Evaluating the Effect of Prompting on a Language Model's Word Understanding Using Cosine Similarity

Sulbha Malviya
Department of Computer Science
Boise State University
sulbhamalviya@u.boisestate.edu

I. ABSTRACT

This project investigates whether prompting a language model with a definitional cue enhances its semantic understanding of words. We used a lightweight generative language model, SmolLM2-135M, to generate definitions for a curated list of 200 English words. First, baseline performance was measured by comparing the model's unprompted definitions to standardized WordNet definitions using cosine similarity. Then, we fine-tuned the same model using Reinforcement Learning with Human Feedback (RLHF), where the reward signal was derived from cosine similarity between model-generated and WordNet definitions. The Proximal Policy Optimization (PPO) algorithm was employed to perform stable and controlled updates during fine-tuning. Post-finetuning, the model was re-evaluated on the same word list, and the results demonstrated a notable improvement in semantic alignment, with 65% of words showing increased similarity scores. These findings suggest that RLHF fine-tuning, guided by definition-based prompting and cosine similarity as a proxy reward, effectively enhances a language model's ability to generate accurate word meanings.

II. INTRODUCTION

Large Language Models (LLMs) such as GPT, BERT, and T5 have revolutionized natural language processing (NLP) by achieving state-of-the-art results across a wide range of tasks, including machine translation, summarization, and question answering. These models are trained on massive corpora and leverage billions of parameters to produce fluent, contextually appropriate responses. Despite these strengths, a persistent limitation is their tendency to produce outputs that are syntactically sound but semantically shallow or even factually incorrect. This challenge becomes particularly evident in tasks that require a deep understanding of word meanings, such as generating dictionary-style definitions.

LLMs often rely on statistical correlations rather than grounded knowledge, which can result in vague, generic, or misleading definitions when prompted with queries like "Define alloy." While prompting techniques offer a practical way to steer the model's behavior without retraining, prompting alone does not guarantee semantic accuracy or alignment with authoritative references like WordNet. This shortfall underscores the need for mechanisms that can refine the model's internal representations and outputs based on task-specific objectives.

To address this gap, our project explores the integration of **Reinforcement Learning with Human Feedback (RLHF)**. RLHF represents a paradigm shift in model fine-tuning: rather than using a fixed set of labeled examples, it optimizes a model based on reward signals derived from human preferences or proxy measures of correctness. In our case, we simulate human feedback using *cosine similarity* between the model-generated definitions and WordNet definitions, treating this similarity score as a scalar reward. This proxy allows us to align the model's outputs with a high-quality lexical standard in a quantifiable manner.

We implement this fine-tuning using **Proximal Policy Optimization (PPO)**, a stable and widely used reinforcement learning algorithm. PPO ensures that updates to the model are not too drastic by penalizing large deviations from the original policy. This balance is crucial when fine-tuning LLMs, as it prevents the model from forgetting prior knowledge while optimizing for new objectives.

The primary objective of this study is to evaluate whether prompting, combined with RLHF-based fine-tuning, can improve the semantic alignment of a small language model (SmolLM2-135M) with expert definitions. By generating definitions for a set of 200 English words and comparing them—before and after fine-tuning—with their WordNet equivalents, we measure the improvement in semantic understanding using cosine similarity. This work not only demonstrates a lightweight and effective approach for improving model performance but also provides insights into how reinforcement learning can be used to enhance LLM interpretability and reliability in definition generation tasks.

III. Related Work

The development of Large Language Models (LLMs) has sparked significant research into how these models can be better aligned with human intent and semantic understanding. These models, while capable of generating coherent and contextually relevant text, often lack a grounded understanding of user expectations or domain-specific nuances. This has led to a growing body of work focused on improving controllability and task alignment through external techniques.

One prominent line of work involves the use of prompting techniques, which guide the model's output using explicit textual instructions. Prompting effectively acts as a soft programming interface for language models, allowing users to influence behavior without modifying the underlying architecture or weights. By carefully crafting prompts—ranging from simple imperative forms like "Translate this sentence" to more structured task descriptions—researchers and practitioners have been able to steer model outputs with impressive precision.

Prompting has proven particularly effective in zero-shot and few-shot learning scenarios, where limited or no labeled training data is available. It allows LLMs to generalize to new tasks based on natural language cues, often achieving surprisingly strong results. A comprehensive survey by [1] categorizes prompting strategies and their applications across various NLP tasks, highlighting how prompt design can impact performance and output quality.

In addition to basic prompting, more advanced formulations have been proposed. One such strategy is the Socratic prompting method, which structures the interaction between the user and the model as a sequence of questions and guided reflections. This method aims to elicit more accurate and interpretable outputs by encouraging the model to reason or self-correct through iterative dialogue rather than respond with a single-shot answer [2].

While prompting offers a lightweight way to steer LLMs, it does not always guarantee semantic alignment, particularly for tasks like generating accurate word definitions. To address this, recent advances incorporate Reinforcement Learning with Human Feedback (RLHF). A landmark study by [3] demonstrated how fine-tuning with human feedback improves instruction-following capabilities. In this paradigm, the model is updated based on reward signals rather than hard labels. RLHF has since been applied in large-scale models such as ChatGPT and OpenAssistant, highlighting its scalability and effectiveness in real-world systems [4].

Despite its success, RLHF presents several practical challenges, including the collection of high-quality human preference data, the risk of reward hacking, and distributional drift between policy outputs and training data. Recent surveys, such as [5], provide an overview of these limitations and suggest improvements such as using preference ranking, critique-based feedback, and simulated annotators to reduce reliance on human annotation.

Our project builds on these ideas by using a proxy reward signal—cosine similarity between model-generated definitions and WordNet definitions—to guide learning. The RLHF implementation in our case uses Proximal Policy Optimization (PPO), a reinforcement learning algorithm explored in detail by [6] for its stability and performance in LLM fine-tuning. PPO remains the most commonly used algorithm in RLHF pipelines due to its simplicity and ability to constrain updates using KL-penalty regularization, making it suitable for fine-tuning models without catastrophic forgetting [7].

WordNet itself continues to be a critical benchmark for evaluating semantic fidelity. Recent work such as [8] investigates how well modern LLMs align with curated dictionary examples, including those from WordNet. Other efforts [9] have examined how LLMs like LLaMA capture lexical-semantic knowledge when tested against WordNet-based tasks, indicating the potential and limitations of pretrained models in understanding word-level semantics.

Further advancements in reward modeling have been proposed to enhance the alignment of LLMs with human preferences. For instance, [10] introduced Reward Learning on Policy (RLP), an unsupervised framework that refines reward models using policy samples to maintain alignment with the evolving data distribution during policy optimization. Other approaches such as Critic-RM [**?**] and Agentic Reward Modeling [**?**] attempt to create more robust, generalizable reward systems by integrating critiques and verifiable correctness signals.

In the realm of fine-tuning pre-trained language models, parameter-efficient techniques have gained traction. [11] proposed Dynamic Parameter Selection (DPS), which adaptively selects subnetworks for fine-tuning, thereby reducing overfitting and improving generalization in low-resource scenarios. Additionally, methods like Low-Rank Adaptation (LoRA) have been employed to fine-tune models with a minimal number of trainable parameters, achieving performance comparable to full fine-tuning [12].

These studies collectively inform our methodology, which integrates prompting, RLHF with cosine similarity-based rewards, and efficient fine-tuning strategies to enhance the semantic accuracy of word definitions generated by LLMs.

IV. Methodology

The goal of this project is to evaluate whether prompting a language model and applying Reinforcement Learning with Human Feedback (RLHF) can improve its semantic understanding of word definitions. The process consists of three main stages: (1) generating baseline definitions using a pretrained model, (2) fine-tuning the model using RLHF based on a similarity-based reward function, and (3) comparing results before and after fine-tuning using cosine similarity.

A. Dataset

The input dataset consists of a curated list of 200 English words selected to represent a range of parts of speech and semantic categories. These words served as the prompts for the language model. For each word, a corresponding ground-truth definition was retrieved from the WordNet lexical database. WordNet definitions were used as the reference for evaluating the semantic accuracy of model-generated definitions.

B. Baseline Definition Generation

To establish a reference point for evaluation, we used the pretrained SmolLM2-135M model as our baseline. This model is a lightweight autoregressive language model with 135 million parameters, designed to perform a variety of generative tasks with reduced computational overhead [13]. We selected this model due to its simplicity, ease of deployment, and relevance for small-scale experimentation, especially in low-resource settings where larger models like GPT-3 may be impractical.

Each of the 200 target words in our dataset was fed into the model using a simple prompt structure: "Define *[word]*". The model, operating in a zero-shot manner, generated one definition per word without access to any external dictionary or prior examples. No additional fine-tuning or instruction conditioning was applied at this stage, ensuring that the model's outputs reflected its pretrained knowledge.

To assess the semantic quality of the generated definitions, we computed cosine similarity between the sentence embedding of each model-generated definition and the corresponding WordNet definition. Sentence embeddings were obtained using a pretrained encoder model (e.g., `all-MiniLM-L6-v2`) from the SentenceTransformers library, which transforms variable-length text into fixed-size dense vectors. The cosine similarity score, ranging from -1 to 1, provided a quantifiable measure of alignment between the baseline model's output and the authoritative reference.

This evaluation allowed us to benchmark the model's raw performance prior to any reinforcement learning or reward-based fine-tuning, serving as a control for later comparison with improved versions of the model.

C. Reinforcement Learning with Human Feedback (RLHF)

Reinforcement Learning with Human Feedback (RLHF) is an increasingly popular paradigm for fine-tuning large language models. Unlike traditional supervised learning, where models are trained on fixed input-output pairs, RLHF optimizes the model's behavior based on feedback signals that reflect human preferences or task-specific quality. The goal is to better align the model's outputs with user intent, especially in cases where supervised labels are scarce, ambiguous, or subjective.

In practice, RLHF is implemented as a reinforcement learning problem where a reward model evaluates the quality of the model's generated outputs. These reward models are often trained on pairwise preference data obtained from human annotators, who rank model responses based on helpfulness, accuracy, or alignment with task instructions. The trained reward model then serves as a proxy evaluator during policy optimization. This approach has been successfully applied in large-scale systems like InstructGPT and ChatGPT, where it was shown to significantly improve the usefulness and controllability of model outputs [3], [4].

In our setup, we simulate human feedback using *cosine similarity* between the model-generated definition and the corresponding WordNet reference definition. This scalar similarity score serves as a reward signal, ranging from -1 to 1. Higher similarity indicates stronger semantic alignment with the ground-truth definition, and the model is optimized to maximize this score. By treating cosine similarity as a continuous reward function, we enable the model to receive fine-grained feedback on its performance rather than a binary pass/fail signal.

While RLHF holds considerable promise, it also introduces challenges such as reward hacking, high annotation costs, and the risk of overfitting to poorly generalized feedback models. However, its flexibility and ability to scale to complex tasks make it a compelling approach for refining LLM behavior in real-world applications.

D. Proximal Policy Optimization (PPO)

To perform reinforcement learning within the RLHF framework, we use Proximal Policy Optimization (PPO) [14], a policy-gradient method widely adopted for its stability and ease of implementation. PPO updates the model (policy) in a way that ensures consistent learning while avoiding overly aggressive policy shifts. This is particularly important in language models, where abrupt updates can lead to performance collapse, loss of fluency, or forgetting of previously learned behaviors.

PPO addresses this by introducing a clipped surrogate objective function, which constrains the new policy from straying too far from the old one. It also incorporates a Kullback–Leibler (KL) divergence penalty term that further regulates how much the policy is allowed to change during training. These mechanisms ensure that the model's behavior evolves gradually in response to reward signals, making the training process more reliable across diverse tasks.

In the context of LLM fine-tuning, PPO has emerged as the de facto standard algorithm used in RLHF pipelines due to its simplicity, generalization ability, and compatibility with gradient-based optimization. Its ability to work with continuous reward signals—such as our cosine similarity-based feedback—makes it especially suitable for tasks involving semantic alignment rather than discrete correctness. As a result, PPO has been used in numerous LLM research efforts, ranging from reward learning for summarization to alignment training in conversational agents.

Overall, PPO provides the stability and control necessary to apply RLHF effectively, particularly in low-data or low-resource setups like our fine-tuning of the SmolLM2-135M model on a definition generation task.

E. RLHF Workflow

As illustrated in Figure 1, our methodology follows a structured pipeline that begins with baseline definition generation and proceeds through reward-based fine-tuning using the RLHF framework.

We used the same pretrained SmolLM2-135M model both as the baseline and as the starting point for RLHF fine-tuning. In the baseline phase, the model was evaluated without any additional training, generating definitions in response to prompts in the format “Define *[word]*”. These outputs were compared to WordNet definitions to compute initial cosine similarity scores, serving as a reference point for evaluation.

In the fine-tuning phase, we applied an RLHF layer to the same SmolLM2-135M model. The RLHF process was driven by a reward model based on semantic similarity: cosine similarity between sentence embeddings of model-generated and WordNet definitions. This reward signal guided the model updates via the PPO algorithm, encouraging it to generate outputs that were more semantically aligned with the target definitions.

The full RLHF training loop in our setup can be summarized in the following steps:

1) The model generates a textual definition in response to a given word prompt.
2) Sentence embeddings for both the generated and the corresponding WordNet definitions are computed.
3) Cosine similarity between the two embeddings is calculated and used as a scalar reward signal.
4) The PPO algorithm updates the policy (i.e., the model parameters) based on the reward, promoting outputs with higher semantic similarity.

This iterative process continues across multiple epochs, gradually optimizing the model toward producing WordNet-aligned definitions with improved semantic accuracy.

F. Post-Fine-Tuning Evaluation

Following the RLHF fine-tuning process, we evaluated the updated model on the same set of 200 input words used during baseline testing. For each word, the model was prompted to generate a definition using the same input format. These outputs were then compared to their corresponding WordNet definitions using cosine similarity between sentence embeddings, identical to the evaluation method used in the baseline phase.

To assess the impact of fine-tuning, we computed the average cosine similarity across all 200 words for both the baseline and fine-tuned models. The difference between these two averages was recorded as the *similarity gain*, serving as a quantitative measure of improvement. Additionally, we calculated the proportion of words for which the fine-tuned model achieved a higher similarity score compared to the baseline, providing insight into the consistency of performance improvement across the dataset.

This post-fine-tuning evaluation enabled a direct comparison between the model’s initial and improved states, allowing us to validate the effectiveness of RLHF in enhancing the semantic quality of model-generated definitions.

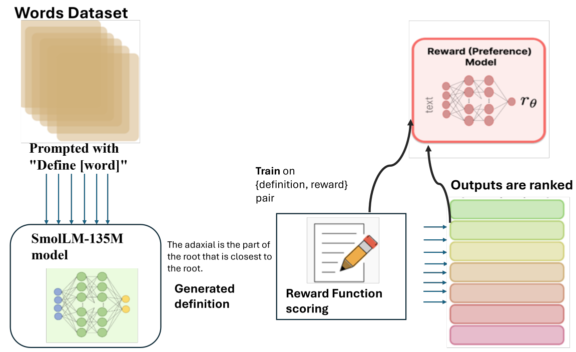

Fig. 1. Project pipeline: prompting, reward modeling, and fine-tuning with PPO.

V. Evaluation and Results

To evaluate the effectiveness of prompting and RLHF-based fine-tuning, we compared the semantic similarity between model-generated definitions and reference definitions from WordNet. The primary metric used was **cosine similarity** between sentence embeddings.

A. Evaluation Metric

We used cosine similarity to quantify the semantic alignment between the generated definition and the WordNet definition for each word. Sentence embeddings were obtained using a pretrained sentence transformer model (e.g., `all-MiniLM-L6-v2`). The cosine similarity score ranges from -1 to 1, with higher values indicating stronger semantic similarity.

B. Baseline vs. WordNet Similarity

For each of the 200 input words, we prompted the baseline SmolLM2-135M model with “Define *[word]*” and recorded its definition. The average cosine similarity between the model-generated definitions and WordNet definitions was:

- **Baseline mean similarity:** 0.1650

This score served as our reference to compare the effectiveness of fine-tuning under different reward schemes.

C. Comparison of Reward Strategies

We experimented with two reward modeling strategies to guide the reinforcement learning process during fine-tuning.

The first approach was a binary reward scheme, where the model was rewarded with a value of 1.0 if the cosine similarity between the generated definition and the corresponding WordNet reference exceeded a predefined threshold of 0.85; otherwise, it received a reward of 0. This strategy introduced a sharp decision boundary, converting a continuous semantic alignment score into a binary signal. While simple to implement, this method had the drawback of providing sparse feedback—many outputs yielded a zero reward, offering little gradient information for effective policy updates. Consequently, the model struggled to learn meaningful improvements from most examples, leading to modest performance gains. The post-fine-tuning evaluation showed only 50.82% of the words exhibited improved similarity to WordNet definitions under this reward formulation.

To address the limitations of binary feedback, we adopted a soft reward strategy. In this scheme, the cosine similarity score itself was used directly as the reward value. This approach preserved the granularity of semantic similarity and provided smooth, continuous feedback to the reinforcement learning algorithm. Each definition generated by the model received a reward proportional to how semantically aligned it was with the WordNet definition, thereby enabling the model to fine-tune its responses even when the alignment was only partial. This richer signal allowed more consistent and informative updates during training, resulting in a significantly higher improvement rate—65.03% of the words demonstrated better alignment post-fine-tuning. Table I summarizes the results for both strategies.

This result shows that a soft reward formulation provides richer guidance during training, leading to a model that is better aligned with WordNet definitions.

TABLE I
COSINE SIMILARITY RESULTS FOR BASELINE AND FINE-TUNED MODELS USING TWO DIFFERENT REWARD STRATEGIES.

Reward Strategy	**Results**	**Words Improved**
Binary Reward (threshold = 0.85)		
Baseline vs WordNet (avg. cosine)	0.1650	–
Fine-tuned vs WordNet (avg. cosine)	0.1695	50.82%
Similarity Gain	0.0045	–
Soft Reward (cosine score)		
Baseline vs WordNet (avg. cosine)	0.1650	–
Fine-tuned vs WordNet (avg. cosine)	0.2446	65.03%
Similarity Gain	0.0796	–

D. Scatter Plot Visualization

Figure 2 illustrates a pointwise comparison of cosine similarity scores between the baseline and fine-tuned models under the soft reward setting. Each point represents one word from the evaluation set, where the x-axis denotes the similarity score achieved by the baseline model, and the y-axis represents the similarity score after RLHF fine-tuning.

The dashed diagonal line indicates the "no-change" threshold—points lying on this line correspond to words for which the similarity did not change between the two models. Notably, a majority of points are located *above* the diagonal, indicating that the fine-tuned model produced definitions that are more semantically aligned with WordNet compared to those from the baseline model. This validates the effectiveness of the soft reward signal in guiding the model toward improved outputs.

The plot also reveals that several words with initially low similarity (e.g., below 0.2 on the x-axis) experienced significant improvement after fine-tuning, as seen by their upward shift in the y-axis. This suggests that the fine-tuned model was not only refining already good predictions, but also learning to correct poorly aligned definitions from the baseline.

While most points follow this positive trend, a small number of outliers fall below the diagonal—indicating instances where the model's definition quality slightly deteriorated after training. However, these cases are relatively few, and their deviations are minor compared to the broad upward trend observed across the dataset.

Overall, this plot provides strong visual evidence that RLHF with soft reward modeling substantially improved the model's semantic understanding, particularly in the low-to-mid similarity range.

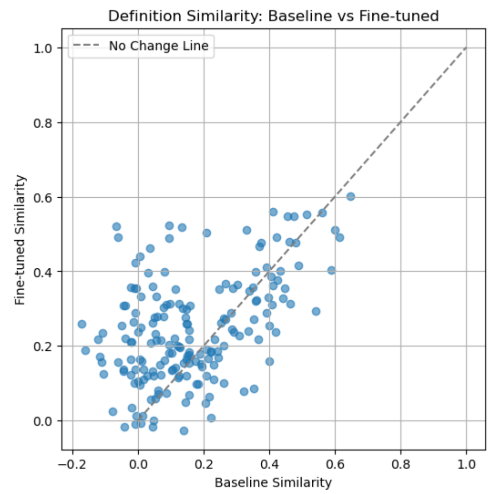

Fig. 2. Cosine similarity scores: baseline vs. fine-tuned (soft reward).

VI. LIMITATIONS AND FUTURE WORK

While the fine-tuned model demonstrated improved semantic alignment with WordNet definitions, several limitations remain that open avenues for future improvement.

One notable limitation is the model's inability to handle words with multiple valid definitions. Many English words are polysemous and can have several distinct meanings depending on context. However, our current prompting approach ("Define [word]") provides no contextual information, and the model is forced to generate a single definition. This often results in generic or dominant-meaning outputs that may not align well with WordNet's more specific senses.

To address this, future work could incorporate context-aware prompting—either by supplying example usage sentences or by explicitly requesting definitions for a particular sense (e.g., "Define the verb form of 'run' in a sports context"). Alternatively, incorporating a word sense disambiguation (WSD) module upstream could help the model generate more targeted definitions.

Another area for improvement involves reward modeling. While cosine similarity served as a useful proxy for semantic alignment, it may not capture finer nuances of meaning or stylistic correctness. Future efforts could explore hybrid reward models that incorporate syntactic fluency, human preference rankings, or example-based scoring.

Finally, training for more epochs or using larger reward datasets could potentially improve generalization. Additionally, evaluating the model on external dictionaries or humanannotated gold standards could provide a more rigorous assessment of semantic quality.

VII. Conclusion

In this project, we investigated whether prompting a language model and fine-tuning it using Reinforcement Learning with Human Feedback (RLHF) could improve its ability to generate semantically accurate word definitions. Starting with a baseline model (SmolLM2-135M), we compared its generated definitions to those in WordNet using cosine similarity as an evaluation metric.

To guide fine-tuning, we experimented with two reward strategies: a binary reward based on a similarity threshold, and a soft reward based on continuous cosine scores. Our results showed that the soft reward approach significantly outperformed the binary method, leading to greater average similarity gains and improvement across a larger portion of the dataset.

The scatter plot and statistical analysis confirm that the fine-tuned model produced outputs more closely aligned with WordNet definitions, especially for words with initially low baseline performance. These findings demonstrate that RLHF with a well-designed reward function can effectively enhance the semantic understanding of language models, even at a small scale.

This work serves as a foundation for future exploration into context-aware prompting, richer reward modeling, and broader evaluation metrics to further improve language model interpretability and alignment.

References

[1] K. Zhou, Y. Yang, Y. Huang, Y. Zhao, C. Qian, and C. C. Loy, "A systematic survey of prompt engineering in large language models: Techniques and applications," *arXiv preprint arXiv:2402.07927*, 2024. [Online]. Available: https://arxiv.org/abs/2402.07927

[2] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. H. Chi, Q. V. Le, and D. Zhou, "Prompting large language models with the socratic method," in *International Conference on Learning Representations (ICLR)*, 2022. [Online]. Available: https://arxiv.org/abs/2212.03827

[3] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray *et al.*, "Training language models to follow instructions with human feedback," *arXiv preprint arXiv:2203.02155*, 2022. [Online]. Available: https://arxiv.org/abs/2203.02155

[4] OpenAI, "Chatgpt: Optimizing language models for dialogue," 2023, https://openai.com/blog/chatgpt.

[5] D. M. Ziegler *et al.*, "A survey of reinforcement learning from human feedback," *arXiv preprint arXiv:2312.14925*, 2023.

[6] W. Xu, J. Li, J. Wang, X. Sun, and J. Wu, "Secrets of rlhf in large language models part i: Ppo," *arXiv preprint arXiv:2307.04964*, 2023. [Online]. Available: https://arxiv.org/abs/2307.04964

[7] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," in *arXiv preprint arXiv:1707.06347*, 2017.

[8] L. Li, A. Marasovic, A. Shevchenko, W. Xu, and G. Glavaš, "Dictionary examples in the era of large language models," in *Proceedings of the 2024 Conference on Language Resources and Evaluation (LREC)*, 2024. [Online]. Available: https://aclanthology.org/2024.lrec-main.1538/

[9] T. Chakrabarty, T. Mihaylov, E. Durmus, and D. Radev, "Wordnet-based model for solving multiple lexical semantic tasks," *arXiv preprint arXiv:2403.09207*, 2024. [Online]. Available: https://arxiv.org/abs/2403.09207

[10] H. Lang, F. Huang, and Y. Li, "Fine-tuning language models with reward learning on policy," in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2024. [Online]. Available: https://aclanthology.org/2024.naacl-long.75/

[11] H. Zhang, G. Li, J. Li, Z. Zhang, Y. Zhu, and Z. Jin, "Fine-tuning pre-trained language models effectively by optimizing subnetworks adaptively," *arXiv preprint arXiv:2211.01642*, 2022. [Online]. Available: https://arxiv.org/abs/2211.01642

[12] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," in *International Conference on Learning Representations (ICLR)*, 2022. [Online]. Available: https://arxiv.org/abs/2106.09685

[13] S. R. Team, "Smollm2-135m: Lightweight language model for generative tasks," 2023, available at: https://github.com/smol-ai.

[14] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017. [Online]. Available: https://arxiv.org/abs/1707.06347