Recurrent Alignment with Hard Attention for Hierarchical Text Rating

Anonymous ACL submission

Abstract

While large language models (LLMs) excel at understanding and generating plain text, they are not tailored to handle hierarchical text structures or directly predict task-specific properties such as text rating. In fact, selectively and repeatedly grasping the hierarchical structure of large-scale text is pivotal for deciphering its essence. To this end, we propose a novel framework for hierarchical text rating utilizing LLMs, which incorporates **R**ecurrent **A**lignment with **H**ard **A**ttention (**RAHA**). Particularly, hard attention mechanism prompts a frozen LLM to selectively focus on pertinent leaf texts associated with the root text and generate symbolic representations of their relationships. Inspired by the gradual stabilization of the Markov Chain, recurrent alignment strategy involves feeding predicted ratings iteratively back into the prompts of another trainable LLM, aligning it to progressively approximate the desired target. Experimental results demonstrate that RAHA outperforms existing state-of-the-art methods on three hierarchical text rating datasets. Theoretical and empirical analysis confirms RAHA's ability to gradually converge towards the underlying target through multiple inferences. Additional experiments on plain text rating datasets verify the effectiveness of this Markov-like alignment. Our data and code can be available in `https://anonymous.4open.science/r/RAHA/`.

1 Introduction

Scaling up LLMs yields significant advances in their ability to mimic human-like text comprehension and generation (Ouyang et al., 2022; Zeng et al., 2023; Touvron et al., 2023; OpenAI, 2023). They demonstrate remarkable aptitude for in-context learning (ICL) (Brown et al., 2020; Min et al., 2022; Kojima et al., 2022) across various natural language processing (NLP) tasks (Qi et al., 2023; Chen et al., 2023a; Wen et al., 2023; Du et al., 2023). In particular, employing chain of thought

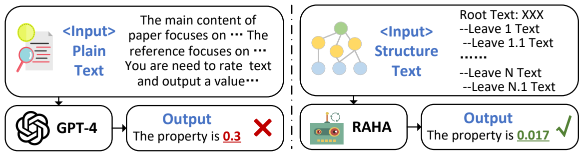


Figure 1: A comparison between a typical LLM and our RAHA in processing hierarchical text rating task. While a typical LLM treats the input as plain text, our RAHA captures hierarchical structures and can straightforwardly provide task-specific rating score.

(CoT) prompts can stimulate the reasoning capabilities of LLMs, enabling them to adeptly navigate and conquer complex downstream tasks (Wei et al., 2022; Wang et al., 2023a).

However, LLMs face a dual challenge. From the perspective of **input**, mainstream LLMs encounter limitations when confronted with extensive and structured textual inputs. While it is possible to extend the input length of LLM (Chen et al., 2023b), this poses additional challenges and complications. For example, excessively long inputs may hinder the attention mechanism of LLM from effectively encompassing the entire context (Liu et al., 2023a). Moreover, a significant proportion of real-world texts (*e.g.*, academic papers, social posts) exhibit hierarchical structures rather than strictly adhering to a linear textual order (Zhao and Feng, 2022; Sun et al., 2023). Figure 1 illustrates an exemplary task to identify groundbreaking score of an academic paper. Placing both the paper and its references within a prompt would result in excessive length and compromise the inherent structural relationship. It is a common approach to model hierarchical text information with a tree structure instead of a plain sequence structure. This involves analyzing the relationship between the root and each leaf individually. However, aggregating all leaf information without proper filtering can introduce noise while also being resource-intensive and time-consuming. Therefore, it is crucial to selectively understand

and integrate valuable relationships.

From the perspective of **output**, while LLMs excel at completing NLP tasks by generating textual responses, practical applications often necessitate directly providing task-required predictions. Despite superiority of parameter-efficient fine-tuning (PEFT) over ICL in terms of speed and performance in few-shot scenarios (Liu et al., 2022), rating tasks that require continuous numerical predictions remain challenging for LLMs. This difficulty arises because LLMs are primarily optimized for discrete text generation rather than precise numerical output, leading to potential inaccuracies and inconsistencies in rating predictions. Thus, further research is needed to effectively enhance LLMs' ability to handle hierarchical text rating.

To this end, this study proposes a novel framework, named **R**ecurrent **A**lignment with **H**ard **A**ttention (**RAHA**) based on LLMs. Firstly, RAHA employs a frozen LLM to manage message passing within the hierarchical structure of the input. For each pair of root and its respective leaf nodes, the LLM discerns and generates symbolic comparative relationships between them. This paired input preserves the structural information of the root and leaf nodes and is much shorter than putting all leaf texts in one prompt. Here, the evaluation guides the LLM to determine whether a particular leaf requires further scrutiny. This decision functions as the hard attention mechanism, effectively reducing the computational load on the LLM and filtering out irrelevant lower-level details. Then, RAHA leverages another trainable LLM to aggregate all selected symbolic relationships that are considered relevant to the root. This LLM is equipped with a trainable adapter followed by a fully connected layer, enabling it to directly predict text ratings. This targeted aggregation supports more effective prediction.

Moreover, inspired by the gradual stabilization seen in Markov Chains, we develop a recurrent alignment strategy to enhance task-specific alignment for the trainable LLM. During the training phase, we introduce a special prompt that incorporates the downstream task score predicted by the trainable LLM. Initially, this value is set to *None* and is subsequently updated with the prediction from the previous training iteration. This dynamic updating allows the trainable parameters to progressively learn and refine the alignment from the currently predicted score to the desired target. Furthermore, consistent with this training methodology, during testing, the trainable LLM performs multiple iterative inferences on the same input. This approach ensures that the predictions become increasingly accurate and aligned with the intended outcomes over successive iterations.

We conduct extensive experiments across three hierarchical text rating benchmarks. Our findings demonstrate that the proposed RAHA outperforms existing state-of-the-art methods in predicting task-specific properties. Furthermore, theoretical and empirical analysis highlights its capacity to incrementally approach the most accurate results through iterative inference processes. Finally, we successfully validate the soundness of our approach on other general rating regression datasets.

The main contributions of this study are summarized as follows:

- We propose a hard attention mechanism to enable LLMs to effectively and efficiently capture hierarchical relationships, thereby addressing the neglect of content structure in long plain text input.
- Drawing inspiration from Markov Chains, we design a recurrent alignment strategy, theoretically and empirically proven to significantly improve the alignment of LLM towards the target value through multiple iterations.
- RAHA exhibits superior performance in understanding hierarchical text input to predict rating score, overcoming the limitations of LLMs in continuous numerical tasks.

2 Related Work

The essence of human intelligence is characterized by the ability to understand abstract concepts, engage in logical reasoning, and make advanced predictions based on existing knowledge (Sternberg et al., 1982; Yu et al., 2023; Huang and Chang, 2022). However, in the era of natural language processing (NLP), despite impressive representation and learning capabilities of neural networks, it is still difficult for them to infer and deduce information from contexts (Duan et al., 2020; Wang et al., 2022). This landscape has been dramatically reshaped with the evolution of large language models (LLMs) (Brown et al., 2020; Workshop et al., 2022), driven by significant upscaling in parameters, data, and computational resources (Ouyang et al., 2022; Zeng et al., 2023; Touvron et al., 2023;

OpenAI, 2023). They exhibit exceptional proficiency for in-context learning (ICL) (Brown et al., 2020; Min et al., 2022; Kojima et al., 2022) across a wide range of NLP tasks (Qi et al., 2023; Chen et al., 2023a; Wen et al., 2023; Du et al., 2023). One of the key advancements in LLMs is the incorporation of strategies like Chain of Thought (CoT) prompting, which empowers these models to generate reasoning steps and tackle more complex downstream application (Liu et al., 2023b; Wei et al., 2022; Wang et al., 2023a).

Notwithstanding the progress made in CoT reasoning (Wei et al., 2022; Wang et al., 2023b; Kojima et al., 2022), there remains a notable deficiency in current methodologies regarding the processing of hierarchical structures within long text. Numerous studies have focused on identifying and correcting specific thought units where the reasoning process may deviate or require additional information, aiming to produce desired outcomes (Yao et al., 2023; Ling et al., 2023; Yang et al., 2023; Wang et al., 2023a). This prevailing research predominantly concentrates on purely textual content, neglecting the intrinsic hierarchical nature of certain text formats (Zhao and Feng, 2022; Sun et al., 2023). In our work, we propose a hard attention mechanism to redress this shortfall by introducing a novel paradigm for enhancing the processing of structured text within CoT reasoning.

The escalation in the scale and adaptability of LLMs has been accompanied by significant advancements in model fine-tuning and adaptation, exemplified by the introduction of various adapter architectures (Houlsby et al., 2019; Pfeiffer et al., 2020; Zaken et al., 2022; Hu et al., 2022). However, these adaptations have primarily focused on enhancing the model's generation capabilities and have not addressed the limitations of LLMs in directly generating continuous prediction values like text rating. Concurrently, recent research within LLMs has increasingly focused on recurrent alignment, primarily through prompting techniques and iterative refinement processes (Huang et al., 2023; Zelikman et al., 2022). Yet, these methodologies have not sufficiently capitalized on employing the properties from predictive tasks as feedback mechanisms for iterative refinement. Our contribution in this regard is the formulation of a Markov-like recurrent alignment strategy. It represents a novel approach in harnessing the model's output for successive iterative enhancements, thereby augmenting the predictive precision and versatility of LLMs.

3 Methodology

The proposed framework, RAHA, is depicted in Figure 2. It includes a tree-based hard attention mechanism that enhances the ability of LLMs to effectively capture hierarchical structures. In addition, a trainable LLM is employed to output hierarchical text rating score. Moreover, we employ a Markov-like recurrent alignment strategy to enable the RAHA to iteratively align with the ground truth of the downstream task.

3.1 Problem Formulation

For each sample in our data collection, we represent its hierarchical structure as a tree, which is denote as $\langle r_i, L_i \rangle$. This structure consists of a textual root r_i and a set of m leaves $L_i = \{l_1^{(i)}, l_2^{(i)}, \cdots, l_m^{(i)}\}$. Each leaf $l_j^{(i)}$ serves as the textual root of its own tree and can have its own associated leaves.

Our framework aims to accomplish an objective with the input $\langle r_i, L_i \rangle$, which is to estimate the text rating y_i. By analyzing the hierarchical structure of the data, RAHA can filter meaningful insights and make accurate predictions according to the recurrent alignment strategy.

3.2 Hard Attention Mechanism

RAHA framework integrates a tree-based hard attention mechanism to facilitate message passing within a tree structure. It eliminates the necessity for LLMs to grasp the intricate interplay between root and individual leaves within extensive plain texts.

To accomplish this goal, this mechanism firstly utilizes a frozen LLM to figure out the comparative relationship between the root r_i and its j-th leaf $l_j^{(i)}$. This process is facilitated by constructing a prompt $p_j^{(i)}$, which contains the following information. Firstly, it provides a clear task description, such as identifying disruptions in papers or predicting potential popularity in social posts. Next, the prompt includes the root text and leaf text along with their respective meta-information. Finally, a well-crafted question is included to extract the necessary features of the root and each leaf that are essential for the task. For a more comprehensive understanding, please refer to the Appendix D.1 for specific formulation and illustrative examples.

With the provided prompt $p_j^{(i)}$, the LLM can derive two critical pieces of information for each pair of root and child $(r_i, l_j^{(i)})$, which are the hard

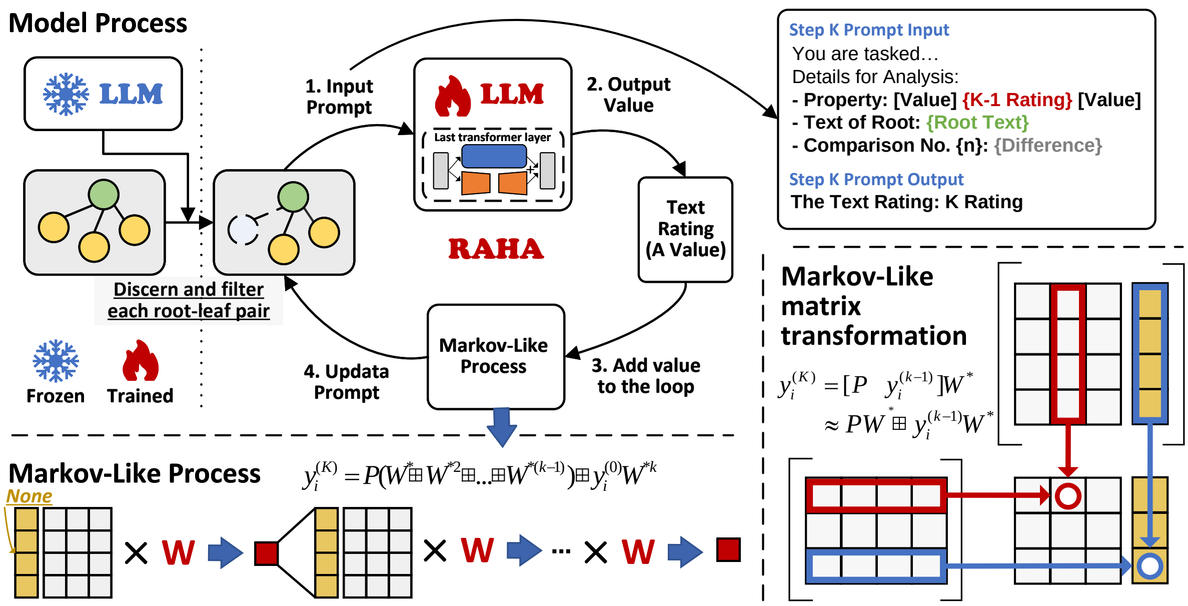


Figure 2: The overview of RAHA architecture. A frozen LLM determines connections and generates updates with hard attention scores to filter noise. RAHA incorporates an adapter and fully connected layer within a trainable LLM to predict text rating scores after aggregating updates. During training and testing, the predicted score is fed back into the trainable LLM prompt, refining predictions over multiple iterations.

attention score $a_j^{(i)}$ and a tailored symbolic representation $d_j^{(i)}$:

$$\begin{aligned} p_j^{(i)} &= f_p^{(1)}(r_i, l_j^{(i)}) \\ a_j^{(i)}, d_j^{(i)} &= \mathcal{F}(p_j^{(i)}) \end{aligned} \tag{1}$$

where $f_p^{(1)}$ represents the heuristics function for constructing the prompt and $\mathcal{F}$ denotes the frozen LLM.

Here, the hard attention score $a_j^{(i)} \in \{0, 1\}$ is a binary value, that determines whether the leaf $l_j^{(i)}$ deserves further aggregation for the root r_i. The symbolic representation $d_j^{(i)}$ serves as an update for the root r_i and provides valuable task-oriented insights. This information captures essential aspects such as the integration, correlation, or distinction between the root r_i and its j-th leaf $l_j^{(i)}$.

Given updates $D_i = [d_1^{(i)}, d_2^{(i)}, \cdots, d_m^{(i)}]$ of the root relative to all leaves, the utilization of hard attention scores $A_i = [a_1^{(i)}, a_2^{(i)}, \cdots, a_m^{(i)}]$ helps filter out potential noise, leading to a reduction in computational consumption:

$$\begin{aligned} D_i^* &= A_i \otimes D_i \\ &= [a_1^{(i)} \otimes d_1^{(i)}, a_2^{(i)} \otimes d_2^{(i)}, \cdots, a_m^{(i)} \otimes d_m^{(i)}] \end{aligned} \tag{2}$$

where $\otimes$ denotes the selection operator and D_i' keeps m' symbolic updates after selection, where $m' \leq m$. The valuable updates D_i^* will be aggregated by the subsequent model.

3.3 Parameter-Efficient Fine-Tuning

We employ a trainable LLM to complete aggregation of the updates within a tree structure. This LLM is enhanced with Parameter-Efficient Fine-Tuning (PEFT) techniques, which improve its alignment with downstream tasks (Houlsby et al., 2019). We integrate trainable parameters $\Delta\boldsymbol{W}$ as an adapter into the original LLM parameters $\boldsymbol{W}_0$ (Hu et al., 2022; Liu et al., 2022). It is represented as:

$$\boldsymbol{W}\boldsymbol{x} = \boldsymbol{W}_0\boldsymbol{x} + \Delta\boldsymbol{W}\boldsymbol{x} = \boldsymbol{W}_0\boldsymbol{x} + \boldsymbol{B}\boldsymbol{A}\boldsymbol{x} \tag{3}$$

where $\boldsymbol{B}$ and $\boldsymbol{A}$ are both trainable low-rank matrices. In addition, we incorporate a fully connected layer following the hidden representation $\boldsymbol{h}$ from the last layer of the LLM.

$$y = \boldsymbol{W}_1\boldsymbol{h} \tag{4}$$

where the $\boldsymbol{W}_1$ is a trainable matrix. This layer facilitates direct prediction of property value for the downstream task. For simplicity, we denote this trainable LLM as $\mathcal{F}^*$.

The prompt for facilitating aggregation of this trainable LLM consists of three key components. Firstly, it includes details about the root r_i of the tree. Secondly, it incorporates the previously filtered updates D_i^*. Next, inspired by Markov Chains, it provides the predicted rating score y_i^* of the text required for the task. Finally, we include the task-related question in the prompt. We aim to iteratively bring the predicted value closer to the true value through prior states. It is important to note that at the initial stage, the model has not started the inference yet. As a result, there is no available predicted value, and therefore, this value is set to *None* in the prompt. The prompt can be represented as p_i:

$$p_i = f_p^{(2)}([r_i, D_i^*, y_i^*]) \tag{5}$$

where $f_p^{(2)}$ denotes heuristic approach for constructing the prompt p_i and the y_i^* is initialized to *None*, denoted as ϕ. Please refer to the Appendix D.2 for specific formulation and illustrative examples.

3.4 Recurrent Alignment Strategy

Many existing studies typically conclude once they complete the previous step. However, we are now considering the possibility of leveraging LLMs to enhance their understanding of inputs based on their previous outputs. Inspired by the principle of Markov Chains, where each state depends on the previous one and converges to a stationary distribution, we propose a recurrent alignment strategy to enhance the learning and inference process of RAHA. Specifically, given the root r_i and filtered updates D_i^*, we perform inference multiple times using trainable LLM $\mathcal{F}^*$. The difference of each step is that we update this rating value y_i^* in the prompt function $f_p^{(2)}$ with the model prediction from the previous step. The formulations are shown as follows:

$$\begin{cases} y_i^{(1)} &= \mathcal{F}^*(f_p^{(2)}(r_i, D_i^*, \phi)) \\ y_i^{(2)} &= \mathcal{F}^*(f_p^{(2)}(r_i, D_i^*, y_i^{(1)})) \\ & \cdots \\ y_i^{(k)} &= \mathcal{F}^*(f_p^{(2)}(r_i, D_i^*, y_i^{(k-1)})) \end{cases} \tag{6}$$

In this context, each iteration can be viewed as a transition in a Markov Chain, progressively refining the state towards convergence. This strategy offers significant benefits to the model's learning process during the training stage. Since the target output of each iteration is considered the ground truth in the downstream task data, the model gradually approaches the true value based on existing assessments.

During the testing phase, we conduct multiple iterations of the model to perform inference on the same input. This iterative approach allows the model to begin with naive information, advancing step by step towards an accurate hidden representation and progressively aligning itself to the true value. This process is analogous to a Markov Chain reaching its steady-state distribution. Since the model parameters remain unchanged during the testing phase, the process can be considered equivalent to the transition matrix of a Markov Chain. The final predicted value can be expressed as:

$$y_i^{(k)} = P(F^* \boxplus F^{*2} \boxplus F^{*3} \boxplus \cdots \boxplus F^{*(k-1)}) \boxplus y_i^{(0)} F^{*k} \tag{7}$$

Assuming that the spectral radius of F^* is less than 1 (Blundell et al., 2015), the value can eventually converge to:

$$\lim_{t\to\infty} y_i^{(k)} = P(I - F^*)^{-1} \tag{8}$$

The detailed theoretical proof is in appendix B.

3.5 Training

Our proposed RAHA integrates two LLMs. The parameters of the first LLM $\mathcal{F}$ remain frozen throughout the process. As for the second LLM $\mathcal{F}^*$, we keep its main parameters $\boldsymbol{W}_0$ fixed. We solely employ training data from downstream tasks to optimize its trainable parameters $\Delta\boldsymbol{W}$ and $\boldsymbol{W}_1$ together, which correspond to the adapter and the fully connected layer, respectively. Specifically, since reasoning s_i has no ground truth, we utilize the property values y_i required by the task to build the mean squared error (MSE) as the objective function:

$$\mathcal{L} = \frac{1}{2M} \sum_{i=1}^{M} (y_i^{(k)} - y_i)^2 \tag{9}$$

where M is the number of training samples and $y_i^{(k)}$ represent the predicted value for the i-the sample in the k-th iteration. We conduct a total of K iterations. After each prediction, we will update the prompts for the next iteration. The target value in each round of loss function is the ground truth of the training data. Appendix C provides detailed steps for RAHA.

4 Experiments

4.1 Datasets and Evaluation Metrics

To evaluate the effectiveness of RAHA, we utilize three hierarchical text rating datasets, namely

Model	DBLP		PubMed		PatentsView		Average	
	MSE ↓	MAE ↓	MSE ↓	MAE ↓	MSE ↓	MAE ↓	MSE ↓	MAE ↓
SciBERT	0.072	0.119	**0.025**	0.116	0.069	0.121	0.055	0.119
Bloom-7B	0.062	0.104	0.044	0.129	0.081	0.162	0.062	0.132
ChatGLM3-6B-32K	0.045	0.091	0.056	0.182	0.042	0.088	0.047	0.120
RAHA	**0.024***	**0.070****	**0.025***	**0.106****	**0.022***	**0.084***	**0.023***	**0.086***
w/o Hard Attention	0.049	0.098	0.035	0.125	0.041	0.089	0.042	0.104
w/o PEFT	0.082	0.101	0.031	0.119	0.034	0.089	0.049	0.103
w/o Recurrent Alignment	0.025	0.085	0.028	0.110	0.023	0.085	0.025	0.093

Table 1: A comparative results of various language models. The performance is measured in terms of MSE and MAE with lower values indicating better performance. The best results are in **bold**. The differences are statistically significant as determined by student-t test and * is significance results for the model.

DBLP, PubMed, and PatentsView. See the Appendix A for detailed introduction. Each dataset is characterized by citation relationships and their respective textual content. Considering the extensive size of these datasets, we randomly select a subset of nearly 10,000 samples from each dataset and allocate 15% of them for validating and 15% for testing purposes. The target text rating score we focus on is the disruption index (Funk and Owen-Smith, 2017; Wu et al., 2019), which measures the novelty and impact of the papers or patents on a scale ranging from -1 to 1. We use Mean Squared Error (MSE) and Mean Absolute Error (MAE) as the main evaluation metrics.

4.2 Baselines

We compare RAHA with four baselines. (1) **SciBERT** (Beltagy et al., 2019) is a pre-trained language model within the scientific domain. (2) **BLOOM-7B** (Workshop et al., 2022) exemplifies advancements in large-scale multi-language processing. (3) **Chatglm3-6B-32K** (Zeng et al., 2023) is a generative language model based on autoregressive blank Infilling. They're all publicly accessible. For all baselines, we simply add a fully connected layer after their last hidden states for property prediction. Here, we don't compare GPT4 since it lacks the ability to map the input to our numerical target.

4.3 Experiment Setup

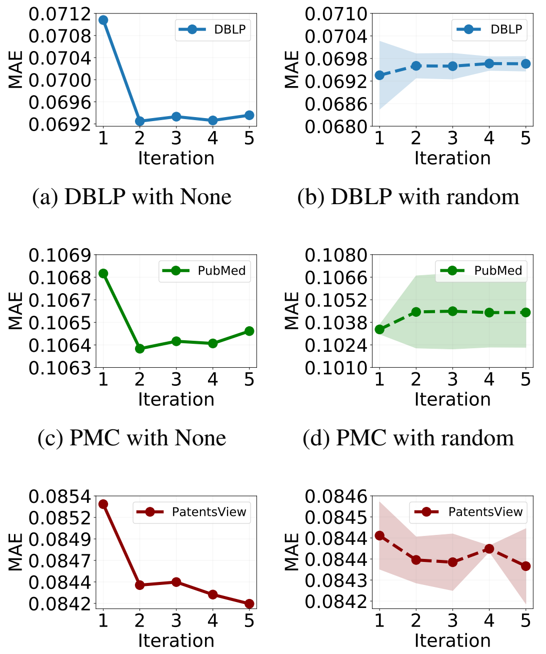
We implement experiments via PyTorch on a single NVIDIA A800 GPU. The two LLMs included in our RAHA are both Chatglm3-32k. Optimization of the models is achieved using AdamW optimizer (Loshchilov and Hutter, 2019), with the learning rate set to 1e-5 and the gradient clipping value fixed to 0.2. We set the model to accommodate a maximum input length of 2560. The batch size is set to 4. The low rank of the adapter in the second LLM is 64. We use the PEFT package to insert the adapter for the last layer of LLM(Mangrulkar et al., 2022). The number of training and testing iterations K of RAHA are set to 3 and 5, respectively. The number of epochs is set to 3 for other baselines. The optimal model checkpoint is selected based on performance metrics obtained from the development set.

4.4 Main Results

We report the main results on DBLP, PubMed, and PatentView in Table 1. Overall, we can observe that our framework RAHA achieves the best MSE and MAE in three datasets.

Specifically, on the DBLP dataset, RAHA demonstrates superior accuracy, reducing MSE and MAE by 0.048 and 0.049, respectively, compared to SciBERT. This improvement underscores RAHA's precision and consistency in interpreting complex academic metadata. Additionally, RAHA shows a marked improvement over Bloom-7b, illustrating its enhanced ability to discern nuanced contextual variations within the DBLP entries.

In the PubMed and PatentView datasets, RAHA maintains its leadership, affirming its robustness and adaptability. The framework's efficacy in these domains can be attributed to its innovative use of a tree-based hard attention mechanism, which methodically navigates through hierarchical data structures, ensuring that significant informational cues are captured and emphasized. Moreover, RAHA's recurrent alignment strategy enhances its ability to discern and interpret the nuanced linguistic and semantic variations that are critical in fields like biomedical research and patent descriptions.



(a) DBLP with None (b) DBLP with random

(c) PMC with None (d) PMC with random

(e) Patent with None (f) Patent with random

Figure 3: Comparison of predictions over multiple iterations during recurrent alignment across three datasets. Figures (a), (c), and (e) show outcomes with the initial prompt set to None. Figures (b), (d), and (f) show results with the initial prompt randomly chosen from -1 to 1.

4.5 Ablation Study

To dissect the contributions of the individual components in our RAHA framework, we conduct ablation studies, as shown in the lower half of Table 1.

(1) RAHA w/o Tree-based hard attention mechanism: Excluding the hard-attention mechanism leads to a decline in performance across all datasets. This mechanism is crucial for RAHA's ability to process and relate different parts of tree-structured data. Without it, RAHA struggles to pinpoint the most relevant parts of the input text for decision-making, highlighting the importance of understanding the information between the root and leaves.

(2) RAHA w/o Parameter-efficient fine-tuning: Removing the adapter results in the most substantial increases in both MAE and MSE. The adapter enables the second LLM to fine-tune its parameters based on training data. Without it, the second LLM struggles to effectively align with downstream tasks, especially those requiring specific property values, demonstrating the adapter's significance in the architecture.

(3) RAHA w/o Recurrent Alignment: The recurrent alignment strategy iteratively refines outputs based on previous predictions, enhancing the learning process. Without this strategy, there is a slight increase in errors, indicating its critical role in maintaining accuracy and performance by learning from previous predictions.

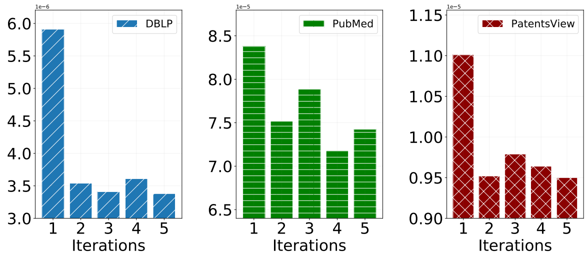
4.6 Predictions over Multiple Iterations

Figure 3 displays the predictions of our RAHA framework over multiple iterations during the test stage. It provides evidence to support our hypothesis that the recurrent alignment strategy allows the fine-tuned LLM to progressively approximate more accurate properties. We use different initialization values in the prompt (see equation 5) to provide broader perspectives for investigating the recurrent alignment strategy. The standard initialization involves using *None* as a value in the prompt. For comparison, we also utilize random initialization with values ranging from -1 to 1.

As shown in Figure 3a, Figure 3c, and Figure 3e, despite fluctuations, the decrease in MAE over gradual iterations demonstrates the ability of RAHA to refine its understanding of the input. This trend suggests that RAHA is not merely fitting to the immediate data but also leveraging its recurrent alignment component to internalize the original input and previous understanding. The ability to improve its performance by iteratively replacing the predicted value in the prompt proves the efficacy of the recurrent alignment strategy.

In contrast, as shown in Figure 3d and Figure 3f, the result of the recurrent alignment strategy initialized with random values is manifested in a random process according to MAE. The lack of the scratch-to-refinement process we set in place results in models making predictions by guessing rather than reasoning from prior knowledge. This random initialization hampers interpretability as the predictions are not based on any discernible pattern or learning process.

Overall, the recurrent alignment strategy plays a critical role in the alignment of RAHA to the downstream task. By replacing the predicted value from the previous round to construct the prompt, this approach allows the model to evolve its knowledge in a logical and transparent manner, which is particularly valuable for applications that require reliability and trustworthiness.



(a) KL of DBLP (b) KL of PubMed (c) KL of Patents

Figure 4: A detailed analysis based on the Kullback-Leibler (KL) divergence over testing iterations across three datasets. It highlights the narrowing gap between the representation of the fine-tuned LLM and the target representation during the recurrent alignment process.

Model	ASAP		Splunk	
	MSE ↓	MAE ↓	MSE ↓	MAE ↓
SciBERT	0.396	0.517	0.208	0.363
Bloom-7b	0.256	0.446	0.214	0.384
GLM3	0.252	0.439	0.214	0.361
RAHA	**0.249**	**0.421**	**0.212**	**0.358**

Table 2: The performance of various language models on two text rating datasets, ASAP and Splunk, using Mean Squared Error (MSE) and Mean Absolute Error (MAE) as metrics.

4.7 Model Representation after Recurrent Alignment

We provide further insight into the role of the recurrent alignment strategy in driving dynamics of model representation. Since our strategy can enable the trainable LLM to learn the alignment capabilities from scratch to pierce, we assume that directly incorporating the task-desired target truth within the prompt (see equation 5) enables the fine-tuned LLM to derive the target's true representation, facilitating subsequent comparisons with the predicted representation. This simulates a situation where the result obtained through previous understanding is completely correct. We employ the Kullback-Leibler (KL) divergence as a metric to gauge the disparity between the predicted representation extracted by the LLM at each iteration and the target representation. Figure 4 delineates the KL divergence trajectories over five test iterations across three datasets. Despite occasional fluctuations, the downward trend suggests that RAHA progressively refines its approximation of the target representation. This highlights the effectiveness of the recurrent alignment process. Combined with the results of specific predictions from the previous step, the fine-tuned LLM can further align with downstream tasks when grasping and aggregating updates. This trend shows a static snapshot of model performance and the significance of the recurrent alignment iterations.

4.8 Experiment on Rating Data without Hierarchical Structure

To enhance the assessment of the generalization of recurrent alignment, we conduct experiments on two plain text rating datasets. Detailed information of the dataset can be found in Appendix A.

The table 2 provides a performance comparison of several models on two text rating datasets, ASAP and Splunk. Generally, RAHA performs better across both the ASAP and Splunk datasets in terms of MAE and nearly best in MSE, suggesting its robustness and suitability for these tasks and its recurrent alignment process's ability to capture the nuances in text rating data effectively.

5 Conclusion

In this paper, we propose a novel framework called RAHA, that leverages two LLMs to analyze hierarchically structured text. RAHA incorporates a tree-based hard attention mechanism and a recurrent alignment strategy. The tree-based attention enables a frozen LLM to understand the associations between the root and each leaf separately and then selectively choose significant updates for aggregation. This results in a reduction of potential noise in the hierarchical structure and improved utilization of computing resources. The iterative recurrent alignment empowers a trainable LLM to revisit insights gained from previous deliberations, progressively aligning itself with the desired property for downstream tasks. In evaluations on three datasets, RAHA outperforms existing baselines in text rating estimation. Theoretical and empirical analysis reveals that by repeated iterations of prompting the results from the preceding step, RAHA produces hidden representations that gradually approach the optimal representation. This study enhances the abilities of LLMs in handling hierarchical text and aligning with specific tasks.

6 Limitation

We list several limitations in this work that could be improved in the future. One limitation of our research is the inference time associated with RAHA. The hard attention and iterative recurrent alignment, while beneficial for progressively refining representations, can lead to increased computational overhead. Future efforts should prioritize optimizing the model framework to reduce inference time, enhancing the broader applicability of RAHA. Additionally, further studies are needed to explore the potential of RAHA in other hierarchical text analysis domains and to validate its performance across a wider range of tasks. A more rigorous investigation into the principles underlying the recurrent alignment strategy is necessary. Understanding the theoretical foundations and the exact mechanisms through which iterative prompting improves representation alignment can provide deeper insights and guide future enhancements to the model.

7 Ethics Statement

We recognize the ethical implications of our work and the importance of developing and using LLMs responsibly. LLMs are powerful tools that need careful monitoring. While our research aims to improve LLMs, these techniques can also be misused to generate harmful content. We emphasize not placing excessive trust in generated content until LLMs are well-regulated.

References

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. 2015. Weight uncertainty in neural network. In *International conference on machine learning*, pages 1613–1622. PMLR.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Hailin Chen, Amrita Saha, Steven Hoi, and Shafiq Joty. 2023a. Personalized distillation: Empowering open-sourced llms with adaptive learning for code generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6737–6749.

Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. 2023b. Longlora: Efficient fine-tuning of long-context large language models. In *The Twelfth International Conference on Learning Representations*.

Chunhui Du, Jidong Tian, Haoran Liao, Jindou Chen, Hao He, and Yaohui Jin. 2023. Task-level thinking steps help large language models for challenging classification task. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2454–2470.

Nan Duan, Duyu Tang, and Ming Zhou. 2020. Machine reasoning: Technology, dilemma and future. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts*, pages 1–6.

Russell J. Funk and Jason Owen-Smith. 2017. A dynamic network measure of technological change. *Manag. Sci.*, 63:791–817.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Jiaxin Huang, Shixiang Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2023. Large language models can self-improve. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1051–1068.

Jie Huang and Kevin Chen-Chuan Chang. 2022. Towards reasoning in large language models: A survey. *arXiv preprint arXiv:2212.10403*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Zhan Ling, Yunhao Fang, Xuanlin Li, Zhiao Huang, Mingu Lee, Roland Memisevic, and Hao Su. 2023. Deductive verification of chain-of-thought reasoning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. 2022. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35:1950–1965.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023a. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023b. Pretrain, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2022. Peft: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064.

OpenAI. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. Mad-x: An adapter-based framework for multi-task cross-lingual transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673.

Jingyuan Qi, Zhiyang Xu, Ying Shen, Minqian Liu, Di Jin, Qifan Wang, and Lifu Huang. 2023. The art of socratic questioning: Recursive thinking with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4177–4199.

Robert J Sternberg, Janet S Powell, and Daniel B Kaye. 1982. The nature of verbal comprehension. *Poetics*, 11(2):155–187.

Chenkai Sun, Jinning Li, Yi Fung, Hou Chan, Tarek Abdelzaher, ChengXiang Zhai, and Heng Ji. 2023. Decoding the silent majority: Inducing belief augmented social graph with large language model for response forecasting. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 43–57.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023a. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2609–2634.

Siyuan Wang, Zhongkun Liu, Wanjun Zhong, Ming Zhou, Zhongyu Wei, Zhumin Chen, and Nan Duan. 2022. From lsat: The progress and challenges of complex reasoning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:2201–2216.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023b. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Jiaxin Wen, Pei Ke, Hao Sun, Zhexin Zhang, Chengfei Li, Jinfeng Bai, and Minlie Huang. 2023. Unveiling the implicit toxicity in large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1322–1338.

BigScience Workshop, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Lingfei Wu, Dashun Wang, and James A. Evans. 2019. Large teams develop and small teams disrupt science and technology. *Nature*, 566:378–382.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. 2023. Large language models as optimizers. *arXiv preprint arXiv:2309.03409*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10601*.

Fei Yu, Hongbo Zhang, and Benyou Wang. 2023. Nature language reasoning, a survey. *arXiv preprint arXiv:2303.14725*.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. 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)*, pages 1–9.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, Weng Lam Tam, Zixuan Ma, Yufei Xue, Jidong Zhai, Wenguang Chen, Zhiyuan Liu, Peng Zhang, Yuxiao Dong, and Jie Tang. 2023. GLM-130b: An open bilingual pre-trained model. In *The Eleventh International Conference on Learning Representations*.

Qihang Zhao and Xiaodong Feng. 2022. Utilizing citation network structure to predict paper citation counts: A deep learning approach. *Journal of Informetrics*, 16(1):101235.

Appendix

A Data analysis

In this study, we utilized five diverse datasets to evaluate the performance of our RAHA: DBLP, PubMed, PatentsView, ASAP, and Splunk. Each dataset was split into training, validation, and test sets to ensure robust evaluation and comparison, which is shown as Table 3.

DBLP: A dataset contains bibliographic information on major computer science journals and proceedings. https://www.aminer.cn/citation

PubMed: PubMed contains citations and abstracts of biomedical literature from several NLM literature resources, including MEDLINE—the largest component of the PubMed database. https://pubmed.ncbi.nlm.nih.gov/download/

PatentsView: PatentsView offers publicly accessible patent research data sets with detailed documentation, which focusing on technological and innovation studies. https://patentsview.org/download/data-download-tables

ASAP: The Automated Student Assessment Prize (ASAP) dataset, sourced from Kaggle, is used for evaluating automated essay scoring systems. https://www.kaggle.com/c/asap-aes/data

Splunk: A Kaggle competition *Predict WordPress Likes* data, is used for operational intelligence tasks. https://www.kaggle.com/c/predict-wordpress-likes/data

Model	Train	Val	Test	Total
DBLP	6945	1488	1488	9921
PubMed	6956	1491	1490	9937
PatentsView	3988	855	854	5697
ASAP	3500	750	750	5000
Splunk	5763	1235	1235	8233

Table 3: Dataset Splits for RAHA. The table displays the number of instances in the training, validation, and test sets for each dataset (DBLP, PubMed, PatentsView, ASAP, and Splunk).

B Formal Proof of Markov-like Process

In our model, we employ a recurrent alignment strategy, analogous to a Markov chain process, by performing multiple iterations on the same input to refine inference. This approach allows the model to start with naive information and progressively refine towards an accurate representation over time.

Given that the model parameters remain unchanged during the testing phase, this iterative process is equivalent to transitions defined by a Markov Chain transition matrix. The mathematical justification proceeds as follows:

B.1 Definitions

- $y_i^{(k)}$: State of the model at the k-th iteration.
- P: Matrix representation of prompt. Fixed.
- F^*: Represents the fixed parameters of the model, analogous to a transition matrix in a Markov chain.
- $\boxplus$: A custom operation defined as follows: $A \boxplus B = (A_1 M + B_1 M) \| (A_2 M + B_2 M)$ Here, A and B are matrices that are split into sub-blocks A_1, A_2 and B_1, B_2, which are then transformed by matrix M and recombined.

B.2 Iterative Process Expansion

The iterative refinement process can be expanded recursively as:

$$
\begin{aligned}
y_i^{(k)} &= [P \quad y_i^{(k-1)}]F^* \\
&= PF^* \boxplus y_i^{(k-1)} F^* \\
&= PF^* \boxplus (PF^* \boxplus y_i^{(k-2)} F^*) F^* \\
&= PF^* \boxplus PF^{*2} \boxplus y_i^{(k-2)} F^{*2} \\
&= \dots \\
&= P(F^* \boxplus F^{*2} \boxplus \dots \boxplus F^{*(k-1)}) \boxplus y_i^{(0)} F^{*k}
\end{aligned}
$$

Define $S = F^* \boxplus F^{*2} \boxplus \dots \boxplus F^{*(k-1)}$, where $\boxplus$ operates similarly to addition. Assuming that the spectral radius of F^* is less than 1, this infinite series converges, meaning the influence of the initial state $y_i^{(0)}$ diminishes over time as k increases. This results in: $\lim_{k \to \infty} S = (I - F^*)^{-1}$ which implies that: $y_i^{(k)} \to P(I - F^*)^{-1} \quad \text{as} \quad k \to \infty$.

The convergence of $y_i^{(k)}$ to $P(I - F^*)^{-1}$ as k approaches infinity can be understood through the lens of stability theory in linear algebra. Since most weights of the neural network are concentrated around zero after training (Blundell et al., 2015), the spectral radius of F^* is less than 1. The spectral radius condition, $\rho(F^*) < 1$, ensures that the effects of F^* dampen over successive iterations, leading to the stabilization of $y_i^{(k)}$. This behavior is analogous to a Markov chain reaching its steady state, where the transition matrix F^* dictates the evolution of states such that the influence of the initial state progressively wanes, eventually stabilizing at a distribution determined by P and $(I - F^*)^{-1}$. This stabilization is crucial in demonstrating that the iterative refinement process under fixed parameters behaves similarly to state transitions in a Markov model, with F^* serving as a transition-like matrix.

Algorithm 1 RAHA

Input: hierarchical text $\langle r_i, L_i \rangle$
Output: task-desired property y_i

```
while 1 ≤ k iteration ≤ K do
    for each root and leaf pair (r_i, s_j^(i)) in ⟨r_i, L_i⟩ do
        p_j^(i) ← construct prompt f_p^(1)(r_i, s_j^(i))
        a_j^(i), d_j^(i) ← conduct inference F(p_j^(i))
    end for
    A_i ← related hard attentions [a_1^(i), a_2^(i), ···, a_m^(i)]
    D_i ← all updates [d_1^(i), d_2^(i), ···, d_m^(i)]
    D'_i ← filter out noise A_i ⊗ D_i
    if k = 1 then
        p_i ← construct aggregation prompt f_p^(2)(r_i, D'_i, φ)
    else
        p_i ← f_p^(2)(r_i, D'_i, y_i^(k-1))
    end if
    y_i^(k) ← conduct inference F*(p_i)
    L ← compute loss between y_i^(k) and y_i
    ΔW, W_1 ← update parameters via AdamW
end while
return y_i^(k)
```

C Pseudo Code

The pseudo-code of our framework is shown in algorithm 1.

D Prompt

In the appendix section, we present a series of detailed tables that outline the prompts used in the various mechanisms of the RAHA framework. These tables are crucial for understanding the intricacies of how the tree-based hard attention mechanism, parameter-efficient fine-tuning, and recurrent alignment strategy are implemented in practice. Each table provides the structure of prompts used in our

Prompt for Tree-based Hard Attention in Academic Paper Analysis
Task1: Determine whether a reference paper is important to a focal paper based on the abstract. Return Import Index is "1" if it is important and "0" if it is not. Don't repeat my inputs, just output the values. Example 1: *Input*: Focal paper abstract: abstract1 Reference paper abstract: reference1 *Output*: 0 *Input*: Focal paper abstract: {abstract} Reference paper abstract: {reference} *Output*:
Task2: You are now tasked with assessing the disruptive potential in the research area of academic papers. Your approach involves contrasting the abstract of a focus paper with the abstracts of its cited references. No need to give me abstract's analysis, just output Contrast and Difference. Focal paper abstract: {abstract} Reference paper abstract: {reference} *Contrast and Difference*:

Table 4: Structured Prompts for Tree-Based Hard Attention in Academic Paper Analysis within the RAHA Framework. This table showcases the input format and elucidates how the prompts direct the LLM's focus and analytical processes in handling the hierarchical structures of academic texts.

Prompt for Tree-based Hard Attention in Patent Analysis
Task1: Assess the importance of a reference patent based on its abstract in relation to a focal patent. Return an Importance Index as "1" if it is important and "0" if it is not. Do not repeat the inputs, only provide the evaluation. Example 1: *Input*: Focal Patent abstract: abstract1 Reference Patent abstract: reference1 *Output*: 0 *Input*: Focal Patent abstract: {abstract} Reference Patent abstract: {reference} *Output*:
Task2: You are tasked with analyzing the innovation gap and potential impact between patents. Your job is to contrast the abstract of a focal patent with the abstracts of its related patents. Avoid providing an analysis of the abstracts themselves; focus instead on the contrast and potential differences. Focal Patent abstract: {abstract} Related Patent Abstract: {reference} *Contrast and Difference*:

Table 5: Structured Prompts for Tree-Based Hard Attention in Patent Analysis within the RAHA Framework. This table presents examples of how prompts are tailored for assessing the importance and innovation gap between patents, demonstrating the framework's adaptability to different domains.

experiments, including examples for academic papers and patents. For specific tasks, prompts should be replaced with content that fits the context of the task.

D.1 Detailed Prompt for Hard Attention

In the RAHA framework, the integration of a tree-based hard attention mechanism significantly enhances the process of message passing within hierarchical structures. This mechanism streamlines the task for LLMs by reducing the complexity involved in understanding the interplay between the root and individual leaves of a tree within extensive texts. To practically implement this mechanism, we utilize structured prompts that direct the LLM's focus and analytical process. Examples of these structured prompts are illustrated in the following table 4.

In addition to academic papers, the RAHA framework's tree-based hard attention mechanism is adeptly applied to patent analysis. The Table 5, showcases structured prompts designed for patent analysis.

D.2 Detailed Prompt for Fine-Tuning and Recurrent Alignment

In this section, we present a detailed example of a prompt designed specifically for the fine-tuning and recurrent alignment components of the RAHA framework. The Property between the [DINDEX] tokens changes iteratively, with the property for

Prompt for Fine-Tuning and recurrent alignment in Academic Paper Analysis
Task: You are tasked with assessing the disruptive potential of academic papers. Your primary tool for this analysis is the Disruption Index, a metric ranging from -1 to 1. This index quantifies the level of innovation or breakthrough a paper represents. A higher positive value on the index indicates a significant breakthrough, while negative values suggest a lower level of innovation. Please provide a detailed analysis based on the contrast and differences between the focus paper and its references. Use the Disruption Index of the focus paper to guide your assessment. Pay special attention to the unique contributions or shortcomings of the focus paper in comparison to the referenced works. *Details for Analysis*: Determine whether the DINDEX predicted in the previous epoch is high or low: [DINDEX]{Property}[DINDEX] Abstract of Focus Paper: {abstract} Comparison with Reference Paper : {reference} Based on the above information, analyze the reason for the disruptive nature (or lack thereof) of the focus paper.

Table 6: Example of a Structured Prompt for Fine-Tuning and recurrent alignment in Academic Paper Analysis within the RAHA Framework. This table demonstrates how prompts are designed to assess the innovation level of papers using the Disruption Index.

this iteration being the output from the previous one. The prompt in Table 6 is tailored for the task of assessing the disruptive potential of academic papers using the Disruption Index. This example illustrates how the prompt structures the analysis process, guiding the model to focus on key indicators and draw meaningful conclusions from the data.

In addition to academic papers, the fine-tuning and recurrent alignment components of the RAHA framework are also effectively applied to the domain of patent analysis. The prompt provided in Table 7 is specifically designed for evaluating the innovation level and potential breakthroughs of patents.

Prompt for Fine-Tuning and recurrent alignment in Patent Analysis
Task: You are tasked with evaluating the innovation level and potential breakthrough of patents. Your primary tool for this analysis is the Disruption Index, a metric ranging from -1 to 1. This index helps quantify the level of novelty and potential market disruption a patent represents. A higher positive value on the index indicates a significant breakthrough, while negative values suggest incremental or less novel innovations. Please provide a detailed assessment based on the comparison between the focal patent and its related patents. Consider the Disruption Index of the focal patent to guide your analysis, focusing on the unique contributions or advancements it offers. *Details for Analysis*: Determine whether the DINDEX predicted in the previous epoch is high or low: [DINDEX]{Property}[DINDEX] Abstract of Focus Patent: {abstract} Comparison with Related Patent: {reference} Based on the above information, predict the Disruption index of the focal patent.

Table 7: Example of a Structured Prompt for Fine-Tuning and recurrent alignment in Patent Analysis within the RAHA Framework. This table demonstrates how prompts are designed to assess the innovation level of patents using the Disruption Index.