Automated Template Engineering

Artur Komaristych, Lukas Wilkens, Mariia Galevskaia, Frederik Schmeißer

Structure

- Introduction to Manual Template Engineering
  - Demonstration Designing
- Automated Template Engineering Techniques
  - Prompt Mining
  - Prompt Paraphrasing
  - Prompt Generation
  - Prompt Scoring
- Experiments
- Deep Dive: BERTese
- Deep Dive: Prefix Tuning
- Conclusion
- Sources

From Manual to Automatic

Last Week: Manual ways of creating prompt templates

Some weaknesses:

1. Manual prompting requires a lot of time and expertise
2. Manually finding optimal prompts is often impossible

The Solution: **Automated Template Generation**

Introduction - Automated Template Engineering

Templates can be created / improved through numerous processes

Some distinctions:

Cloze vs. Prefix

Static vs. Dynamic

Discrete vs. Continuous

Cloze vs. Prefix

Cloze = Masked Token inside of the prompt

“I love this movie, it is a [Z] movie”

Prefix = Prompt precedes answer-space

“I love this movie. What’s the sentiment of the review? [Z]”

Static vs. Dynamic

Static = Use the same prompt-template for each input

“[X] founded [Z]”

“Jeff Bezos founded [Z]”

“Bill Gates founded [Z]”

Dynamic = Generate a custom template for each input

“Mark Zuckerberg founded [Z]”

“Jeff Bezos is the founder of [Z]”

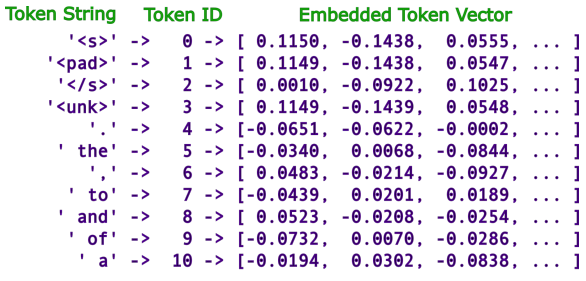
“Bill Gates created [Z]”

Discrete vs. Continuous

Discrete = Strings of text

“Who is the president of the United States? [Z]”

Continuous = Prompt is described in embedding-space



How would you automate Template Generation?

Which aspects / parts of prompts could be improved automatically?

What methods / tools could be used for this?

Demonstration Designing

Connection: Template Design and ICL ➜ Few Shot Prompting

How do you find the best examples for ICL?

➜ You automate the process!

Split into Demonstration Organization and Formatting

Demonstration Organization

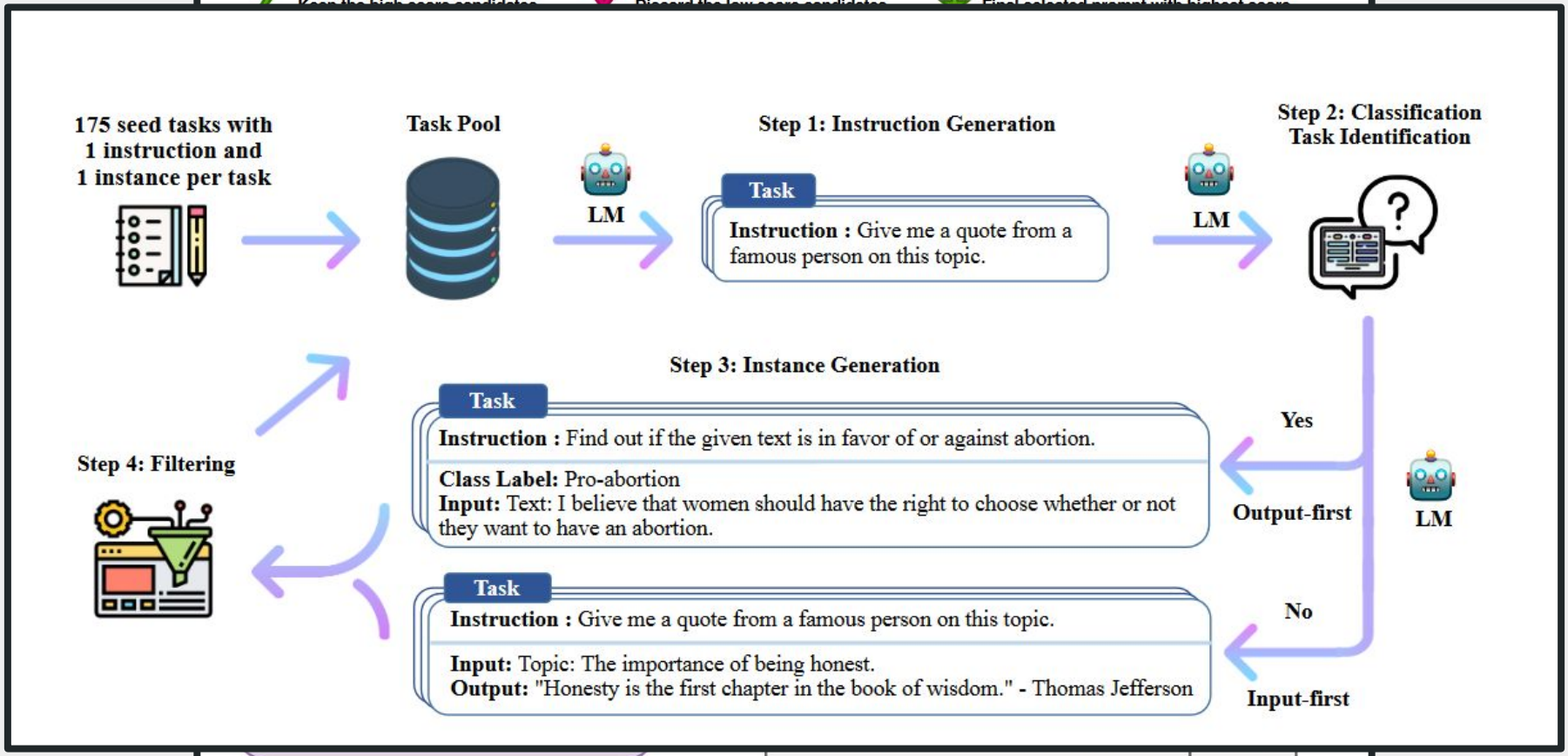
= Which examples in what order

Demonstration Selection = Finding good examples

- Unsupervised methods
  - Nearest Neighbours
  - Information overlap
  - Perplexity and diversity
  - Let LLMs generate the examples

Demonstration Ordering = Finding an effective order

- Automatic sorting based on proximity and entropy scores


175 seed tasks with
1 instruction and
1 instance per task
Task Pool
LM
Step 1: Instruction Generation
Task
Instruction : Give me a quote from a famous person on this topic.
LM
Step 2: Classification Task Identification
Step 3: Instance Generation
Task
Instruction : Find out if the given text is in favor of or against abortion.
Class Label: Pro-abortion
Input: Text: I believe that women should have the right to choose whether or not they want to have an abortion.
Yes
Output-first
LM
Task
Instruction : Give me a quote from a famous person on this topic.
Input: Topic: The importance of being honest.
Output: "Honesty is the first chapter in the book of wisdom." - Thomas Jefferson
No
Input-first
Step 4: Filtering

Automated Template Engineering Techniques

Prompt Mining

Distant Supervision Assumption: If two words x <subject>, y <object> participate in a relation r, then any sentence that contain those two words might express r.

Idea: Mine sentences containing x and y from Websites, e.g., Wikipedia, and extract them as templates.

WIKIPEDIA

WIKIDATA

Input: x <subject> and y <object>
Output: Template describing the relation r

Source: How Can We Know What Language Models Know? (2021)

Prompt Mining

Example: (Barack Obama, Hawaii)

1. Search for sentences that contain Barack Obama and Hawaii.
    a. Wikipedia: Barack Obama was born in Hawaii.
2. Substitute Barack Obama and Hawaii with placeholders.
    a. Template: **[X]** was born in **[Y]**.

Middle-word Prompts: [X] middle words [Y]

Source: How Can We Know What Language Models Know? (2021)

Prompt Mining

Dependency-based Prompts

Idea: Build Prompts based on the shortest dependency path between the subject and object in the dependency path.

1. Sentence: The capital of France is Paris
2. Build dependency path from sentence
   a. France (pobj) ← of (prep) ← capital (nsubj) ← is (attr) → Paris
3. Assemble prompt as the path from the leftmost word to the rightmost word in the dependency path.

Template: capital of [X] is [Y]

Source: How Can We Know What Language Models Know? (2021)

Prompt Paraphrasing

Idea: Take in an existing seed prompt, paraphrase it into a set of other candidate prompts, and then selects the one that achieves the highest training accuracy on the target task.

- Round-trip translation: Translate prompt into another language then back
- Replacement of phrases from a thesaurus
- Neural prompt rewriters, e.g., PRewrite

Source: How Can We Know What Language Models Know? (2020), BARTScore: Evaluating Generated Text as Text Generation (2021)

Prompt Paraphrasing

Round-trip translation: Translate existing Prompt to different languages, e.g., EN ➜ DE ➜ RU ➜ EN

1. [X] was born in the city of [Y]
2. [X] wurde in der Stadt [Y] geboren
3. [X] е роден в град [Y]
4. [X] was born in [Y]

Source: How Can We Know What Language Models Know? (2020), BARTScore: Evaluating Generated Text as Text Generation (2021)

Prompt Paraphrasing

Replacement of phrases from a thesaurus: Replace words with Synonyms and Antonyms

1. [X] was born in the city of [Y]
2. [X] was born in the **metropolitan area** of [Y]
3. [X] was **brought into this world** in the metropolitan area of [Y]

Source: How Can We Know What Language Models Know? (2020), BARTScore: Evaluating Generated Text as Text Generation (2021)

Prompt Generation

Idea: Treat the generation of prompts as a text generation task and use standard natural language generation models, e.g., seq2seq T5, to generate prompts.

T5 functionality:

Input: “Thank you <X> me to your party <Y> week”

Output: “<X> for inviting <Y> last <Z>”

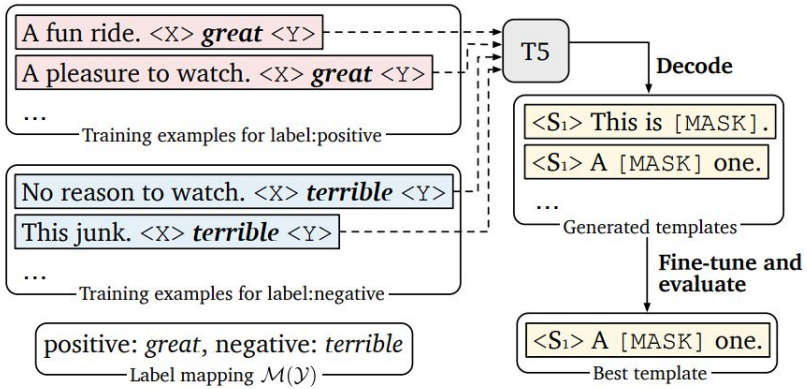
Source: PADA: Example-based Prompt Learning for on-the-fly Adaptation to Unseen Domains (2022), Making Pre-trained Language Models Better Few-shot Learners (2021)

Prompt Generation (Finetuning)

Step 1: Find ideal label words

Template	Label words	Accuracy
SST-2 (positive/negative)		mean (std)
<S_1> It was [MASK] .	great/terrible	**92.7 (0.9)**
<S_1> It was [MASK] .	good/bad	92.5 (1.0)
<S_1> It was [MASK] .	cat/dog	91.5 (1.4)
<S_1> It was [MASK] .	dog/cat	86.2 (5.4)
<S_1> It was [MASK] .	terrible/great	83.2 (6.9)
Fine-tuning	-	81.4 (3.8)

Step 2: Generate good templates based on label words



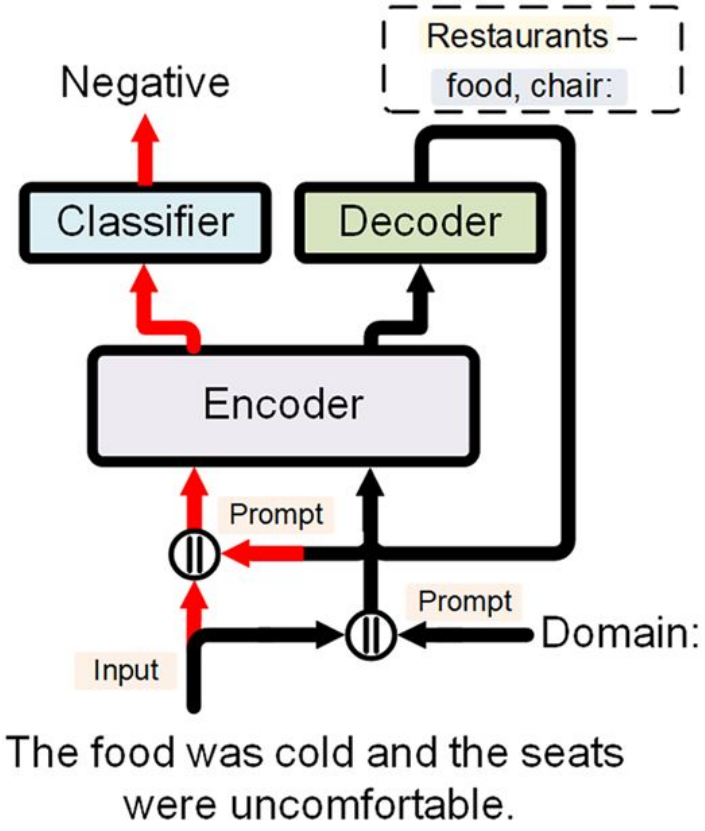
Source: PADA: Example-based Prompt Learning for on-the-fly Adaptation to Unseen Domains (2022), Making Pre-trained Language Models Better Few-shot Learners (2021)

Prompt Generation (PADA)

DRFs = Domain Related Features

First Step: Generate DRFs and Domain

Second Step: Append DRFs to Prompt



Source: PADA: Example-based Prompt Learning for on-the-fly Adaptation to Unseen Domains (2022), Making Pre-trained Language Models Better Few-shot Learners (2021)

Prompt Scoring

Idea: Build templates from head-relation-tail triple x = (h, r, t)

Input: (musician, CapableOf, play musical instrument)

- “HEAD is capable of TAIL”
- “HEAD is able to TAIL”
- “HEAD has the ability to TAIL”
- ....

Source: Commonsense Knowledge Mining from Pretrained Models (2019)

Prompt Scoring

Idea: Build templates from head-relation-tail triple x = (h, r, t)

Input: (musician, CapableOf, play musical instrument)

- “HEAD is capable of TAIL”
  - “Musician is capable of play musical instrument”
- “HEAD is able to TAIL”
  - “Musician is able to play musical instrument”
- “HEAD has the ability to TAIL”
  - “Musician has the ability to play musical instrument”
- ....

Source: Commonsense Knowledge Mining from Pretrained Models (2019)

Prompt Scoring

Resulting “raw” prompts are then refined in order to fix grammar issues etc.

- “HEAD is capable of TAIL”
  - “Musician is capable of play**ing a** musical instrument”
- “HEAD is able to TAIL”
  - “Musician is able to play **a** musical instrument”
- “HEAD has the ability to TAIL”
  - “Musician has the ability to play **a** musical instrument”

Source: Commonsense Knowledge Mining from Pretrained Models (2019)

Prompt Scoring

Lastly, all prompts are evaluated by an LMM and the one with the highest likelihood for producing the correct result is selected.

Candidate Sentence S_i	$\log p(S_i)$
“musician can playing musical instrument”	-5.7
“musician can be play musical instrument”	-4.9
“musician often play musical instrument”	-5.5
“a musician can play a musical instrument”	$\mathbf{-2.9}$

Source: Commonsense Knowledge Mining from Pretrained Models (2019)

Prompt Scoring

Lastly, all prompts are evaluated by an LMM and the one with the highest likelihood for producing the correct result is selected.

Candidate Sentence S_i	$\log p(S_i)$
"musician can playing musical instrument"	-5.7
"musician can be play musical instrument"	-4.9
"musician often play musical instrument"	-5.5
"a musician can play a musical instrument"	$\mathbf{-2.9}$

Source: Commonsense Knowledge Mining from Pretrained Models (2019)

Experiments

BERTese: Learning to Speak to BERT

Adi Haviv, Jonathan Berant and Amir Globerson (2021)

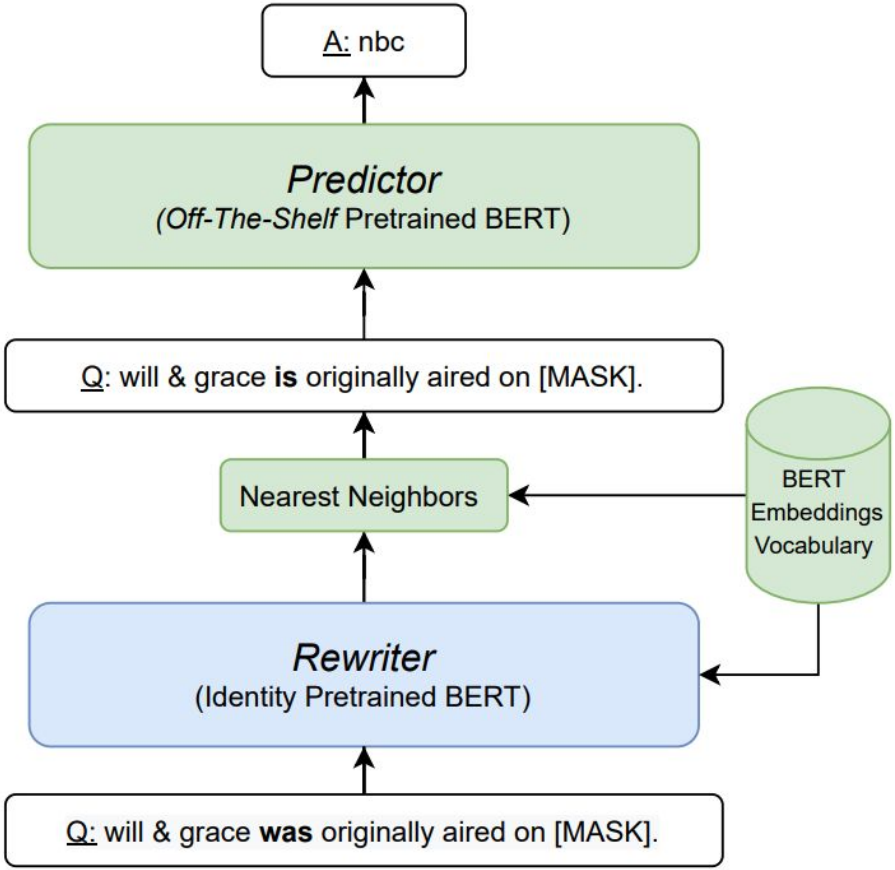
Why was BERTese created?

- Previous approaches to knowledge retrieval often struggled with determining why the BERT model could not correctly answer a query – whether due to a lack of knowledge or a misunderstanding of the query.

- To address this issue, the authors proposed an automatic query reformulation method, enabling the model to better understand and process queries, leading to more accurate responses.

Structure of the BERTese Model

1. **Anfragen-Umformulierer (Rewriter)**
2. **Prädiktor (Predictor)**



Rewriter Pre-training

Loss functions:

- Valid Token Loss (VTL)
- Single [MASK] Loss (SML)
- Prediction loss

Valid Token Loss (VTL)

- Encourages the use of correct language tokens.

(original masked query) *(rewritten masked query)*

Q:“The capital of France is [MASK]” ⟶ Q:“France’s capital is [MASK]”

“France’s capittal is [MASK]”

(“capittal” is an incorrect token)

Single [MASK] Loss (SML)

- Ensures that exactly one [MASK] token is included in the reformulated query.
- Signals the model, what word should be replaced or completed.
- The precise place of token plays a crucial role in the accuracy of an answer.

Prediction Loss + Straight-Through Estimator (STE)

- Minimises the difference between the predicted and correct answers. **STE** allows the training of non-differentiable operations, which is crucial for working with discrete tokens.

(rewritten masked query)
Q: “France’s capital is [MASK]”

A: Paris ✅

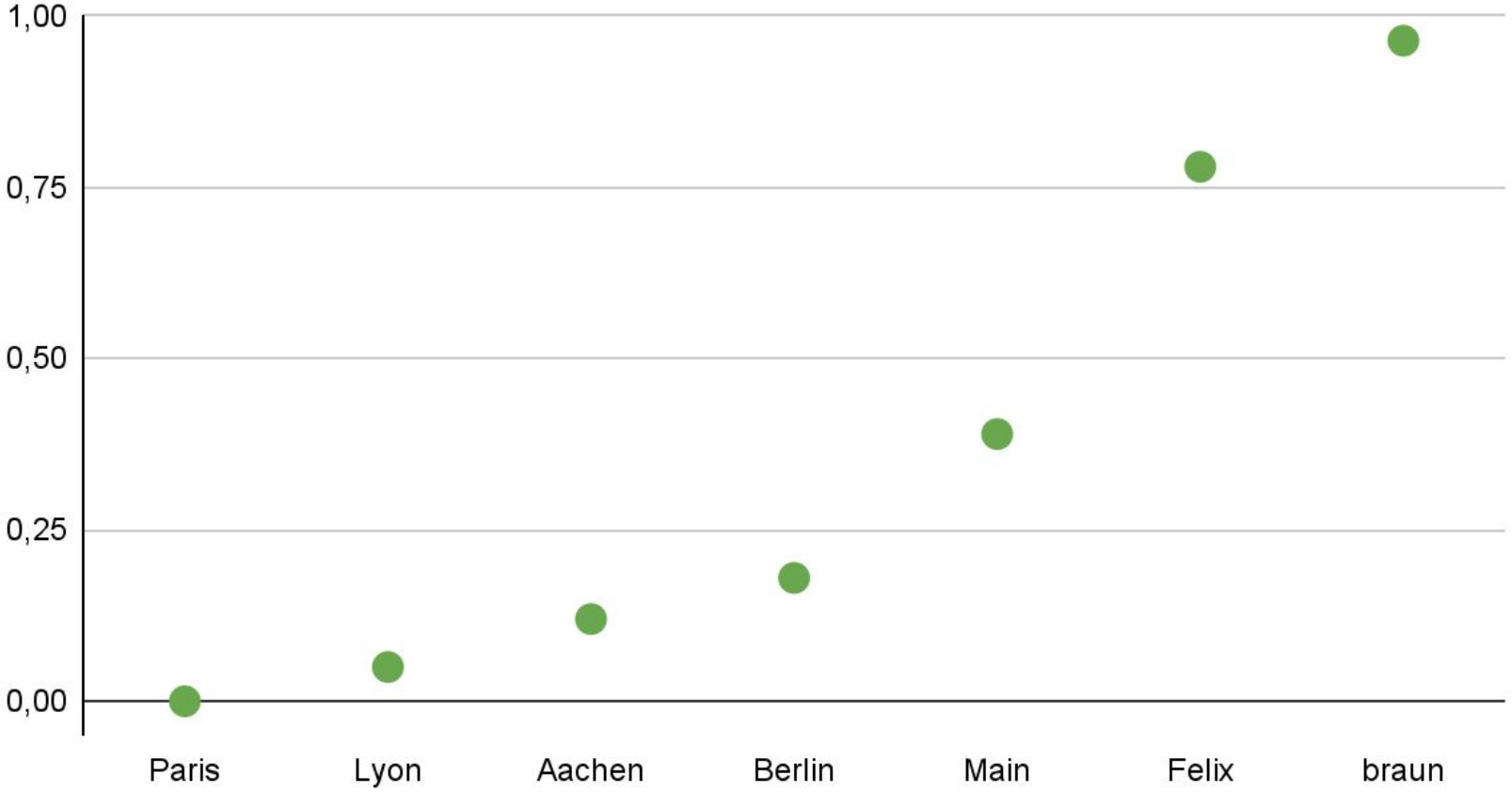
$$\mathcal{L}(y_{\text{true}}, y_{\text{pred}}) \to 0 \quad \textbf{if} \quad y_{\text{pred}} \to y_{\text{true}}$$

(rewritten masked query)
Q: “France’s capital is [MASK]”

A: Lyon ❌

$$\mathcal{L}(y_{\text{true}}, \text{“Lyon”}) > \mathcal{L}(y_{\text{true}}, y_{\text{true}})$$

Prediction loss



Evaluation of Functionality and Efficiency of the Model

Evaluierung:

- **LAMA** (Petroni et al., 2019) – collection of cloze-style queries about relational facts with a single token answer.

- **T-Rex** (Elsahar et al., 2018) – constructed out of 41 relations, each associated with at most 1000 queries from Wikidata.

For model's training:

- **T-Rex-train** (Jiang et al., 2020) – constructed from Wikidata, no overlap with original T-Rex.

Baselines

Corpus	BERT	FT-BERT	LPAQA	BERTese
T-REx	31.1	36	34.1	**38.3**

- **BERT$_{base}$** (Petroni et al., 2019) – no fine-tuning.
- **LPAQA** (Jiang et al., 2020) – based on mining additional paraphrase queries.
- **FT-BERT** – end-to-end differentiable BERT$_{base}$ model, fine-tuned on T-Rex-train to output the correct answer.

Modifications made by BERTese

Modification	Original Masked Query	Bertese Masked Query
"!" removed	yahoo**!** tech is owned by [MASK].	yahoo tech is owned by [MASK].
verb patterns	**working** dog is a subclass of [MASK].	**work** dog is a subclass of [MASK].
was → is	will & grace **was** originally aired on [MASK].	will & grace **is** originally aired on [MASK].
a → the	tom terriss is **a** [MASK] by profession.	tom terriss is **the** [MASK] by profession.
rephrasing	istanbul hezarfen **airfield** is named after [MASK].	istanbul hezarfen **airport** is named after [MASK].
token → [SEP]	**lub**ka kolessa plays [MASK].	**[SEP]**ka kolessa plays [MASK].

Ablation Study

- here the general functionality of the model is checked
- at this step, some components or functions are being taken out from the model, to look what effect it has on its functionality
- **P@1** = precision at one

Ablation	P@1
No auxilary losses	25.3
SML	36.6
VTL	37.5
SML + VTL (BERTese)	38.3

Part of Speech Analysis

- **more than 70%** – nouns (NN) and verbs (VBN)
- **15%** – determiners (DT)

POS Tag	Frequency
NN	47.6%
VBN	23%
DT	15.3%
JJ	4.4%
CD	3%
NNP	1.7%
NNS	1.3%

**JJ – adjectives, CD – numerals, NNP – proper noun sg., NNS – proper noun pl.*

Conclusion

- This approach substantially improves the interaction with LLMs.
- By automatically reformulating queries, the understandability for the model is increased, resulting in more accurate answers.
- The results of the experiments confirm the effectiveness and utility of this approach in various application scenarios.

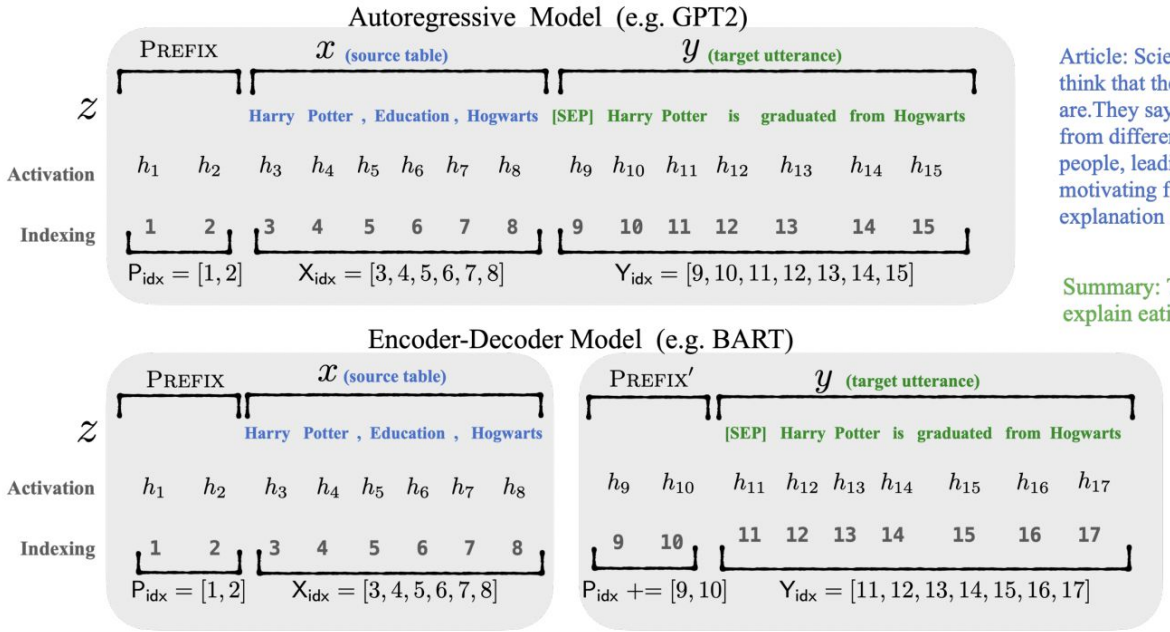
Prefix-Tuning: Optimizing Continuous Prompts for Generation

Xiang Lisa Li, Percy Liang (2021)

Prefix Tuning - Idea

- **Goal:** Optimize LLM for multiple downstream tasks at once
- Similar to prompting > Adds continuous, task-specific vectors, not real tokens
  - Not limited to real tokens, more “precise”
  - Because the prefix-vectors are continuous, there are no examples for them
- Alternative to fine-tuning, LLM parameters are frozen
- **Intuition**: Steer the LLM in the right direction

Prefix Tuning - “Example”



Summarization Example

Article: Scientists at University College London discovered people tend to think that their hands are wider and their fingers are shorter than they truly are.They say the confusion may lie in the way the brain receives information from different parts of the body.Distorted perception may dominate in some people, leading to body image problems ... [ignoring 308 words] could be very motivating for people with eating disorders to know that there was a biological explanation for their experiences, rather than feeling it was their fault."

Summary: The brain naturally distorts body image - a finding which could explain eating disorders like anorexia, say experts.

Table-to-text Example

Table: name[Clowns] customer-rating[1 out of 5] eatType[coffee shop] food[Chinese] area[riverside] near[Clare Hall]

Textual Description: Clowns is a coffee shop in the riverside area near Clare Hall that has a rating 1 out of 5 . They serve Chinese food .

Li, Liang (2021)

Prefix Tuning - Evaluation

- Evaluated for summarization (BART) & table to text (GPT-2)
- One dataset for summarization
- Three datasets for table to text
- Compared to fine-tuning, lightweight fine-tuning, adapter tuning
- Metrics:
  - Table to text: **BLEU, METEOR, TER**, ... (Others depending on dataset)
  - Summarization: **ROUGE**
- **Prefix size**: 0.1% new, task-specific parameters (ca. 250k)

Prefix Tuning - Table to Text

	E2E					WebNLG									DART					
	BLEU	NIST	MET	R-L	CIDEr	BLEU			MET			TER ↓			BLEU	MET	TER ↓	Mover	BERT	BLEURT
						S	U	A	S	U	A	S	U	A						
GPT-2$_{\text{MEDIUM}}$																				
FT-FULL	68.8	8.71	46.1	71.1	2.43	**64.7**	26.7	45.7	**0.46**	0.30	0.38	**0.33**	0.78	0.54	46.2	**0.39**	**0.46**	**0.50**	**0.94**	**0.39**
FT-TOP2	68.1	8.59	46.0	70.8	2.41	53.6	18.9	36.0	0.38	0.23	0.31	0.49	0.99	0.72	41.0	0.34	0.56	0.43	0.93	0.21
ADAPTER(3%)	68.9	8.71	46.1	71.3	**2.47**	60.5	**47.9**	54.8	0.43	**0.38**	**0.41**	0.35	**0.46**	**0.39**	45.2	0.38	**0.46**	**0.50**	**0.94**	**0.39**
ADAPTER(0.1%)	66.3	8.41	45.0	69.8	2.40	54.5	45.1	50.2	0.39	0.36	0.38	0.40	0.46	0.43	42.4	0.36	0.48	0.47	**0.94**	0.33
PREFIX(0.1%)	**70.3**	**8.82**	**46.3**	**72.1**	2.46	62.9	45.3	**55.0**	0.44	0.37	**0.41**	0.35	0.51	0.42	**46.4**	0.38	**0.46**	**0.50**	**0.94**	**0.39**
GPT-2$_{\text{LARGE}}$																				
FT-FULL	68.5	8.78	46.0	69.9	2.45	**65.3**	43.1	55.5	**0.46**	0.38	**0.42**	**0.33**	0.53	0.42	**47.0**	**0.39**	0.46	**0.51**	**0.94**	**0.40**
Prefix	**70.3**	**8.85**	**46.2**	**71.7**	**2.47**	63.4	**47.7**	**56.3**	0.45	**0.39**	**0.42**	0.34	**0.48**	**0.40**	46.7	**0.39**	**0.45**	**0.51**	**0.94**	**0.40**
SOTA	68.6	8.70	45.3	70.8	2.37	63.9	52.8	57.1	0.46	0.41	0.44	-	-	-	-	-	-	-	-	-

Li, Liang (2021)

Prefix Tuning - Summarization Results

	R-1 ↑	R-2 ↑	R-L ↑
FT-FULL(Lewis et al., 2020)	45.14	22.27	37.25
PREFIX(2%)	43.80	20.93	36.05
PREFIX(0.1%)	42.92	20.03	35.05

Performance on XSUM Dataset Li, Liang (2021)

Prefix Tuning - Results (Summarized)

Table to text:

- Outperforms other light-weight approaches
- Similar performance to full fine-tuning
- Prefix tuning outperforms fine-tuning in low data setting
- More efficient: similar performance, with fewer parameters (30x)

Summarization:

- Fine-tuning outperforms prefix tuning
- Prefix tuning outperforms fine-tuning in low data setting

Fine tuning seems more appropriate for complex tasks and more data
Prefix tuning may be a good alternative to lightweight fine-tuning

Conclusion

- Automated Template Engineering has advantages over Manual
  - Faster template generation
  - More efficient / “ideal” templates
  - More and better options compared to manual engineering
  - => Better knowledge elicitation
- However: Not necessarily intuitive / understandable (e.g. prefix tuning)
- A good response is relative to the goal, real life use cases commonly more challenging
- Human-Machine interaction as knowledge bottleneck

Sources

Dong, Qingxiu, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. “A Survey on In-Context Learning,” 2023. https://arxiv.org/abs/2301.00234.

Eyal Ben-David, Nadav Oved, Roi Reichart. “PADA: Example-based Prompt Learning for on-the-fly Adaptation to Unseen Domains.” Transactions of the Association for Computational Linguistics 2022; 10 414–433. doi: https://doi.org/10.1162/tacl_a_00468

Haviv, Adi, Jonathan Berant, and Amir Globerson. “BERTese: Learning to Speak to BERT.” In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, edited by Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty, 3618–23. Online: Association for Computational Linguistics, 2021. https://doi.org/10.18653/v1/2021.eacl-main.316.

Li, Xiang Lisa, and Percy Liang. “Prefix-Tuning: Optimizing Continuous Prompts for Generation.” In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, edited by Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, 4582–97. Online: Association for Computational Linguistics, 2021. https://doi.org/10.18653/v1/2021.acl-long.353.

Liu, Pengfei, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. “Pre-Train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing.” *ACM Comput. Surv.* 55, no. 9 (January 2023). https://doi.org/10.1145/3560815.

Tianyu Gao, Adam Fisch, and Danqi Chen. “Making pre-trained language models better few-shot learners.” In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL’21)*. 2021.

Wang, Yizhong, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022b. Self-instruct: Aligning language model with self generated instructions. ArXiv preprint, abs/2212.10560. https://arxiv.org/abs/2212.10560

Zhou, Denny, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Olivier Bousquet, Quoc Le, and Ed Chi. 2022. Least-to-most prompting enables complex reasoning in large language models. ArXiv preprint, abs/2205.10625. https://arxiv.org/abs/2205.10625

Zhou, Yongchao, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2022c. Large language models are human-level prompt engineers. ArXiv preprint, abs/2211.01910. https://arxiv.org/abs/2211.01910