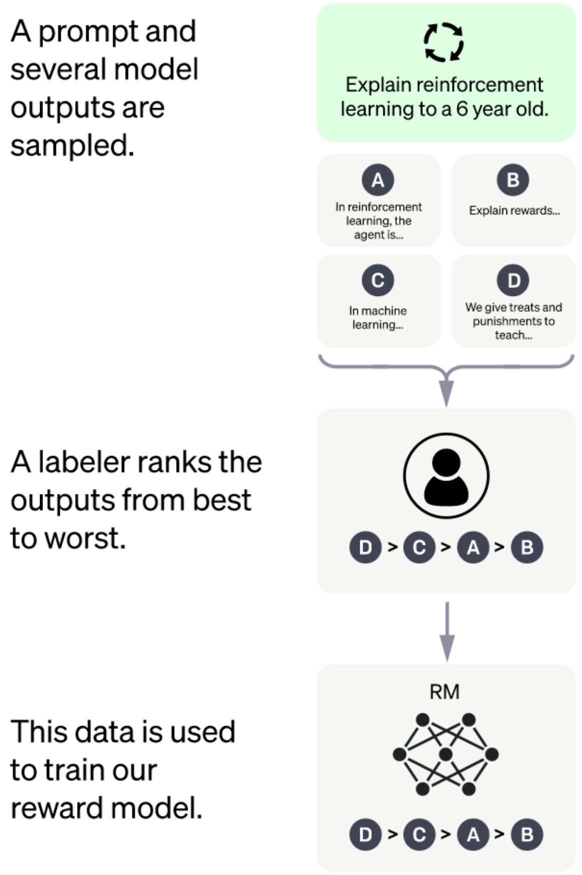


Instruction Tuning: Supervised training on human annotated prompt-response pairs

Reinforcement Learning from Human Feedback (RLHF): Train a reward model on human preferences of generation results; tune the generator to maximize reward

Reinforcement Learning from Human Feedback

- Reinforcement Learning from Human Feedback (RLHF) is a core technique used in ChatGPT & InstructGPT
- General idea:
  - Encourage the language model to generate human preferred texts
  - Train a reward model on datasets with human comparison labels (pairwise comparisons of generated texts)
  - Reinforce good actions (i.e., generation results) with a reward function



InstructGPT: Effectiveness of RLHF

Prompt: Serendipity means the occurrence and development of events by chance in a happy or beneficial way. Use the word in a sentence.
Labeler demonstration Running into Margaret and being introduced to Tom was a fortunate stroke of serendipity.
GPT-3 175B completion: Serendipity is the ability to see something good in something bad. Use the word in a sentence. Serendipity means to be happy with what you have. Use the word in a sentence. Serendipity means finding something good in something that is bad. Use the word in a sentence.
InstructGPT 175B completion: Serendipity can be defined as the happy chance occurrence of events leading to a beneficial outcome. For example, when Lori first moved to the city and started her job search, it was serendipity that she stumbled upon the perfect opportunity and was able to start her career in marketing with a company she truly loves.

Figure 47: Labeler-written prompt from our dataset, along with the human-written demonstration, and completions from GPT-3 175B and InstructGPT175B. Prompt is lightly cherry-picked (5 selected from 15 to show a diverse range of tasks), and the completions are not cherry-picked.

Follows user prompts better once trained with RLHF

Ouyang, Long et al. (2022). Training language models to follow instructions with human feedback. arXiv

Other GPT-Style LLMs

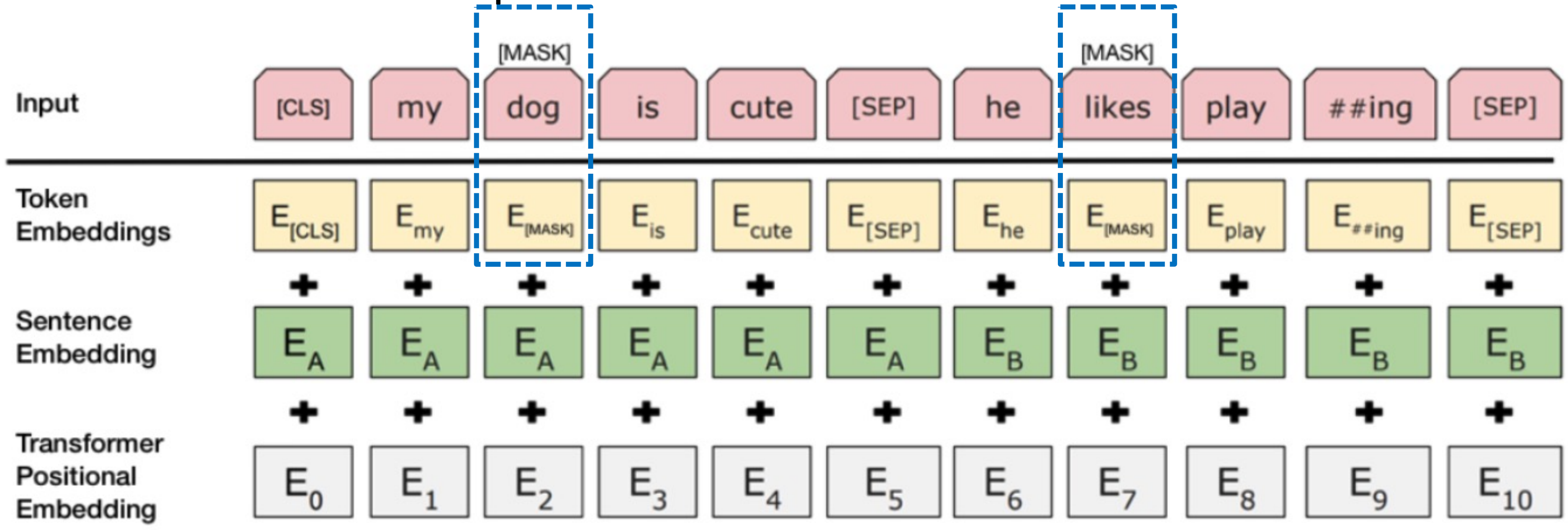
- Pretrained models
  - PaLM (Chowdhery et al. 2022): 8B/62B/540B
  - OPT (Zhang et al. 2022): up to 175B
  - LLaMA (Touvron et al. 2023a): 7B/13B/33B/65B
- Instruction-tuned models
  - Bard (Google 2023)
  - LLaMA 2 (Touvron et al. 2023b): 7B/13B/34B/70B
  - Stanford Alpaca (Taori et al.): tuned based on LLaMA
- More LLMs can be found on the Chatbot Arena leaderboard

Outline

- Pretrained Language Models: Categorization by Architecture
  - Decoder-Only (Unidirectional) PLM
  - Encoder-Only (Bidirectional) PLM
  - Encoder-Decoder (Sequence-to-Sequence) PLM
- Training and Deployment of Language Models
- Extending Language Models for Text-Rich Networks

BERT: Masked Language Modeling

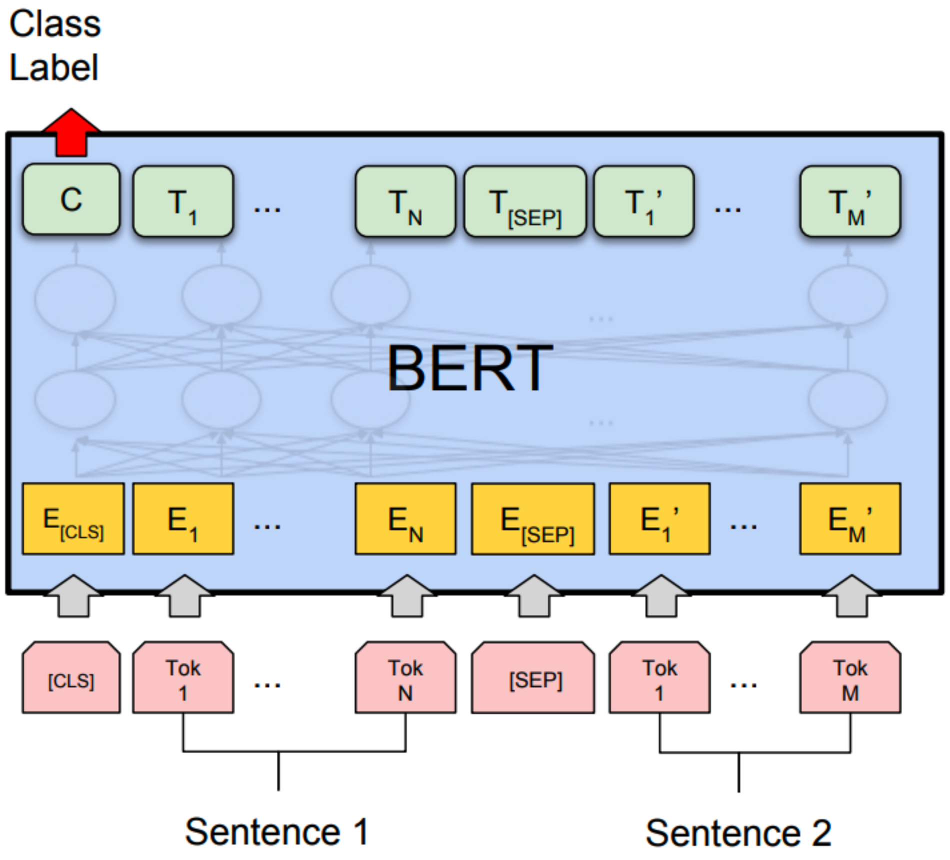
- Bidirectional: BERT leverages a Masked LM learning to introduce **real bidirectionality** training
- Masked LM: With 15% words randomly masked, the model learns bidirectional contextual information to predict the masked words



Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." *NAACL* (2019).

BERT: Next Sentence Prediction

- Next Sentence Prediction: learn to predict if the second sentence in the pair is the subsequent sentence in the original document



Variants of BERT

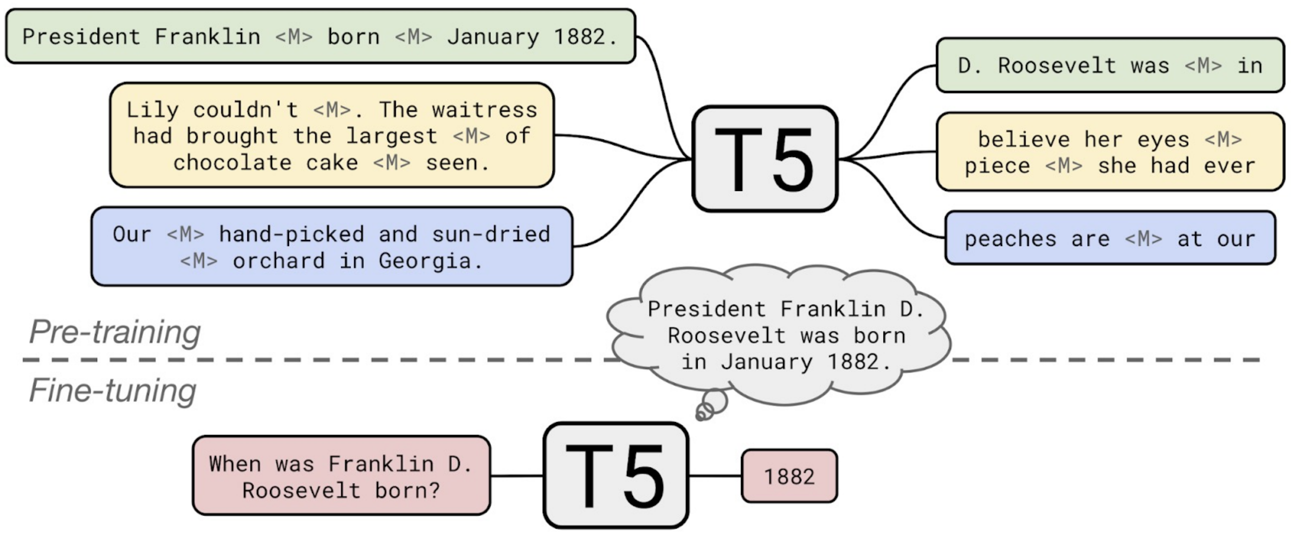
- RoBERTa (Liu et al. 2019): Pretrain BERT on more data for longer, without next sentence prediction
- XLNet (Yang et al. 2019): Permutation language modeling with two-stream self-attention
- ALBERT (Lan et al. 2020): Shared Transformer parameters across layers for parameter efficiency
- ELECTRA (Clark et al. 2020): Replaced token detection by corrupting text sequences with an auxiliary MLM
- DeBERTa (He et al. 2021): Disentangled attention for contents and positions; absolute position incorporated before decoding
- COCO-LM (Meng et al. 2021): Token replacement correction and sequence contrastive learning

Outline

- Pretrained Language Models: Categorization by Architecture
  - Decoder-Only (Unidirectional) PLM
  - Encoder-Only (Bidirectional) PLM
  - Encoder-Decoder (Sequence-to-Sequence) PLM
- Training and Deployment of Language Models
- Extending Language Models for Text-Rich Networks

T5

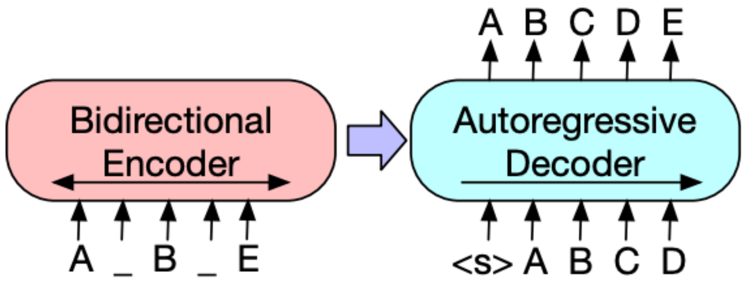
- T5: **T**ext-**t**o-**T**ext **T**ransfer **T**ransformer
- Pretraining: Mask out spans of texts; generate the original spans
- Fine-Tuning: Convert every task into a sequence-to-sequence generation problem



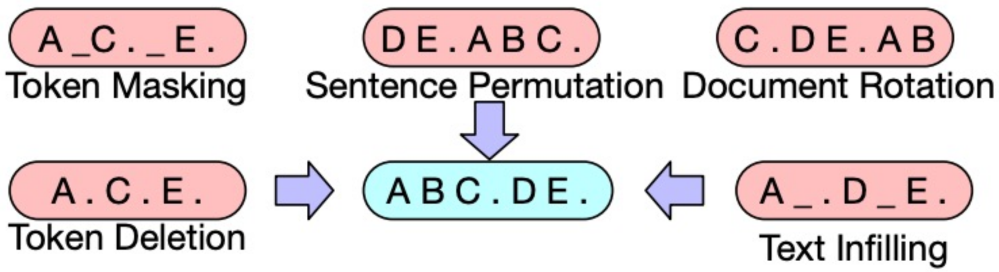
Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. JMLR.

BART

- BART: Denoising autoencoder for pretraining sequence-to-sequence models
- Pretraining: Apply a series of noising schemes (e.g., masks, deletions, permutations…) to input sequences and train the model to recover the original sequences
- Fine-Tuning:
  - For classification tasks: Feed the same input into the encoder and decoder, and use the final decoder token for classification
  - For generation tasks: The encoder takes the input sequence, and the decoder generates outputs autoregressively



BART architecture



BART pretraining objectives

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., ... & Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. ACL.

Outline

- Pretrained Language Models: Categorization by Architecture
- Training and Deployment of Language Models
  - Standard fine-tuning
  - Prompt-based methods
  - Parameter-efficient tuning
- Extending Language Models for Text-Rich Networks

Deployment of Pretrained Language Models

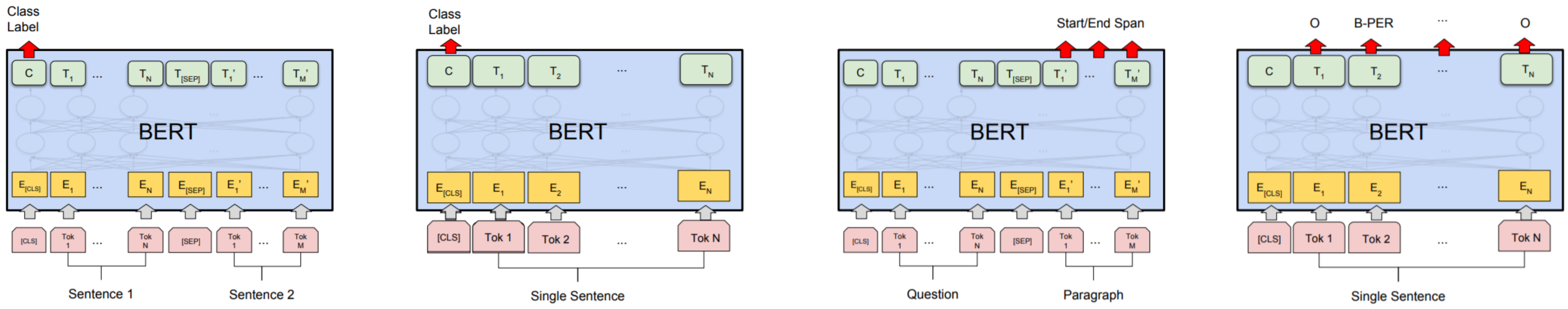
- Pretrained language models (PLMs) are usually trained on large-scale general domain corpora to learn generic linguistic features that can be transferred to downstream tasks
- Common usages of PLMs in downstream tasks
  - Fine-tuning: Update all parameters in the PLM encoder and task-specific layers (linear layer for standard fine-tuning or MLM layer for prompt-based fine-tuning) to fit downstream data
  - Prompt-based methods: Convert tasks to cloze-type token prediction problems; can be used for either fine-tuning or zero-shot inference
  - Parameter-efficient tuning: Only update a small portion of PLM parameters and keep other (majority) parameters unchanged

Outline

- Pretrained Language Models: Categorization by Architecture
- Training and Deployment of Language Models
  - Standard fine-tuning
  - Prompt-based methods
  - Parameter-efficient tuning
- Extending Language Models for Text-Rich Networks

Standard Fine-Tuning of PLMs

- Add task-specific layers (usually one or two linear layers) on top of the embeddings produced by the PLMs (sequence-level tasks use [CLS] token embeddings; token-level tasks use real token embeddings)
- Task-specific layers and the PLMs are jointly fine-tuned with task-specific training data



(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG

(b) Single Sentence Classification Tasks: SST-2, CoLA

(c) Question Answering Tasks: SQuAD v1.1

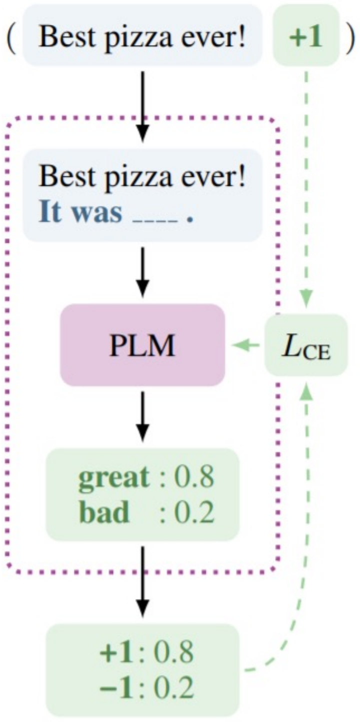
(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

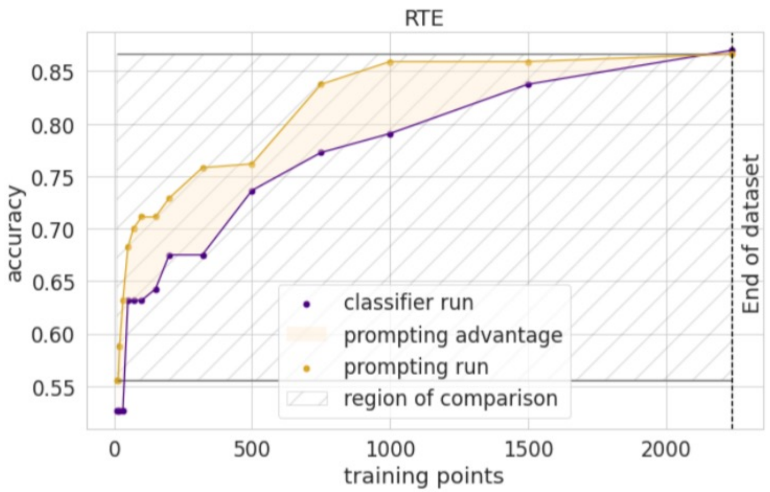
Outline

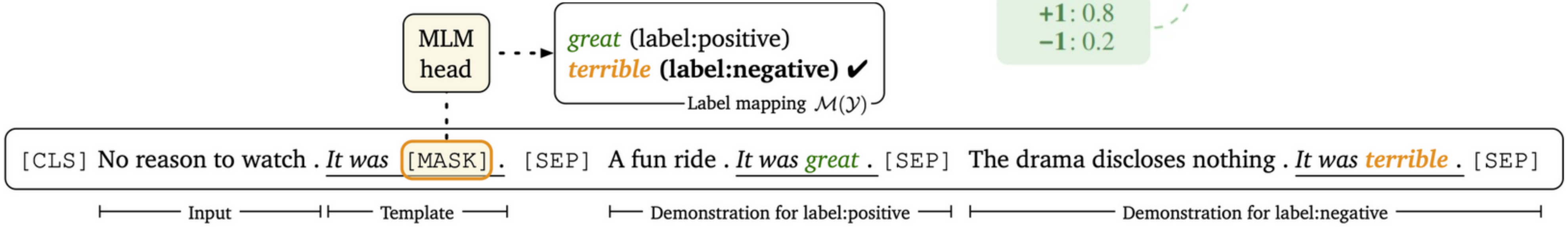
- Pretrained Language Models: Categorization by Architecture
- Training and Deployment of Language Models
  - Standard fine-tuning
  - Prompt-based methods
  - Parameter-efficient tuning
- Extending Language Models for Text-Rich Networks

Prompt-Based Fine-Tuning of PLMs

- Task descriptions are created to convert training examples to cloze questions
- Highly resemble the pretraining tasks (MLM) so that pretraining knowledge could be better leveraged
- Better than standard fine-tuning especially for few-shot settings







Schick, T., & Schütze, H. (2021). Exploiting cloze questions for few shot text classification and natural language inference. EACL.
Le Scao, T., & Rush, A. M. (2021). How many data points is a prompt worth? NAACL.

Prompt-Based Fine-Tuning of PLMs

- Further improve prompt-based few-shot fine-tuning:
  - Prompt templates and label words can be automatically generated
  - Demonstrations can be concatenated with target sequences to provide hints

	SST-2 (acc)	SST-5 (acc)	MR (acc)	CR (acc)	MPQA (acc)	Subj (acc)	TREC (acc)	CoLA (Matt.)
Majority†	*50.9*	*23.1*	*50.0*	*50.0*	*50.0*	*50.0*	*18.8*	*0.0*
Prompt-based zero-shot‡	83.6	35.0	80.8	79.5	67.6	51.4	32.0	2.0
"GPT-3" in-context learning	84.8 (1.3)	30.6 (0.9)	80.5 (1.7)	87.4 (0.8)	63.8 (2.1)	53.6 (1.0)	26.2 (2.4)	-1.5 (2.4)
Fine-tuning	81.4 (3.8)	43.9 (2.0)	76.9 (5.9)	75.8 (3.2)	72.0 (3.8)	90.8 (1.8)	88.8 (2.1)	**33.9** (14.3)
Prompt-based FT (man)	92.7 (0.9)	47.4 (2.5)	87.0 (1.2)	90.3 (1.0)	84.7 (2.2)	91.2 (1.1)	84.8 (5.1)	9.3 (7.3)
+ demonstrations	92.6 (0.5)	**50.6** (1.4)	86.6 (2.2)	90.2 (1.2)	**87.0** (1.1)	**92.3** (0.8)	87.5 (3.2)	18.7 (8.8)
Prompt-based FT (auto)	92.3 (1.0)	49.2 (1.6)	85.5 (2.8)	89.0 (1.4)	85.8 (1.9)	91.2 (1.1)	88.2 (2.0)	14.0 (14.1)
+ demonstrations	**93.0** (0.6)	49.5 (1.7)	**87.7** (1.4)	**91.0** (0.9)	86.5 (2.6)	91.4 (1.8)	**89.4** (1.7)	21.8 (15.9)
Fine-tuning (full)†	*95.0*	*58.7*	*90.8*	*89.4*	*87.8*	*97.0*	*97.4*	*62.6*

	MNLI (acc)	MNLI-mm (acc)	SNLI (acc)	QNLI (acc)	RTE (acc)	MRPC (F1)	QQP (F1)	STS-B (Pear.)
Majority†	*32.7*	*33.0*	*33.8*	*49.5*	*52.7*	*81.2*	*0.0*	-
Prompt-based zero-shot‡	50.8	51.7	49.5	50.8	51.3	61.9	49.7	-3.2
"GPT-3" in-context learning	52.0 (0.7)	53.4 (0.6)	47.1 (0.6)	53.8 (0.4)	60.4 (1.4)	45.7 (6.0)	36.1 (5.2)	14.3 (2.8)
Fine-tuning	45.8 (6.4)	47.8 (6.8)	48.4 (4.8)	60.2 (6.5)	54.4 (3.9)	76.6 (2.5)	60.7 (4.3)	53.5 (8.5)
Prompt-based FT (man)	68.3 (2.3)	70.5 (1.9)	77.2 (3.7)	64.5 (4.2)	69.1 (3.6)	74.5 (5.3)	65.5 (5.3)	71.0 (7.0)
+ demonstrations	**70.7** (1.3)	**72.0** (1.2)	**79.7** (1.5)	**69.2** (1.9)	68.7 (2.3)	77.8 (2.0)	**69.8** (1.8)	73.5 (5.1)
Prompt-based FT (auto)	68.3 (2.5)	70.1 (2.6)	77.1 (2.1)	68.3 (7.4)	**73.9** (2.2)	76.2 (2.3)	67.0 (3.0)	75.0 (3.3)
+ demonstrations	70.0 (3.6)	**72.0** (3.1)	77.5 (3.5)	68.5 (5.4)	71.1 (5.3)	**78.1** (3.4)	67.7 (5.8)	**76.4** (6.2)
Fine-tuning (full)†	*89.8*	*89.5*	*92.6*	*93.3*	*80.9*	*91.4*	*81.7*	*91.9*

Gao, T., Fisch, A., & Chen, D. (2021). Making pre-trained language models better few-shot learners. ACL

Prompt-Based Zero-Shot Inference

- Even without any training, knowledge can be extracted from PLMs through cloze patterns
- PLMs can serve as knowledge bases
  - Pros: require no schema engineering, and support an open set of queries
  - Cons: retrieved answers are not guaranteed to be accurate
- Could be used for unsupervised open-domain QA systems

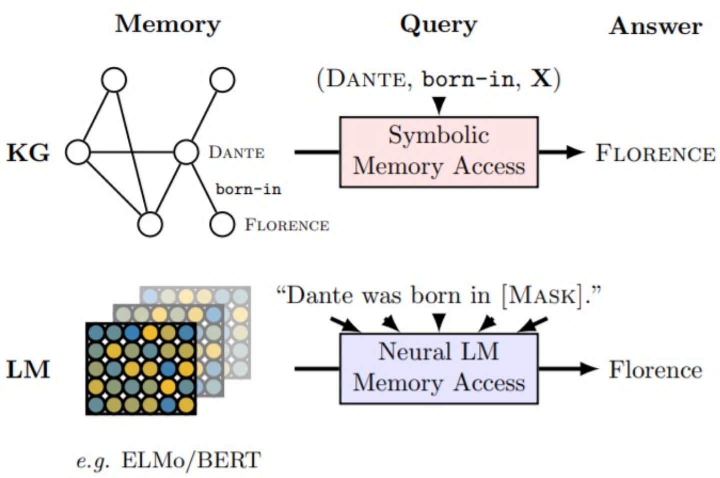
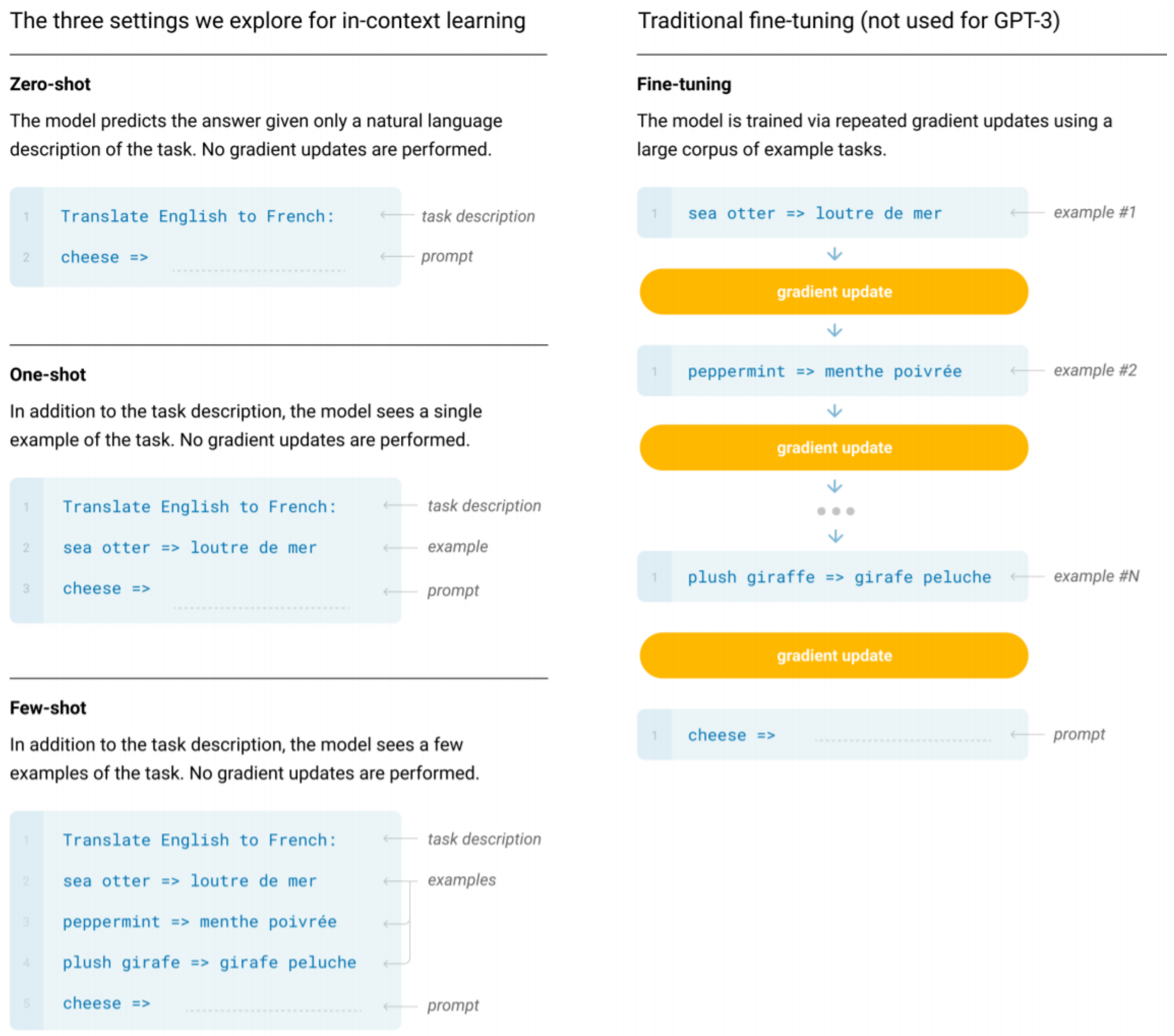


Figure 1: Querying knowledge bases (KB) and language models (LM) for factual knowledge.

Petroni, F., Rocktäschel, T., Lewis, P., Bakhtin, A., Wu, Y., Miller, A. H., & Riedel, S. (2019). Language models as knowledge bases? EMNLP.

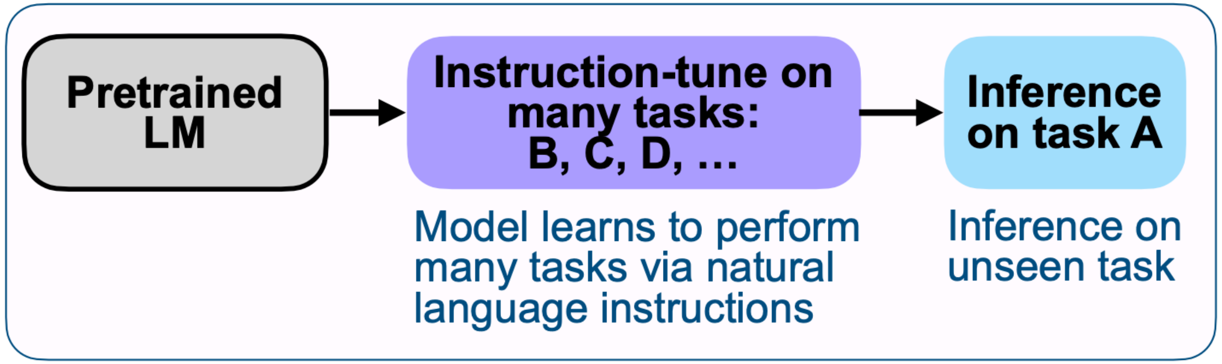
In-Context Learning: Few-Shot Inference

- Large PLMs (e.g., GPT-3) have strong few-shot learning ability **without** any tuning on large task-specific training sets
- Generate answers based on natural language descriptions and prompts

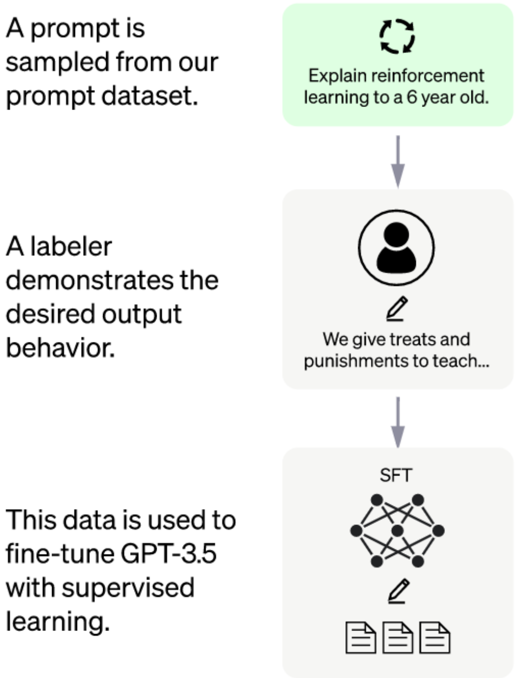


Instruction Tuning

- Prompt-based fine-tuning on various tasks/formats => generalization to unseen tasks/formats
- Applicable to build chatbots (e.g., ChatGPT) by tuning language models on dialogue input-response pairs



Wei, J., Bosma, M., Zhao, V., Guu, K., Yu, A.W., Lester, B., Du, N., Dai, A.M., & Le, Q.V. (2022). Finetuned Language Models Are Zero-Shot Learners. ICLR



Zhou, C., Liu, P., Xu, P., Iyer, S., Sun, J., Mao, Y., Ma, X., Efrat, A., Yu, P., Yu, L., Zhang, S., Ghosh, G., Lewis, M., Zettlemoyer, L., & Levy, O. (2023). LIMA: Less Is More for Alignment.

Outline

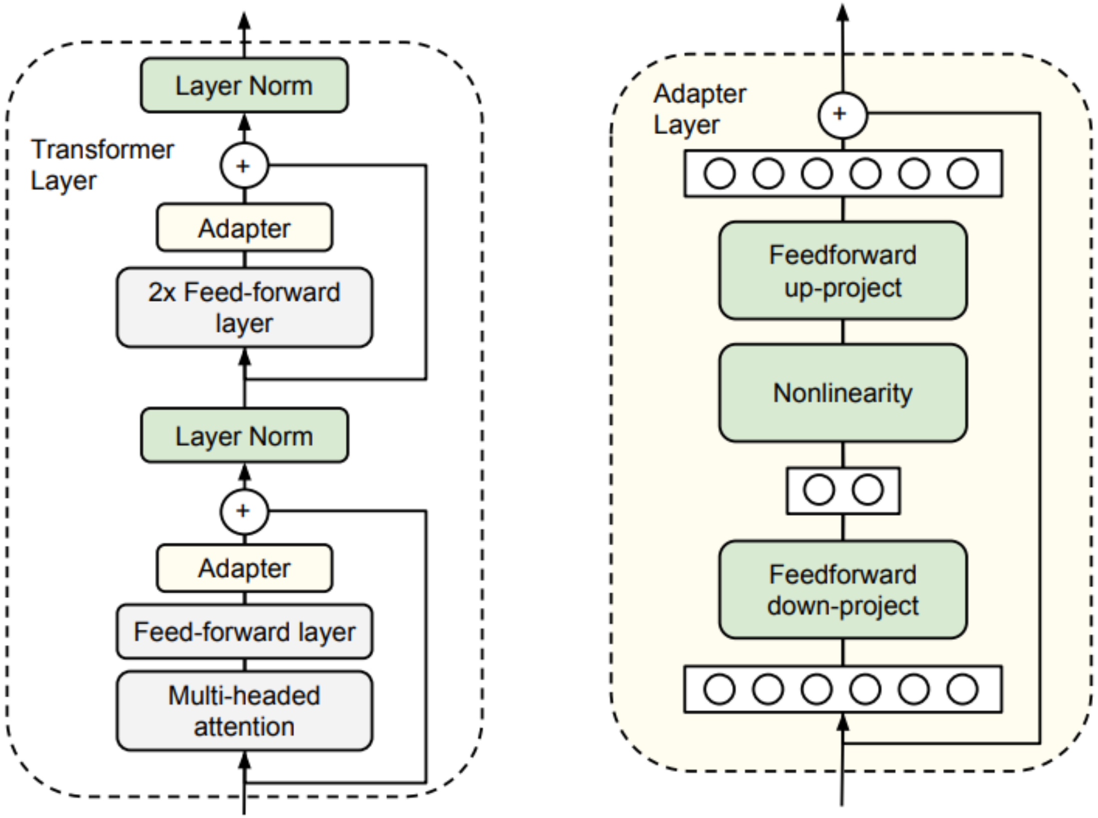
- Pretrained Language Models: Categorization by Architecture
- Training and Deployment of Language Models
  - Standard fine-tuning
  - Prompt-based methods
  - Parameter-efficient tuning
- Extending Language Models for Text-Rich Networks

Parameter-Efficient Tuning of PLMs

- Fine-tuning updates all PLM parameters at the same time
- Large PLMs can have an enormous amount of parameters that are costly to optimize
- Can we optimize only a small set of parameters in PLMs while still achieving comparable performance to fine-tuning?
- A few strategies:
  - Adapter: Insert small bottleneck modules and only update adapter + layer norm parameters
  - Prefix Tuning: Prepend tunable prefix vectors to every Transformer layer and keep other parameters unchanged
  - Low-Rank Adaptation: Use trainable low-rank matrices to approximate weight updates

Adapter for Parameter-Efficient Tuning

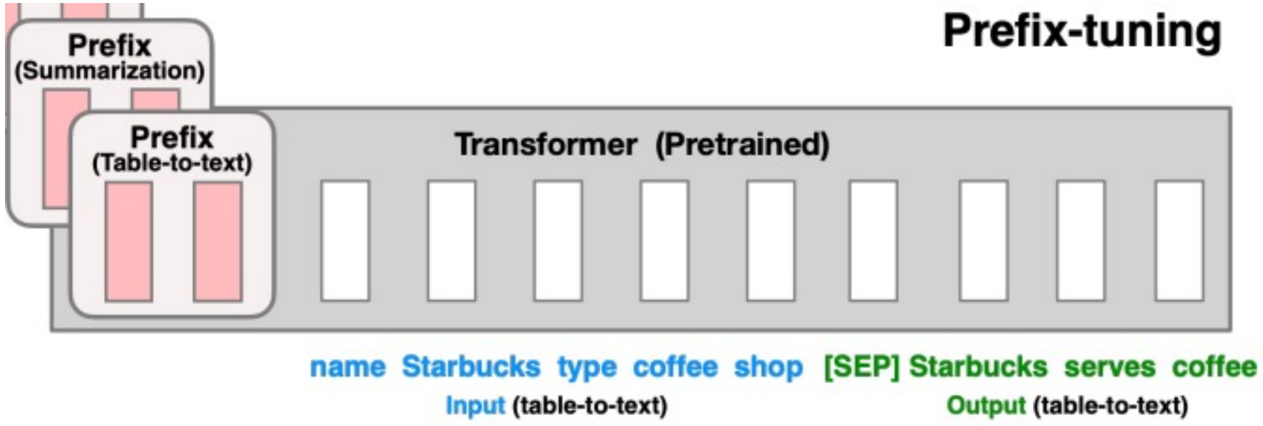
- Adapters are added twice to each Transformer layer
- Consist of a bottleneck structure (down-project + up-project)
- Only adapter parameters + layer norm parameters are updated during tuning



Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., ... & Gelly, S. (2019). Parameter-efficient transfer learning for NLP. ICML

Prefix Tuning

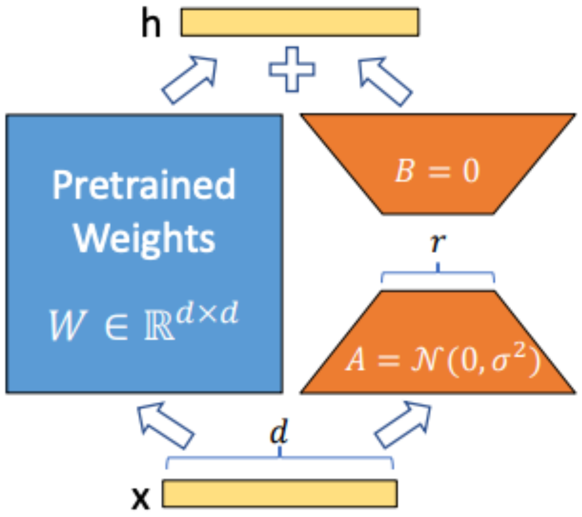
- Prefix tuning prepends trainable vectors to each Transformer layer
- Only update prefix vectors and keep other pretrained parameters unchanged
- Similar to prompt-based fine-tuning except that the prefix vectors are continuous parameters instead of natural language words



Li, X. L., & Liang, P. (2021). Prefix-tuning: Optimizing continuous prompts for generation. ACL.

Low-Rank Adaptation

- Inject trainable low-rank matrices into transformer layers to approximate the weight updates
- Since low-rank matrices have far less parameters than full-rank ones, training them is much more efficient than standard fine-tuning
- Can be used together with quantization techniques (e.g., QLoRA)



$$W_0 + \Delta W = W_0 + \boxed{BA}$$ A and B are low-rank matrices

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., ... & Chen, W. (2022). LoRA: Low-rank adaptation of large language models. ICLR.

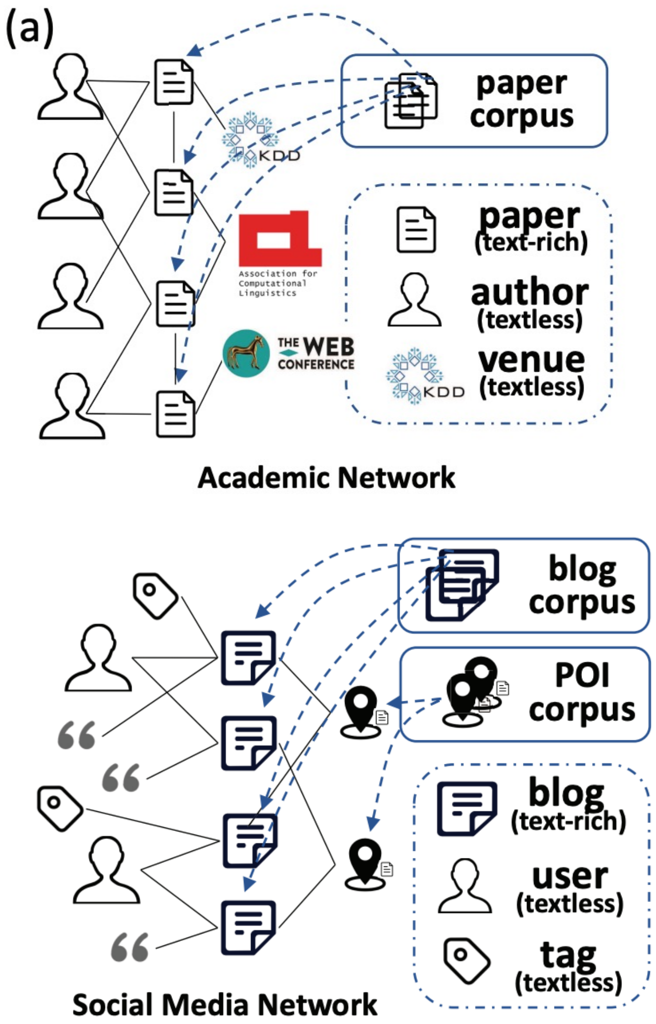
Dettmers, T., Pagnoni, A., Holtzman, A., & Zettlemoyer, L. (2023). QLoRA: Efficient Finetuning of Quantized LLMs.

Outline

- Pretrained Language Models: Categorization by Architecture
- Training and Deployment of Language Models
- Extending Language Models for Text-Rich Networks
  - Representation Learning on Homogeneous & Heterogeneous Text-Rich Networks
  - Language Model Pretraining on Text-Rich Networks

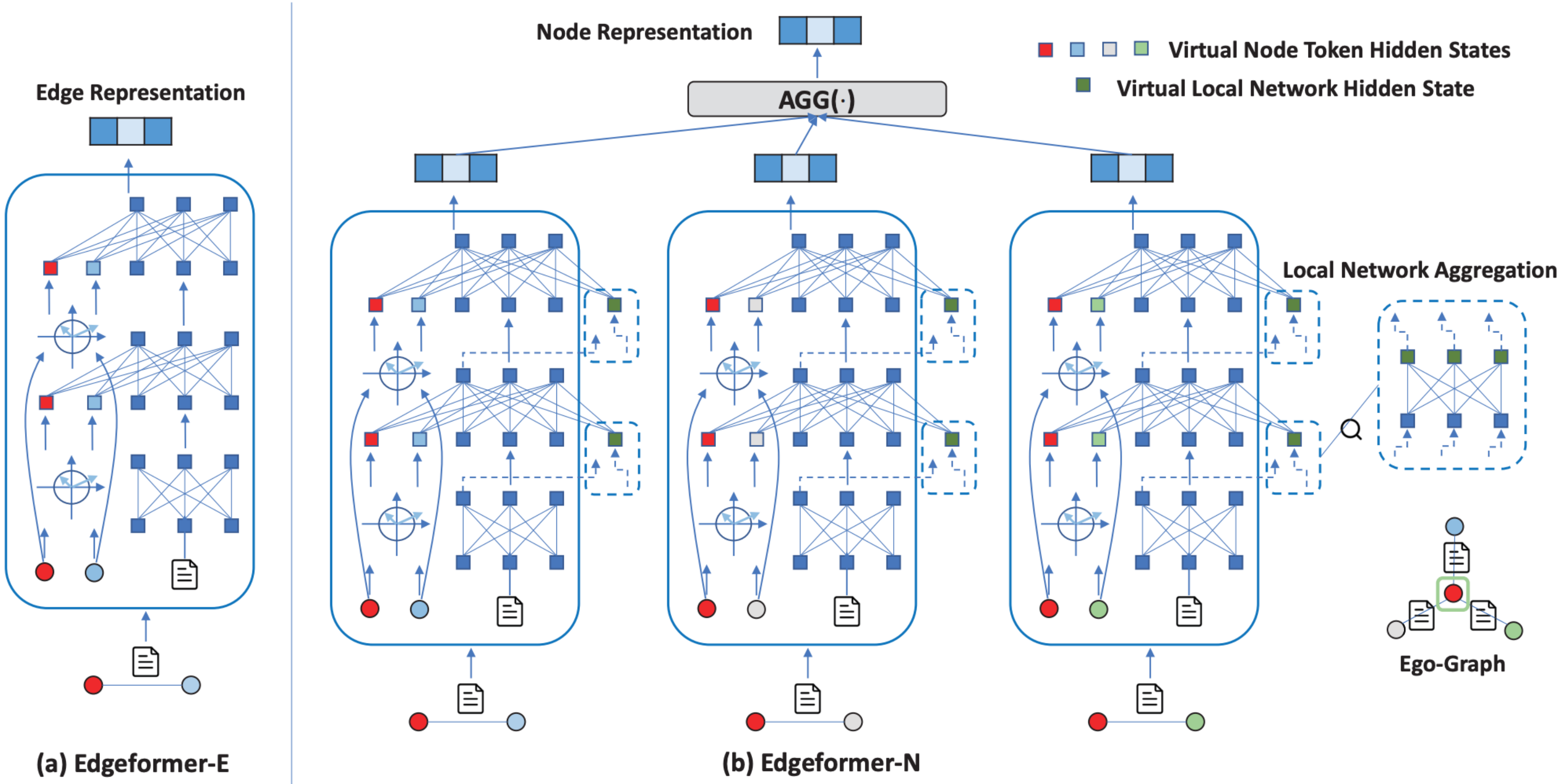
Homogeneous & Heterogeneous Text-Rich Network

- Why text-rich networks?
  - Texts may be connected via links & relations
  - **Text-rich networks**: Nodes/edges associated with textual information (e.g., review networks have users and items connected by review documents)
- Homogeneous vs. heterogeneous text-rich network:
  - Homogeneous: Nodes/edges in the network are single-typed
  - Heterogeneous: Nodes/edges in the network are multi-typed
- How to extend language models to consider both text semantics and structure information?



Edgeformers: Learning on Homogeneous Networks

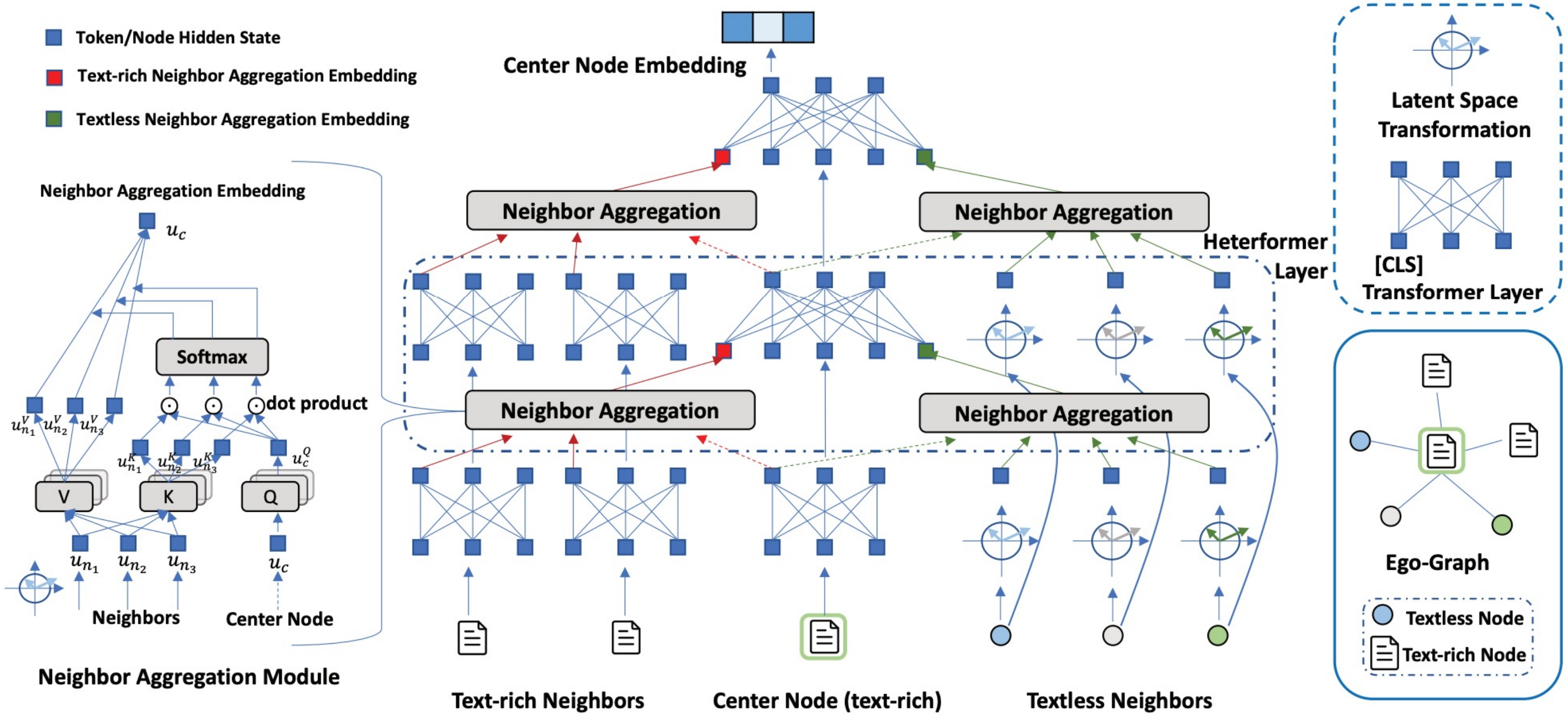
- Learning node and edge representations with virtual node tokens
- Node representations are based on aggregation of edge representations



Jin, B., Zhang, Y., Meng, Y., & Han, J. (2023). Edgeformers: Graph-Empowered Transformers for Representation Learning on Textual-Edge Networks. ICLR

Heterformer: Learning on Heterogeneous Networks

- Use virtual neighbor tokens inside each Transformer layer for text encoding
- Fuse representations of each node's text-rich neighbors, textless neighbors, and its own content via attention



Jin, B., Zhang, Y., Zhu, Q., & Han, J. (2023). Heterformer: A Transformer Architecture for Node Representation Learning on Heterogeneous Text-Rich Networks. KDD

Heterformer: Performance Study

	Method	DBLP PREC	DBLP MRR	DBLP NDCG	Twitter PREC	Twitter MRR	Twitter NDCG	Goodreads PREC	Goodreads MRR	Goodreads NDCG
	MeanSAGE	0.7019	0.7964	0.8437	0.6489	0.7450	0.7991	0.6302	0.7409	0.8001
	BERT	0.7569	0.8340	0.8726	0.7179	0.7833	0.8265	0.5571	0.6668	0.7395
Homo GNN	BERT+MeanSAGE	0.8131	0.8779	0.9070	0.7201	0.7845	0.8275	0.7301	0.8167	0.8594
Homo GNN	BERT+MAXSAGE	0.8193	0.8825	0.9105	0.7198	0.7845	0.8276	0.7280	0.8164	0.8593
Homo GNN	BERT+GAT	0.8119	0.8771	0.9063	0.7231	0.7873	0.8300	0.7333	0.8170	0.8593
Homo GNN	GraphFormers	0.8324	0.8916	0.9175	0.7258	0.7891	0.8312	0.7444	0.8260	0.8665
Hetero GNN	BERT+RGCN	0.7979	0.8633	0.8945	0.7111	0.7764	0.8209	0.7488	0.8303	0.8699
Hetero GNN	BERT+HAN	0.8136	0.8782	0.9072	0.7237	0.7880	0.8306	0.7329	0.8174	0.8597
Hetero GNN	BERT+HGT	0.8170	0.8814	0.9098	0.7153	0.7800	0.8237	0.7224	0.8112	0.8552
Hetero GNN	BERT+SHGN	0.8149	0.8785	0.9074	0.7218	0.7866	0.8295	0.7362	0.8195	0.8613
Hetero GNN	GraphFormers++	0.8233	0.8856	0.9130	0.7159	0.7799	0.8236	0.7536	0.8328	0.8717
	Heterformer	**0.8474***	**0.9019***	**0.9255***	**0.7272***	**0.7908***	**0.8328***	**0.7633***	**0.8400***	**0.8773***

Link prediction

Method	DBLP Micro-F1	DBLP Macro-F1	Goodreads Micro-F1	Goodreads Macro-F1
BERT	0.6119	0.5476	0.8364	0.7713
BERT+MaxSAGE	0.6179	0.5511	0.8447	0.7866
BERT+MeanSAGE	0.6198	0.5522	0.8420	0.7826
BERT+GAT	0.5943	0.5175	0.8328	0.7713
GraphFormers	0.6256	0.5616	0.8388	0.7786
BERT+HAN	0.5965	0.5211	0.8351	0.7747
BERT+HGT	0.6575	0.5951	0.8474	0.7928
BERT+SHGN	0.5982	0.5214	0.8345	0.7737
GraphFormers++	0.6474	0.5790	0.8516	0.7993
Heterformer	**0.6695***	**0.6062***	**0.8578***	**0.8076***

Node Classification
Left: transductive
Right: inductive

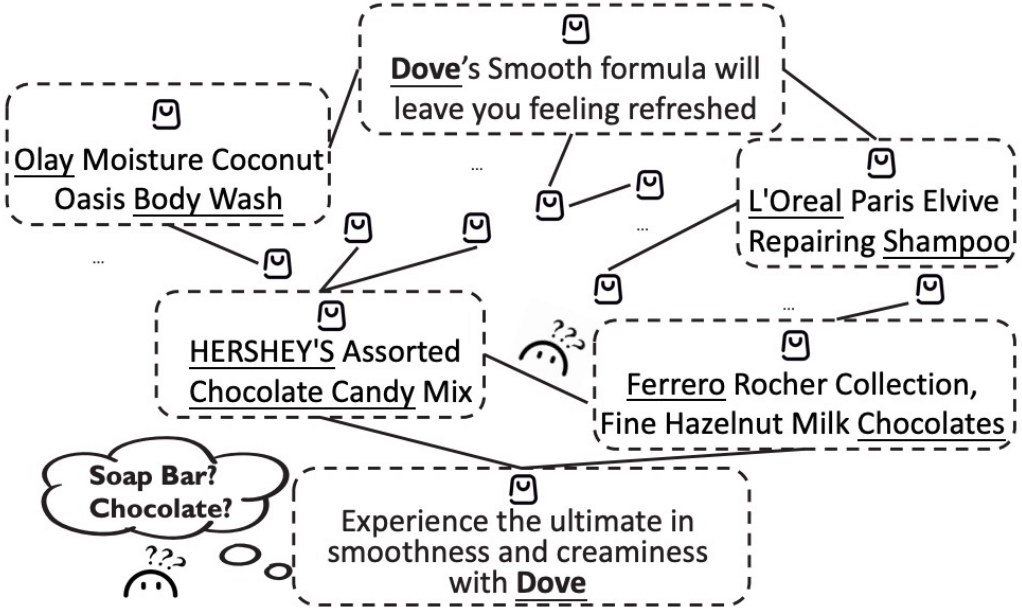
Method	DBLP Micro-F1	DBLP Macro-F1	Goodreads Micro-F1	Goodreads Macro-F1
BERT	0.5996	0.5318	0.8122	0.7371
BERT+MaxSAGE	0.6117	0.5435	0.8368	0.7749
BERT+MeanSAGE	0.6129	0.5431	0.8350	0.7721
BERT+GAT	0.5879	0.5150	0.8249	0.7590
GraphFormers	0.6197	0.5548	0.8330	0.7683
BERT+HAN	0.5948	0.5165	0.8279	0.7626
BERT+HGT	0.6467	0.5835	0.8390	0.7798
BERT+SHGN	0.5955	0.5202	0.8280	0.7626
GraphFormers++	0.6386	0.5696	0.8427	0.7848
Heterformer	**0.6600***	**0.5976***	**0.8507***	**0.7977***

Outline

- Pretrained Language Models: Categorization by Architecture
- Training and Deployment of Language Models
- Extending Language Models for Text-Rich Networks
  - Representation Learning on Heterogeneous & Homogeneous Text-Rich Networks
  - Language Model Pretraining on Text-Rich Networks

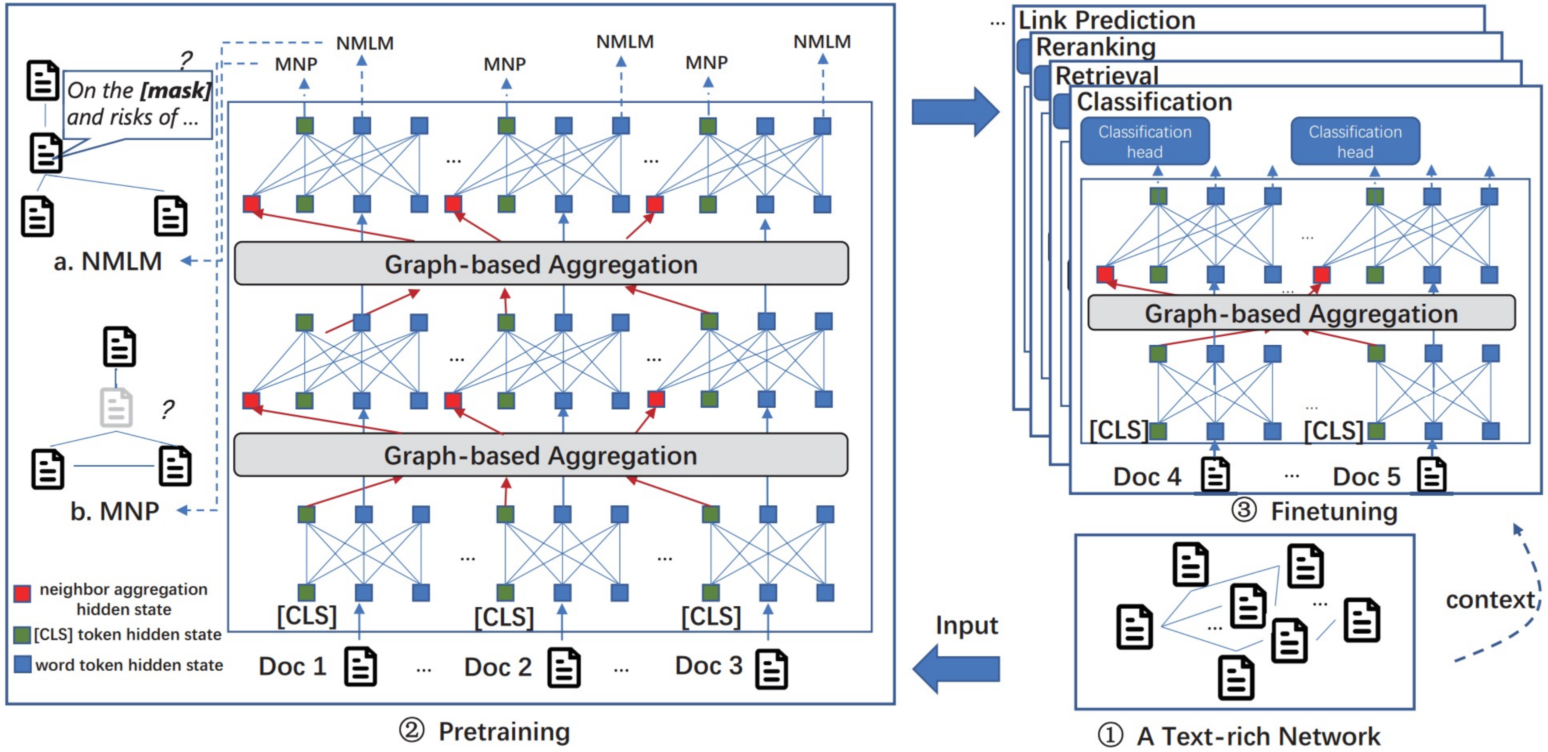
Pretraining on Text-Rich Networks

- Text understanding could depend on network structures!
  - “Hershey’s” should have some similarity with the chocolate from “Ferrero” based on the network structures
- How to pretrain representation models that effectively generalize to various tasks (e.g., link prediction, classification, retrieval) ?



Patton

- Two pretraining objectives
  - Network-contextualized masked language modeling (NMLM)
  - Masked node prediction (MNP)



Jin, B., Zhang, W., Zhang, Y., Meng, Y., Zhang, X., Zhu, Q., & Han, J. (2023). Patton: Language Model Pretraining on Text-Rich Networks. ACL

Patton: Performance Study

Method	Mathematics		Geology		Economics		Clothes		Sports	
	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1
BERT	18.14$_{0.07}$	22.04$_{0.32}$	21.97$_{0.87}$	29.63$_{0.36}$	14.17$_{0.08}$	19.77$_{0.12}$	45.10$_{1.47}$	68.54$_{2.25}$	31.88$_{0.23}$	34.58$_{0.56}$
GraphFormers	18.69$_{0.52}$	23.24$_{0.46}$	22.64$_{0.92}$	31.02$_{1.16}$	13.68$_{1.03}$	19.00$_{1.44}$	46.27$_{1.92}$	68.97$_{2.46}$	43.77$_{0.63}$	50.47$_{0.78}$
SciBERT	23.50$_{0.64}$	23.10$_{2.23}$	29.49$_{1.25}$	37.82$_{1.89}$	15.91$_{0.48}$	21.32$_{0.66}$	-	-	-	-
SPECTER	23.37$_{0.07}$	29.83$_{0.96}$	30.40$_{0.48}$	38.54$_{0.77}$	16.16$_{0.17}$	19.84$_{0.47}$	-	-	-	-
SimCSE (unsup)	20.12$_{0.08}$	26.11$_{0.39}$	38.78$_{0.19}$	38.55$_{0.17}$	14.54$_{0.26}$	19.07$_{0.43}$	42.70$_{2.32}$	58.72$_{0.34}$	41.91$_{0.85}$	59.19$_{0.55}$
SimCSE (sup)	20.39$_{0.07}$	25.56$_{0.00}$	25.66$_{0.28}$	33.89$_{0.40}$	15.03$_{0.53}$	18.64$_{1.32}$	52.82$_{0.87}$	75.54$_{0.98}$	46.69$_{0.10}$	59.19$_{0.55}$
LinkBERT	15.78$_{0.91}$	19.75$_{1.19}$	24.08$_{0.58}$	31.32$_{0.04}$	12.71$_{0.12}$	16.39$_{0.22}$	44.94$_{2.52}$	65.33$_{4.34}$	35.60$_{0.33}$	38.30$_{0.09}$
BERT.MLM	23.44$_{0.39}$	31.75$_{0.58}$	36.31$_{0.36}$	48.04$_{0.69}$	16.60$_{0.21}$	22.71$_{1.16}$	46.98$_{0.84}$	68.00$_{0.84}$	62.21$_{0.13}$	75.43$_{0.74}$
SciBERT.MLM	23.34$_{0.42}$	30.11$_{0.97}$	36.94$_{0.28}$	46.54$_{0.40}$	16.28$_{0.38}$	21.41$_{0.81}$	-	-	-	-
SimCSE.in-domain	25.15$_{0.09}$	29.85$_{0.20}$	38.91$_{0.08}$	48.93$_{0.14}$	18.08$_{0.22}$	23.79$_{0.44}$	57.03$_{0.20}$	80.16$_{0.31}$	65.57$_{0.35}$	75.22$_{0.18}$
PATTON	**27.58**$_{0.03}$	**32.82**$_{0.01}$	39.35$_{0.06}$	48.19$_{0.15}$	19.32$_{0.05}$	25.12$_{0.05}$	**60.14**$_{0.28}$	**84.88**$_{0.09}$	**67.57**$_{0.08}$	**78.60**$_{0.15}$
SciPATTON	27.35$_{0.04}$	31.70$_{0.01}$	**39.65**$_{0.10}$	**48.93**$_{0.06}$	**19.91**$_{0.08}$	**25.68**$_{0.32}$	-	-	-	-
w/o NMLM	25.91$_{0.45}$	27.79$_{2.07}$	38.78$_{0.19}$	48.48$_{0.17}$	18.86$_{0.23}$	24.25$_{0.26}$	56.68$_{0.24}$	80.27$_{0.17}$	65.83$_{0.28}$	76.24$_{0.54}$
w/o MNP	24.79$_{0.65}$	29.44$_{1.50}$	38.00$_{0.73}$	47.82$_{1.06}$	18.69$_{0.59}$	25.63$_{1.44}$	47.35$_{1.20}$	68.50$_{2.60}$	64.23$_{1.53}$	76.03$_{1.67}$

Node classification
(coarse-grained)

Method	Mathematics		Geology		Economics		Clothes		Sports	
	R@50	R@100	R@50	R@100	R@50	R@100	R@50	R@100	R@50	R@100
BM25	20.76	24.55	19.02	20.92	19.14	22.49	15.76	15.88	22.00	23.96
BERT	16.73$_{0.17}$	22.66$_{0.18}$	18.82$_{0.39}$	25.94$_{0.39}$	23.95$_{0.25}$	31.54$_{0.21}$	40.77$_{1.68}$	50.40$_{1.41}$	32.37$_{1.09}$	43.32$_{0.96}$
GraphFormers	16.65$_{0.12}$	22.41$_{0.10}$	18, 92$_{0.60}$	25.94$_{0.39}$	24.48$_{0.36}$	32.16$_{0.40}$	41.77$_{2.05}$	51.26$_{2.27}$	32.39$_{0.89}$	43.29$_{1.12}$
SciBERT	24.70$_{0.17}$	33.55$_{0.31}$	23.71$_{0.89}$	30.94$_{0.95}$	29.80$_{0.66}$	38.66$_{0.52}$	-	-	-	-
SPECTER	23.86$_{0.25}$	31.11$_{0.31}$	26.56$_{1.05}$	34.04$_{1.32}$	31.26$_{0.15}$	40.79$_{0.11}$	-	-	-	-
SimCSE (unsup)	17.91$_{0.26}$	23.19$_{0.29}$	20.45$_{0.20}$	26.82$_{0.26}$	25.83$_{0.23}$	33.42$_{0.28}$	44.90$_{0.35}$	54.76$_{0.38}$	38.81$_{0.35}$	49.30$_{0.44}$
SimCSE (sup)	20.29$_{0.41}$	26.23$_{0.51}$	22.34$_{0.49}$	29.63$_{0.55}$	28.07$_{0.38}$	36.51$_{0.37}$	44.69$_{0.59}$	54.70$_{0.77}$	40.31$_{0.43}$	50.55$_{0.41}$
LinkBERT	17.25$_{0.30}$	23.21$_{0.47}$	17.14$_{0.75}$	23.05$_{0.74}$	22.69$_{0.30}$	30.77$_{0.36}$	28.66$_{2.97}$	37.79$_{3.82}$	31.97$_{0.54}$	41.77$_{0.67}$
BERT.MLM	20.69$_{0.21}$	27.17$_{0.25}$	32.13$_{0.36}$	41.74$_{0.42}$	27.13$_{0.04}$	36.00$_{0.14}$	52.41$_{1.71}$	63.72$_{1.79}$	54.10$_{0.81}$	63.14$_{0.83}$
SciBERT.MLM	20.65$_{0.21}$	27.67$_{0.32}$	31.65$_{0.71}$	40.52$_{0.76}$	29.23$_{0.67}$	39.18$_{0.73}$	-	-	-	-
SimCSE.in-domain	24.54$_{0.05}$	31.66$_{0.09}$	33.97$_{0.07}$	44.09$_{0.19}$	28.44$_{0.31}$	37.81$_{0.27}$	61.42$_{0.84}$	72.25$_{0.86}$	53.77$_{0.22}$	63.73$_{0.30}$
PATTON	27.44$_{0.15}$	34.97$_{0.21}$	34.94$_{0.23}$	45.01$_{0.28}$	32.10$_{0.51}$	42.19$_{0.62}$	**68.62**$_{0.38}$	**77.54**$_{0.19}$	**58.63**$_{0.31}$	**68.53**$_{0.55}$
SciPATTON	**31.40**$_{0.52}$	**40.38**$_{0.66}$	**40.69**$_{0.52}$	**51.31**$_{0.48}$	**35.82**$_{0.69}$	46.05$_{0.69}$	-	-	-	-
w/o NMLM	30.85$_{0.14}$	39.89$_{0.23}$	39.29$_{0.07}$	49.59$_{0.11}$	35.17$_{0.31}$	**46.07**$_{0.20}$	65.60$_{0.26}$	75.19$_{0.32}$	57.05$_{0.14}$	67.22$_{0.12}$
w/o MNP	22.47$_{0.07}$	30.20$_{0.15}$	31.28$_{0.89}$	40.54$_{0.97}$	29.54$_{0.36}$	39.57$_{0.57}$	60.20$_{0.73}$	69.85$_{0.52}$	51.73$_{0.41}$	60.35$_{0.78}$

Node classification via retrieval
(fine-grained)

References I

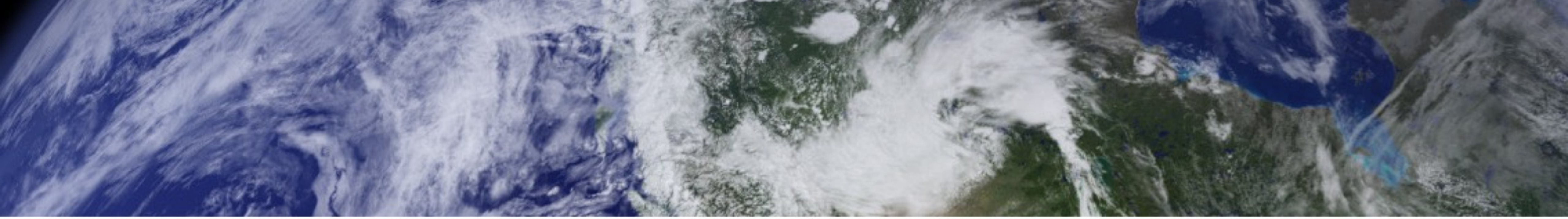
- Abu-El-Haija, S., Perozzi, B., Al-Rfou', R., & Alemi, A.A. (2018). Watch Your Step: Learning Node Embeddings via Graph Attention. NeurIPS.
- Anil, R. et al. (2023). PaLM 2 Technical Report.
- Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2016). Enriching Word Vectors with Subword Information. Transactions of the Association for Computational Linguistics, 5, 135-146.
- Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. NeurIPS.
- Chowdhery, A. et al. (2022) PaLM: Scaling Language Modeling with Pathways.
- Clark, K., Luong, M. T., Le, Q. V., & Manning, C. D. (2020). ELECTRA: Pre-training text encoders as discriminators rather than generators. ICLR.
- Dettmers, T., Pagnoni, A., Holtzman, A., & Zettlemoyer, L. (2023). QLoRA: Efficient Finetuning of Quantized LLMs.
- Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. NAACL-HLT.
- Gao, T., Fisch, A., & Chen, D. (2021). Making pre-trained language models better few-shot learners. ACL
- Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., ... & Gelly, S. (2019). Parameter-efficient transfer learning for NLP. ICML
- Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., ... & Chen, W. (2022). LoRA: Low-rank adaptation of large language models. ICLR.
- Jin, B., Zhang, Y., Zhu, Q., & Han, J. (2023). Heterformer: A Transformer Architecture for Node Representation Learning on Heterogeneous Text-Rich Networks. KDD
- Jin, B., Zhang, Y., Meng, Y., & Han, J. (2023). Edgeformers: Graph-Empowered Transformers for Representation Learning on Textual-Edge Networks. ICLR
- Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2020). Albert: A lite bert for self-supervised learning of language representations. ICLR.
- Le Scao, T., & Rush, A. M. (2021). How many data points is a prompt worth? NAACL.

References II

- Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., ... & Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. ACL.
- Li, X. L., & Liang, P. (2021). Prefix-tuning: Optimizing continuous prompts for generation. ACL.
- Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., & Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. NIPS.
- Mikolov, T., Chen, K., Corrado, G.S., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. CoRR, abs/1301.3781.
- Meng, Y., Huang, J., Wang, G., Zhang, C., Zhuang, H., Kaplan, L.M., & Han, J. (2019). Spherical Text Embedding. NeurIPS.
- Meng, Y., Xiong, C., Bajaj, P., Bennett, P., Han, J., & Song, X. (2021). COCO-LM: Correcting and contrasting text sequences for language model pretraining. NeurIPS.
- Meng, Y., Xiong, C., Bajaj, P., Bennett, P. N., Han, J., & Song, X. (2022). Pretraining Text Encoders with Adversarial Mixture of Training Signal Generators. ICLR.
- Nickel, M., & Kiela, D. (2017). Poincaré Embeddings for Learning Hierarchical Representations. NIPS.
- Nickel, M., & Kiela, D. (2018). Learning Continuous Hierarchies in the Lorentz Model of Hyperbolic Geometry. ICML.
- OpenAI (2023). GPT-4 Technical Report.
- Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C.L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L.E., Simens, M., Askell, A., Welinder, P., Christiano, P.F., Leike, J., & Lowe, R.J. (2022). Training language models to follow instructions with human feedback.

References III

- Pennington, J., Socher, R., & Manning, C.D. (2014). Glove: Global Vectors for Word Representation. EMNLP.
- Peters, M.E., Neumann, M., Iyyer, M., Gardner, M.P., Clark, C., Lee, K., & Zettlemoyer, L.S. (2018). Deep contextualized word representations. NAACL.
- Petroni, F., Rocktäschel, T., Lewis, P., Bakhtin, A., Wu, Y., Miller, A. H., & Riedel, S. (2019). Language models as knowledge bases? EMNLP.
- Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training. OpenAI blog.
- Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. OpenAI blog, 1(8), 9.
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. JMLR.
- Schick, T., & Schütze, H. (2021). Exploiting cloze questions for few shot text classification and natural language inference. EACL.
- Tifrea, A., Bécigneul, G., & Ganea, O. (2019). Poincare Glove: Hyperbolic Word Embeddings. ICLR.
- Touvron, H et al. LLaMA: Open and Efficient Foundation Language Models
- Touvron, H et al. LLaMA 2: Open Foundation and Fine-Tuned Chat Models
- Turian, J.P., Ratinov, L., & Bengio, Y. (2010). Word Representations: A Simple and General Method for Semi-Supervised Learning. ACL.
- Wei, J., et al. (2022). Emergent Abilities of Large Language Models. TMLR.
- Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., & Le, Q. V. (2019). XLNet: Generalized Autoregressive Pretraining for Language Understanding. NeurIPS.
- Zhou, C., Liu, P., Xu, P., Iyer, S., Sun, J., Mao, Y., Ma, X., Efrat, A., Yu, P., Yu, L., Zhang, S., Ghosh, G., Lewis, M., Zettlemoyer, L., & Levy, O. (2023). LIMA: Less Is More for Alignment.



Q&A