Soft Thinking: Enhancing Knowledge Base Completion through Trainable Prompts in the Chain of Thought

Aldan Creo[1,*,†], Christophe Guéret[2], Alberto Bernardi[2], Adrianna Janik[2] and Luca Costabello[2]

[1] *Independent Author, Dublin, Ireland*
[2] *Accenture Labs, 7 Hanover Quay, Dublin, Ireland*

Abstract

We introduce Soft Thinking, a parameter-efficient approach for knowledge base completion that inserts trainable soft prompts within the chain-of-thought reasoning space of language models. Unlike traditional soft prompting methods that prepend tokens to inputs, our approach embeds relation-specific trainable parameters between `<think>` and `</think>` tokens, allowing models to develop specialized reasoning pathways for different relation types while keeping base model parameters frozen. We evaluated our method using a dataset of six different relations, achieving a macro F1-score of 0.3977, over two times higher than that of baseline prompting (0.186). Our approach particularly excels at entity-based relations, substantially improving geographic and organizational query results while maintaining parameter efficiency (less than 0.001% of total model parameters). Notably, we achieve optimal performance using only subject entities as input, demonstrating that soft prompts can autonomously develop effective reasoning strategies without manual prompt engineering. Our code is available at https://github.com/ACMCMC/soft-thinking.

1. Introduction

Intuition.

Several recent works have shown that LLMs can be prompted to think in a chain-of-thought manner, generating intermediate reasoning steps before arriving at the final answer.

Those "thoughts" are expressed in words. **But what if thoughts could go beyond words?**

Like a painter who draws a picture without needing to write down the steps of how to do it, can we help the model express its reasoning in a more abstract way?

We do that by inserting trainable embeddings in the model's reasoning process, allowing it to **learn relation-specific reasoning pathways** without needing to express them in words.

(We formalize this intuition in the next sections.)

Large Language Models (LLMs) have demonstrated remarkable capabilities in knowledge-intensive tasks [1], but they often struggle with complex reasoning and knowledge base completion tasks that require precise relation understanding. While prompt engineering techniques like Chain of Thought (CoT) [2] have improved reasoning capabilities, they typically rely on fixed prompt templates that are not optimized for specific relation types.

Recent advances in parameter-efficient fine-tuning methods, such as prompt tuning [3] and prefix tuning [4], have shown promising results in adapting language models to downstream tasks without

Joint proceedings of KBC-LM and LM-KBC @ ISWC 2025

[*] Corresponding author.

[†] Work done while at Accenture Labs.

✉ research@acmc.fyi (A. Creo); christophe.gueret@accenture.com (C. Guéret); alberto.bernardi@accenture.com (A. Bernardi); adrianna.janik@accenture.com (A. Janik); luca.costabello@accenture.com (L. Costabello)
🌐 https://acmc.fyi (A. Creo); https://adrijanik.github.io (A. Janik)
iD 0000-0002-7401-5198 (A. Creo); 0000-0002-8914-6107 (C. Guéret); 0000-0003-1573-3234 (A. Bernardi); 0000-0002-4387-4563 (A. Janik); 0000-0002-0720-9347 (L. Costabello)

modifying all model parameters. Liu et al. [5] showed that prompt tuning can be as effective as fine-tuning the model, while only training a fraction of the weights, typically 0.1%-3% of parameters; Qin and Eisner [6] utilized mixtures of soft prompts for question answering on LMs.

However, these approaches typically insert trainable parameters at the input level or across all layers, without a specific focus on enhancing the model's reasoning process. Only very recently has soft prompting started to be explored: Xu et al. [7] propose utilizing an assistant language model trained to inform CoT reasoning processes. Similarly, COCONUT (Chain of Continuous Thought) [8] directly feeds the last layer's hidden states into the next time step, thus creating a continuous flow of information that does need to be mapped back into a discrete representation in the form of tokens.

Nonetheless, our work differs in that both approaches aim to generate intermediate thoughts that are only expressed in the latent space without any explicit tokenization, while we aim to find a set of soft input embeddings that are not generated on the fly, but rather trained to optimize the model's reasoning process. We opt for such a design choice to improve scalability and efficiency, as generating soft thoughts on the fly would require substantial additional computational resources. This is further motivated by the fact that predicting the object entity for a given subject, where the relation does not change, can be expected to follow similar reasoning patterns (in contrast to, e.g., solving mathematical problems, where there is a greater degree of variability). Therefore, a single chain of soft thoughts can be trained to cover all such cases while avoiding the additional cost of inference-time generation.

In this paper, we introduce **Soft Thinking**, a novel approach that combines the strengths of CoT prompting with trainable soft prompts specifically inserted in the "thinking" section of the assistant response. Our approach creates relation-specific reasoning pathways that guide the model's thought process while maintaining the model's parameters frozen. We implement this by:

1. Adding soft chain-of-thought tokens that create a dedicated reasoning space between the question and answer generation
2. Inserting trainable soft prompt embeddings in this thinking section
3. Training these embeddings for each relation type in a knowledge base completion task

Our contributions are as follows:

- A novel framework for knowledge base completion using trainable "soft thinking" prompts in the model's reasoning space
- An automatic optimization approach that eliminates manual prompt engineering by maximizing answer likelihood, removing human bias and variability
- An efficient implementation requiring minimal trainable parameters (less than 0.001% of total model parameters)
- Empirical validation achieving third place in the LM-KBC Challenge @ ISWC 2025, with an average macro F1-score of 0.398 across six relations of different nature
- Insights into relation-specific reasoning pathways, showing particular effectiveness on entity-based relations

2. Methods

2.1. Task definition

Formally, we can define the sets of all possible entities $\mathcal{E}$ and relations $\mathcal{R}$. The cartesian product $\mathcal{E} \times \mathcal{R} \times \mathcal{E}$ defines the space $\mathcal{A}$ of all possible assertions. Given a boolean function $c(x)$ that is true if and only if x is a "true fact", a Knowledge Base $\mathcal{KB}$ satisfies the condition $\forall t \in \mathcal{KB} \subseteq \mathcal{A} : c(t)$, i.e., the Knowledge Base is a subset of all possible assertions such that each assertion is a true fact.

Our inputs are a set of tuples $i = (s, p) \in \mathcal{E} \times \mathcal{R}$, and we aim to find a Knowledge Base $\mathcal{KB}$ such that for each input tuple i, we can find a set of entities $o \in \mathcal{E}$ such that the assertion $t = (s, p, o)$ is true, i.e., $c(t) = \text{True}$.

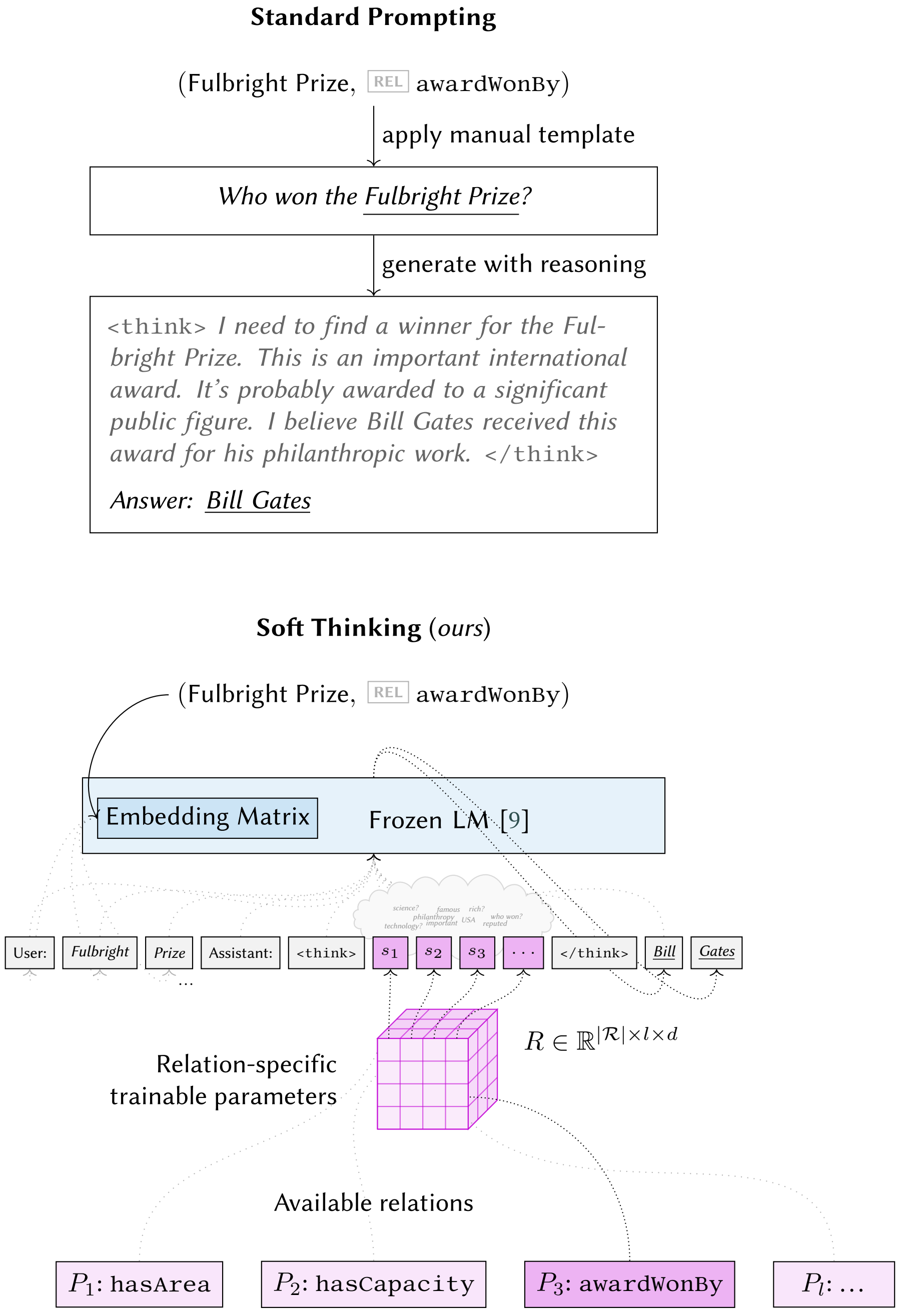
Standard Prompting
(Fulbright Prize, REL awardWonBy)
apply manual template
Who won the Fulbright Prize?
generate with reasoning
<think> I need to find a winner for the Fulbright Prize. This is an important international award. It's probably awarded to a significant public figure. I believe Bill Gates received this award for his philanthropic work. </think>
Answer: Bill Gates
Soft Thinking (ours)
(Fulbright Prize, REL awardWonBy)
Embedding Matrix
Frozen LM [9]
science?
famous
rich?
philanthropy
USA
who won?
technology?
important
reputed
User:
Fulbright
Prize
...
Assistant:
<think>
s_1
s_2
s_3
...
</think>
Bill
Gates
Relation-specific trainable parameters
$R \in \mathbb{R}^{|\mathcal{R}| \times l \times d}$
Available relations
P_1: hasArea
P_2: hasCapacity
P_3: awardWonBy
P_l: ...

Figure 1: Soft Thinking Approach for Knowledge Base Completion. Standard prompting (top) uses explicit, token-based chain-of-thought reasoning with readable text. In contrast, our Soft Thinking approach (bottom) introduces trainable soft prompts in a dedicated <think> section. For each relation $r \in \mathcal{R}$, we optimize l embeddings of size d (the model's hidden size), forming a matrix $R \in \mathbb{R}^{|\mathcal{R}| \times l \times d}$ that guide the model's reasoning process but can't be represented as readable tokens, creating relation-specific reasoning pathways while keeping the language model's parameters frozen.

For example, given the input tuple (Ireland, hasArea), we want to find all entities o such that the assertion $t = (\text{Ireland}, \text{hasArea}, o)$ is true. In this case, there is only one object o that satisfies this condition: 84,421. Therefore, we can represent the knowledge base as $KB = \{t\}$, where $t = (\text{Ireland}, \text{hasArea}, 84{,}421)$.

2.2. System Architecture

Our approach builds upon the `Qwen3-8B` language model [9], which we keep frozen to maintain its pre-trained knowledge while introducing trainable parameters specifically for knowledge base completion tasks. Figure 1 illustrates our architecture compared to standard prompting approaches.

The base architecture consists of the frozen `Qwen3-8B` model with its standard tokenization pipeline. We use the model's existing embedding matrix to convert input tokens to continuous representations, but we augment this process with relation-specific trainable embeddings inserted at strategic positions in the input sequence.

Unlike traditional soft prompting methods that prepend trainable tokens to the input [3], our approach inserts soft prompt embeddings within the chain-of-thought reasoning section. We structure prompts using special tokens: `<think>` to begin the reasoning phase, followed by relation-specific soft prompt embeddings, and `</think>` to conclude the reasoning before answer generation.

For each relation type r in our knowledge base, we maintain a trainable parameter matrix $P_r \in \mathbb{R}^{l \times d}$, where l is the number of soft prompt tokens (typically 5-10) and d is the model's embedding dimension (4096 for `Qwen3-8B`). These parameters are randomly initialized and optimized during training to develop relation-specific reasoning pathways.

Our prompt templates follow a simple structure: we start with the subject entity, add the `<think>` token, insert the relation-specific soft prompt embeddings, close with `</think>`, and allow the model to generate the answer. This creates a dedicated reasoning space where the soft prompts can guide the model's thought process without interfering with the input question or output answer.

During forward passes, we combine embeddings from three sources: (1) the frozen word embeddings for input tokens and special tokens, (2) the trainable soft prompt embeddings for the current relation, and (3) position-specific attention masks to ensure proper sequence modeling. The combined embedding sequence is then processed by the frozen language model to generate answers.

2.3. Training Methodology

Our training approach focuses exclusively on optimizing the relation-specific soft prompt parameters while maintaining the base language model in evaluation mode with frozen weights. This helps to keep the training computationally efficient and avoiding catastrophic forgetting of the pre-trained knowledge.

We organize training data by relation type and process each relation separately within each epoch. This allows the optimizer to focus on relation-specific patterns and develop specialized reasoning strategies for different types of knowledge queries. For each relation, we shuffle the training examples to prevent overfitting to example ordering.

We employ the Adam optimizer with a learning rate of 0.05 and weight decay of 0.01 for regularization. Gradient clipping with a threshold of 2.5 prevents exploding gradients, which can occur with soft prompt training. We also implement gradient accumulation to simulate larger effective batch sizes when GPU memory constraints limit the actual batch size.

Our training objective uses cross-entropy loss computed only on the answer tokens that follow the `</think>` token. This ensures the model learns to generate accurate answers based on the reasoning process guided by the soft prompts, rather than simply memorizing the training examples. We mask padding tokens and use the model's end-of-sequence token for proper termination.

During training, we evaluate on a fixed validation set of 20 examples per relation to ensure consistent performance tracking across epochs. We compute macro and micro F1 scores, precision, and recall metrics, and save the best-performing soft prompts for each relation based on relation-specific macro F1 scores.

We implement several memory optimization strategies including gradient checkpointing, careful tensor management, and periodic garbage collection. These techniques allow training on standard GPU hardware while maintaining training stability and speed.

2.4. Experimental Setup

We train our model for 300 epochs with a batch size of 10 and a prompt length of 10 trainable tokens per relation. While we observed diminishing returns after 5 tokens, we chose 10 to capture any remaining improvements, though results would likely be similar with 5 tokens. The model generates up to 15 new tokens during both training and inference. We set the random seed to 2262 for reproducibility and run validation after every epoch to monitor training progress.

Our approach uses simple subject-entity prompt templates and processes 477 training examples across 7 relation types. We position the soft prompts after the input prompt within the chain-of-thought reasoning section, creating dedicated reasoning pathways for each relation type without external knowledge access.

With respect to the relation types considered in our experimental setup, we focus on six relation types across diverse knowledge domains. While the shared task organizers provided detailed prompt templates for each relation, we chose to use only the subject entity without additional prompt information to allow the model more flexibility in reasoning.

The relation types are[1]:

- REL `awardWonBy`: Recipients of awards
- REL `companyTradesAtStockExchange`: Financial market listings
- REL `countryLandBordersCountry`: International border relationships
- REL `hasArea`: Geographic area information (e.g., countries, islands)
- REL `hasCapacity`: Capacity measurements (e.g., stadiums, venues)
- REL `personHasCityOfDeath`: Biographical information about death locations

3. Results

3.1. Competition Performance

Our Soft Thinking approach achieved third place in the LM-KBC Challenge at the 24th International Semantic Web Conference (ISWC) [10]. Table 1 shows the final competition rankings based on the average macro F1-score across all relations.

Table 1
LM-KBC Challenge @ ISWC 2025 Final Rankings

Rank	Team	Average Macro F1	Δ to Baseline
1	edarsem	0.444	+0.232
2	JingboHe	0.405	+0.193
3	**acmc (ours)**	**0.398**	**+0.186**
4	isam	0.241	+0.029
5	aclay	0.216	+0.004
6	lm-kbc	0.212	0.000
7	emilia-maria	0.106	−0.106
–	Baseline	0.212	–

The baseline row corresponds to directly prompting Qwen3-8B to generate a list of answers, without any optimization or relation-specific adaptation. Our approach achieved an average macro F1-score of 0.398, representing a 0.186 point improvement over the baseline and establishing our method as competitive in the knowledge base completion domain.

[1] We denote relation types as REL `relationName` throughout this paper.

3.2. Detailed Performance Analysis

Table 2 presents the detailed performance breakdown across all relation types in the challenge. Our method exhibits substantial variation in performance depending on the relation type, with particularly strong results for entity-based relations and more challenging performance on numeric relations.

Table 2
Detailed Performance Results by Relation Type

Relation	Prec.	Recall	Macro F1	Baseline F1	Δ to Baseline
REL `awardWonBy`	0.161	0.037	0.057	0.117	−0.060
REL `companyTradesAtStockExchange`	0.667	0.647	0.555	0.167	+0.388
REL `countryLandBordersCountry`	0.817	0.791	0.771	0.702	+0.069
REL `hasArea`	0.190	0.190	0.190	0.240	−0.050
REL `hasCapacity`	0.090	0.090	0.090	0.040	+0.050
REL `personHasCityOfDeath`	0.930	0.600	0.540	0.080	+0.460
All Relations	**0.512**	**0.432**	**0.398**	**0.212**	**+0.186**

Our approach demonstrates particularly strong performance on geographic and organizational relations, achieving F1-scores above 0.5 for REL `countryLandBordersCountry` (0.771), REL `companyTradesAtStockExchange` (0.555), and REL `personHasCityOfDeath` (0.540). These relations benefit from the structured reasoning pathways that our soft prompts develop during training.

Conversely, numeric relations (REL `hasArea`, REL `hasCapacity`) show substantially lower performance, with F1-scores of 0.190 and 0.090 respectively.

The precision-recall balance varies significantly across relations, with higher precision than recall for most entity-based relations (e.g., REL `personHasCityOfDeath`: 0.930 precision vs. 0.600 recall), suggesting our method tends toward conservative predictions that favor accuracy over completeness.

Compared to the baseline, our method also shows substantial improvements on several relation types, with the most notable gains on REL `personHasCityOfDeath` (+0.460) and REL `companyTradesAtStockExchange` (+0.388).

4. Discussion

Next, we analyze our experimental results and discuss the effectiveness of relation-specific soft prompts for knowledge base completion.

Our approach achieved competitive performance compared to the other two best teams. We secured third place with a macro F1-score of 0.398. Notably, our score is practically identical to second place (0.405), with only a 0.075-point difference; the top-performing methods are very close to each other, which points to the inherent difficulty of the task.

The dataset presents inherent challenges that affect all participating methods. We observed instances of incorrectly formatted subjects, questions with ambiguous or multiple valid answers, and entities that appear to be non-existent or poorly documented. These issues highlight the fundamental difficulty of the task: if human experts with internet access struggle to answer certain questions accurately, we cannot reasonably expect a language model without external knowledge access to perform better on such cases.

It is interesting to consider the role of general knowledge in the performance of our model. We got our best results in the case of relations that a human person with general knowledge, whose language abilities this general-purpose LM tries to model, would be able to predict or guess.

To illustrate this, consider a persona P with a general knowledge of the world and capable of making deductions even if they don't know the specific answer to a question. We expect P to be able to answer most of the REL `countryLandBordersCountry` questions, and to make "reasonable" guesses based on the textual representation of s. For instance, when asked about "Where did Maeve Binchy die?"

(($s =$ Maeve Binchy, $p =$ REL `personHasCityOfDeath`)), P might assume that such a person with a name of Irish origin has died in Dublin or Cork, both Irish cities, even if P has no prior knowledge of that person. There is a nontrivial probability that such guess would turn out to be correct, and indeed, our model achieves a high F1-score of 0.540 for this relation.

Conversely, in the context of relations that require specific numeric answers or names, the probability of guessing correctly is close to impossible, so the performance is severely impacted (e.g. REL `hasArea`, REL `hasCapacity`, REL `awardWonBy`). We believe that this shows promising results of our approach, as it seems to be able to mimic the "common sense" reasoning of a human person with general knowledge.

Another key idea to highlight is that we observed the best performance even after removing the original prompt templates provided by the challenge organizers and using only the subject entity as input. While we take more training steps to converge to the optimal solution, removing the prompt templates eliminates human bias and variability expressed in the specific wording and structure of the prompts. This allows the model to "explore" the reasoning space more freely and develop its own understanding of how to approach the task. We see this as one of the key advantages of our approach; we remove the frequent dilemmas prompt engineers face when trying to find the best prompt structure.

5. Conclusion

We introduced Soft Thinking, a novel approach for knowledge base completion that combines Chain of Thought prompting with trainable soft prompts inserted in the model's reasoning space. Our method automatically optimizes relation-specific reasoning pathways without manual prompt engineering, requiring less than 0.001% of total model parameters. We achieved third place in the LM-KBC Challenge @ ISWC 2025 with a macro F1-score of 0.398, representing a 0.186 point improvement over baseline and practically identical performance to second place.

Our analysis revealed that the approach particularly excels at entity-based relations, with substantial improvements over baseline methods. Notably, we achieved optimal performance using only subject entities as input, demonstrating that our soft prompts develop effective reasoning strategies autonomously while avoiding human bias. We believe Soft Thinking represents a promising direction for parameter-efficient knowledge base completion that balances performance, efficiency, and automation.

Limitations and Future Work

While our approach shows promising results, several limitations warrant acknowledgment. Performance varies significantly across relation types, with particularly poor results on numerical relations such as REL `hasArea` (F1-score: 0.190) and REL `hasCapacity` (F1-score: 0.090). This suggests that our current soft prompt configuration struggles to capture reasoning patterns for precise numerical queries requiring exact knowledge retrieval.

Future work could address these limitations through more sophisticated architectures for numerical reasoning or hybrid approaches combining soft thinking with external knowledge retrieval. Investigating the transferability of learned soft prompts across related relations could improve efficiency. Multi-task learning approaches may yield better performance and more generalizable reasoning patterns. We also see potential in exploring a mixture of a fixed set of soft prompts common across multiple relations that are combined with relation-specific ones.

Declaration on Generative AI

During the preparation of this work, the authors used Claude 4.0 Sonnet and GPT-4.1 in order to: Drafting content, Paraphrase and reword, Improve writing style, Abstract drafting, Grammar and spelling check, Formatting assistance. The authors reviewed and edited the content to ensure its quality and integrity. The use of these tools was limited to specific tasks, and the final work is a product of the authors' own efforts.

References

[1] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, volume 33, Curran Associates, Inc., 2020, pp. 1877–1901. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[2] J. Wei, X. Wang, D. Schuurmans, M. Bosma, brian ichter, F. Xia, E. H. Chi, Q. V. Le, D. Zhou, Chain of thought prompting elicits reasoning in large language models, in: A. H. Oh, A. Agarwal, D. Belgrave, K. Cho (Eds.), Advances in Neural Information Processing Systems, 2022. URL: https://openreview.net/forum?id=_VjQlMeSB_J.

[3] B. Lester, R. Al-Rfou, N. Constant, The power of scale for parameter-efficient prompt tuning, in: M.-F. Moens, X. Huang, L. Specia, S. W.-t. Yih (Eds.), Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 2021, pp. 3045–3059. URL: https://aclanthology.org/2021.emnlp-main.243/. doi:10.18653/v1/2021.emnlp-main.243.

[4] X. L. Li, P. Liang, Prefix-tuning: Optimizing continuous prompts for generation, in: C. Zong, F. Xia, W. Li, R. Navigli (Eds.), 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), Association for Computational Linguistics, Online, 2021, pp. 4582–4597. URL: https://aclanthology.org/2021.acl-long.353/. doi:10.18653/v1/2021.acl-long.353.

[5] X. Liu, K. Ji, Y. Fu, W. Tam, Z. Du, Z. Yang, J. Tang, P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks, in: S. Muresan, P. Nakov, A. Villavicencio (Eds.), Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 61–68. URL: https://aclanthology.org/2022.acl-short.8/. doi:10.18653/v1/2022.acl-short.8.

[6] G. Qin, J. Eisner, Learning how to ask: Querying LMs with mixtures of soft prompts, in: K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tur, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, Y. Zhou (Eds.), Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Online, 2021, pp. 5203–5212. URL: https://aclanthology.org/2021.naacl-main.410/. doi:10.18653/v1/2021.naacl-main.410.

[7] Y. Xu, X. Guo, Z. Zeng, C. Miao, SoftCoT: Soft chain-of-thought for efficient reasoning with LLMs, in: W. Che, J. Nabende, E. Shutova, M. T. Pilehvar (Eds.), Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Vienna, Austria, 2025, pp. 23336–23351. URL: https://aclanthology.org/2025.acl-long.1137/. doi:10.18653/v1/2025.acl-long.1137.

[8] S. Hao, S. Sukhbaatar, D. Su, X. Li, Z. Hu, J. Weston, Y. Tian, Training large language models to reason in a continuous latent space, 2024. URL: https://arxiv.org/abs/2412.06769. arXiv:2412.06769.

[9] Q. Team, Qwen3 technical report, 2025. URL: https://arxiv.org/abs/2505.09388. arXiv:2505.09388.

[10] Lm-kbc challenge @ iswc 2025, 2025. URL: https://lm-kbc.github.io/challenge2025/, accessed: 2025-08-08.