CSE 587: Deep Learning for Natural Language Processing

Lecture 7. Prompt-based Methods

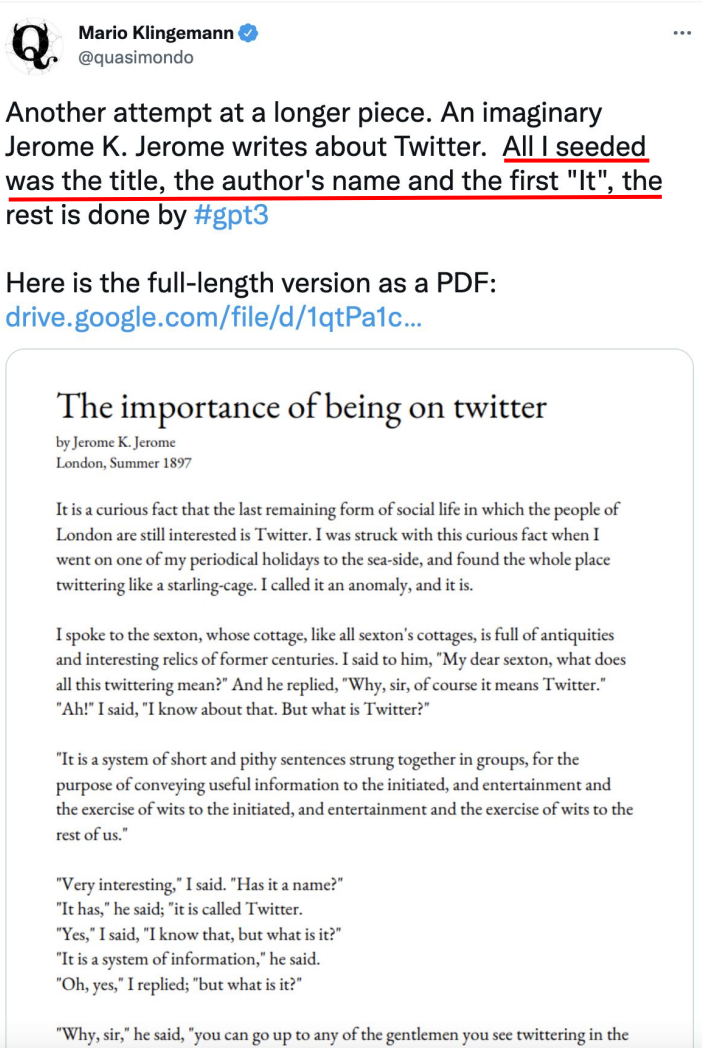
Rui Zhang

Spring 2023

Many slides from Pengfei Liu and Graham Neubig

What is Prompt?

OpenAI GPT-3 Writes Entire Articles



Mario Klingemann
@quasimondo

Another attempt at a longer piece. An imaginary Jerome K. Jerome writes about Twitter. All I seeded was the title, the author's name and the first "It", the rest is done by #gpt3

Here is the full-length version as a PDF:
drive.google.com/file/d/1qtPa1c...

The importance of being on twitter

by Jerome K. Jerome
London, Summer 1897

It is a curious fact that the last remaining form of social life in which the people of London are still interested is Twitter. I was struck with this curious fact when I went on one of my periodical holidays to the sea-side, and found the whole place twittering like a starling-cage. I called it an anomaly, and it is.

I spoke to the sexton, whose cottage, like all sexton's cottages, is full of antiquities and interesting relics of former centuries. I said to him, "My dear sexton, what does all this twittering mean?" And he replied, "Why, sir, of course it means Twitter."
"Ah!" I said, "I know about that. But what is Twitter?"

"It is a system of short and pithy sentences strung together in groups, for the purpose of conveying useful information to the initiated, and entertainment and the exercise of wits to the initiated, and entertainment and the exercise of wits to the rest of us."

"Very interesting," I said. "Has it a name?"
"It has," he said; "it is called Twitter.
"Yes," I said, "I know that, but what is it?"
"It is a system of information," he said.
"Oh, yes," I replied; "but what is it?"

"Why, sir," he said, "you can go up to any of the gentlemen you see twittering in the

Prompt

Downstream tasks are reformulated to look more like those solved during the original LM training with the help of a textual prompt. For example,

- Natural Language Generation. e.g., GPT-3 in the previous slides.
- Sentiment Analysis:
  - Input: “I love this movie.”
  - Prompt: “I love this movie. It was a __ movie”.
  - Output: a sentiment word, e.g., good
- Machine Translation:
  - Input: “I love this movie.”
  - Prompt: “English: I love this movie. French:”
  - Output: the French translation

Workflow of Prompting

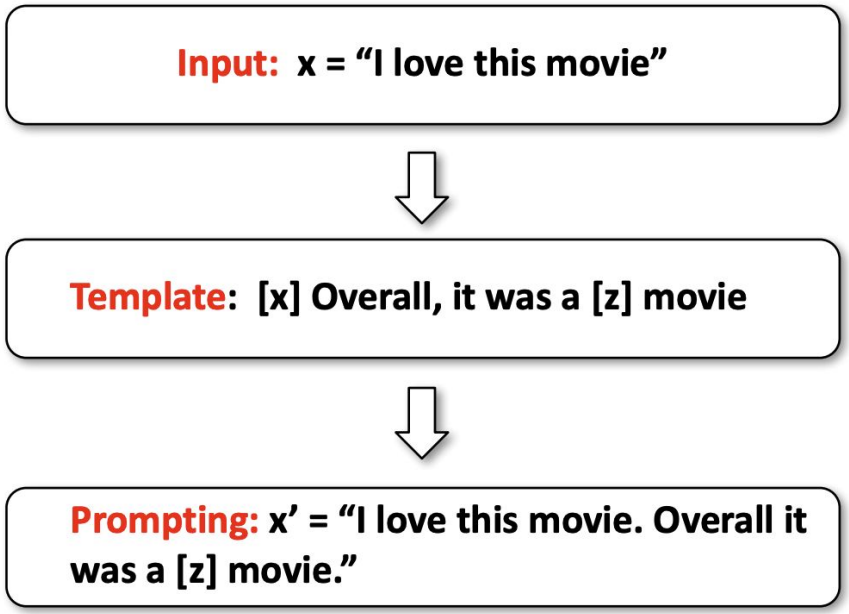
- Step 1: Prompt Addition
- Step 2: Answer Search
- Step 3: Answer Mapping

Prompt Addition

Given the input, construct the prompt following a template.

1. Apply a *template*, which is a textual string that has two slots: an *input slot* `[X]` for input $\boldsymbol{x}$ and an *answer slot* `[Z]` for an intermediate generated *answer* text $\boldsymbol{z}$ that will later be mapped into $\boldsymbol{y}$.
2. Fill slot `[X]` with the input text $\boldsymbol{x}$.

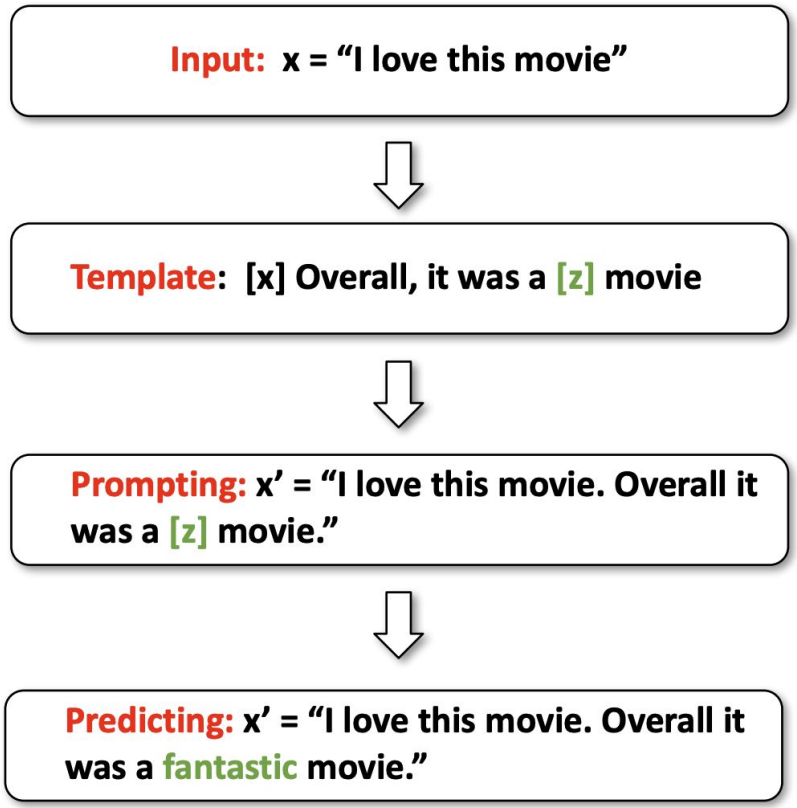
Example: Sentiment Analysis



Answer Prediction

Given the prompt, predict the answer [z]

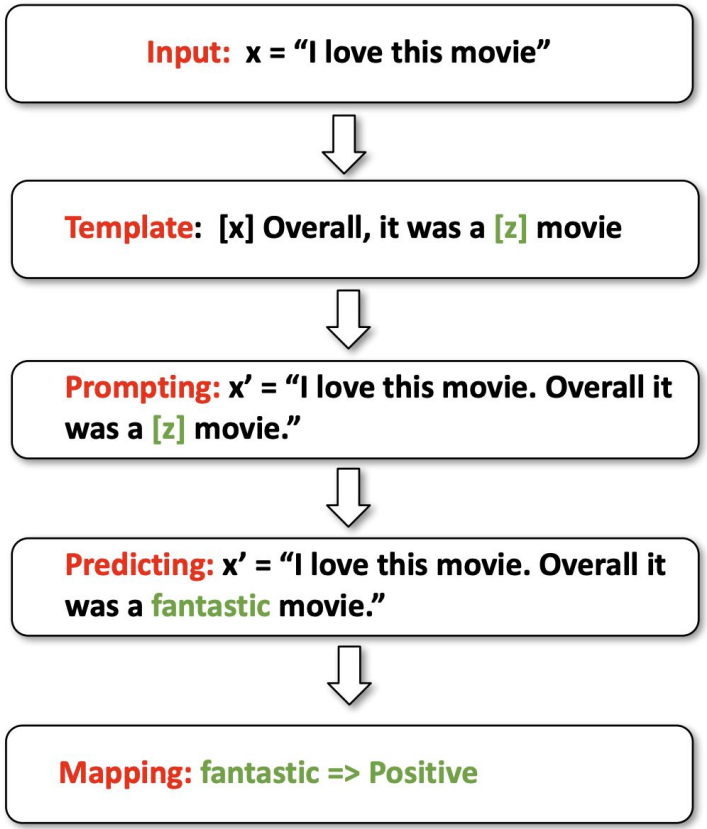
Example: Sentiment Analysis



Answer Mapping

Given the answer, map it to a class label.

Example: Sentiment Analysis



Prompts for Different Tasks

Type	Task	Input ([X])	Template	Answer ([Z])
Text CLS	Sentiment	I love this movie.	[X] The movie is [Z].	great fantastic ...
	Topics	He prompted the LM.	[X] The text is about [Z].	sports science ...
	Intention	What is taxi fare to Denver?	[X] The question is about [Z].	quantity city ...
Text-span CLS	Aspect Sentiment	Poor service but good food.	[X] What about service? [Z].	Bad Terrible ...
Text-pair CLS	NLI	[X1]: An old man with ... [X2]: A man walks ...	[X1]? [Z], [X2]	Yes No ...
Tagging	NER	[X1]: Mike went to Paris. [X2]: Paris	[X1][X2] is a [Z] entity.	organization location ...
Text Generation	Summarization	Las Vegas police ...	[X] TL;DR: [Z]	The victim ... A woman ... ...
	Translation	Je vous aime.	French: [X] English: [Z]	I love you. I fancy you. ...

Table 3: Examples of *input*, *template*, and *answer* for different tasks. In the **Type** column, "CLS" is an abbreviation for "classification". In the **Task** column, "NLI" and "NER" are abbreviations for "natural language inference" (Bowman et al., 2015) and "named entity recognition" (Tjong Kim Sang and De Meulder, 2003) respectively.

Prompt Engineering: Design of Prompts

Prompt engineering is the process of creating a prompting function f_prompt(x) that results in the most effective performance on the downstream task.

Manual Template Engineering

- Configure the manual template based on the characteristics of the task

Automated Template Learning

- Discrete Prompt: Search in discrete space
- Continuous Prompt: Search in continuous space

Automated Template Learning

Discrete Prompt: Search in discrete space

- Prompt Mining
- Prompt Paraphrasing
- Gradient-based Search

Continuous Prompt: Search in continuous space

- Prompt/Prefix Tuning

Prompt Mining (Jiang et al. 2020)

Mine prompts given a set of questions/answers

Middle-word Prompts Following the observation that words in the middle of the subject and object are often indicative of the relation, we directly use those words as prompts.

Barack Obama was born in Hawaii. → [X] was born in [Y].

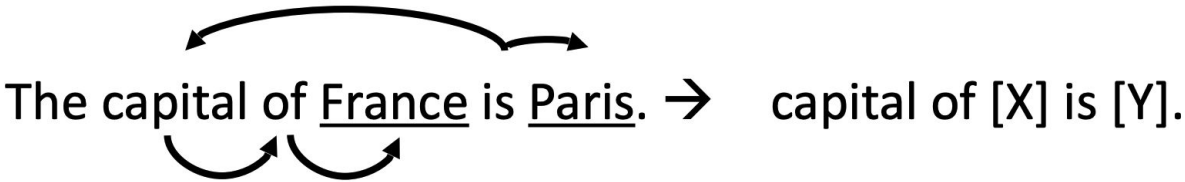
Prompt Mining (Jiang et al. 2020)

Mine prompts given a set of questions/answers

Middle-word Prompts Following the observation that words in the middle of the subject and object are often indicative of the relation, we directly use those words as prompts.

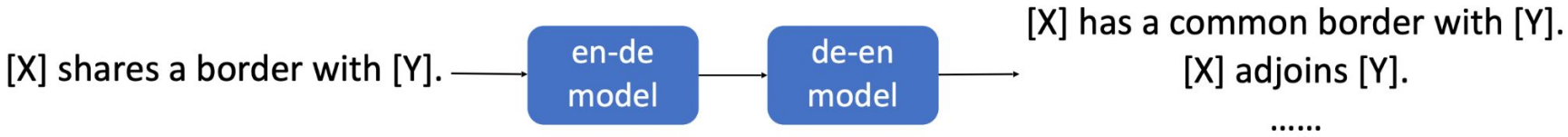
Barack Obama was born in Hawaii. → [X] was born in [Y].

Dependency-based Prompts We parse sentences with a dependency parser to identify the shortest dependency path between the subject and object, then uses the phrase spanning from the leftmost word to the rightmost word in the dependency path as a prompt.



Prompt Paraphrasing (Jiang et al. 2020)

Paraphrase an existing prompt to get other candidates, e.g. back translation with beam search



Editing-based Search (Prasad et al., 2022)

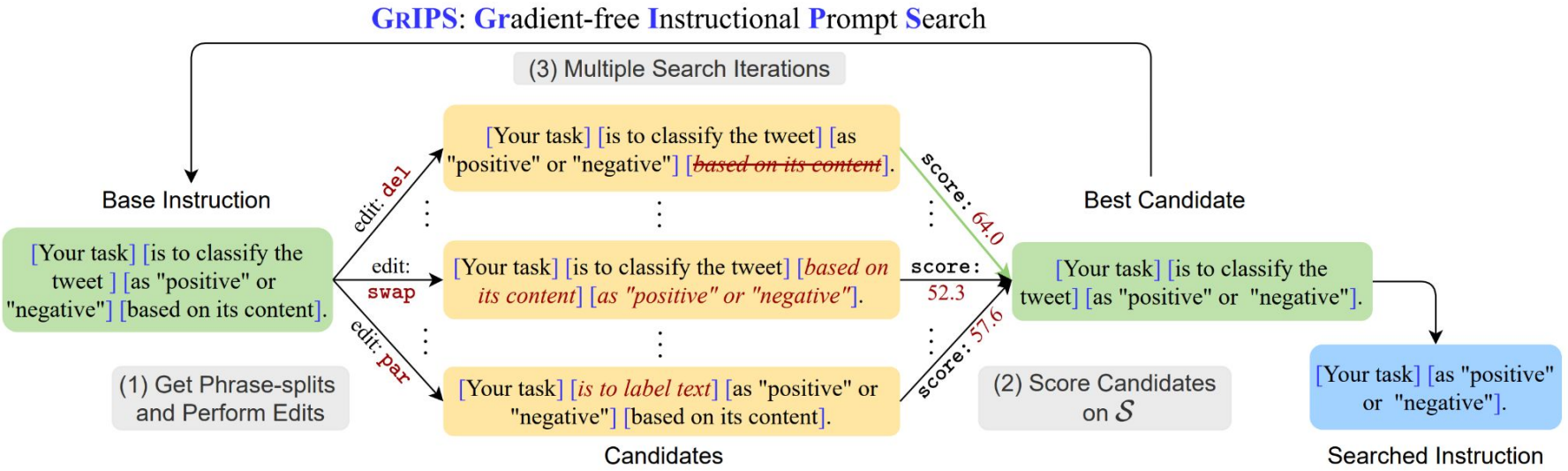


Figure 1: Overall Pipeline of GRIPS. The main steps are numbered. Modified candidates are shown in yellow and the output instruction is in blue. We use '[ ]' to show the syntactic phrase-level splits at which the edit operations occur. Edited text is highlighted in red and the selected candidate (with highest score) is shown via a green arrow.

Gradient-based Search (Shin et al. 2020)

We develop AUTOPROMPT, an automated method to create prompts for a diverse set of tasks, based on a gradient-guided search.

The trigger tokens are shared across all inputs and determined using a gradient-based search.

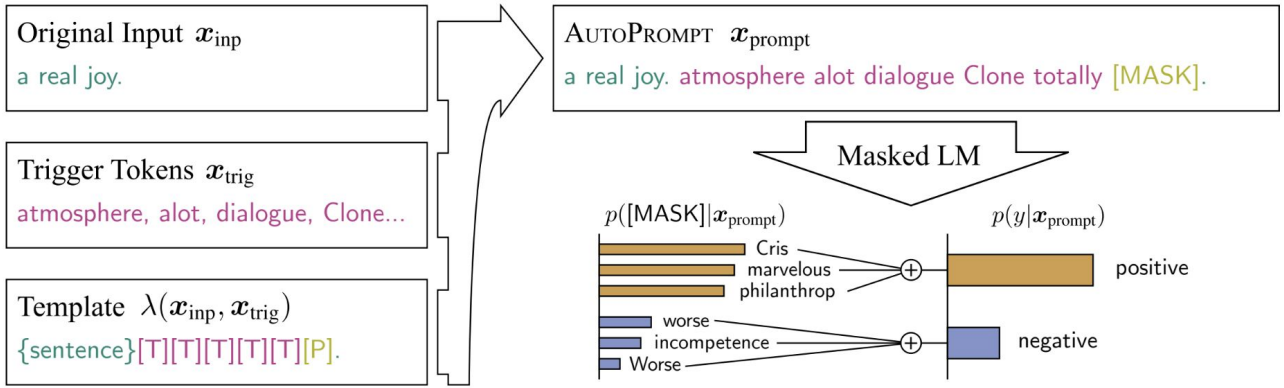


Figure 1: **Illustration of AUTOPROMPT** applied to probe a masked language model's (MLM's) ability to perform sentiment analysis. Each input, x_{inp}, is placed into a natural language prompt, x_{prompt}, which contains a single [MASK] token. The prompt is created using a template, λ, which combines the original input with a set of trigger tokens, x_{trig}. The trigger tokens are shared across all inputs and determined using a gradient-based search (Section 2.2). Probabilities for each class label, y, are then obtained by marginalizing the MLM predictions, $p([MASK]|x_{prompt})$, over sets of automatically detected label tokens (Section 2.3).

Prefix Tuning (Li and Liang 2021)

Prefix-Tuning: Optimizing Continuous Prompts for Generation

Optimize the embeddings of a prompt, instead of the words.

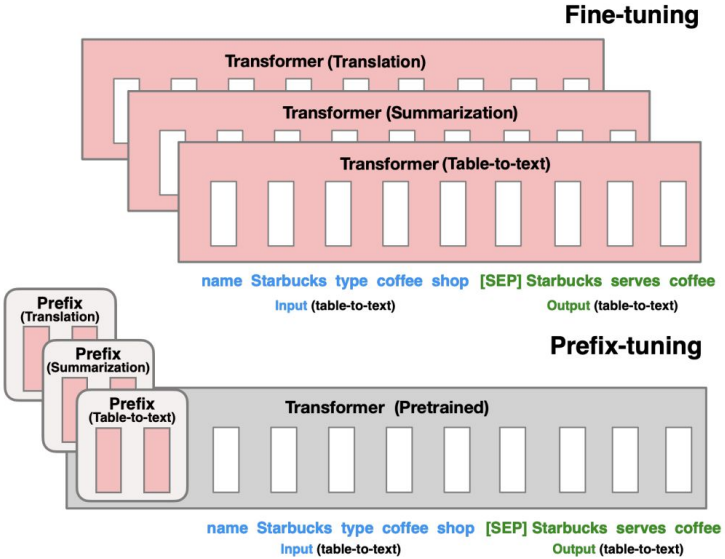


Figure 1: Fine-tuning (top) updates all LM parameters (the red Transformer box) and requires storing a full model copy for each task. We propose prefix-tuning (bottom), which freezes the LM parameters and only optimizes the prefix (the red prefix blocks). Consequently, we only need to store the prefix for each task, making prefix-tuning modular and space-efficient. Note that each vertical block denote transformer activations at one time step.

Prompt Tuning (Lester et al. 2021)

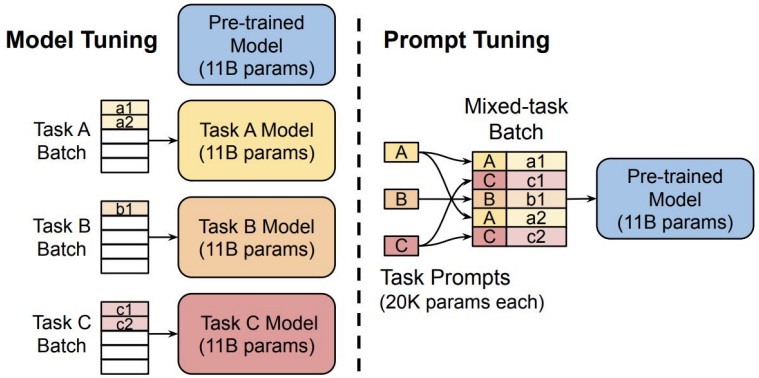


Figure 2: **Model tuning** requires making a task-specific copy of the entire pre-trained model for each downstream task and inference must be performed in separate batches. **Prompt tuning** only requires storing a small task-specific prompt for each task, and enables mixed-task inference using the original pre-trained model. With a T5 "XXL" model, each copy of the tuned model requires 11 billion parameters. By contrast, our tuned prompts would only require 20,480 parameters per task—a reduction of *over five orders of magnitude*—assuming a prompt length of 5 tokens.

Prompt Tuning (Lester et al. 2021)

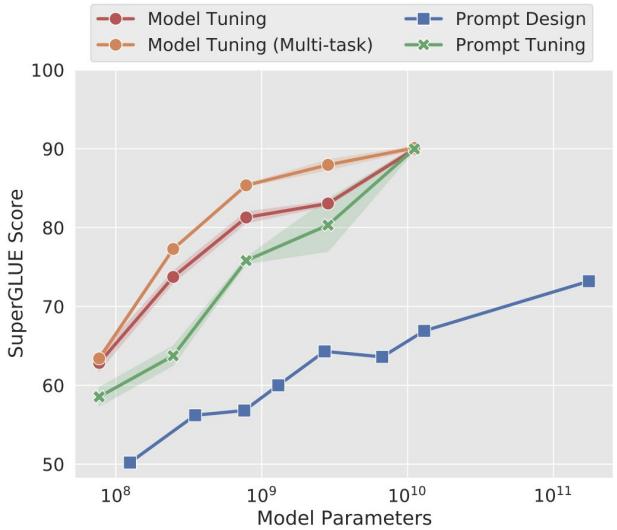


Figure 1: Standard **model tuning** of T5 achieves strong performance, but requires storing separate copies of the model for each end task. Our **prompt tuning** of T5 matches the quality of model tuning as size increases, while enabling the reuse of a single frozen model for all tasks. Our approach significantly outperforms few-shot **prompt design** using GPT-3. We show mean and standard deviation across 3 runs for tuning methods.

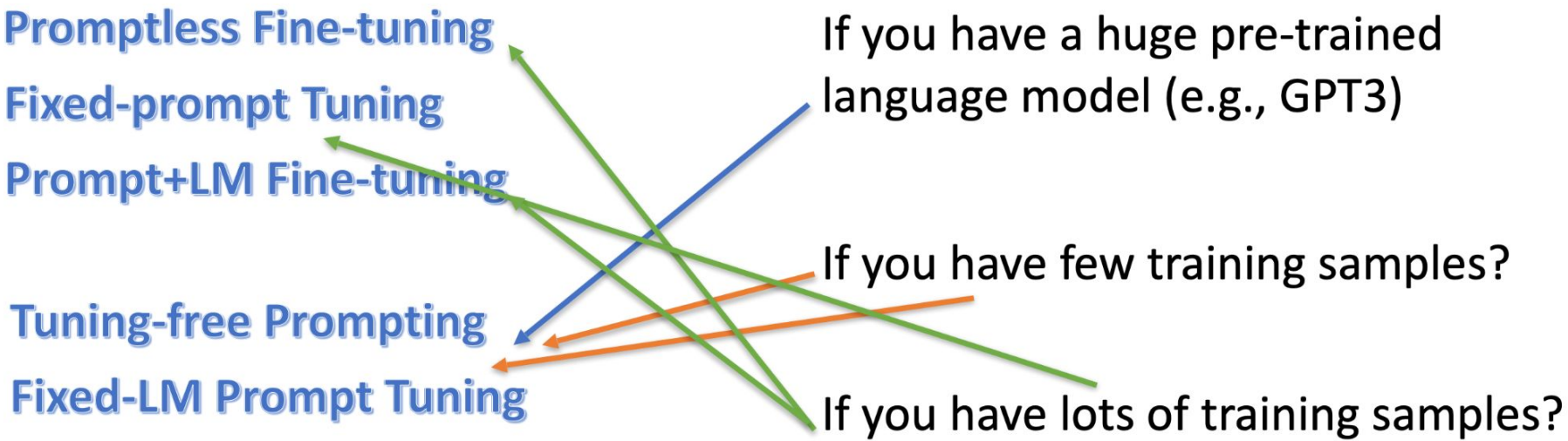
Prefix Tuning vs Prompt Tuning

- "Prefix Tuning" optimizes prefix of all layers.
- "Prompt Tuning" optimizes only the embedding layer.

Prompt-based Training Strategies

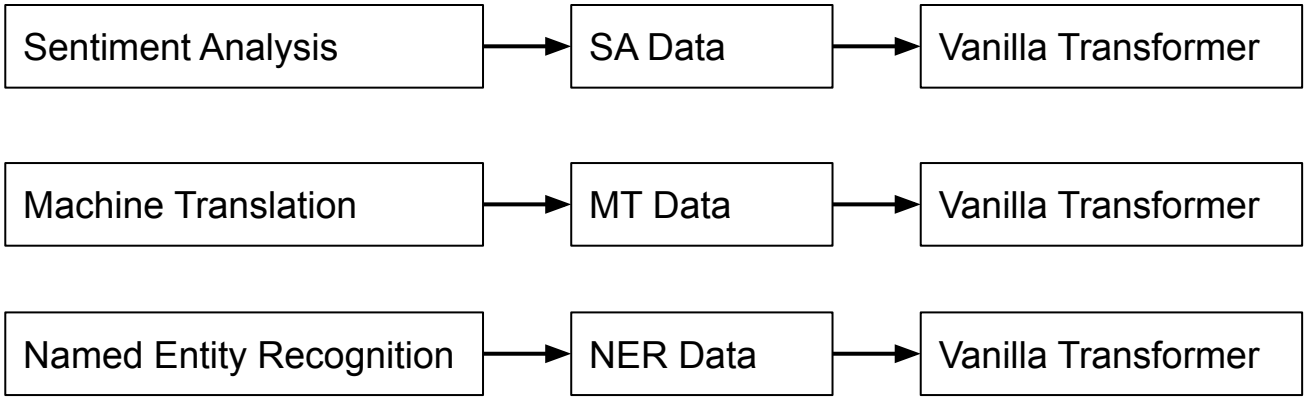
Strategy	LM Params Tuned	Additional Prompt Params	Prompt Params Tuned	Examples
Promptless Fine-Tuning	Yes	N/A	N/A	BERT Fine-tuning
Tuning-free Prompting	No	No	N/A	GPT-3
Fixed-LM Prompt Tuning	No	Yes	Yes	Prefix Tuning
Fixed-prompt LM Tuning	Yes	No	N/A	PET
Prompt+LM Fine-tuning	Yes	Yes	Yes	PADA

Prompt-based Training Strategies



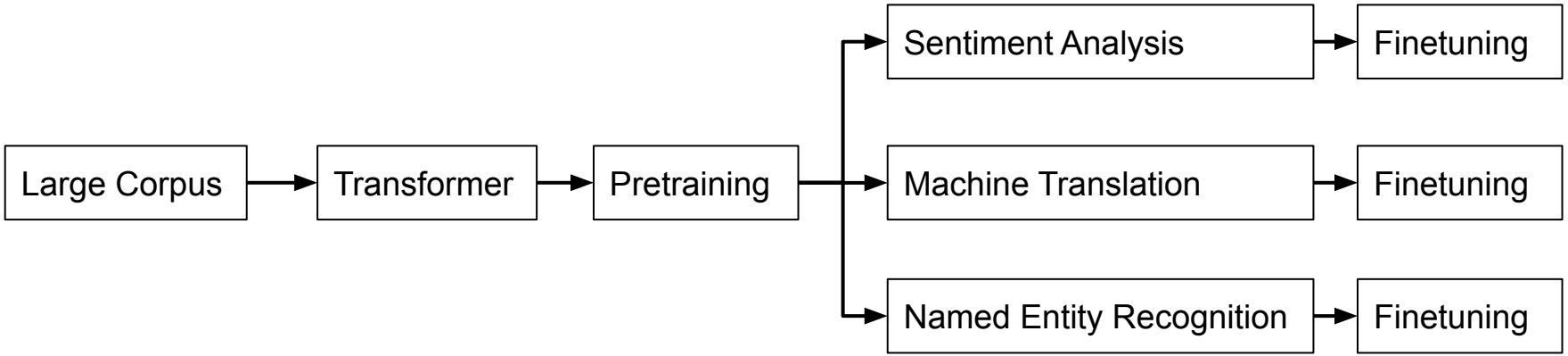
How is Transformer used

We can use Transformer separately for each task.
Transformer is initialized randomly, and trained for each dataset using supervised learning.



Pretraining

- First train Transformer using a lot of general text using *unsupervised* learning. This is called **pretraining**.
- Then train the pretrained Transformer for a specific task using *supervised* learning.This is called **finetuning.**
- The whole process can be called **transfer learning**.



Four Paradigms of NLP

Paradigm
a. Fully Supervised Learning (Non-Neural Network)
b. Fully Supervised Learning (Neural Network)
c. Pre-train, Fine-tune
d. Pre-train, Prompt, Predict

Fully Supervised Learning (Non-Neural Network)

Model: Logistic Regression, Decision Tree, SVM, CRF

Engineering: Feature Engineering

Time Period: Most popular through 2015

Characteristics:

- Non-neural machine learning models mainly used
- Require manually defined feature extraction

Representative Work:

- Manual features -> linear or kernelized support vector machine (SVM)
- Manual features -> conditional random fields (CRF)

Fully Supervised Learning (Neural Network)

Model: Neural Networks (RNN, CNN, LSTM, Transformers)

Engineering: Architecture Engineering

Time Period: About 2013-2018

Characteristics:

- Rely on neural networks
- Do not need to manually define features, but should modify the network structure (e.g.: LSTM v.s CNN)
- Sometimes used pre-training of LMs, but often only for shallow features such as embeddings

Representative Work

- CNN for Text Classification

Pretrain, Finetune

Model: Pretrained Language Models (discussed in the last lecture)

Engineering: Objective Engineering

Time Period: 2017-Now

Characteristics:

- Pre-trained LMs (PLMs) used as initialization of full model - both shallow and deep features
- Less work on architecture design, but engineer objective functions

Representative Work:

- BERT → Fine Tuning

Pretrain, Prompt, Predict

Model: Pretrained Language Models

Engineering: Prompt Engineering

Date: 2019-Now

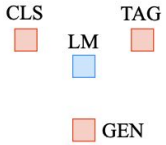
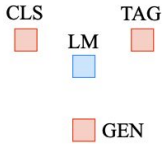
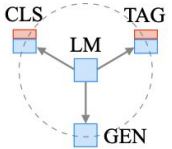
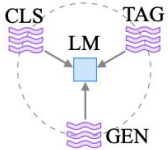
Characteristic:

- NLP tasks are modeled entirely by relying on LMs
- The tasks of shallow and deep feature extraction, and prediction of the data are all given to the LM
- Engineering of prompts is required

Representative Work:

- GPT-3

Four Paradigms of NLP

Paradigm	Engineering	Task Relation
a. Fully Supervised Learning (Non-Neural Network)	Features (e.g. word identity, part-of-speech, sentence length)	
b. Fully Supervised Learning (Neural Network)	Architecture (e.g. convolutional, recurrent, self-attentional)	
c. Pre-train, Fine-tune	Objective (e.g. masked language modeling, next sentence prediction)	
d. Pre-train, Prompt, Predict	Prompt (e.g. cloze, prefix)	

: fully unsupervised training.

: fully supervised training.

indicates a textual prompt.

Dashed lines suggest that different tasks can be connected by sharing parameters of pre-trained models.

“LM→Task”: adapting LMs (objectives) to downstream tasks.

“Task→LM”: adapting downstream tasks (formulations) to LMs

Recommended Reading

Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing

Pengfei Liu
Carnegie Mellon University
pliu3@cs.cmu.edu

Weizhe Yuan
Carnegie Mellon University
weizhey@cs.cmu.edu

Jinlan Fu
National University of Singapore
jinlanjonna@gmail.com

Zhengbao Jiang
Carnegie Mellon University
zhengbaj@cs.cmu.edu

Hiroaki Hayashi
Carnegie Mellon University
hiroakih@cs.cmu.edu

Graham Neubig
Carnegie Mellon University
gneubig@cs.cmu.edu

https://arxiv.org/pdf/2107.13586.pdf

A useful library for Prompt Engineering

https://github.com/thunlp/OpenPrompt