프롬프트 러닝

20180376 안제준

Pretrain, Prompt, Predict: A New Paradigm for NLP

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

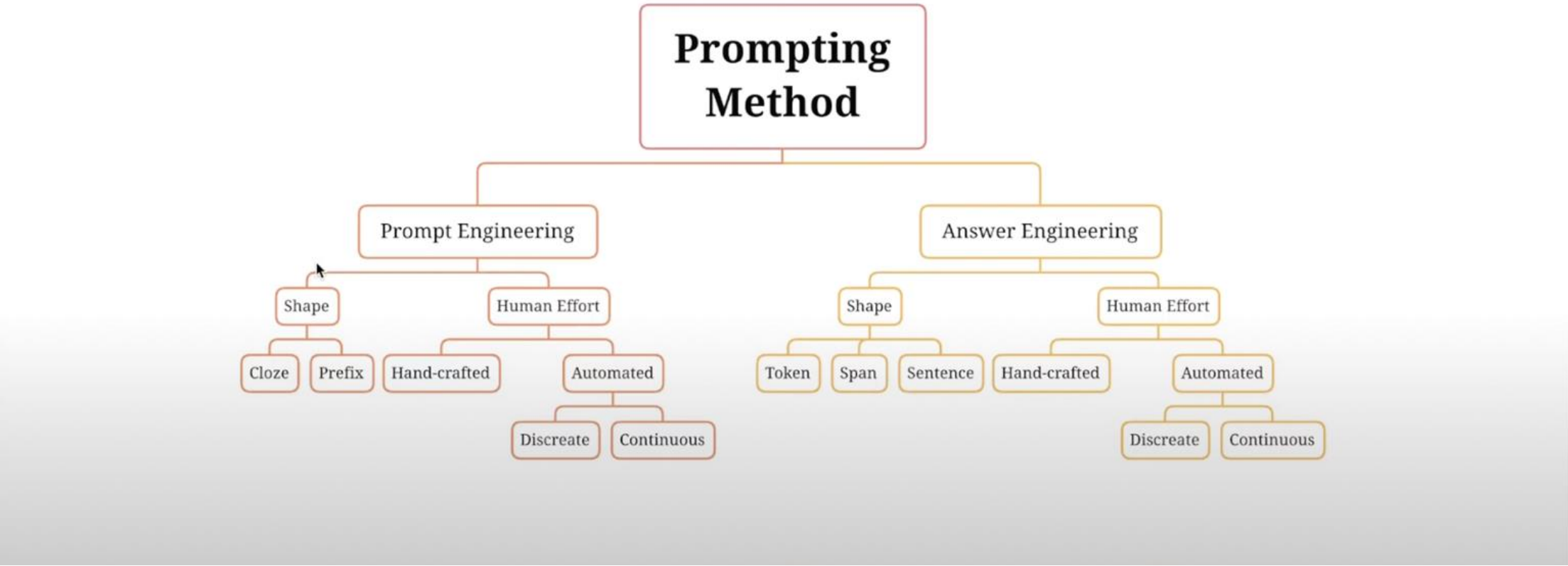
Hiroaki Hayashi
Carnegie Mellon University
hiroakih@cs.cmu.edu

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

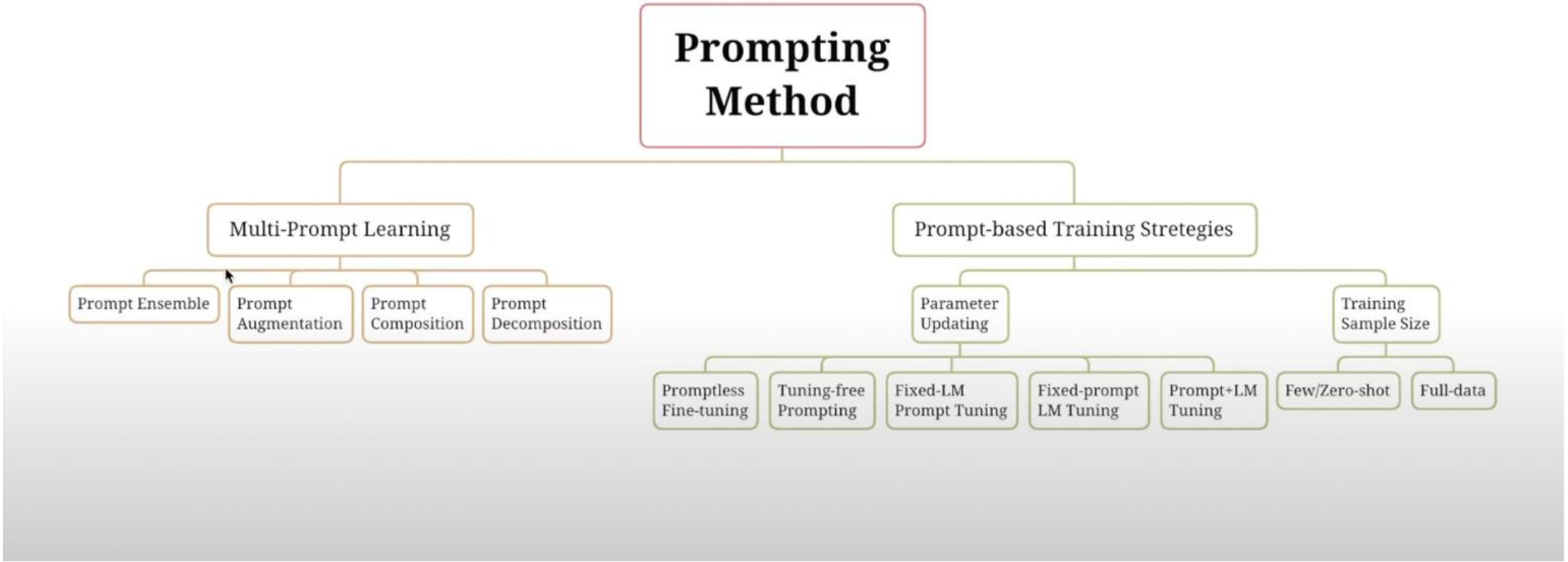
Downstram Task에 적합한 Prompt를 설계하기 위해 고려할 것들

1. Pre-trained Model Choice - 어떤 LM을 선택할 것 인가.
2. Prompt Engineering - 다운스트림 태스크에서 최고의 성능을 내는 프롬프트 템플릿을 찾는 방법에 대한 연구
3. Answer Engineering - 예측되는 토큰과 다운스트림 태스크의 레이블과의 관계를 mapping하는 것에 대한 연구
4. Multi-prompt Learning - 하나의 prompt learning으로 풀기 어려운 문제를 여러 개의 prompt를 활용해서 푸는 연구.
5. Prompt-based Training Strategies - 프롬프트를 이용한 학습을 할 때 선택할 수 있는 다양한 학습 기법들에 대한 선택지

Preview

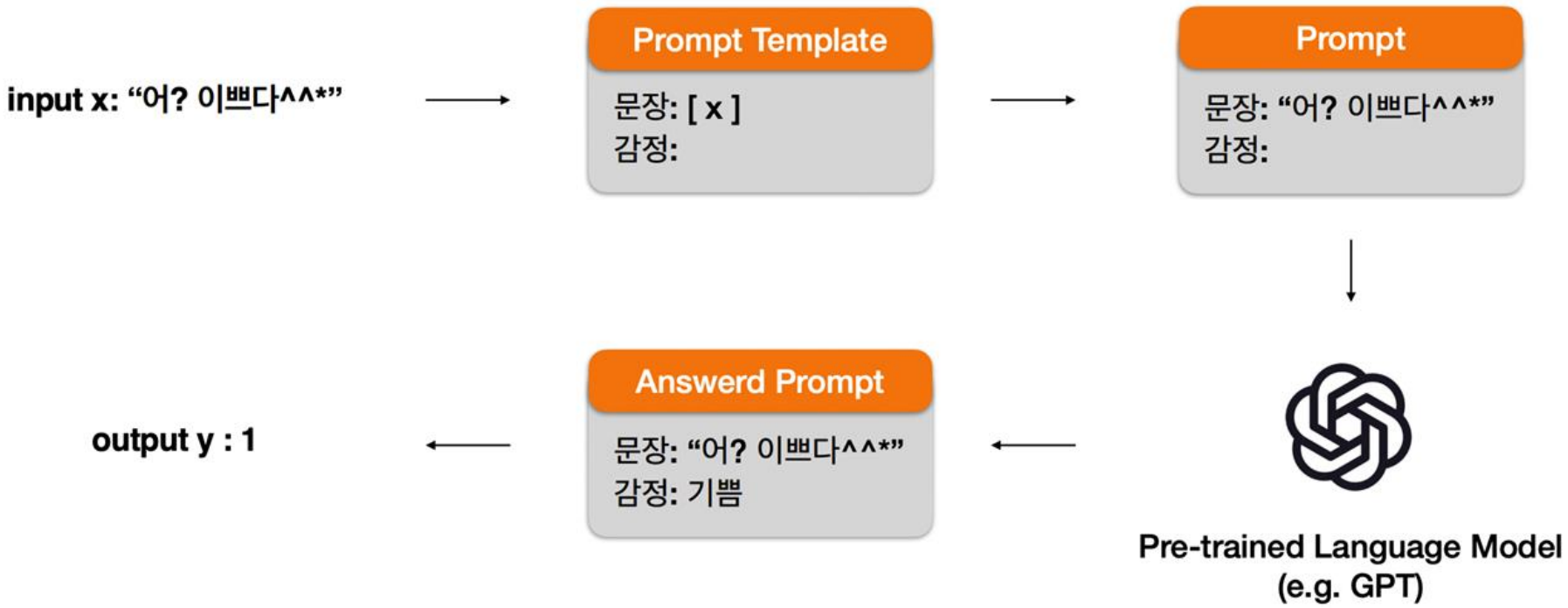

Prompting Method
Prompt Engineering
Answer Engineering
Shape
Human Effort
Shape
Human Effort
Cloze
Prefix
Hand-crafted
Automated
Token
Span
Sentence
Hand-crafted
Automated
Discreate
Continuous
Discreate
Continuous

Preview


Prompting Method
Multi-Prompt Learning
Prompt Ensemble
Prompt Augmentation
Prompt Composition
Prompt Decomposition
Prompt-based Training Stretegies
Parameter Updating
Promptless Fine-tuning
Tuning-free Prompting
Fixed-LM Prompt Tuning
Fixed-prompt LM Tuning
Prompt+LM Tuning
Training Sample Size
Few/Zero-shot
Full-data

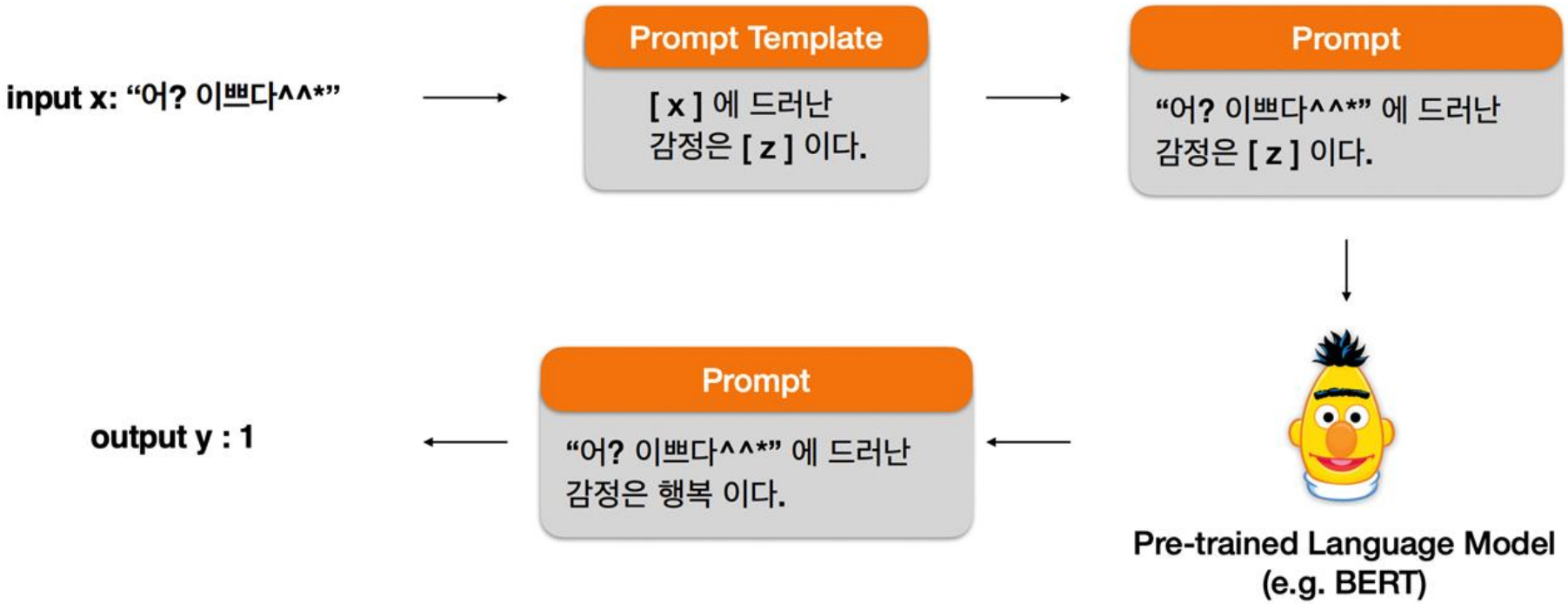
Pre-trained Model Choice

- Prefix Prompt



Pre-trained Model Choice

- Cloze Prompt



Prompt Engineering

- Downstream task에 가장 효과적인 prompt template을 만드는 과정

- 최적의 Prompt Template를
  - 사람이 직접 설계하여 찾거나(Manual Template Engineering)
  - 자동화된 방법으로 탐색(Automated Template Learning)

Prompt Engineering - Manual

- 사람이 직접 설계하여 찾는 방식

- LAMA(Petroni et al., 2019)
  - Factual Probing task를 위한 cloze prompt template
  - e.g. Patrick Oboya plays in _____ position

- GPT-3(Brown et al., 2020)
  - 다양한 task에 대한 prefix prompt template
  - e.g. Translation English to French:
  - cheese => ____

Prompt Engineering - Auto

- 최적의 prompt template을 자동으로 탐색

- Prompt template을 사람이 문자 그대로 해석할 수 있는지에 따라
  - Discrete Prompts(a.k.a. Hard Prompts)
  - Continuous Prompts(a.k.a. Soft Prompts)

Discreate Prompts(a.k.a. Hard Prompts)

- Vocab에서 최적의 prompt template을 위한 token들의 조합 탐색한다.

- 주요 방법론
  - 1. Prompt Mining
  - 2. Prompt Paraphrasing
  - 3. Gradient-based Search
  - 4. Prompt Generation
  - 5. Prompt Scoring

Discreate Prompts(a.k.a. Hard Prompts)

- Prompt Mining
  - Wikipedia의 각 문장에 대해서 다음 두 가지 규칙에 따라 prompts를 추출
- 1. Middle-word Prompts
  - Barack Obama was born in Hawaii" ➔ "x was born in y"
- 2. Dependency-based Prompts
  - "The capital of France is Paris" ➔ "capital of x is y"

Discreate Prompts(a.k.a. Hard Prompts)

- Prompt Paraphrasing
  - LPAQA(Jiang et al., 2020)에서 제안된 방법
  - “x shares a boarder with y” ➔ “x has a common border with y” / “x adjoins y”

Discreate Prompts(a.k.a. Hard Prompts)

- Gradient-based Search
  - AutoPrompt(Shin et al., 2020)
- Prompt Generation
  - LM-BFF(Gao et al. 2021)
  - PADA(Ben-David et al., 2021)
- Prompt Scoring
  - Coherency Ranking(Davison et al., 2019)

Continuous Prompts(a.k.a Soft Prompts)

- Prompt가 반드시 사람이 이해할 수 있는 자연어 형태일 필요는 없음

- 주요 방법론
  - 1. Prefix-Tuning
  - 2. Tuning Initialized with Discrete Prompts
  - 3. Hard-SoftPromptHybridTuning

Continuous Prompts(a.k.a Soft Prompts)

- Prefix-Tuning(GPT-2 구조)

→Learnable parameter에 prompt token <p1>,<p2>, … <pn>을 입력 받아

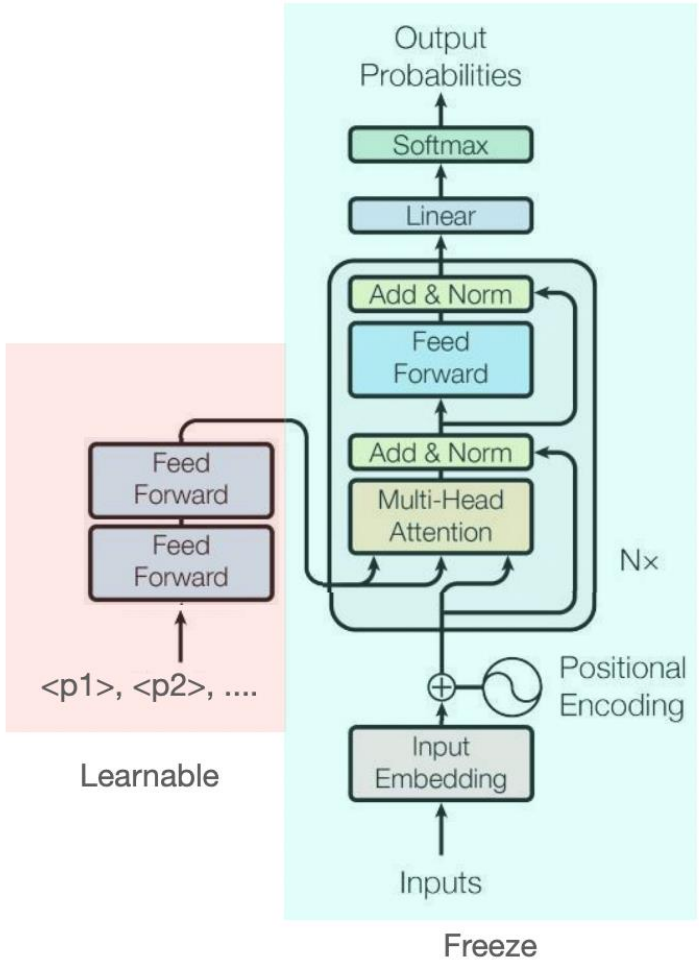
→Feed Foward Network 거쳐 나온 key와 value값을 모델에 입력

→ 때문에 각각에 task마다 앞에 붙는 Learnable parameter가 학습

→ 학습이 끝나면 어떠한 task에 대해 이 모델을 적용할 때

→ 얻어진 Task-specific vector(key, value)를 가지고 task마다 이 LM을 적합하게 바꿔 사용이 가능

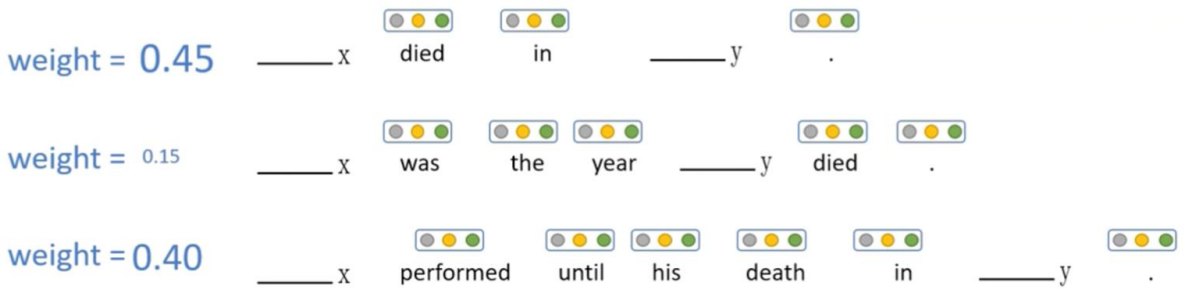
→ task-specific vectors를 출력한 후 input과 cross attention 적용



Continuous Prompts(a.k.a Soft Prompts)

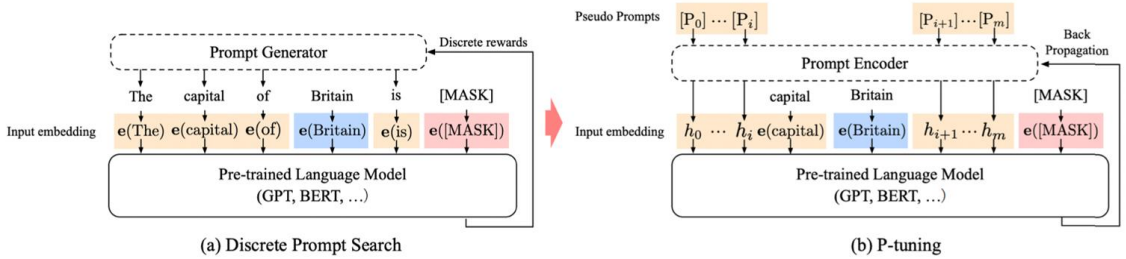
- Tuning Initialized with Discrete Prompts
  - Discrete Prompts를 초기값으로 하여 Downstream task에 tuning
  - OptiPrompt(Zhong et al., 2021)

  - Soft(Qin and Eisner, 2021)

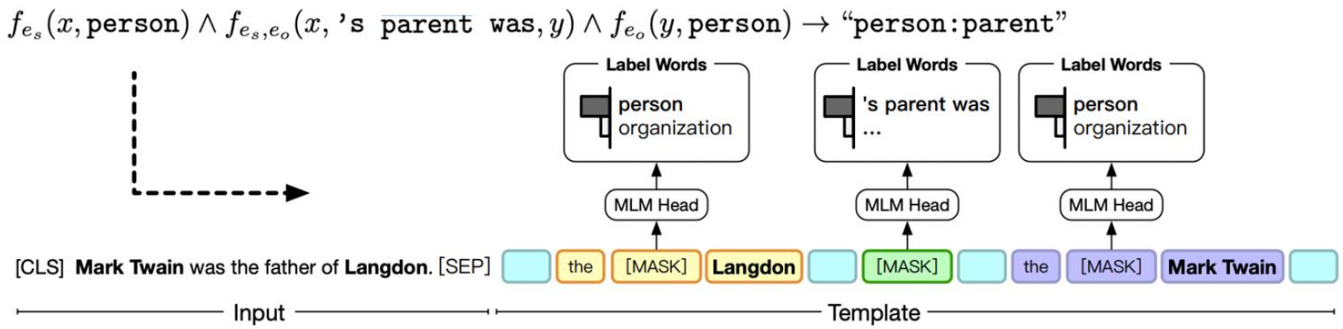


Continuous Prompts(a.k.a Soft Prompts)

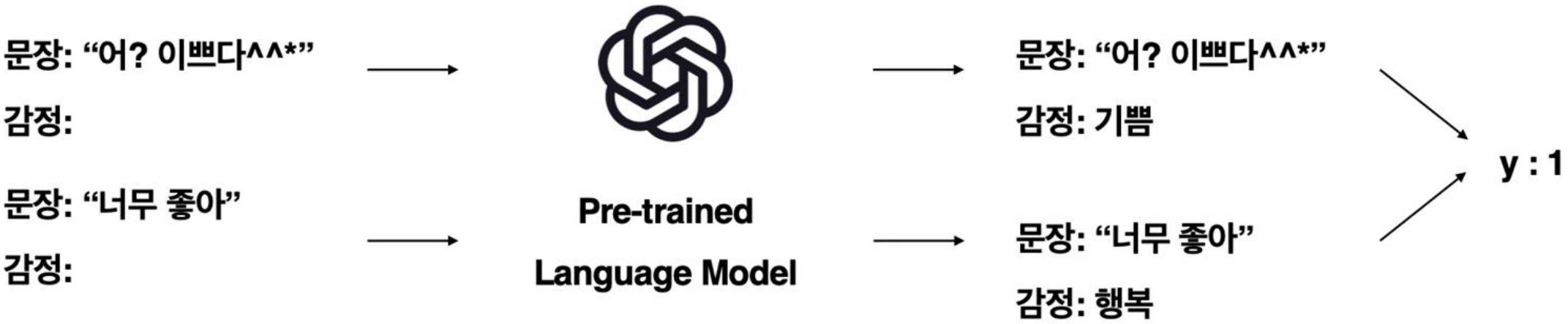
- Hard-Soft Prompt Hybrid Tuning
  - P-Tuning(Liu et al., 2021)



  - PTR:Prompt Tuning with Rules(Han et al., 2021)



Answer Engineering



- Answer Engineering
  - 가능한 answer z의 집합을 answer space Z, output label y의 집합을 Y라 할 때 task의 성능을 높이는 answer space Z와 Z ➔ Y의 mapping을 탐색하는 것
  - NLG 형태의 task인 경우 출력된 answer token z가 그 자체로 task의 output y

Answer Engineering

- Manual Design - 사람이 직접 answer z와 output y의 대응 관계를 지정
  - TemplateNER(Chi et al., 2021)
    - Ex) z : "location" -> y : "LOC" 로 mapping

Answer Engineering

- Automated Design - 최적의 answer space & mapping을 자동으로 탐색

- 사람이 해석가능한지의 여부에 따라
  - Discrete Answer
  - Continuous Answer

Discreat Answer

- 주요 방법론
  - 1. Answer Paraphrasing
  - 2. Label Decomposition
  - 3. Prune-then-Search

Discreat Answer

- Answer Paraphrasing
  - LPAQA(Jiang et al., 2020)

- Label Decomposition
  - Adaprompt(Chen et al., 2021)
    - Ex) per:city_of_death ➔ {person, city, death}
    - per:city_of_death의 확률 = 각 token(person, city, death)의 확률의 합

Discreat Answer

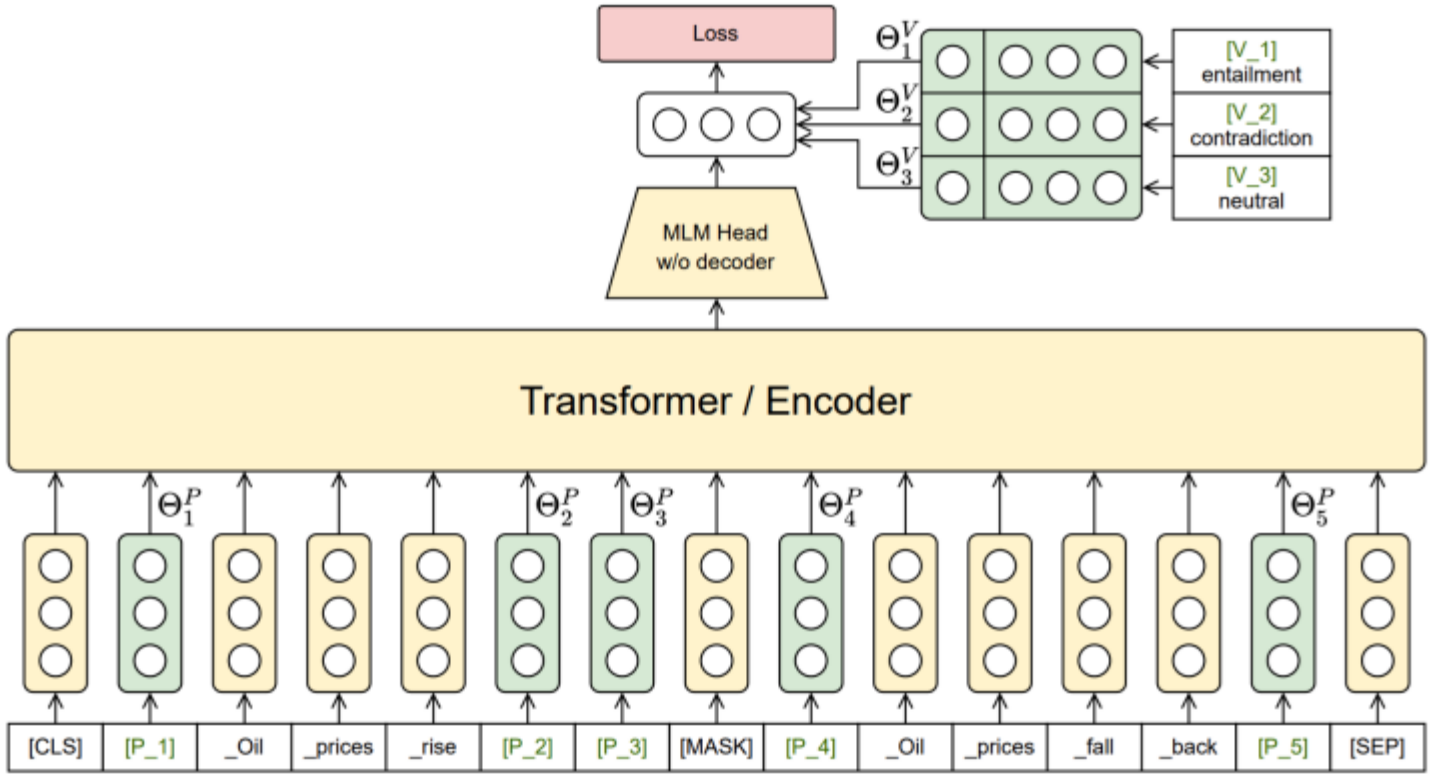
- Prune-then-Search
  - Answer token 후보를 추려낸 다음(Prune), 알고리즘을 이용해 적절한 것을 선택(Search)

	PET (Schink, 2020)	**AutoPrompt** (Shin et al., 2020)	**LM-BFF** (Gao et al., 2021)
Prune	2개 이상의 알파벳을 포함하는 token z 중에 빈도 수가 높은 것들로 구성	[MASK] token의 contextualized embedding을 이용한 분류기를 학습 시킴	[Step1] token z가 들어갈 위치에서의 생성 확률이 가장 높은 top-k개 token 선택 [Step2] zero-shot accuracy가 높은 것들로 구성
Search	label y에 대한 LM의 likelihood를 최대화하는 token z	Token z를 학습된 분류기에 입력했을 때 각 label에 대하여 확률이 가장 높은 top-k개 token을 answer token z로 선택	LM fine-tuning 이후 dev set의 accuracy 기준으로 최종 answer token z 선택

Continuous Answer

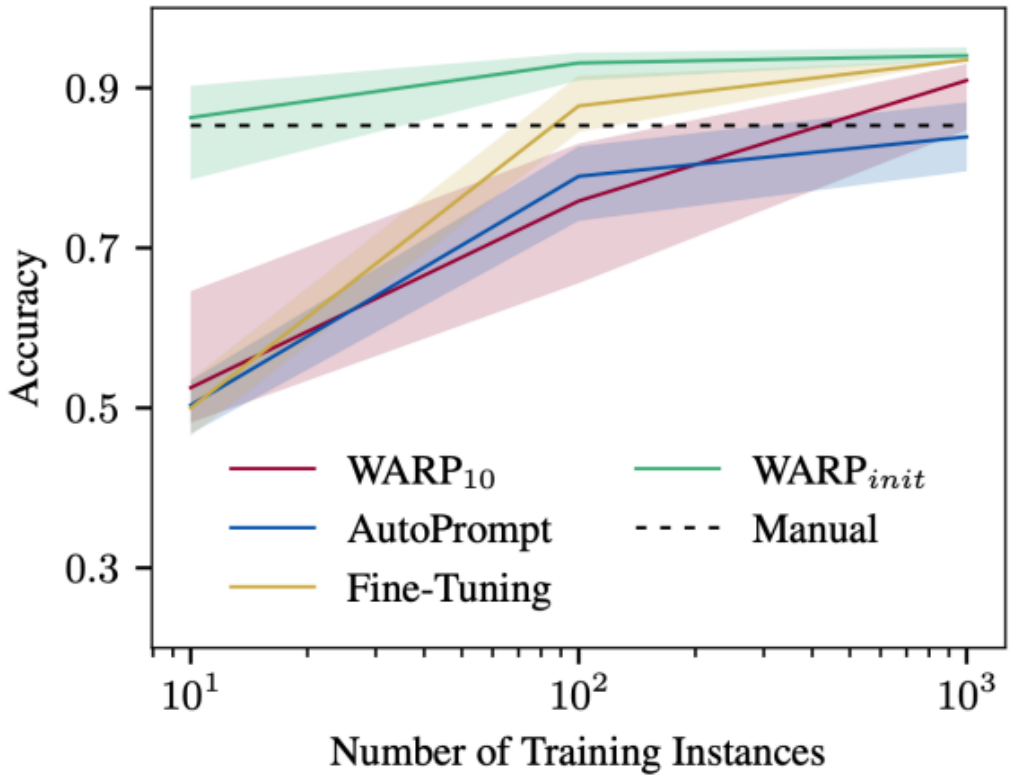
- Answer token z가 반드시 사람이 이해할 수 있는 형태일 필요는 없다.
- WARP(Hambardzumyan et al., 2021)

WRAP


Loss
Θ_1^V
Θ_2^V
Θ_3^V
[V_1] entailment
[V_2] contradiction
[V_3] neutral
MLM Head w/o decoder
Transformer / Encoder
Θ_1^P
Θ_2^P
Θ_3^P
Θ_4^P
Θ_5^P
[CLS]
[P_1]
_Oil
_prices
_rise
[P_2]
[P_3]
[MASK]
[P_4]
_Oil
_prices
_fall
_back
[P_5]
[SEP]

Result

	Model	CB F_1 / Acc.	RTE Acc.
dev	GPT-3 Small	26.1 / 42.9	52.3
	GPT-3 Med	40.4 / 58.9	48.4
	GPT-3	57.2 / 82.1	72.9
	PET (ALBERT)	59.4 / 85.1	69.8
	iPET (ALBERT)	92.4 / 92.9	74.0
	$\text{WARP}_{\text{init}}$ (ALBERT)	84.0 / 87.5	71.8
test	GPT-3	52.0 / 75.6	69.0
	PET (ALBERT)	60.2 / 87.2	67.2
	iPET (ALBERT)	79.9 / 88.8	70.8
	$\text{WARP}_{\text{init}}$ (ALBERT)	70.2 / 82.4	69.1

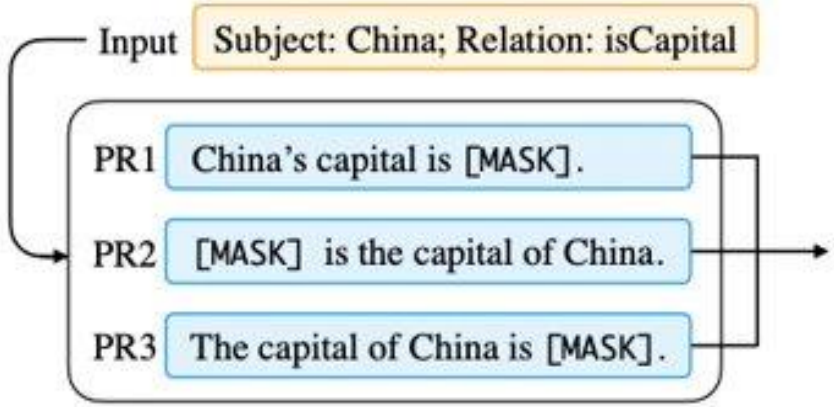


Multi-Prompt Learning

- 1. Prompt Ensembling
- 2. Prompt Augmentation
- 3. Prompt Composition
- 4. Prompt Decomposition

Prompt Ensembling

- 여러개의 prompt를 사용하여 predict한 결과를 ensemble

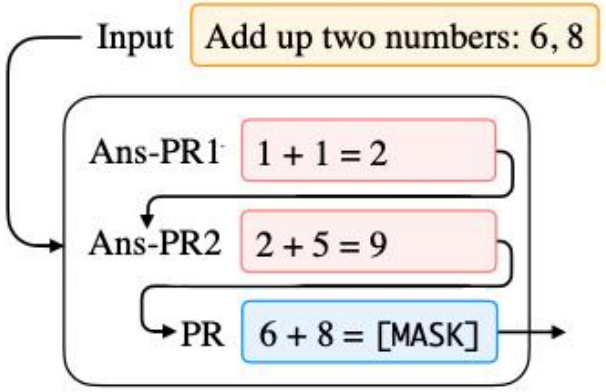


(a) Prompt Ensembling.

Prompt Augmentation

- 정답이 포함된 prompt를 예시로 추가

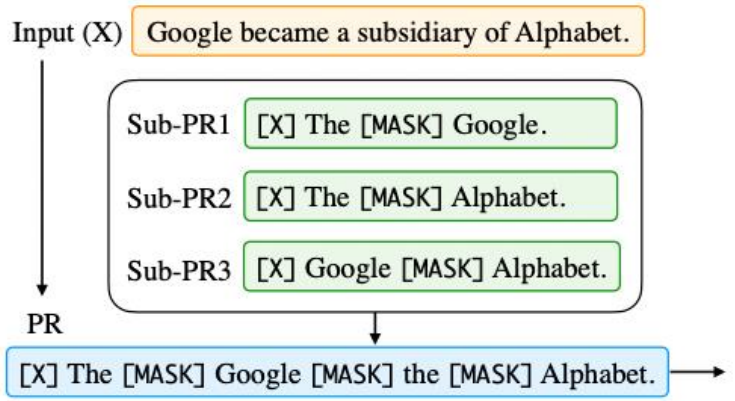
- Sample 선택과 제시 순서가 성능에 크게 영향을 미침
  - Sample Selection
    - LM-BFF(Gao et al., 2021), KATE(Liu et al., 2020)
  - Sample Ordering
    - OrderEntropy(Lu et al., 2021)



(b) Prompt Augmentation.

Prompt Composition

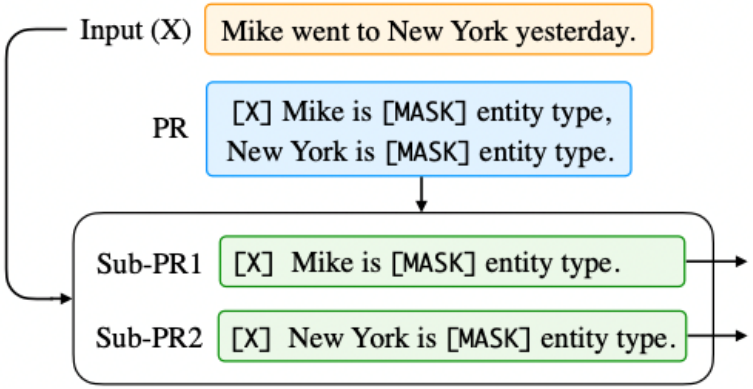
- 하나의 task를 이를 구성하는 sub task의 조합으로 표현
  - PTR(Han et al., 2021)



(c) Prompt Composition.

Prompt Decomposition

- 하나의 prompt를 여러개의 sub-prompt로 분할
  - TemplateNER(Cui et al., 2021)



(d) Prompt Decomposition.

Training Strategies for Prompting Methods

Strategy	LM Params	Prompt Params		Example
		Additional	Tuned	
Promptless Fine-tuning	Tuned	-		ELMo [130], BERT [32], BART [94]
Tuning-free Prompting	Frozen	✗	✗	GPT-3 [16], AutoPrompt [159], LAMA [133]
Fixed-LM Prompt Tuning	Frozen	✓	Tuned	Prefix-Tuning [96], Prompt-Tuning [91]
Fixed-prompt LM Tuning	Tuned	✗	✗	PET-TC [153], PET-Gen [152], LM-BFF [46]
Prompt+LM Fine-tuning	Tuned	✓	Tuned	PADA [8], P-Tuning [103], PTR [56]

Table 6: Characteristics of different tuning strategies. "Additional" represents if there are additional parameters beyond LM parameters while "Tuned" denotes if parameters are updated.

Fixed-LM Prompt Tuning

- LM의 parameter는 고정, prompt와 관련된 parameter를 추가해 이것만 업데이트(fine-tuning)
  - 관련 연구
    - Prefix-Tuning(Li and Liang., 2021), Prompt-Tuning(Lester et al., 2021), WARP(Hambardzumyan et al., 2021)
  - 장점
    - Catastrophic forgetting을 피할 수 있음(LM이 고정되어 있기 때문에 LM이 이전에 학습한 데이터를 잊지않는다.)
    - 학습되는 parameter가 적어 few-shot setting에 적합하며 종종 tuning-free prompting(zero-shot setting)보다 더 나은 성능을 보임
  - 단점
    - Zero-shot setting(학습 데이터가 전혀 없는)에서는 적용 불가
    - Hyperparameter 또는 random seed 설정 필요(추가 학습이 이루어지기 때문)
    - Continuous Prompt를 최적화하므로 사람이 해석하거나 조작하기 어려움

Fixed-prompt LM Tuning

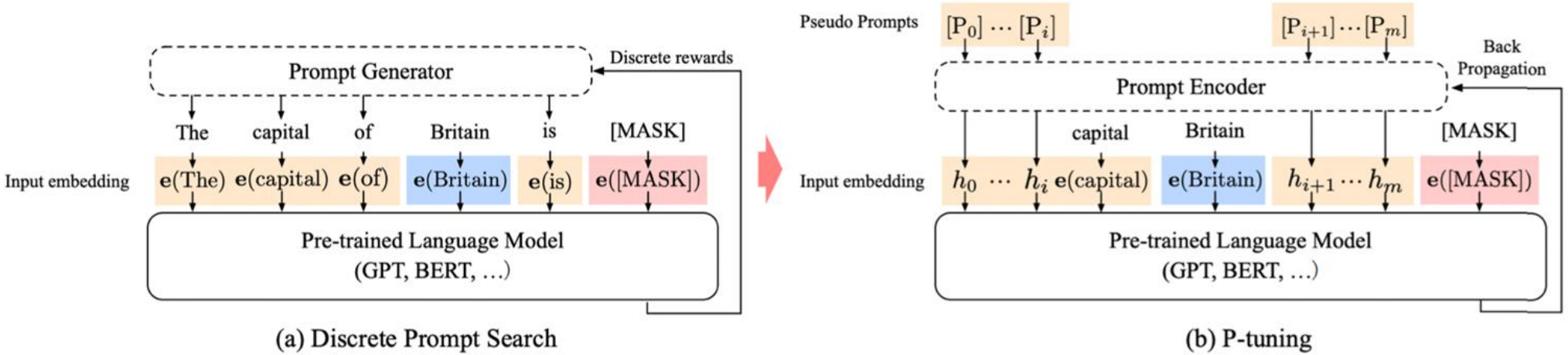
Prompt에 대한 파라미터 없이 LM을 Prompt에 사용하면서 fine-tuning하는 방식

- Tuning-free Prompting 방식에서 LM의 parameter를 fine-tuning하는 방법
- 관련 연구
  - PET-TC(Schick and Shütze, 2021), PET-Gen(Schick and Shütze, 2020), LM-BFF(Gao et al. 2021) 등
  - Logan IV et al.(2021)
- 장점
  - Prompting method를 이용해서 모델에 task를 보다 명확하게 지시하여 효과적인 학습 가능(특히 few-shot learning)
- 단점
  - Prompt 또는 Answer engineering 필요
  - LM을 Fine-tuning을 하기 때문에 한 downstream task에 학습된 모델은 다른 task에 좋은 성능을 얻지 못 할 수 있음

Prompt+LM Fine-tuning

- Prompt parameter + LM의 parameter 일부 또는 전부를 fine-tuning
  - 기존 pre-train & fine-tune 방식과 비슷하지만, 서로 다른 prompt가 model 학습에 bootstrapping 효과
- 관련 연구
  - PADA(Ben-David et al., 2021), P-Tuning(Liu et al., 2021)
- 장점
  - 가장 expressive한 방법(표현력이 좋은 방법), dataset이 충분할 때 적합
- 단점
  - model의 모든 parameters를 학습하고 저장해야함
  - 작은 데이터셋에 과적합 가능성

P-Tuning



(a) Discrete Prompt Search

(b) P-tuning

Prompt type	Model	P@1
Original (MP)	BERT-base	31.1
	BERT-large	32.3
	E-BERT	36.2
Discrete	LPAQA (BERT-base)	34.1
	LPAQA (BERT-large)	39.4
	AutoPrompt (BERT-base)	43.3
P-tuning	BERT-base	48.3
	BERT-large	**50.6**

Model	MP	FT	MP+FT	P-tuning
BERT-base (109M)	31.7	51.6	52.1	52.3 (+20.6)
-AutoPrompt (Shin et al., 2020)	-	-	-	45.2
BERT-large (335M)	33.5	54.0	55.0	54.6 (+21.1)
RoBERTa-base (125M)	18.4	49.2	50.0	49.3 (+30.9)
-AutoPrompt (Shin et al., 2020)	-	-	-	40.0
RoBERTa-large (355M)	22.1	52.3	52.4	53.5 (+31.4)
GPT2-medium (345M)	20.3	41.9	38.2	46.5 (+26.2)
GPT2-xl (1.5B)	22.8	44.9	46.5	54.4 (+31.6)
MegatronLM (11B)	23.1	OOM*	OOM*	**64.2** (+41.1)

* MegatronLM (11B) is too large for effective fine-tuning.

Table 2. Knowledge probing Precision@1 on LAMA-34k (left) and LAMA-29k (right). P-tuning outperforms all the discrete prompt searching baselines. And interestingly, despite fixed pre-trained model parameters, P-tuning overwhelms the fine-tuning GPTs in LAMA-29k. (MP: Manual prompt; FT: Fine-tuning; MP+FT: Manual prompt augmented fine-tuning; PT: P-tuning).

Challenges

1. PromptDesign
   a. Tasks beyond Classification and Generation(기존의 prompt연구가 Classification과 Generation 위주로 연구가 진행됨) → 정보 추출 또는 텍스트 분석 등의 task에 확장
   b. Prompting with Structured Information(구조화된 데이터에 대해서도 처리할 수 있는 prompt 연구도 필요하다)

      →NLP task에서는 Tree, graph, table 등의 형태에 대해서 처리할 수 있는 prompt 또는 answer engineering이 필요

         →Htlm(Aghajanyan et al., 2021): HTML을 이용한 구조화된 prompt

   c. Entanglement of Template and Answer

      모델의 성능이 template과 answer 둘 모두에 영향을 받기 때문에 (높은 성능을 낼 수 있는)둘의 최적 조합을 찾는 것이 문제

      최근 연구들은 template을 선택하기 전에 answer를 선택(LM-BFF, AutoPrompt)

      WARP(Hambardzumyan et al., 2021)에서는 이 둘을 동시에 학습할 수 있다는 가능성을 보임

Challenges

2. AnswerEngineering
    a. Many-class and Long-answer Classification Tasks

        →분류해야할 class가 매우 많은 경우, 적절한 answer space를 설정하기 어려움

        →여러개의 token으로 이뤄진 긴 answer의 경우, 어떻게 decoding할 것인지

    b. Multiple Answer for Generation Tasks

        →Text 생성 문제에서 동일한 의미의 문장이 다양한 구문 구조로 표현될 수 있음

        →현재까진 대부분의 연구가 하나의 정답만을 두고 평가했음

        학습시에 다양한 레퍼런스(문장)를 참조하도록 하는 방법 연구 필요
3. Selection of Tuning Strategy

    Prompt와 LM의 parameter를 tuning하는 방법의 tradeoff에 대한 체계적인 탐구 필요

Challenges

4. Multiple Prompt Learning
    a. Prompt Ensembling

        더 많은 Prompt를 ensemble할 수록 시공간 복잡도 증가 ➔ 이를 완화하는 방법(PET → iPET)

        ensemble을 적용할 때 효과적인 prompt를 선택하는 방법에 대한 연구

        텍스트 생성 task에 대한 ensemble에 대한 연구

    b. Prompt Composition & Decomposition

        Span relation prediction(e.g. entity coreference)에는 composition이 Token 또는 span prediction(e.g. NER)에는 decomposition이 경험적으로 더 낫다고 알려져 있음

        더 다양한 상황에서 이 둘을 적용하는 것에 대한 연구

    c. **Prompt Augmentation**

        효과적인 예시를 선택하는 방법과 적절한 순서로 제시하는 방법에 대한 연구 필요

    d. Prompt Sharing

        서로 다른 task나 domain, 언어에 대해서 공유하여 사용될 수 있는 prompt에 대한 연구