Part I: A Brief Introduction to Pretrained Language Models

WWW 2023 Tutorial

Turning Web-Scale Texts to Knowledge: Transferring Pretrained Representations to Text Mining Applications

Yu Meng, Jiaxin Huang, Yu Zhang, Jiawei Han

Computer Science, University of Illinois at Urbana-Champaign

Tutorial Website:

April 30, 2023

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

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

Categorization of Pretrained Language Models

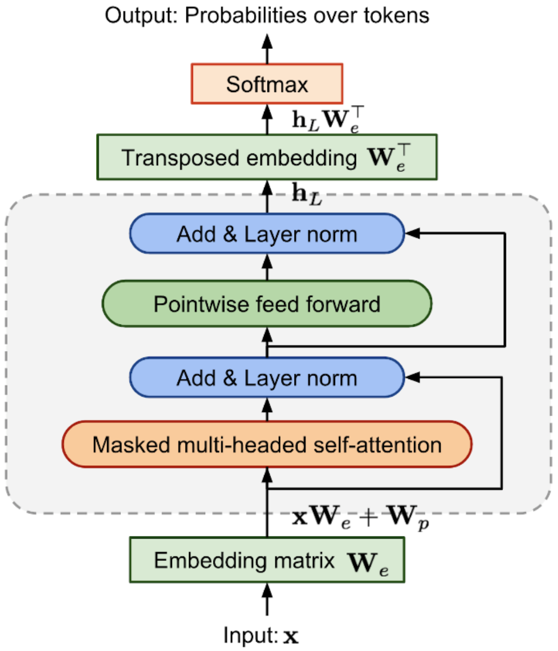
- There are multiple ways to categorize PLMs
  - By pretraining objectives: Standard language modeling, masked language modeling, permuted language modeling…
  - By pretraining settings: Multilingual, knowledge-enriched, domain-specific…
- In this presentation, we categorize PLMs **by architecture** 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)
  - **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

GPT-Style Pretraining: Introduction

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



k previous tokens as context

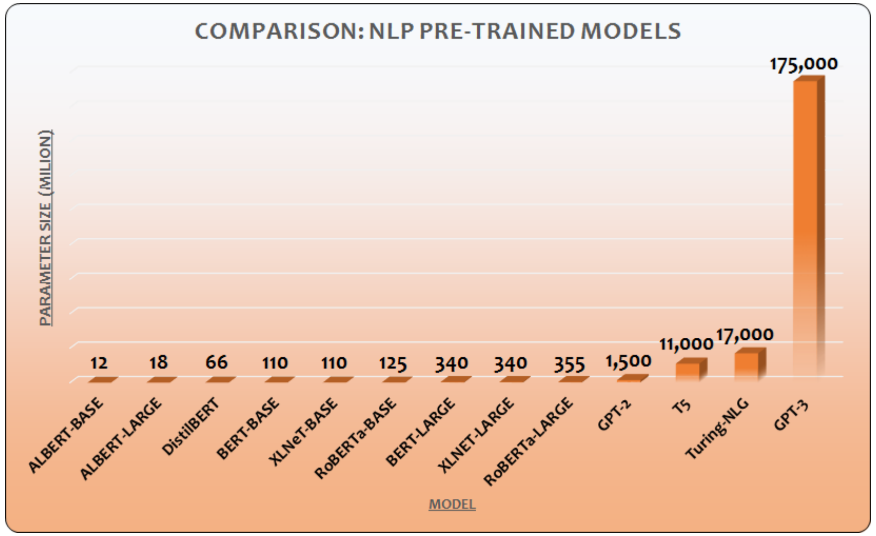
$$\mathcal{L}_{\mathrm{LM}} = -\sum_i \log p(x_i \mid \boxed{x_{i-k}, \ldots, x_{i-1}})$$

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

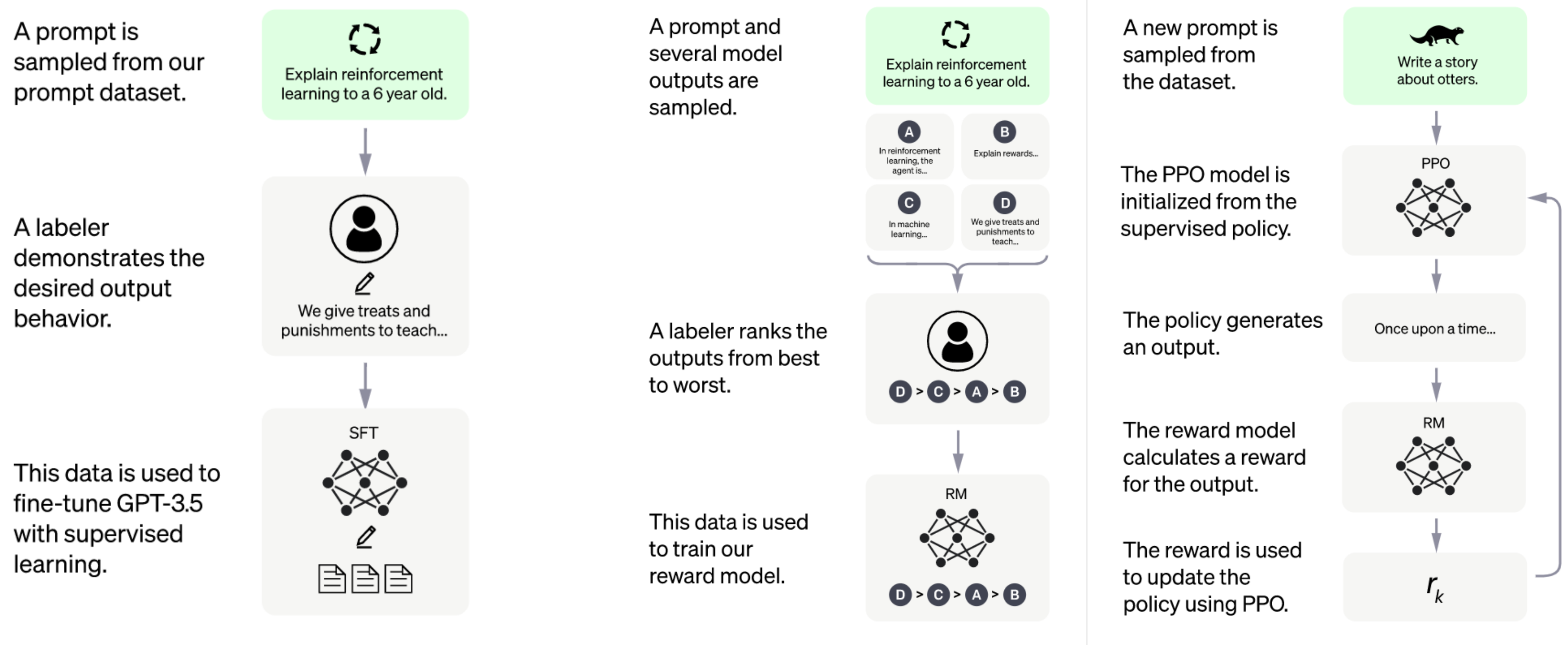
[1] Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training. OpenAI blog
[2] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. OpenAI blog, 1(8), 9.
[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!) and have very strong text generation abilities



ChatGPT: GPT + Instruction Tuning + RLHF



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

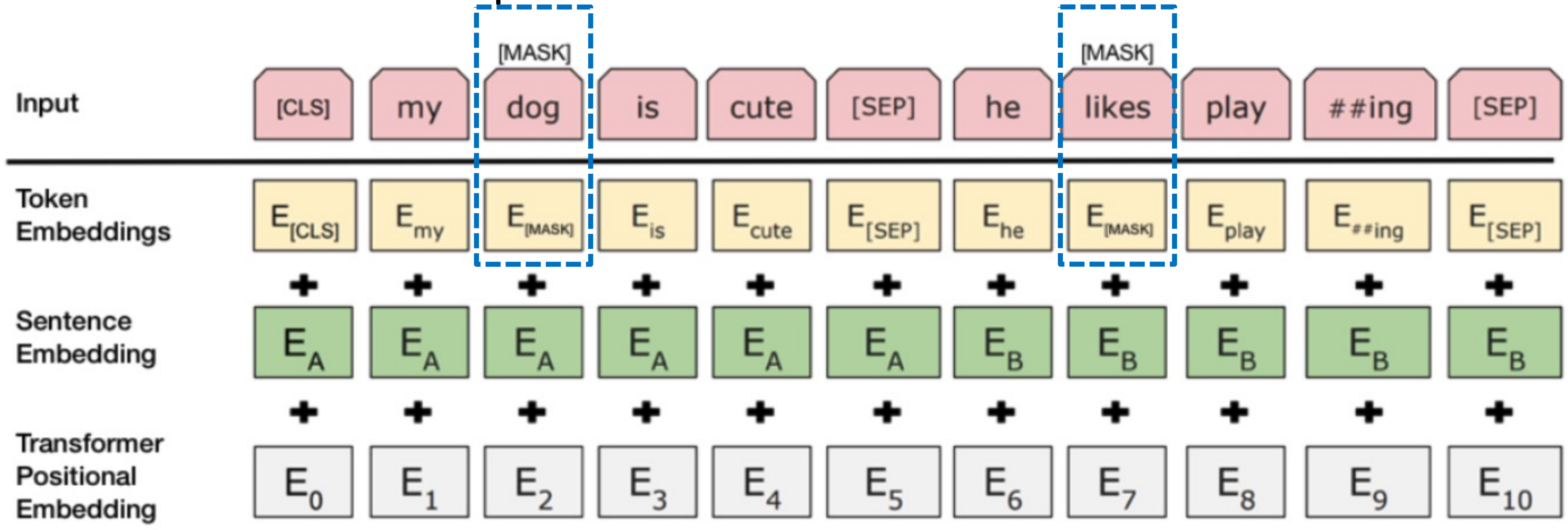
Reinforcement Learning from Human Feedback (RLHF): Train a reward model on human preferences of generation results; tune the generator to maximize reward

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

BERT: Masked Language Modeling

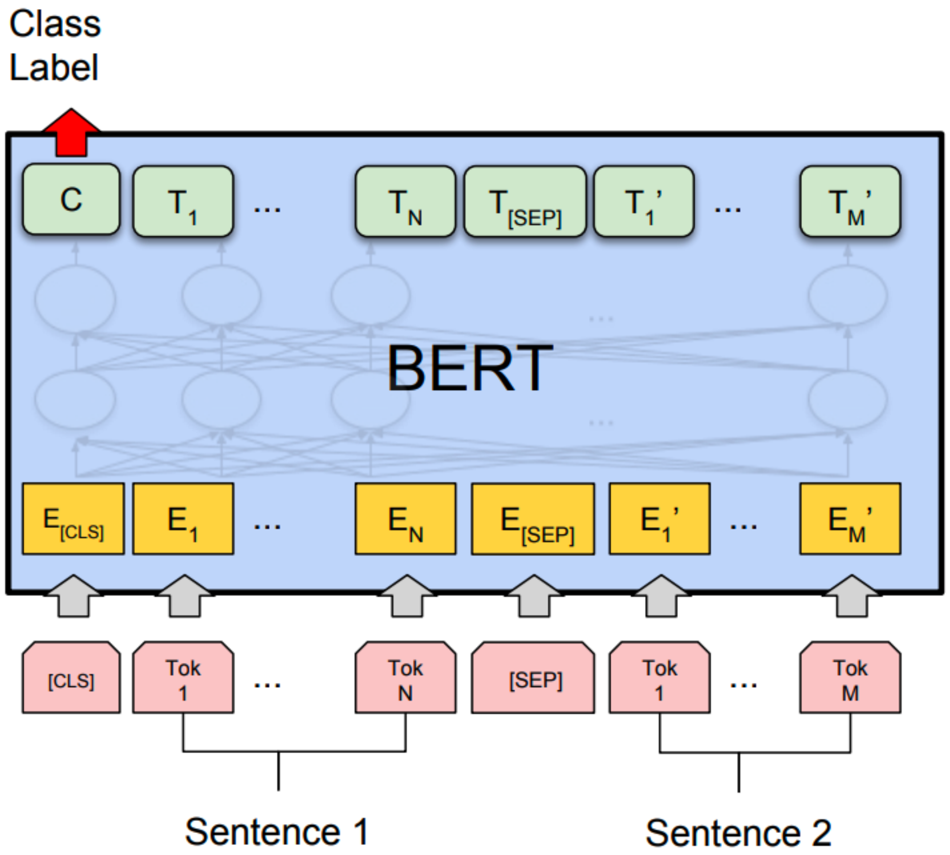
- Bidirectional: BERT leverages a Masked LM learning to introduce **real bidirectionality** training
- Masked LM: With 15% words randomly masked, the model learns bidirectional contextual information to predict the masked words



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

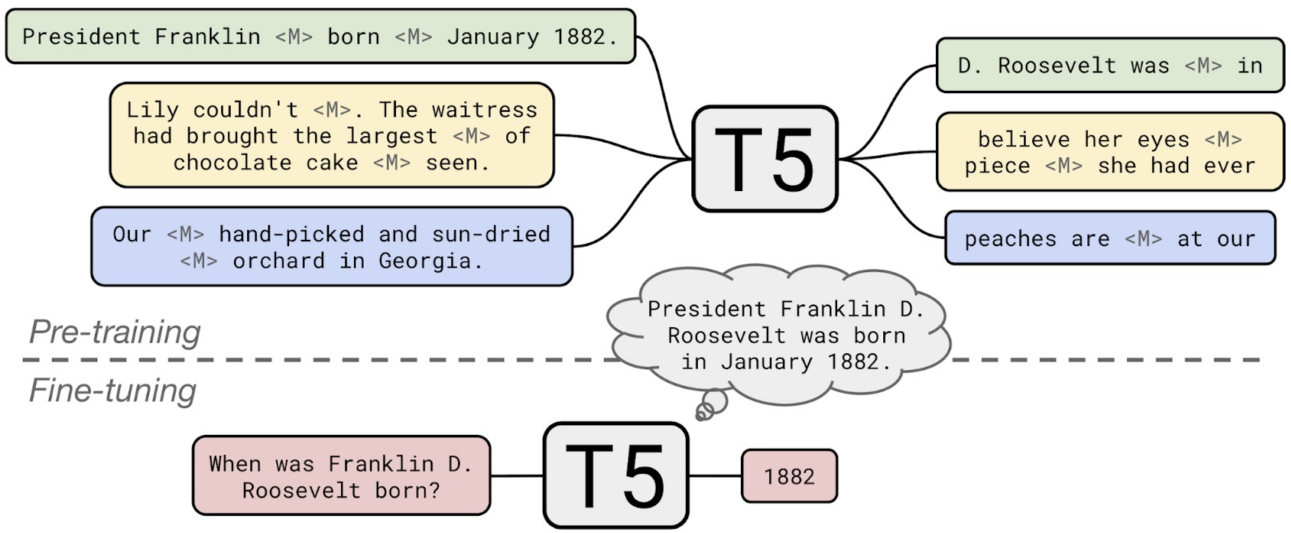
- RoBERTa (Liu et al. 2019): Pretrain BERT on more data for longer, without next sentence prediction
- XLNet (Yang et al. 2019): Permutation language modeling with two-stream self-attention
- ALBERT (Lan et al. 2020): Shared Transformer parameters across layers for parameter efficiency
- ELECTRA (Clark 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

T5

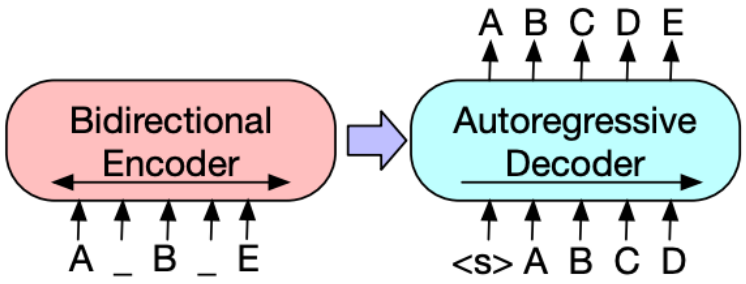
- T5: **T**ext-**t**o-**T**ext **T**ransfer **T**ransformer
- Pretraining: Mask out spans of texts; generate the original spans
- Fine-Tuning: Convert every task into a sequence-to-sequence generation problem



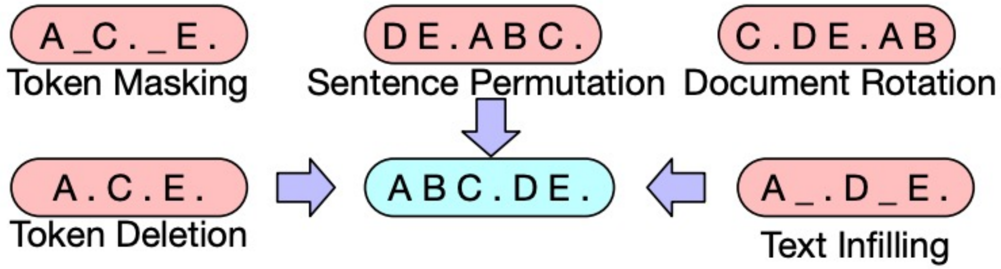
Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. JMLR.

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

Deployment of Pretrained Language Models

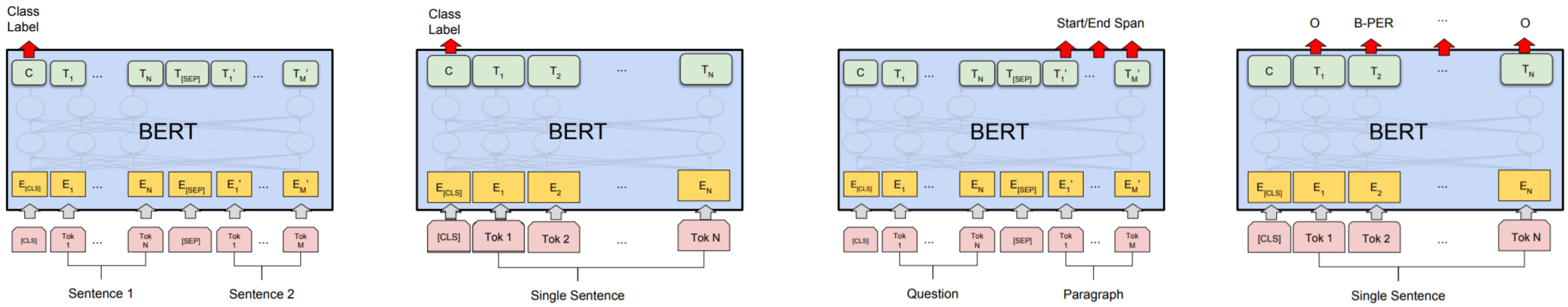
- Pretrained language models (PLMs) are usually trained on large-scale general domain corpora to learn generic linguistic features that can be transferred to downstream tasks
- Common usages of PLMs in downstream tasks
  - Fine-tuning: Update all parameters in the PLM encoder and task-specific layers (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
  - Reinforcement learning from human feedback:Reinforce good actions (i.e., generation results) with a reward function

Outline

- Pretrained Language Models: Categorization by Architecture
- Training and Deployment of Language Models
  - Standard fine-tuning
  - Prompt-based methods

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

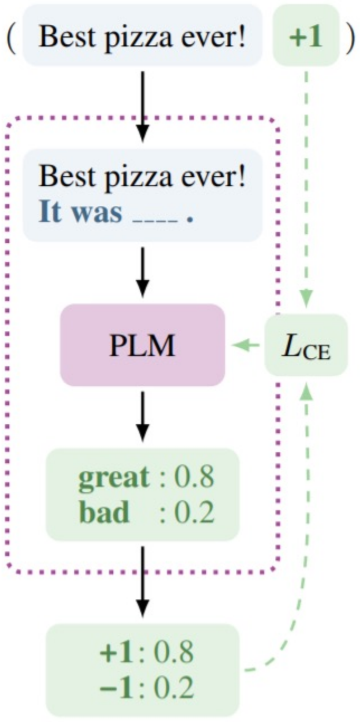
(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

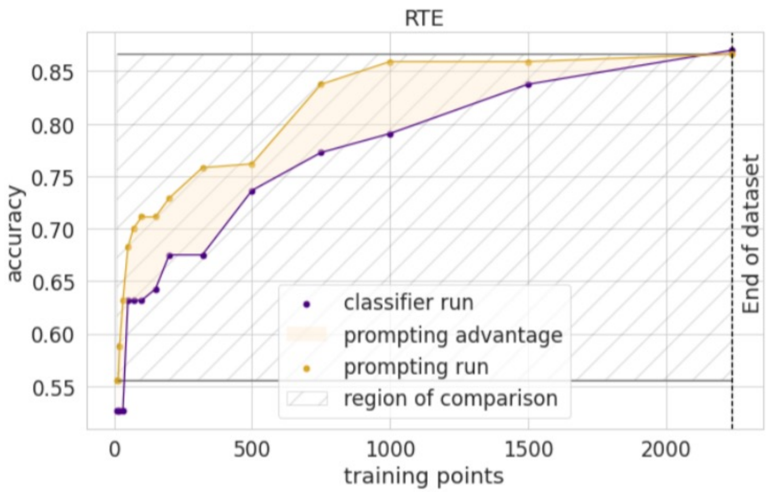
Outline

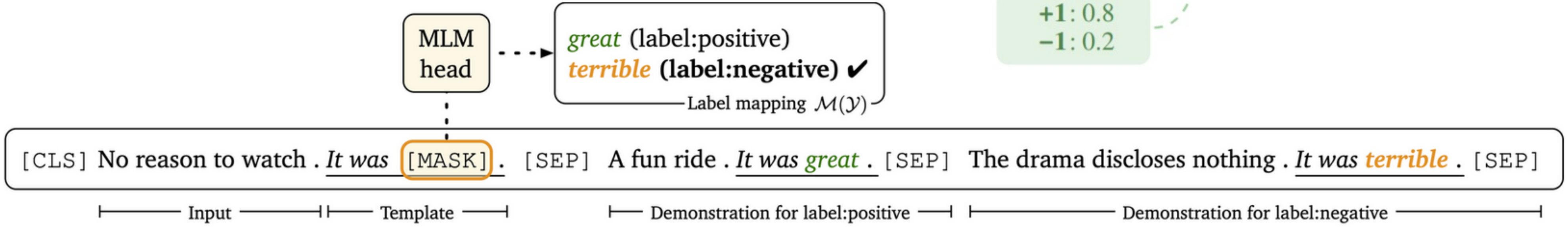
- Pretrained Language Models: Categorization by Architecture
- Training and Deployment of Language Models
  - Standard fine-tuning
  - Prompt-based methods

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

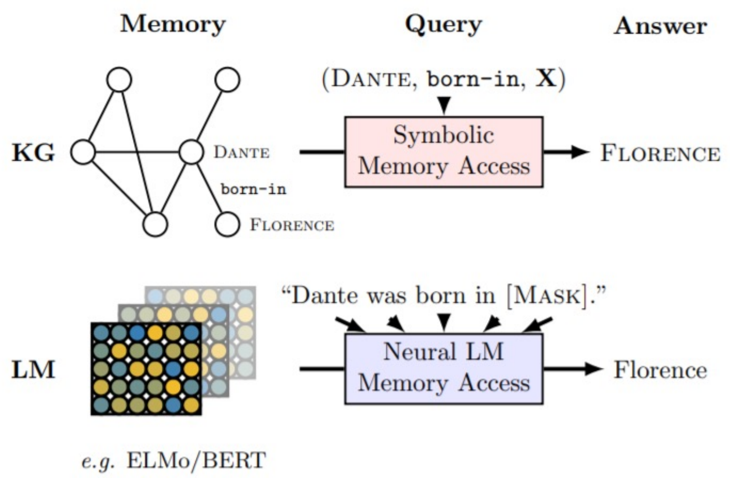
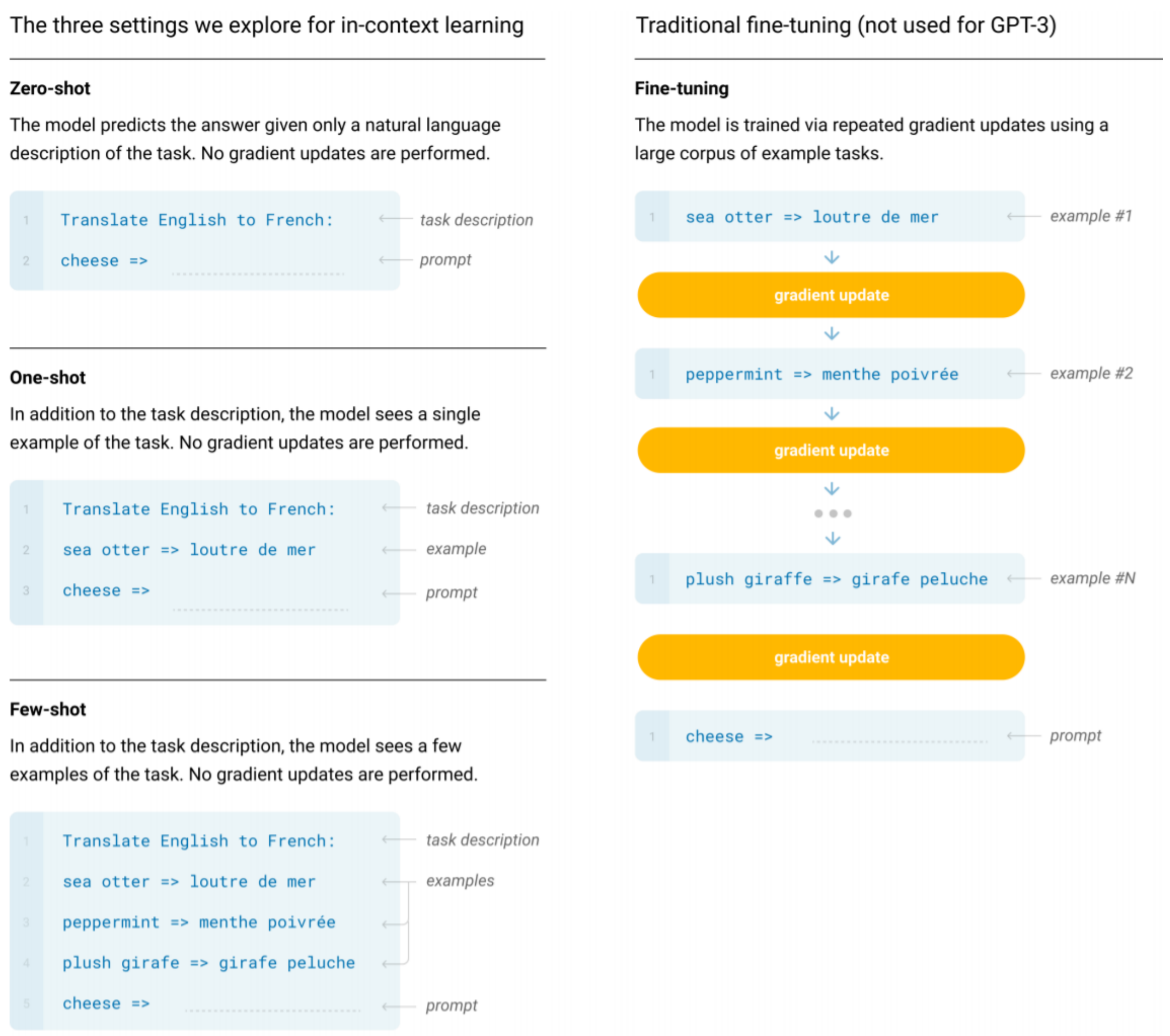


Figure 1: Querying knowledge bases (KB) and language models (LM) for factual knowledge.

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



Zero-Shot Fine-Tuning of PLMs

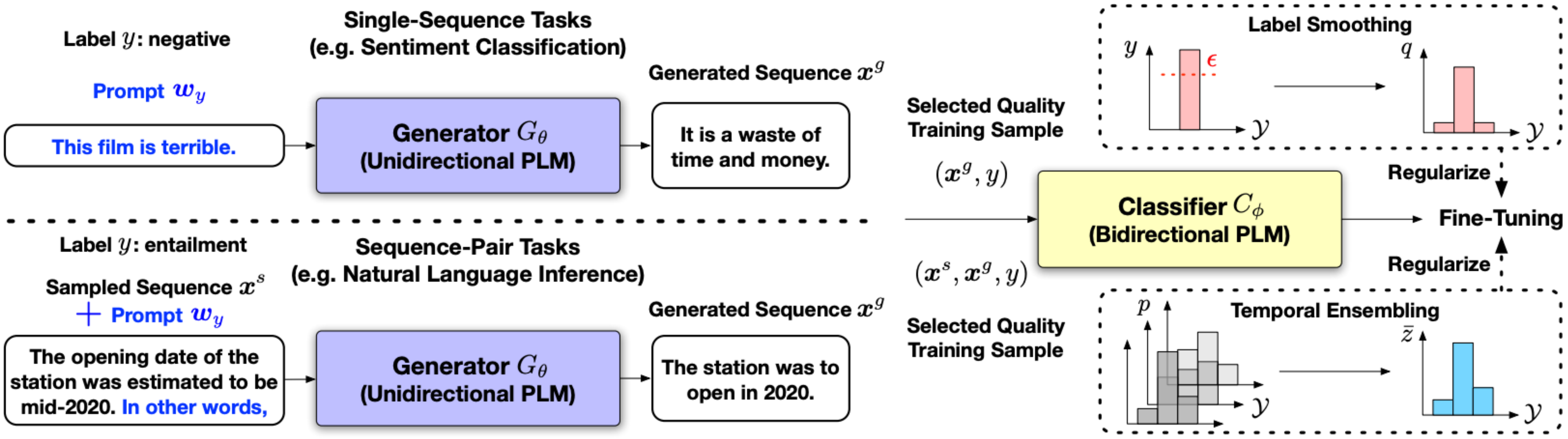
- Prompt-based approaches have remarkable few-shot fine-tuning performance, but their zero-shot performance is significantly worse
- Without any task-specific samples, it is challenging for PLMs to interpret the prompts that come in different formats and are unseen in the pretraining data
- The current mainstream of zero-shot learning is based on transfer learning
  - Train PLMs on a large variety of different tasks with abundant annotations, and transfer to unseen tasks
  - Require many **cross-task annotations** and **gigantic model sizes** which are not practical for common application scenarios

Zero-Shot Fine-Tuning of PLMs

- Can we do fully zero-shot learning, without any task-related or cross-task annotations?
- When there are no training data, we can create them from scratch using PLMs!
- Humans can generate training data pertaining to a specific label upon given a label-descriptive prompt (e.g., “write a negative review:”)
- We can leverage the strong text generation power of PLMs to do the same job

Prompt-Based Zero-Shot Training Data Generation

- SuperGen: A **Super**vision **Gen**eration approach
- Use a unidirectional PLM to generate class-conditioned texts guided by prompts
- Fine-tune a bidirectional PLM on the generated data for the corresponding task



Meng, Y., Huang, J., Zhang, Y., & Han, J. (2022). Generating Training Data with Language Models: Towards Zero-Shot Language Understanding. NeurIPS.

Zero-Shot Learning Results

- Using the same prompt-based fine-tuning method, zero-shot SuperGen (fine-tuned on generated training data) is comparable or even better than strong few-shot methods (fine-tuned on 32 manually annotated training samples per class)

Method	MNLI-(m/mm) (Acc.)	QQP (F1)	QNLI (Acc.)	SST-2 (Acc.)	CoLA (Matt.)	RTE (Acc.)	MRPC (F1)	AVG
Zero-Shot Setting: No task-specific data (neither labeled nor unlabeled).								
Prompting†	$50.8_{0.0}$/$51.7_{0.0}$	$49.7_{0.0}$	$50.8_{0.0}$	$83.6_{0.0}$	$2.0_{0.0}$	$51.3_{0.0}$	$61.9_{0.0}$	50.1
SuperGen	$\mathbf{72.3}_{0.5}$/$\mathbf{73.8}_{0.5}$	$\mathbf{66.1}_{1.1}$	$\mathbf{73.3}_{1.9}$	$\mathbf{92.8}_{0.6}$	$\mathbf{32.7}_{5.5}$	$\mathbf{65.3}_{1.2}$	$82.2_{0.5}$	**69.4**
- data selection	$63.7_{1.5}$/$64.2_{1.6}$	$62.3_{2.2}$	$63.9_{3.2}$	$91.3_{2.0}$	$30.5_{8.8}$	$62.4_{1.5}$	$81.6_{0.2}$	65.1
- label smooth	$70.7_{0.8}$/$72.1_{0.7}$	$65.1_{0.9}$	$71.4_{2.5}$	$91.0_{0.9}$	$9.5_{1.0}$	$64.8_{1.1}$	$\mathbf{83.0}_{0.7}$	65.2
- temporal ensemble	$62.0_{4.6}$/$63.6_{4.8}$	$63.9_{0.3}$	$72.4_{2.0}$	$92.5_{0.9}$	$23.5_{7.0}$	$63.5_{1.0}$	$78.8_{2.2}$	65.3
Few-Shot Setting: Use 32 labeled samples/class (half for training and half for development).								
Fine-tuning†	$45.8_{6.4}$/$47.8_{6.8}$	$60.7_{4.3}$	$60.2_{6.5}$	$81.4_{3.8}$	$\mathbf{33.9}_{14.3}$	$54.4_{3.9}$	$76.6_{2.5}$	59.1
Manual prompt†	$68.3_{2.3}$/$70.5_{1.9}$	$65.5_{5.3}$	$64.5_{4.2}$	$92.7_{0.9}$	$9.3_{7.3}$	$69.1_{3.6}$	$74.5_{5.3}$	63.6
+ demonstration†	$\mathbf{70.7}_{1.3}$/$\mathbf{72.0}_{1.2}$	$\mathbf{69.8}_{1.8}$	$\mathbf{69.2}_{1.9}$	$92.6_{0.5}$	$18.7_{8.8}$	$68.7_{2.3}$	$77.8_{2.0}$	66.9
Auto prompt†	$68.3_{2.5}$/$70.1_{2.6}$	$67.0_{3.0}$	$68.3_{7.4}$	$92.3_{1.0}$	$14.0_{14.1}$	$\mathbf{73.9}_{2.2}$	$76.2_{2.3}$	65.8
+ demonstration†	$70.0_{3.6}$/$72.0_{3.1}$	$67.7_{5.8}$	$68.5_{5.4}$	$\mathbf{93.0}_{0.6}$	$21.8_{15.9}$	$71.1_{5.3}$	$\mathbf{78.1}_{3.4}$	**67.3**

References I

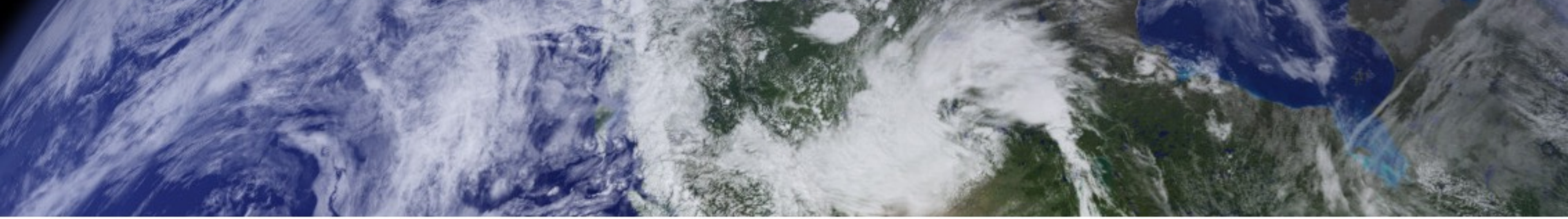
- Abu-El-Haija, S., Perozzi, B., Al-Rfou', R., & Alemi, A.A. (2018). Watch Your Step: Learning Node Embeddings via Graph Attention. NeurIPS.
- Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2016). Enriching Word Vectors with Subword Information. Transactions of the Association for Computational 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.
- Clark, K., Luong, M. T., Le, Q. V., & Manning, C. D. (2020). ELECTRA: Pre-training text encoders as discriminators rather than generators. ICLR.
- 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.
- 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.
- Meng, Y., Huang, J., Zhang, Y., & Han, J. (2022). Generating Training Data with Language Models: Towards Zero-Shot Language Understanding. arXiv preprint arXiv:2202.04538.
- 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.

References III

- 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. arXiv.
- 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.
- Turian, J.P., Ratinov, L., & Bengio, Y. (2010). Word Representations: A Simple and General Method for Semi-Supervised Learning. ACL.
- Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., & Le, Q. V. (2019). XLNet: Generalized Autoregressive Pretraining for Language Understanding. NeurIPS.



Q&A