GPS: Genetic Prompt Search for Efficient Few-shot Learning

CS410.P21

Presented by:

Tang Nhat[1] Le Minh Nhut[1] Tran Dinh Khanh Dang[1]

Ho Trong Duy Quang[1] Vo Dinh Khanh[1]

Instructor: PhD. Luong Ngoc Hoang[1]

[1]Department of Computer Science
University of Information Technology

CS410 Final Project Presentation, June 13th 2025

Table of Contents

Table of Contents

Pre-trained Language Model (PLMs)

- Pre-trained Language Models (PLMs) like **BERT** [1], **T5** [2], **GPT** series have become foundational in Natural Language Processing (NLP). These models are trained on massive text corpora, thus acquiring broad linguistic knowledge.
- The standard paradigm involves **pretraining** followed by **fine-tuning**. While effective, the fine-tuning approach requires a **substantial labeled datasets** for each downstream task to achieve high performance.

PLMs Enhancing Methods

- Manual Prompt Engineering
- Parameters-Based Methods
- Parameter-Frozen Methods

Prompting

- Another paradigm, "prompting", particularly catalyzed by large models **GPT-3** [3]. Instead of reformulating downstream tasks to fit the model's pretraining objectives, prompting reformulates the input to match the model's original pretraining format. This is done by adding a textual "prompt" or template to the input example.
  - For instance, a sentiment analysis task might be framed as: "Input: [Sentence]. Sentiment: [MASK]".
- Prompting has shown remarkable potential for **In-Context learning**, allowing PLMs to perform tasks with minimal or no task-specific examples, simply by providing the right instructions or demonstrations within the prompt.

The Challenge of Manual Prompt Engineering

- Prompting's effectiveness heavily depends on the quality of the prompt template. It requires significant human effort, domain expertise, and extensive trial-and-error. Furthermore, some research indicates that:
    1. Manually crafted prompts are frequently **suboptimal**.
    2. PLM performance can be **highly sensitive to minor changes** in prompt phrasing, leading to unstable results.
    3. Optimal prompts often **vary significantly across different tasks** and even different PLMs.
- Recent efforts have tried to mitigate this by collecting diverse prompts or using human feedback, often involving large-scale data collection or model fine-tuning.

Parameters-based Tuning

These methods adapt the PLM to the downstream task using limited data, involving gradient-based updates.

- **Model Fine-Tuning**: Some few-shot tuning methods focus on template design and update all the parameters of pretrained language models.
- **Parameter-Efficient Fine-Tuning (PEFT)**: These methods aim to reduce the computational and storage cost of fine-tuning by updating only a small subset of parameters or adding small auxiliary modules. Some methods fall into this category are:

Parameter-Efficient Fine-Tuning (PEFT)

Parameter-Efficient Fine-Tuning (PEFT): These methods aim to reduce the computational and storage cost of fine-tuning by updating only a small subset of parameters or adding small auxiliary modules. Some methods fall into this category are:

- **Prompt Tuning [4]**
- **Adapters [5], Bit-Fit [6], LoRA [7]**: Involve adding or modifying specific modules or parameters within the model
- **Black-Box Tuning [8]**

Prompt Tuning

Prompt Tuning [4]: Optimizes continuous vector tokens (soft prompts) in prompts via gradient-based optimization, while the pretrained language model remains frozen

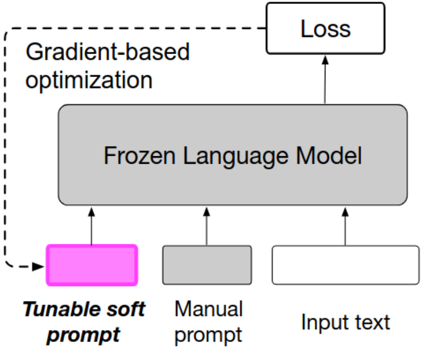

Figure 1: Prompt Tuning Pipeline

Black-Box Tuning [8]: A gradient-free optimization method for prompt tuning, but it searches for continuous prompt embeddings rather than discrete text prompts

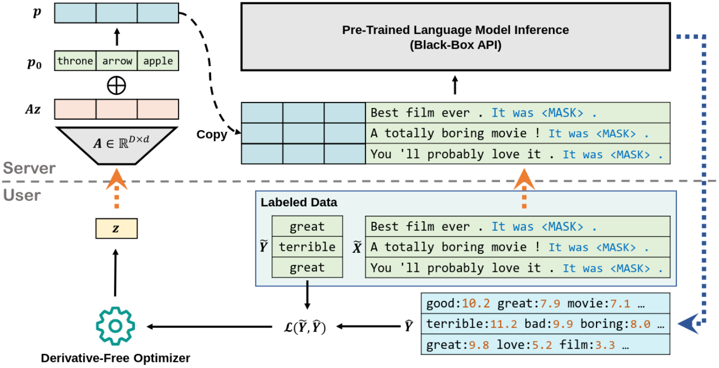

Figure 2: BPT Pipeline

Parameters-based Tuning

These methods adapt the PLM to the downstream task using limited data, involving gradient-based updates.

- **Model Fine-Tuning**: Some few-shot tuning methods focus on template design and update all the parameters of pretrained language models.
- **Parameter-Efficient Fine-Tuning (PEFT)**: These methods aim to reduce the computational and storage cost of fine-tuning by updating only a small subset of parameters or adding small auxiliary modules. Some methods fall into this category are:

Limitations: While it can improve performance, it is time-consuming and requires subjective interpretation.

Parameter-Frozen Tuning

These methods focus on improving performance by finding better discrete prompts or demonstrations, without changing model weights.

- **In-Context Learning (ICL)**: Requires human efforts to provide manual prompts, and sensitive to labeled examples.
- **GRIPS (Gradient-Free) [9]**: Concurent method.
- **GPS (Gradient-Free)**: Proposed method.

GRIPS - Gradient-free Instructional Prompt Search

- Is an **edit-based search** approach that focuses on refining existing natural language instructions within prompts.
- GRIPS splits the input into syntactic phrases and applies edit operations: delete (`del`), swap (`swap`), paraphrase (`par`), and add (`add`) previously deleted phrases at random positions.

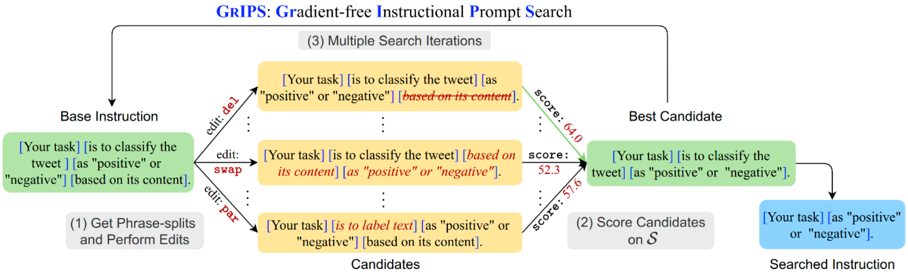

Figure 3: GRIPS Pipeline

Table of Contents

GPS: Genetic Prompt Search

- The paper proposes **Genetic Prompt Search (GPS)** as an automated, efficient method to discover high-performing prompts specifically for few-shot learning scenarios.
- **GPS** aims to address the limitations of manual prompting by:
  1. Automating Search.
  2. Easy and Cost-efficient.
  3. Low data Requirement.
  4. Improved Performance.

GPS: Genetic Prompt Search (cont.)

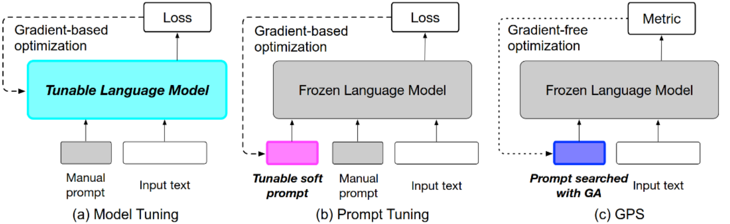

Figure 4: The paradigms of Model Tuning, Prompt Tuning, and GPS.

- **Model Tuning**: Gradient-based, update all PLM parameters, task specific, need a training set.
- **Prompt Tuning**: Gradient-based, PLM frozen, task specific, need a tranining set.
- **GPS**: Gradient-free, PLM frozen, need a small validation set.

GPS: Genetic Prompt Search

- GPS employs a **genetic algorithm** to automatically search for high-performing "hard prompts" (prompts in the discrete word space) to enhance few-shot learning.
- Unlike GRIPS's direct edits, GPS focuses on generating new prompt formulations through "reproduction" strategies
- GPS starts with a set of handcrafted prompts for initialization. It then iteratively "reproduces the current generation of prompts" and selects candidates based on their performance on a small validation set.

Genetic Prompt Search Algorithm

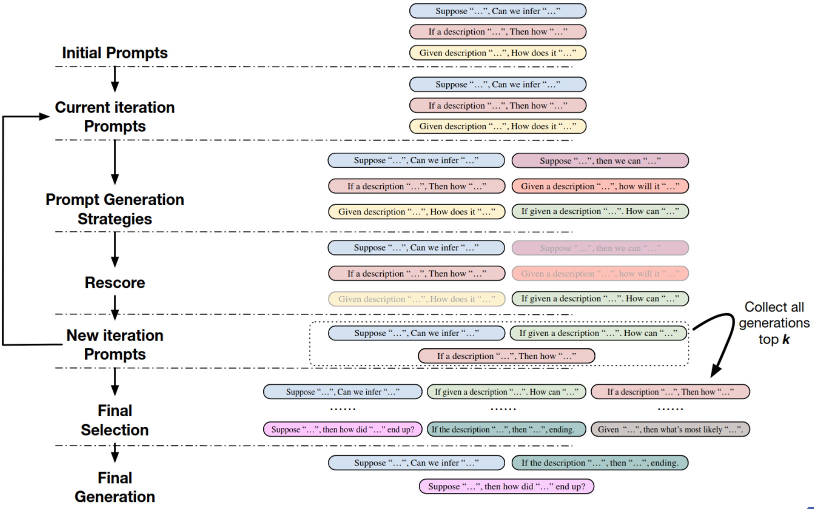

Figure 5: Overall pipeline of GPS algorithm.

Genetic Prompt Search Algorithm

Algorithm 1 Genetic Prompt Search

Require: $G^0, D_{\text{dev}}, f_{\text{GPS}}, g_{\text{GPS}}, T, K$;
Ensure: Final optimized prompts, G^{T+1}

1: obtain handcrafted prompts G^0 as initialization
2: **for** each $t \in [0, T]$ **do do**
3: store G^t
4: calculate score for each prompt in G^t using f_{GPS},
5: from G^t, select top K prompts as reproductive group G^t_*,
6: generate G^{t+1} based on G^t_* using g_{GPS},
7: **end for**
8: from stored $\{G^0_*, \ldots, G^T_*\}$, select top K prompts as optimal prompts group G^{T+1} using g_{GPS}.
9: **return** G^{T+1};

Prompt Generation Strategies

Three main strategies have been evaluated:

- **Back Translation**
- **Cloze**
- **Sentence Continuation**

Back Translation

- A common technique for data augmentation in NLP.
- Applied here for **prompt reproduction**.
- Steps:
  - Translate prompts from English to **11 languages**: **Chinese, Japanese, Korean, French, Spanish, Italian, Russian, German, Arabic, Greek, Cantonese**.
  - Then, translate back to English.
- **Prompt scoring**: score each prompt based on its **accuracy on** D_{dev}.

Example: Back Translation

Original Prompt:

"Summarize the following paragraph in one sentence."

Step 1 – Translate to Other Languages:

- Spanish: *Resume el siguiente párrafo en una oración.*
- French: *Résumez le paragraphe suivant en une seule phrase.*
- German: *Fassen Sie den folgenden Absatz in einem Satz zusammen.*

Step 2 – Translate Back to English:

- *"Summarize this paragraph in one sentence."*
- *"Provide a one-sentence summary of the paragraph below."*
- *"Condense the paragraph into a single sentence."*

Cloze

- A prompt generation approach based on the **cloze task** and **pretrained language models**.
- Initially follows **LM-BFF** (Gao et al., 2021b) for few-shot learning.
- Uses **T5** to automatically generate prompts by filling in templates with placeholders.
- This method performs poorly in a **no-parameter-update setting**.
- Instead, the authors:
  - Manually design prompts with some tokens replaced by placeholders.
  - Use **T5** to fill in the blanks.
- **Prompt scoring**: score the prompts with **average logits on** D_{dev}.

Example: Cloze

Manual Template:

*"The sentence of the sentence: '**The movie was amazing**' is ___________."*

T5 Fills the Placeholder:

- *"positive"*
- *"great"*
- *"favorable"*

Sentence Continuation

- An alternative approach for **prompt augmentation**.
- Inspired by **DINO** (Schick and Schütze, 2021).
- Uses a **pretrained language model** to generate new prompts.
- Uses the following template as input:
    - *"Write two sentences that mean the same thing: Sentence 1: Manual Prompt, Sentence 2:"*
- The model continues the prompt to generate **Sentence 2** as a new prompt.
- Experiments conducted with:
    - **GPT2-XL** (1.5B parameters)
    - **T5LM-XXL** (11B parameters)
- **Prompt scoring**: score each prompt using **accuracy on** D_{dev}.

Example: Sentence Continuation

Input Template:

"Write two sentences that mean the same thing:
Sentence 1: Classify the sentiment of the sentence.
Sentence 2:"

Generated Prompts (Sentence 2):

- *"Determine whether the sentiment is positive or negative."*
- *"Identify the emotional tone of the sentence."*
- *"Analyze the sentiment expressed in the sentence."*

Table of Contents

Evaluation Tasks

Evaluation Protocol:

- Use the **10 test tasks from T0**, which are *not included* in the training prompt set.
- The goal is to evaluate the performance of (**GPS**) and compare it to other baselines.
- For each task, we compute the **average accuracy** over different generated prompts.

- **Natural Language Inference:**
  - ANLI R1, ANLI R2, ANLI R3
  - CB, RTE
- **Coreference Resolution:**
  - WSC, Winogrande
- **Sentence Completion:**
  - COPA, HellaSwag
- **Word Sense Disambiguation:**
  - WiC

Setting up

- Due to computational cost constraints, we use **T0-3B** and **T5-XL**, instead of **T0 (11.1B)** and **T5-XXL** as used in the original paper.
- In each training run:
  - **T0-3B** is used for *prompt evaluation* via downstream metrics.
  - **T5-XL** is used for *prompt generation*.
- We reduce the **batch size** from **8 to 2**.
- The original paper sets `max_step` $= 9$, while we only experiment with values from **1 to 7**.

Overall Comparision

Methods	Serving Efficiency	Tunable Parameters	Performance	Computation Cost†
Model Tuning	✗	100%	61.73 (Paper results)	11.1x
Prompt Tuning	✓	$\sim 0.01\%$	58.56 (Paper results)	11.1x
Black-Box Tuning	✓	$\sim 0.001\%$	57.82 (Paper results)	9.3x
In-Context Learning	✗‡	0%	51.28 (Paper results)	0x
GRIPS	✓	0%	58.66 (Paper results)	Not mentioned
GPS	✓	0%	60.12 (Paper results)	1.0x
GPS	✓	0%	50.39 (Our Results)	1.0x

Table 1: Comparison of few-shot learning methods in efficiency, tunable parameters, performance, and computation cost. †: Includes training and prompt search. ‡: In-context learning incurs high inference cost due to long sequences.

GPS Experiment

Dataset \ max_step	Author	Our Results						
	9	1	2	3	4	5	6	7
anli.r1	44.06	33.81	33.98	33.94	34.28	33.70	33.52	33.65
anli.r2	38.10	33.11	32.55	31.91	31.53	31.71	32.12	32.14
anli.r3	41.51	33.97	34.44	34.14	34.54	34.73	34.68	34.91
hellaswag	38.85	27.28	27.04	26.67	27.05	27.60	28.01	28.28
super_glue.cb	80.12	43.56	55.36	56.67	57.03	56.89	56.18	56.31
super_glue.copa	93.50	73.09	74.98	75.67	76.24	74.08	75.98	75.41
super_glue.rte	84.22	64.55	69.66	72.53	72.09	73.68	73.25	73.79
super_glue.wic	57.65	50.69	52.34	53.09	53.44	53.76	55.08	55.01
super_glue.wsc	63.62	61.50	66.06	65.34	64.42	63.96	64.85	64.00
winogrande.winogrande.xl	59.59	51.20	51.44	51.44	50.76	50.14	50.60	50.60
Avg.	**60.12**	**47.81**	**50.08**	**50.19**	**50.22**	**50.27**	**50.36**	**50.39**

Table 2: Comparison between Author (max_step = 9) and Our Results (max_step = 1 $\rightarrow$ 7) on GPS

GPS Performance

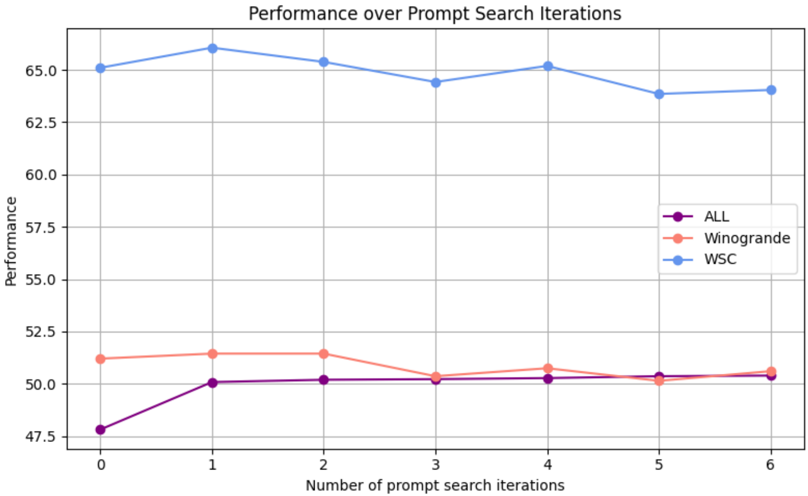

Figure 6: Performance over prompt search iteration (Our Results)

GPS Results I

Examples of original prompts and generated prompts by GPS (in red) for each dataset:

- **ANLI R1**: "{{premise}} Using only the above description and what you know about the world, "{{hypothesis}}" is definitely correct, incorrect, or inconclusive? ||| {{ answer_choices[label] }}"
- **ANLI R1**: "Given {{premise}} a test of "{{hypothesis}}" What is the conclusion of this test? ||| {{ answer_choices[label] }}"
- **ANLI R2**: {{premise}} Using only the above description and what you know about the world, "{{hypothesis}}" is definitely correct, incorrect, or inconclusive? ||| {{ answer_choices[label] }}
- **ANLI R2**: "{{premise}} What other alternative "{{hypothesis}}" is more likely to be true. ||| {{ answer_choices[label]

GPS Results

- **ANLI R3**: "{{premise}} Using only the above description and what you know about the world, "{{hypothesis}}" is definitely correct, incorrect, or inconclusive? ||| {{ answer_choices[label] }}
- **ANLY R3**: "{{premise}} What should be the "{{hypothesis}}" of this test? ||| {{ answer_choices[label]}}"
- **CB**: "Suppose {{premise}} Can we infer that "{{hypothesis}}"? Yes, no, or maybe? ||| {% if label !=-1 %}{{ answer_choices[label] }}{% endif %} "
- **CB**: "Inferred {{premise}} : "{{hypothesis}}" is true. No, no, or maybe? ||| {% if label != -1 %}{{ answer_choices[label] }}{% endif %}"

GPS Results

- **RTE**: "{{premise}} Using only the above description and what you know about the world, is "{{hypothesis}}" definitely correct? Yes or no? ||| {% if label != -1 %}{{ answer_choices[label] }}{% endif %}"
- **RTE**: "Yes, given that {{premise}} Therefore, it must be true that "{{hypothesis}}" ? Yes or no? ||| {% if label != -1 %}{{ answer_choices[label] }}{% endif %}"
- **WSC**: "{{ text }}In the previous sentence, does the pronoun "{{ span2_text.lower() }}" refer to {{ span1_text }}? Yes or no? ||| {% if label != -1 %}{{ answer_choices[label]}}{% endif %}"
- **WSC**: "{{ text }} In the above sentence, can the pronoun "{{ span2_text }}" be replaced with "{{ span1_text }}" ? Yes or no? ||| {% if label != -1 %}{{ answer_choices[label] }}{% endif %}"

GPS Results

- **Winogrande**: "{{ sentence }} _ refers to my brother {{ option1 }} or {{ option2 }}? ||| {% if answer == "1" %} {{option1}} {% else %} {{ option2 }} {% endif %}"
- **Winogrande**: "{{ sentence }} In the previous sentence, does _ refer to {{ option1 }} or {{ option2 }}? ||| {% if answer == "1" %} {{option1}} {% else %} {{ option2 }} {% endif %}"

GPS Results

- **COPA**: "Exercise: choose the most plausible alternative.
  {{ premise }} {% if question == "cause" %} because... {% else %}
  so... {% endif %}
  - {{choice1}}
  - {{choice2}} ||| {% if label != -1 %}{{ answer_choices[label]
  }}{%endif%}"
- **COPA**: "{{ premise }} {% if question == "cause" %} This
  happened because... {% else %} As a consequence... {% endif %}
  What about this scenario?
  - {{choice1}}
  - {{choice2}} ||| {% if label != -1 %} {{ answer_choices[label]
  }}{%endif%}"

GPS Results

- **HellaSwag**: "Complete the description with an appropriate ending:
  First, {{ ctx_a.lower() }} Then, {{ ctx_b.lower() }} ...
  (a) {{ answer_choices[0] }}
  (b) {{ answer_choices[1] }} (c) {{ answer_choices[2] }} (d) {{ answer_choices[3] }}
  |||
  {{ answer_choices[label | int()] }}"
- **HellaSwag**: "the question ends with a phrase {{ctx}}
  (a) {{answer_choices[0]}}
  (b) {{answer_choices[1]}}
  (c) {{answer_choices[2]}}
  (d) {{answer_choices[3]}}
  Hint: the topic of the sentence is {{activity_label}}
  |||
  {{answer_choices [label | int()]}}"

GPS Results

- **WiC**: "Does the word "{{word}}" have the same meaning in these two sentences?
  Yes, No?
  {{sentence1}}
  {{sentence2}}
  ||| {% if label != -1%}
  {{answer_choices[label]}}
  {% endif %}"
- **WiC**: "Where is the word '{{word}}'
  {{sentence1}}
  {{sentence2}}
  ||| {% if $\backslash\backslash label! = -1\%$}

{{answer_choices[label]}}
{% endif %}"

Demo

References I

[1] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds., Association for Computational Linguistics, 2019, pp. 4171–4186. [Online]. Available: https://doi.org/10.18653/v1/n19-1423.

[2] C. Raffel, N. Shazeer, A. Roberts, *et al.*, *Exploring the limits of transfer learning with a unified text-to-text transformer*, 2023. arXiv: 1910.10683 [cs.LG]. [Online]. Available: https://arxiv.org/abs/1910.10683.

References II

[3] T. B. Brown, B. Mann, N. Ryder, *et al.*, *Language models are few-shot learners*, 2020. arXiv: 2005.14165 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2005.14165.

[4] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, *Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing*, 2021. arXiv: 2107.13586 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2107.13586.

[5] N. Houlsby, A. Giurgiu, S. Jastrzebski, *et al.*, "Parameter-efficient transfer learning for NLP," in *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, K. Chaudhuri and R. Salakhutdinov, Eds., ser. Proceedings of Machine Learning Research, vol. 97, PMLR, 2019, pp. 2790–2799. [Online]. Available: http://proceedings.mlr.press/v97/houlsby19a.html.

References III

[6] E. B. Zaken, Y. Goldberg, and S. Ravfogel, "Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, S. Muresan, P. Nakov, and A. Villavicencio, Eds., Association for Computational Linguistics, 2022, pp. 1–9. [Online]. Available: https://doi.org/10.18653/v1/2022.acl-short.1.

[7] E. J. Hu, Y. Shen, P. Wallis, *et al.*, "Lora: Low-rank adaptation of large language models," in *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, OpenReview.net, 2022. [Online]. Available: https://openreview.net/forum?id=nZeVKeeFYf9.

References IV

[8] L. Yu, Q. Chen, J. Lin, and L. He, "Black-box prompt tuning for vision-language model as a service," in *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China*, ijcai.org, 2023, pp. 1686–1694. [Online]. Available: https://doi.org/10.24963/ijcai.2023/187.

[9] A. Prasad, P. Hase, X. Zhou, and M. Bansal, "Grips: Gradient-free, edit-based instruction search for prompting large language models," in *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2023, Dubrovnik, Croatia, May 2-6, 2023*, A. Vlachos and I. Augenstein, Eds., Association for Computational Linguistics, 2023, pp. 3827–3846. [Online]. Available: https://doi.org/10.18653/v1/2023.eacl-main.277.