Understand the Implication: Learning to Think for Pragmatic Understanding

Settaluri Lakshmi Sravanthi[*†], **Kishan Maharaj**[*†], **Sravani Gunnu**[†],
Abhijit Mishra[‡], **Pushpak Bhattacharyya**[†]
†Indian Institute of Technology Bombay, Mumbai, India
‡University of Texas at Austin, Texas, United States
{sravanthi.settaluri, kishan.maharaj.iitb, sravi.gunnu}@gmail.com,
abhijitmishra@utexas.edu, pb@cse.iitb.ac.in

Abstract

Pragmatics, the ability to infer meaning beyond literal interpretation, is crucial for social cognition and communication. While LLMs have been benchmarked for their pragmatic understanding, improving their performance remains underexplored. Existing methods rely on annotated labels but overlook the reasoning process humans naturally use to interpret implicit meaning. To bridge this gap, we introduce a novel pragmatic dataset **ImpliedMeaningPreference** that includes *explicit reasoning ('thoughts')* for both correct and incorrect interpretations. Through preference-tuning and supervised fine-tuning, we demonstrate that thought-based learning significantly enhances LLMs' pragmatic understanding, improving accuracy by 11.12% across model families. We further discuss a transfer-learning study where we evaluate the performance of *thought*-based training for the other tasks of pragmatics (presupposition, deixis) that are not seen during the training time and observe an improvement of 16.10% compared to *label* trained models. Code and data are available in the repo [1]

1 Introduction

Human interactions shape relationships through shared understandings, influenced not just by explicit words but by emotional and pragmatic nuances that convey implicit meanings. The ability to interpret beyond the literal meaning of language, known as *pragmatics*, is essential for social cognition, interpersonal awareness, and emotional intelligence. It allows individuals to navigate conversations fluidly, recognising intentions, cultural contexts, and unspoken implications.

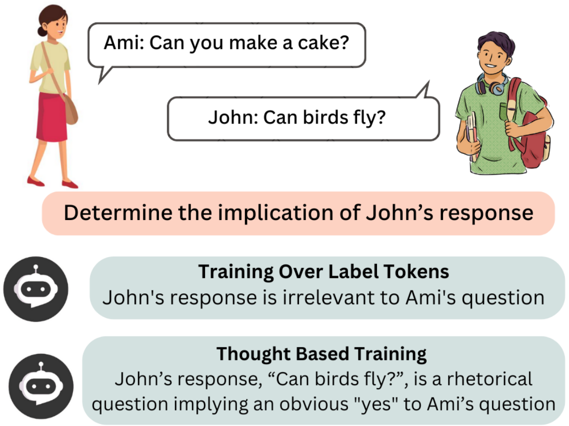
Ami: Can you make a cake?
John: Can birds fly?
Determine the implication of John's response
Training Over Label Tokens
John's response is irrelevant to Ami's question
Thought Based Training
John's response, "Can birds fly?", is a rhetorical question implying an obvious "yes" to Ami's question

Figure 1: An example of implicature in a response which demonstrates the effectiveness of thought-based training by capturing the correct pragmatic meaning

Recent progress in large language models (Brown et al., 2020; Team et al., 2024; Yang et al., 2024a; Achiam et al., 2023; Dubey et al., 2024; Team et al., 2023) has advanced the capabilities of conversational AI. These systems exhibit robust performance in natural language generation, reasoning tasks like math word problems, code generation (Wang et al., 2019; Cobbe et al., 2021; Geva et al., 2021; Clark et al., 2018), etc., largely due to the exploitation of extensive computational resources and vast language datasets. Despite these strengths, current LLMs struggle with effective communication, specifically in capturing the pragmatic and ambiguous dimensions of user inputs. Additionally, conventional training strategies prioritise the production of responses that are safe, objective, and widely acceptable (Glaese et al., 2022). This approach, while ensuring reliability, diverges from the goal of replicating truly human-like conversational behaviour, where the subtleties of context, emotion, and cultural nuance are critical.

While humans naturally engage in pragmatic reasoning, LLMs often struggle with this skill, especially the small LLMs (SLMs) (Amirizaniani et al., 2024), which are often used in practical scenarios due to their lower inference costs, reduced latency, and suitability for local deployment. Given the increased interaction between humans and LLMs, it

[*] Equal Contributions
[1] Code and Datasets

Findings of the Association for Computational Linguistics: ACL 2025, pages 23778–23790
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

is very important for the LLMs to obtain substantial pragmatic understanding of human language and intent. Recent work has primarily focused on evaluating LLMs' pragmatic understanding, yet efforts to enhance their performance on such tasks remain limited (Van Dijk et al., 2023). Approaches that try to improve LLMs in pragmatic reasoning rely on label-based supervision or policy optimisation over annotated datasets (Wu et al., 2024), but these methods do not explicitly incorporate the reasoning process that humans use to grasp implicit meaning. This is mainly due to the absence of training mechanisms which can explicitly incorporate the reasoning process. For instance, as shown in Figure 1, interpreting the response *"Can birds fly?"* as *"Yes"* to the question *"Can you make a cake?"* requires recognising it as a rhetorical question with an obvious affirmative answer—implying that the speaker's answer to the original question is also an obvious *"Yes"*.

To address this gap, we introduce a novel approach that leverages explicit reasoning, or *thoughts*, to improve LLMs' pragmatic comprehension. Specifically, we perform thought-based training for the task of implicature recovery, understanding what is implied in a statement even though it is not literally expressed. We then show generalizability on multiple pragmatics domains, which include implicature, presupposition and reference. Unlike reasoning tasks such as math word problems or coding challenges, pragmatic reasoning often lacks definitive answers, making it more challenging. The correct interpretation in a given scenario is highly influenced by context, culture, and the individuals involved. This interpretation is often not described in the raw training data explicitly and can not be easily captured during the training process. To mitigate this, an explicit intermediate reasoning process must be provided during the training time along with the correct label, which details the intermediate reasoning process, mimicking how humans derive correct interpretation by deliberate system-2 thinking (Weston and Sukhbaatar, 2023). Hence, we present a first-of-its-kind pragmatic dataset where each instance includes a *thought* explaining the reasoning behind the correct label, along with a plausible yet incorrect *negative thought* justifying the incorrect label. We integrate this thought-based data into both preference-tuning and supervised fine-tuning settings, demonstrating an absolute improvement of **11.12%** in accuracy across three model families. Our findings establish the effectiveness of thought-based learning in advancing LLMs' ability to interpret implicit meaning in language. Our contributions are:

- A training framework incorporating explicit reasoning (*thoughts*) [2], leading to an 11.12% improvement in implicature recovery compared to label-based training approaches (Figure 2).
- A transfer learning analysis examining the effects of thought-based supervised fine-tuning (SFT) and direct preference optimisation (DPO) on unseen tasks, showing an improvement of 16.10% over label-based training approaches (Section 7.2).
- Synthetic QA datasets; **Syn-Circa** and **Syn-ludwig**, consisting of ∼33.75K, created by extending CIRCA and LUDWIG to improve understanding of implicit responses (Section 3.2).
- A novel dataset, named **ImpliedMeaning-Preference**, for thought based implicature recovery consisting of ∼*66.2K* instances. This dataset is developed through a human-LLM collaboration integrating multiple implicature recovery datasets (Section 3.1).

2 Related Work

Implicature recovery is a central topic in pragmatics, attracting significant attention from linguists and computational researchers alike. One of the most influential theoretical contributions to this field is the formulation of the *Gricean Maxims* (Grice, 1975), which outline principles governing conversational implicature through Quality, Quantity, Relevance, and Manner.

Various approaches have been proposed to analyse and recover implicatures. For instance, Louis et al. (2020); Ruis et al. (2023) study indirect answers in polar questions, shedding light on how conversational participants infer unstated meanings. Zheng et al. (2021) leverage hierarchical grammar models to interpret both implicatures and deictic references in structured dialogues. Additionally, Jeretic et al. (2020) explores the role of Natural Language Inference (NLI) in understanding scalar implicatures, while Deng et al. (2014) integrate

[2] Here, thoughts do not imply human cognition.

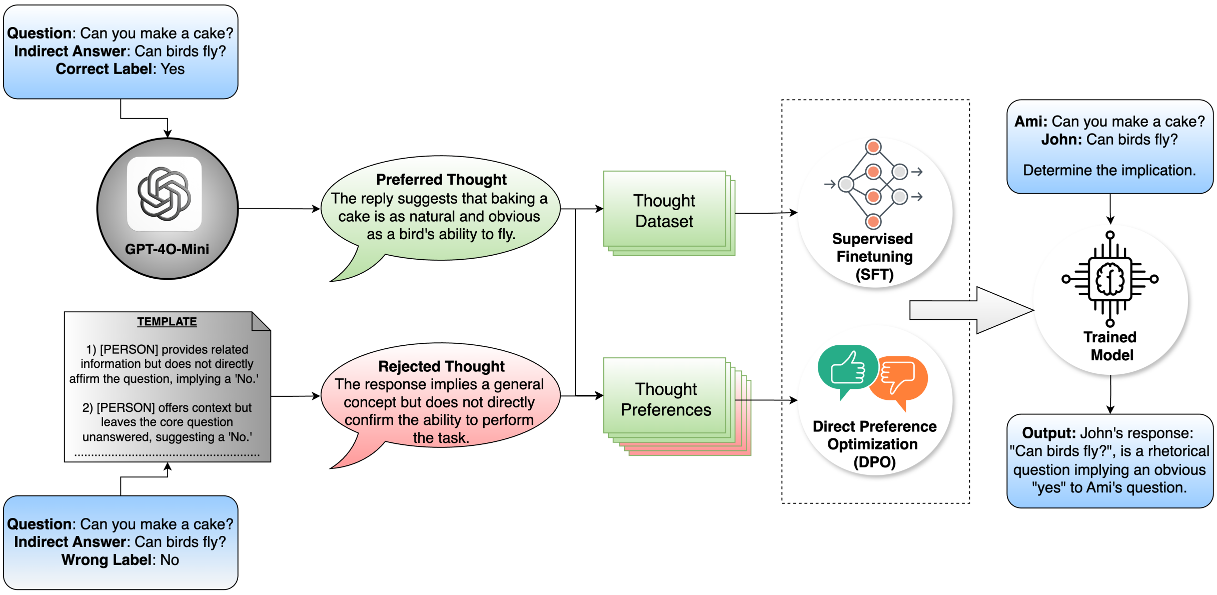
Question: Can you make a cake?
Indirect Answer: Can birds fly?
Correct Label: Yes
GPT-4O-Mini
Preferred Thought
The reply suggests that baking a cake is as natural and obvious as a bird's ability to fly.
Thought Dataset
Supervised Finetuning (SFT)
TEMPLATE
1) [PERSON] provides related information but does not directly affirm the question, implying a 'No.'
2) [PERSON] offers context but leaves the core question unanswered, suggesting a 'No.'
Rejected Thought
The response implies a general concept but does not directly confirm the ability to perform the task.
Thought Preferences
Direct Preference Optimization (DPO)
Question: Can you make a cake?
Indirect Answer: Can birds fly?
Wrong Label: No
Ami: Can you make a cake?
John: Can birds fly?
Determine the implication.
Trained Model
Output: John's response: "Can birds fly?", is a rhetorical question implying an obvious "yes" to Ami's question.

Figure 2: This diagram shows the proposed thought-based training framework with two different training mechanisms: 1) SFT (Supervised Finetuning) and 2) DPO (Direct Preference Optimisation). The left side of the diagram shows the preference data generation steps, and the right side of the diagram shows the training pipeline. We use preferred thought+label for SFT and preference tune with the rejected thought+incorrect label and preferred thought+correct label in DPO.

implicature-based reasoning into sentiment analysis.

Further contributions in this domain include corpus-based studies such as Lahiri (2015), which provide sentence-level annotations for implicature detection. Work by Schuster et al. (2019) and Li et al. (2021) focuses on employing neural networks and linguistic signals to predict scalar inferences, highlighting the potential of machine learning in implicature comprehension. Despite these advancements, recent benchmarking efforts (Hu et al., 2023; Sravanthi et al., 2024) consistently reveal a persistent performance gap between human reasoning and LLM capabilities in pragmatics.

Building upon these findings, Wu et al. (2024) introduces an open-ended evaluation framework to assess LLMs' pragmatic abilities, showing the superiority of preference-based learning over supervised fine-tuning when label-based data is considered. Going forward, our work incorporates the intermediate reasoning steps (thoughts) in fine-tuning and preference optimisation process for pragmatic reasoning. Unlike conventional approaches that reward only label accuracy, our method explicitly incorporates thought processes into model training, enabling LLMs to develop a deeper understanding of pragmatics. In the following sections, we present our datasets, methodology and evaluate the effectiveness of structured reasoning in enhancing LLM performance in pragmatics tasks like implicature recovery, presupposition and deixis.

3 Datasets

In this section, we discuss the process of generating ImpliedMeaningPreference data and synthetic QA datasets (Syn-Circa and Syn-ludwig).

3.1 Preference Data Generation

Gathering high-quality preference data typically requires substantial resources and significant human effort. Existing pragmatic QA datasets, such as Circa and Ludwig, include human-annotated mappings between indirect answers and their corresponding direct interpretations (i.e., labels such as yes and no). To minimise human efforts in preference construction, we leverage these existing label mappings: the original mapped label is treated as the preferred label, while its complement is considered the *rejected label*.

Preferred thought generation: As shown in Figure 2 we generate the *thoughts* supporting the *correct label* by prompting *gpt-4o-mini*. <*Question, Indirect answer, Correct label*> are given as input to the model, and the model is tasked to generate an intermediate reasoning step that helps in mapping the indirect answer to the label.

Rejected thought generation: We attempted to generate rejected thoughts using a similar approach by providing <*Question, Indirect answer, Wrong label*> to the model. However, we observed that most of the time, the model generated thoughts supporting the *correct label*, which may be due to the inherent safety guardrails present in the model (Achiam et al., 2023). Therefore, for rejected thought generation, a linguistic expert is tasked to write templates. The templates are made to capture the wrong reasoning that mimics the misunderstanding humans can have when one or more of the Gricean maxims are flouted. For example, in the Figure 1, the *maxim of relevance* is flouted, and it can be understood as *John* is giving an irrelevant reply to the question asked by *Ami*. Each *rejected thought* is generated by randomly selecting one of the 50 templates written by the linguist. Prompts for the data generation and sample templates are given in the Appendix, Section 7.2. A sample of preferred and rejected thoughts is verified by linguistic experts. Details discussing the human evaluation can be found in Appendix A.

3.2 Synthetic QA Datasets

To enhance our preference dataset, we expanded the existing QA dataset, facilitating the generation of additional preference annotations. We construct our synthetic QA datasets based on existing polar questions and indirect answer datasets (Louis et al., 2020; Ruis et al., 2023). Circa (Louis et al., 2020) and ludwig (Ruis et al., 2023) consist of 3,345 and 601 unique questions. For generating *syn-circa* and *syn-ludwig*, we take unique questions from both the datasets and generate indirect answers that can be mapped to polar direct answers *i.e.*, each indirect answer conveys a "yes" or "no" reply to the question. The responses were generated using *gpt-4o-mini* (Achiam et al., 2023) in a few-shot prompting setting. For each unique question, we generate five answers using five different temperature values -0.0, 0.2, 0.4, 0.6 and 0.8 for generating varied and creative responses. To guide the model effectively, 3 to 6 examples were randomly selected from a curated set of 50 examples, serving as contextual prompts to steer the generation process. The prompts for generation are given in Appendix B. To evaluate that the generated responses adhered to the desired criteria of being indirect, we use a BERT-based classifier (Devlin, 2018) trained on classifying declarative sentences of the questions and indirect answers. Out of 33.75K instance only five examples were classified as indirect answers by the classifier. The effects of this data augmentation using synthetic datasets is discussed in Section 7.3.

4 Methodology

This section discusses our approach in detail. We aim to study the impact of incorporating thought training in two settings: 1) Supervised Fine-Tuning (SFT) and 2) Direct Preference Optimization (DPO) (Rafailov et al., 2023). We formulate the task as follows: Given an input consisting of an indirect answer to a question along with a context, output the pragmatic interpretation. Let $P(x)$ be the initial prompt which contains the task description T_{desc} and input description I_{desc} where $x = [T_{desc}, I_{desc}]$. Let G be the generated output, which contains a thinking process $\mathcal{T}_{thought}$ followed by a predicted label $\mathcal{P}_{label}$. Here, $G = [\mathcal{T}_{thought}; \mathcal{P}_{label}]$ consisting of tokens $(g_1, g_2, \ldots, g_{t-1}, g_t)$.

In the general supervised fine-tuning process, we aim to maximize the conditional log-likelihood of the output tokens given the input tokens. In the context of our setting, this corresponds to:

$$\mathcal{L}_{sft} = -\sum_{t=1}^{|G|} log P_\theta(g_t \mid P(x), g_1, g_2, \ldots, g_{t-1})$$

Here $\mathcal{L}$ is the total loss (negative log-likelihood of the sequence), $|G|$ is the length of the output sequence G, $P_\theta(g_t \mid P(x), g_1, g_2, \ldots, g_{t-1})$ is the model's predicted probability of the token g_t at position t, given the input prompt $P(x)$ and all previous tokens $g_1, g_2, \ldots, g_{t-1}$ and θ is the model parameters being optimized.

Contrary to SFT, in standard Reinforcement Learning from Human Feedback (RLHF) setup, we use the structure of the Markov Decision Process consisting of 4 tuples: (States S, Actions A, Transition Probabilities T_p, Rewards R). Here, we define a function policy π, which maps states to actions ($\pi : S \rightarrow A$). The goal is to optimize the policies to maximize the rewards. In our context, given the current state (Input Prompt), we would like to optimize the policy (language model) to select the actions (which token to predict next) such that the reward function (a function which scores the generated output based on human preferences) yields the maximum value. We aim to study the effects of optimizing policy over the thoughts and labels together.

Dataset Name	Train Set			Val Set			Test Set		
	Yes	No	Total	Yes	No	Total	Yes	No	Total
Circa	11,996	9,310	21,306	2,957	2,251	5,208	1,675	1,272	2,947
Synthetic_Circa	9,955	9,517	19,472	2,502	2,468	4,970	1,322	1,394	2,716
Synthetic_Ludwig	2,401	2,330	4,731	592	608	1,200	327	333	660

Table 1: Class distribution and totals for Train, Validation, and Test datasets.

This means that the probability of winning generation (G_W) preferred by humans should be more than the probability of losing generation (G_L), which humans do not prefer. Therefore, the Bradley Terry Model for our setup is:

$$P(G_W > G_L) = \frac{e^{R(P(x),G_W)}}{e^{R(P(x),G_W)} + e^{R(P(x),G_L)}} \tag{1}$$

This finally yields the adapted DPO loss for our setting, incorporating policy optimization over thought and labels.

Specifically, $L_{DPO}(\pi_\theta; \pi_{ref})$:

$$-\mathbb{E}_{(x,G_W,G_L)\sim\mathbb{D}}[log(\sigma(\beta\psi(G_W) - \beta\psi(G_L))] \tag{2}$$

where

$$\psi(G) = log(\frac{\pi_\theta(G|P(x))}{\pi_{ref}(G|P(x))}) \tag{3}$$

In the above equation, π_{ref} is the reference model instantiated with the initial version of the model, π_θ is the model obtained after preference tuning, and β is the regularizing parameter used for penalizing the scenario when the resulting model is very far from the base version resulting to the loss of prior knowledge.

From a linguistic perspective, our approach is motivated by the need to model pragmatic competence in language understanding. Pragmatic reasoning involves interpreting implied meanings that go beyond the literal content of utterances, as theorized in Grice's maxims of conversation. Traditional models often struggle with implicature resolution because they lack an explicit mechanism for reasoning about contextually inferred meanings. By integrating structured thought processes into both fine-tuning and preference optimization, our method provides a computational analog to human inferential processes in discourse interpretation. This, we hope, should enable LLMs to better grasp implicatures, handle indirect responses, and align with human-like conversational norms, thereby improving their effectiveness in pragmatic language tasks.

5 Experimental Setup

For our experiments, we consider models from three different families: 1) Llama-3.2-1B (Dubey et al., 2024) 2) Qwen-2.5-1.5B (Yang et al., 2024b), 3) Gemma-2-2B (Team et al., 2024). We also report the zero-shot performance of Llama3.1 70B for comparison with a large language model. Our experiments kept the learning rate at $5e-7$ with warmup steps of 500 iterations. We use RMSprop (Ruder, 2016) as our optimiser following Wu et al. (2024). All models in both settings are trained for one epoch (till convergence), and the greedy decoding mechanism was used throughout the experiments. For Qwen-2.5-1.5B and Gemma-2-2B, we use the global batch size of 32; for llama-3.2-1B, the global batch size was set to 64. For regularization, we use gradient clipping of 1 in DPO and weight decay of 0.01 in SFT. We use 4 NVIDIA H100 80GB HBM3 GPUs for all the experiments in this work, with a total train time of 8 GPU hours. For all other hyper-parameters, we use the default values. We report all the training and evaluation prompts in Appendix, Sections C and D respectively. We use macro precision (P), recall (R) and F1 scores for evaluation.

6 Results

In Table 2, we report the results of our experiments after training with QA datasets. We note significant improvement after the inclusion of the thoughts in SFT and DPO for Llama-3.2-1B and Qwen2.5-1.5B. For Gemma2-2B, we observe significant gains in SFT with thought settings and a slight performance decline when thought is incorporated in the DPO setting. We note that DPO was not originally used in the training process of Gemma-2B, unlike Llama-3.2-1B and Qwen2.5-1.5B. We conjecture that since the model was not exposed to DPO in the general training, our training for implicature recovery could not induce the thoughts as effectively as in other models.

We note that training with just labels provides an edge to DPO over SFT across models, aligning with Wu et al. (2024). While training with thought alongside labels provides significantly higher gains for SFT when compared to DPO, with thought-based SFT outperforming thought-based DPO in most cases. The 'thoughts' contain more explicit signals and the interpretation of the reasoning required to reach the right answer, which may be captured in a more straightforward way in the SFT setup compared to DPO. Intuitively, the thought-based training mechanism would require higher updates in the parameters than the scenario when we have to optimise over only label tokens. In general, the optimisation objective for SFT does not have any constraints and is more flexible compared to DPO, which requires a regularising parameter β for the KL constraint to prevent divergence from the base model (untrained) during the training.

Another perspective in the context of this observation was suggested by Feng et al., 2024; Pal et al., 2024, which shows the gradient of the DPO loss with respect to preferred (winning) response is lower compared to the dispreferred (losing) response which essentially hinders the learning capacity of LLMs to generate the actual human preferences while introducing the tendency of avoiding human dispreferred responses. This effect may have been further magnified in our setting which has more tokens compared to the only label setting.

We also note that our best-performing model, Gemma2-2B, supervised-fine-tuned with thoughts, yields comparable performance to LLama3.1-70B, which highlights the effectiveness of incorporating thoughts in the training mechanism. In general, the thought-based training mechanism yielded better results compared to the setting, which just incorporates labels, highlighting the importance of learning thought generation.

7 Analysis

In this section, we discuss various insights about the proposed method, which describes the advantages of thought-based learning and some general error cases.

7.1 Predictive Analysis

Here, we describe the general predictive trends observed in our framework. In general, we observe a significant improvement after incorporating thought in the generated output. Intuitively, the causal models are optimized to generate the appropriate explanations first and then derive the predictions based on the generated explanation. Probabilistically, the next token for prediction is conditioned on the 'thought' and 'input' tokens which can act as a guide for reaching the correct predictions more accurately compared to the scenario when only input tokens are considered. We discuss an example in Illustration 1 where the task is to determine if the given response to a question implies a “Yes" or 'No". We observe that the model is correctly predicting the output of 'Yes' by generating thought which are used for resolving the final predictions compared to the scenario when the model trained on just labels (without thought) is considered. The generated thought is also helpful in understanding if the model is genuinely predicting the correct output based on the right understanding or predicting the correct output randomly (further explanations in Section 7.5).

Given Input: Determine whether a given response to a question implies a "Yes" or "No"
Context: Y has just told X that he/she is thinking of buying a flat in New York.
Question: Is it crowded and busy in that area?
Response: It never sleeps.

With Thought Prediction: The response "It never sleeps" suggests that the area is always active and bustling with people, implying high population density and activity levels. This strongly indicates that the area would indeed be crowded and busy.
Prediction: Yes
Without Thought Prediction: No
Correct Label: Yes

Illustration 1: General Prediction for QA

7.2 Transfer Learning Analysis

This section discusses whether thought learning is transferable to the other datasets and tasks which are not seen during the training process. The primary motivation behind this study is to understand if the thought training done for one of the pragmatic tasks is helpful in learning other pragmatic tasks in different datasets.

Specifically, we evaluate our models trained for implied question answering with the following datasets: 1) FigQA (figurative Natural Language Inference) 2) Flute (figurative Natural Language Inference) 3) IMPPRES (figurative Natural Language Inference) 4) Ludwig 5) *Pub-presupposition*

Setting	Circa			Synthetic_Circa			Synthetic_Ludwig			Overall F1
	P	R	F1	P	R	F1	P	R	F1	
Llama-3.2 1B										
Zero-Shot	73.52	60.67	50.83	79.35	62.90	57.49	79.38	65.60	61.24	56.52
SFT	68.28	55.57	42.72	72.53	53.84	42.39	64.68	56.32	49.27	44.79
SFT+Thought	81.32	81.87	81.40 (+38.68)	86.37	86.39	86.37 (+43.98)	82.01	79.42	79.09 (+29.82)	82.29 (+37.5)
DPO	81.17	60.22	55.37	77.67	61.73	54.62	94.14	93.83	93.78	67.92
DPO+Thought	75.39	73.81	74.16 (+18.79)	81.77	80.95	80.62 (+26)	89.01	88.82	88.78 (-5)	81.85 (+13.93)
Qwen-2.5-1.5B										
Zero-Shot	75.15	63.15	54.55	83.51	75.78	74.78	86.73	81.96	81.49	70.94
SFT	74.39	62.53	53.76	85.04	79.23	78.75	87.00	82.11	81.63	71.38
SFT+Thought	76.85	70.13	64.99 (+11.23)	87.90	84.30	84.24 (+5.49)	88.72	86.10	85.96 (+4.33)	78.40 (+7.02)
DPO	76.96	72.49	68.50	88.10	87.23	87.31	88.77	86.72	86.63	80.81
DPO+Thought	77.77	74.93	71.80 (+3.3)	90.13	89.02	89.11 (+1.8)	88.22	87.52	87.51 (+0.88)	82.14 (+1.33)
Gemma2-2B										
Zero-Shot	77.52	71.94	67.46	79.18	67.82	64.91	83.93	77.38	76.39	69.58
SFT	83.25	81.60	79.15	93.42	92.26	92.38	84.75	78.14	77.20	82.24
SFT+Thought	90.69	91.10	**90.85** (+11.7)	95.61	95.63	**95.58** (+3.2)	93.48	93.49	**93.48** (+16.28)	**93.30** (+11.06)
DPO	87.55	82.88	83.77	89.45	87.61	87.18	94.57	94.54	94.54	88.50
DPO+Thought	87.48	80.63	81.54 (-2.33)	84.75	80.00	78.87 (-8.31)	92.59	92.17	92.10 (-2.44)	84.17 (-4.33)
Llama3.1-70B										
Zero-Shot	94.37	93.88	94.09	93.61	93.63	93.59	91.79	91.69	91.66	93.11

Table 2: Comparison of P (Precision), R (Recall), and F1 scores across Circa, Synthetic_Circa, and Synthetic_Ludwig datasets under various settings for QA dataset. The last column reports the mean F1 score across datasets.

Setting	Circa (%)			Synthetic_Circa (%)			Synthetic_Ludwig (%)			Mean F1 (%)
	P	R	F1	P	R	F1	P	R	F1	
DPO	71.43	67.05	66.91	78.93	78.56	78.34	73.58	72.96	72.72	72.66
DPO+Thought	74.42	53.36	43.61	71.10	52.68	38.83	75.27	51.95	37.33	39.92
SFT	17.13	16.63	16.86	12.40	12.44	12.40	31.30	31.40	31.29	20.18
SFT+Thought	80.12	79.21	77.12	87.33	86.00	86.07	89.45	89.41	89.39	84.19

Table 3: **Data Ablation on Gemma-2B**: We report the Precision (P), Recall (R) and F1 scores on all four settings by training the model with just the Circa dataset.

task 6) *Pub-reference* task.

Presupposition, implicature, and reference are pragmatic phenomena that rely on context, shared background knowledge, and the interactive nature of communication to convey meaning beyond the literal content of an utterance. Intuitively, models trained with explicit reasoning for performing implicature recovery should also perform better on these related linguistic phenomena. Specially, reference is a special case of implicature where the only difference is the usage of *deixis* terms.

For these experiments, we chose our best-performing model, Gemma2-2B. We observe significant improvement in performance when the thought is incorporated into the training mechanism for both the SFT and DPO. We report these results in Table 4 for three NLI datasets and Table 5 for other pragmatics tasks with a mean improvement of 16.10%.

We observe significant improvements across all the datasets with thought-based training approaches when compared to the label-based training approaches. This highlights that the learning for implicature recovery is also transferable to other datasets and pragmatic tasks.

Given Input:
Does Sentence 2 entail or contradict Sentence 1?
Sentence 1: The girl moved as fast as a turtle
Sentence 2: The girl was quick

With Thought Prediction: Sentence 1 states that the girl moved at a speed similar to a turtle (slow), while Sentence 2 indicates that she was quick. These descriptions contradict each other.
Prediction: Contradiction
Without Thought Prediction: Entailment
Correct Label: Contradiction

Illustration 2: General Prediction for NLI

In the Illustration 2, we describe a general scenario where the model is able to resolve the figurative language of *as fast as a turtle* to *slow*, finally arriving at the correct prediction of Contradiction.

Setting	**FigQA (%)**			**Flute (%)**			**IMPPRES (%)**			**Mean F1 (%)**
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	
Zero-Shot	62.14	62.13	62.13	77.77	77.77	75.70	43.16	37.62	39.19	59.00
DPO	61.77	61.69	61.63	76.86	75.55	72.56	43.22	37.55	39.03	57.74
DPO + Thought	63.21	62.98	62.80	72.20	72.74	71.62	44.47	41.06	42.08	58.83 (+1.09)
SFT	59.59	58.99	58.35	76.28	73.35	69.40	49.37	48.75	44.27	57.34
SFT + Thought	**64.09**	**63.85**	**63.69**	**78.13**	**78.30**	**76.36**	**49.72**	**49.48**	**46.72**	**62.25** (+4.91)

Table 4: **Transfer Learning for NLI** on figurative sentences: FigQA, Flute, and IMPPRES.

Setting	**Ludwig (%)**			**Presupposition (%)**			**Reference (%)**			**Mean F1 (%)**
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	
Zero-Shot	70.92	65.21	61.42	52.10	50.55	17.41	11.31	26.33	15.82	31.55
DPO	73.35	70.29	69.99	53.89	53.19	26.02	22.58	35.62	22.87	39.62
DPO+Thought	74.07	72.71	72.75	**53.36**	**55.77**	**51.51**	68.41	62.64	62.88	62.38 (+22.76)
SFT	61.68	60.96	59.73	52.47	50.92	19.02	3.27	6.30	4.31	27.68
SFT+Thought	**76.42**	**76.53**	**76.40**	54.23	57.64	44.27	**69.08**	**69.77**	**69.34**	**63.33** (+35.65)

Table 5: **Transfer Learning** for the Presupposition, Ludwig and Reference dataset

7.3 Data Ablations

In this section, we discuss the effects of introducing our synthetically created data in the training. Specifically, we perform the experiments without training with any synthetic data and use the Circa data (similar to the original setting) to train our best-performing model, Gemma, in all four settings. We report the results in Table 3 and note that the performances across all settings are significantly lower compared to the original training setup (as reported in Table 2), where we also include the synthetic data. We observe a reduction of 9.11% in the SFT+Thought setting and 15.84% in the DPO setting, respectively. A drastic reduction is observed in the DPO+Thought and SFT settings, primarily due to the poor legibility of model predictions where the generated text has very high hallucination. This shows the utility of synthetic data, highlighting its role in enhancing the robustness of training mechanisms by providing diverse and well-aligned examples that may be difficult to cover otherwise.

7.4 Error Analysis

This section describes the most prominent cases of error in the given task. We observe that the majority of errors occur when there is a use of a complex linguistic phenomenon that requires an additional layer of interpretation.

In the Illustration 3, we describe one such scenario. Specifically, sentence 2 shows the presence of sarcasm, which is evident in the latter part, which uses the adjective ‘lovely’ for work. The model fails to understand that the speaker is being sarcastic about the situation, and the implication is similar to sentence 1, which leads to an incorrect prediction of ‘Contradiction’. Similar predictive behaviour is observed when other special linguistic phenomena like metaphors and hyperbole are present.

Given Input:
Does Sentence 2 entail or contradict Sentence 1?
Sentence 1: I was going to have dinner with friends today but I was stuck at work.
Sentence 2: I was so looking forward to dinner with friends today, but unfortunately, I got stuck at work.

With Thought Prediction: The first sentence states that the person had plans for dinner with friends but couldn’t go due to being stuck at work. The second sentence also mentions having dinner with friends but getting stuck at work, indicating similar circumstances. However, the first sentence uses "stuck," while the second uses "lovely."
Prediction: Contradiction
Correct Label: Entailment

Illustration 3: Misinterpretation of sarcasm

7.5 Thought Analysis

In this section, we discuss various insights related to the thought generations. The primary aim of this study is to understand if the model is deriving the predictions from the correct thought or getting the predictions right with the incorrect logic. To analyze this quantitatively, we consider our best-performing model, Gemma2-2B-SFT, for evaluating thoughts using GPT4O-mini. To perform a human evaluation to ensure the quality of GPT4omini predictions, we asked two linguistic experts to annotate if they agreed with the predictions, leading to a Cohen kappa score of 0.79 on 85 examples randomly sampled from the data.

We observe that in the cases of correct predictions, the model generates correct thoughts in 96.41 % of the instances and 3.59 % of incorrect thoughts. For incorrect predictions, we observe 70.92% instances with wrong thoughts while 29.08% instances have the correct thoughts where the model generates the correct reasoning but could not resolve the correct predictions. We discuss an example in Illustration 4 pertaining to the scenario where the model is correctly generating the contrast in both the sentences but provides prediction as 'Entailment' instead of 'Contradiction'.

In general, we observe that most of the correct predictions have correct thoughts, and a significant amount of incorrect predictions also have correct thoughts. This highlights that the model is able to generate thoughts to a reasonable extent but cannot cut the threshold of reaching the correct answer.

Given Input: Determine whether a given response to a question implies a "Yes" or "No".
Context: X and Y are colleagues leaving work on a Friday at the same time.
Question: Do you have any big plans for the weekend?
Response: I have a number of things to do this weekend

With Thought Prediction: The response indicates that the person has several activities planned for the weekend, suggesting they likely have some significant commitments.
Prediction: No
Correct Label: Yes

Illustration 4: Correct Thought but Incorrect Prediction

7.6 Thought Perturbation Analysis

In this section, we describe our experiments, which aim to understand the if the improvements are observed due to the presence of correct thoughts leading to the right label in the training data or is it just some spurious correlation. For this experiment, we perturb the correct thought with the incorrect thought: for SFT, we replace the correct thought with the incorrect thought and for DPO, we flip the correct (preferred) and incorrect (dispreferred) thoughts in preference data. In general, we observe a significant decrement in the scores compared to the original setting where we consider the correct thought. The decrease in the SFT is very drastic, and the F1-scores went down to as low as 2%. In the DPO, we also see a considerable decline in the performance ($25\% - 30\%$) across all QA tasks compared to the DPO+Thought settings. Even though, in both cases, there is a decrease in the performance, DPO models did not suffer a tragic decline in the accuracies due to the presence of the regularizing constant (β) in DPO. In other words, the regularizing constant β prevents the large updates in the model, which is not the case in SFT, where the weight updates are unconstrained.

8 Conclusion and Future Work

In this work, we highlighted the effectiveness of integrating explicit thought processes into two training paradigms: 1) Supervised Fine-Tuning (SFT) and 2) Direct Preference Optimization (DPO). Our findings indicate that while thought integration benefits both training approaches, thought-based SFT consistently outperforms its DPO counterpart in pragmatic reasoning tasks. Through a detailed analysis of model predictions, we uncover key patterns in implicature resolution and identify specific failure cases that illuminate areas for further improvement. To reinforce the role of structured reasoning, we investigate the impact of perturbing thought generation, revealing a notable decline in performance when the reasoning process is disrupted. Furthermore, our transfer learning experiments demonstrate the adaptability of thought-based training, showing its efficacy in generalizing across previously unseen datasets and pragmatic tasks. In the future, our goal is to refine this approach by developing a process-based reward mechanism that better aligns LLMs with human pragmatic inference, ultimately bridging the gap between computational and human-like language understanding.

9 Limitations

While our approach improves implicature recovery, it also presents several limitations. The reliance on explicit thought generation may introduce biases, as the quality and accuracy of generated thoughts depend on both the model's prior knowledge and human annotations. Additionally, the increased computational complexity associated with training thought-based models may limit scalability in resource-constrained settings. Further, while transfer learning results are promising, the generalizability of our approach across diverse linguistic domains, including highly contextual or culturally specific implicatures, remains an open question. In Future, we plan to explore more efficient training strategies and broader evaluation frameworks to enhance the robustness and applicability of thought-based learning.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Maryam Amirizaniani, Elias Martin, Maryna Sivachenko, Afra Mashhadi, and Chirag Shah. 2024. Can llms reason like humans? assessing theory of mind reasoning in llms for open-ended questions. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 34–44.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv*, abs/1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168.

Lingjia Deng, Janyce Wiebe, and Yoonjung Choi. 2014. Joint inference and disambiguation of implicit sentiments via implicature constraints. In *COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, August 23-29, 2014, Dublin, Ireland*, pages 79–88. ACL.

Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Duanyu Feng, Bowen Qin, Chen Huang, Zheng Zhang, and Wenqiang Lei. 2024. Towards analyzing and understanding the limitations of dpo: A theoretical perspective. *arXiv preprint arXiv:2404.04626*.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? A question answering benchmark with implicit reasoning strategies. *Trans. Assoc. Comput. Linguistics*, 9:346–361.

Amelia Glaese, Nat McAleese, Maja Trębacz, John Aslanides, Vlad Firoiu, Timo Ewalds, Maribeth Rauh, Laura Weidinger, Martin Chadwick, Phoebe Thacker, et al. 2022. Improving alignment of dialogue agents via targeted human judgements. *arXiv preprint arXiv:2209.14375*.

Herbert P Grice. 1975. Logic and conversation. In *Speech acts*, pages 41–58. Brill.

Jennifer Hu, Sammy Floyd, Olessia Jouravlev, Evelina Fedorenko, and Edward Gibson. 2023. A fine-grained comparison of pragmatic language understanding in humans and language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 4194–4213. Association for Computational Linguistics.

Paloma Jeretic, Alex Warstadt, Suvrat Bhooshan, and Adina Williams. 2020. Are natural language inference models imppressive? learning implicature and presupposition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8690–8705. Association for Computational Linguistics.

Shibamouli Lahiri. 2015. Squinky! A corpus of sentence-level formality, informativeness, and implicature. *CoRR*, abs/1506.02306.

Elissa Li, Sebastian Schuster, and Judith Degen. 2021. Predicting scalar inferences from “or” to “not both” using neural sentence encoders. In *Proceedings of the Society for Computation in Linguistics 2021*, pages 446–450.

Annie Louis, Dan Roth, and Filip Radlinski. 2020. " i'd rather just go to bed": Understanding indirect answers. *arXiv preprint arXiv:2010.03450*.

Arka Pal, Deep Karkhanis, Samuel Dooley, Manley Roberts, Siddartha Naidu, and Colin White. 2024. Smaug: Fixing failure modes of preference optimisation with dpo-positive. *arXiv preprint arXiv:2402.13228*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn.

2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741.

Sebastian Ruder. 2016. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.

Laura Eline Ruis, Akbir Khan, Stella Biderman, Sara Hooker, Tim Rocktäschel, and Edward Grefenstette. 2023. The goldilocks of pragmatic understanding: Fine-tuning strategy matters for implicature resolution by llms. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Sebastian Schuster, Yuxing Chen, and Judith Degen. 2019. Harnessing the linguistic signal to predict scalar inferences. *arXiv preprint arXiv:1910.14254*.

Settaluri Lakshmi Sravanthi, Meet Doshi, Tankala Pavan Kalyan, Rudra Murthy, Pushpak Bhattacharyya, and Raj Dabre. 2024. Pub: A pragmatics understanding benchmark for assessing llms' pragmatics capabilities. *arXiv preprint arXiv:2401.07078*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Bram Van Dijk, Tom Kouwenhoven, Marco R Spruit, and Max J van Duijn. 2023. Large language models: The need for nuance in current debates and a pragmatic perspective on understanding. *arXiv preprint arXiv:2310.19671*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 3261–3275.

Jason Weston and Sainbayar Sukhbaatar. 2023. System 2 attention (is something you might need too). *arXiv preprint arXiv:2311.11829*.

Shengguang Wu, Shusheng Yang, Zhenglun Chen, and Qi Su. 2024. Rethinking pragmatics in large language models: Towards open-ended evaluation and preference tuning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22583–22599.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2024a. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024b. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Zilong Zheng, Shuwen Qiu, Lifeng Fan, Yixin Zhu, and Song-Chun Zhu. 2021. GRICE: A grammar-based dataset for recovering implicature and conversational reasoning. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 2074–2085. Association for Computational Linguistics.

A Evaluation of GPT-Generated Thoughts

To evaluate the quality of the generated thoughts, we sampled 500 data points, including 200 from the CIRCA dataset, 150 from a synthetic CIRCA dataset, and 150 from a synthetic LUDWIG dataset. These data points were annotated by an external annotator and one of the authors of this paper. The evaluation focused on assessing the alignment between the correct label and its corresponding generated thought (Correct Label Thoughts - CLT), as well as the wrong label and its corresponding generated thought (Wrong Label Thoughts - WLT). Additionally, the confidence of alignment was rated on a three-point scale: 1 (Poor), 2 (Average), and 3 (Good).

The results indicate that the correct label and its corresponding thought aligned 99% of the time, and similarly, the wrong label and its corresponding thought also aligned 99% of the time. Furthermore, both annotators agreed on the alignment 99% of the time and showed 97% agreement in confidence ratings. The external annotator was compensated at a standard rate for the annotation task. Table 6 presents the agreement percentages between the two annotators for alignment and confidence evaluations.

Evaluation Metric	Agreement Percentage
CLT Alignment	99.80%
CLT Confidence	97.24%
WLT Alignment	100.00%
WLT Confidence	98.82%

Table 6: Agreement percentages between the two annotators for alignment and confidence evaluations.

B Prompts and Templates Used for Generating Thoughts

In this section, we provide the prompts used for generating thoughts corresponding to the correct labels using GPT-4o mini, as well as the structured templates employed for generating thoughts related to the wrong labels.

Prompt: Generate a one line rational to support [label] label to the answer-Y:
context : [context]
question-X : [question]
answer-Y : [answer]

Prompt For Generating Correct Label Thoughts

Example Templates for label “Yes”:
1) Y’s response directly states their interest or intent, making the affirmative answer obvious.
2) Y explicitly agrees with the question, providing an unambiguous ’yes.’
3) The reply given by Y is positive and directly answers the question, ensuring clarity in the response.

Example Templates for label “No”:
1) Y’s response touches on related information but does not directly affirm the question, suggesting the answer may be ’no.’
2) While Y provides some context, the core question remains unanswered, implying that the response could be interpreted as a ’no.’
3) Y offers additional information but avoids directly addressing the question, indicating an implicit negative response.

Templates for Generating Wrong Label Thoughts

C Prompts Used for Training and Evaluation

In this section, we provide the prompts used for training the models using Supervised Fine-Tuning (SFT) with label tokens and SFT with thoughts. We first present a generic prompt that serves as a foundational structure for training. This prompt is then adapted to fit the specific chat templates of each model, ensuring compatibility with their respective architectures and tokenization formats. The modifications may include adjustments in prompt wording, system instructions, formatting, or token placement to optimize the model’s performance across different setups.

Prompt for Training Data With Context :

QA Context Input:
You are reasoning driven assistant.
Given the following **context, question** and **response**, you task is to determine whether the response to a question implies a "Yes" or "No." Focus on the meaning implied in the response.
Pretext : [pretext]
Question: Does the response imply a "Yes" or a "No"? Do not output anything other than "Yes" or "No".

QA context Output:
prediction: [label]

Prompt for Training Data Without Context:

QA Input:
You are reasoning driven assistant.
Given the following **question** and a **response**, you task is to determine whether the response to a question implies a "Yes" or "No". Focus on the meaning implied in the response.
Pretext : [pretext]
Question: Does the response imply a "Yes" or a "No"? Do not output anything other than "Yes" or "No".

QA Output:
prediction: [label]

Prompt for Training QA Data with Label Tokens

Prompt for Evaluating NLI :

NLI Input:
You are reasoning driven assistant.
Your task is to analyze the relationship between two sentences by first providing an explanation. Use the explanation to derive the prediction, which can be either "Entailment" or "Contradiction".
Pretext : [pretext]
Question: Analyze the relationship between the two sentences below and provide an explanation of your reasoning process. Derive the final prediction based on your explanation and give it in the below format:

NLI Output:
Explanation: [rationale],
prediction: [label]

Prompt for evaluating NLI tasks

Similarly, for the Direct Preference Optimization (DPO) task, we incorporate both the correct label thought and the wrong label thought during training. By including both perspectives, the model learns to distinguish between well-reasoned cor-

Prompt for Training Data to Generate Thought :

QA Context Thought Input:
You are reasoning driven assistant.
Your task is to analyze whether a given response to a question implies a "Yes" or "No" by providing a one-line thought. The thought should focus on the reasoning process and should not include the final prediction.
Input: [pretext]
Task: Analyze the given context, question, and response. Provide a one-line reasoning thought without deriving the final prediction. Use the format below:
Thought: " "

QA context Thought Output:
Thought: [rationale]

Prompt for Training Data to Generate both Thought and Label

QA Context Thought Input:
You are reasoning driven assistant.
Your task is to determine whether a given response to a question implies a "Yes" or "No" by first providing a one-line explanation. Use the explanation to derive the prediction, which can be either "Yes" or "No” **Input**: [pretext]
Task: Analyze the given context, question, and response. Provide a one-line of your reasoning process. Use the explanation to derive the prediction: "Yes" or "No" and give in the below format:
Explanation: ""
Prediction: ""

QA Context Thought Output:
Explanation: [rationale],
Prediction: [label]

Prompt for Training QA Data with Thoughts

rect responses and incorrect alternatives, thereby improving its ability to align with human preferences. This approach enhances the model’s reasoning capabilities, ensuring that it not only recognizes correct answers but also understands why certain responses are less appropriate.

D Prompts Used for Evaluating Model Generated Thoughts

In this section, we provide the prompts used for evaluating the thoughts generated by models which are trained using a thought-based training mechanism.

Prompt: Your task is to verify whether the given sentences follow the ground truth. Only output yes or no.
Ground Truth: [Input]
Reasoning: [Correct Thought]
Given Sentences: [model generated thought]

Prompt For Evaluating Model Generated Thoughts