Maieutic Prompting: Logically Consistent Reasoning with Recursive Explanations

Jaehun Jung[†] **Lianhui Qin**[†] **Sean Welleck**[†‡]
Faeze Brahman[†‡] **Chandra Bhagavatula**[‡] **Ronan Le Bras** [‡] **Yejin Choi** [†‡]
[†]Paul G. Allen School of Computer Science & Engineering, University of Washington
[‡]Allen Institute for Artificial Intelligence
hoony123@cs.washington.edu

Abstract

Pre-trained language models (LMs) struggle with consistent reasoning; recently, prompting LMs to generate explanations that self-guide the inference has emerged as a promising direction to amend this. However, these approaches are fundamentally bounded by the correctness of explanations, which themselves are often noisy and inconsistent. In this work, we develop MAIEUTIC PROMPTING, which aims to infer a correct answer to a question even from the unreliable generations of LM. MAIEUTIC PROMPTING induces a tree of explanations *abductively* (e.g. *X is true, because* ...) and *recursively*, then frames the inference as a satisfiability problem over these explanations and their logical relations. We test MAIEUTIC PROMPTING for true/false QA on three challenging benchmarks that require complex commonsense reasoning. MAIEUTIC PROMPTING achieves up to 20% better accuracy than state-of-the-art prompting methods, and as a fully unsupervised approach, performs competitively with supervised models. We also show that MAIEUTIC PROMPTING improves robustness in inference while providing interpretable rationales.[1]

1 Introduction

Following the remarkable success of few-shot prompting over large language models (e.g. Brown et al., 2020), recent studies on prompting methods suggest that LMs' reasoning capability can be further promoted by generating a sequence of explanation for a given problem, prior to inferring the answer (Wei et al., 2022; Wang et al., 2022; Liu et al., 2021). The so-called *explanation-based prompting* helps an LM better elicit its knowledge and reason by leveraging its own generated explanations - whether it be commonsense knowledge (Liu et al., 2021), a solution for a math word problem (Wei et al., 2022), or the intermediate steps of program execution (Nye et al., 2021a).

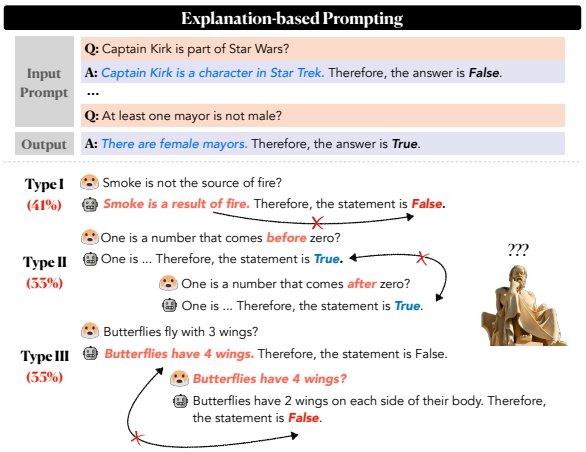


Figure 1: Logical errors in explanation-based prompting: (1) explanation does not logically lead to the answer, (2) model is invariant to negation, and (3) falsifies its own explanation. We prompt 175B GPT-3 with 100 questions sampled from Talmor et al. (2021).

Explanation-based prompting is intuitively motivated by the reasoning steps humans typically employ to solve a problem (Hausmann and VanLehn, 2007). However, we find that this intuition is faulty in practice, as model-generated explanations are often logically inconsistent and unreliable. For example, we manually inspected 100 samples from a QA task (Figure 1) and found that for a considerable number of cases, (1) the explanation does not logically lead to the inferred answer, (2) the model infers the same label for a statement and its negation (Kassner and Schütze, 2020), and (3) falsifies its own generated explanation. These findings raise fundamental questions on the role of explanations in LM inference: If the explanation is correct - is there a guarantee that the LM will infer a label consistent with the explanation? And if the explanation is wrong - is there a way to make use of even the wrong explanation in inferring the correct answer?

To this end, we propose MAIEUTIC PROMPT-

[1]We share our code at https://github.com/jaehunjung1/Maieutic-Prompting.

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 1266 - 1279
December 7-11, 2022

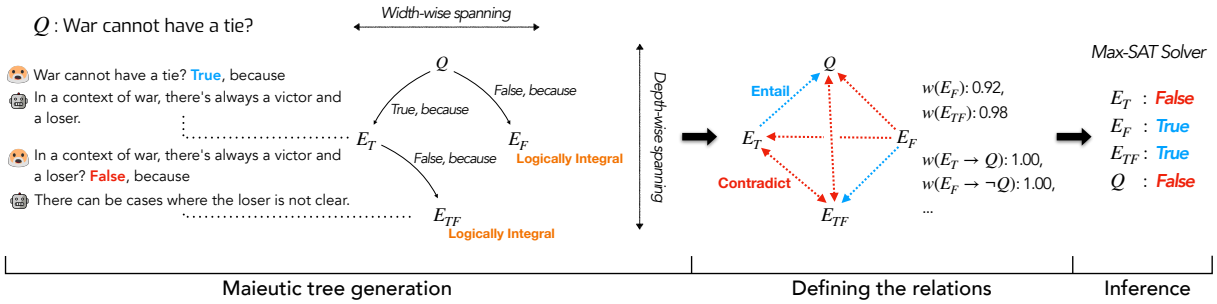


Figure 2: An overview of MAIEUTIC PROMPTING. Given a question Q, we generate *maieutic tree* consisting of abductive and recursive explanations, define the relations between them, and employ MAX-SAT to find the best truth-value assignments to the explanations and Q.

ING, a novel few-shot inference method that infers a correct answer by enumerating a structure of explanations — possibly noisy and contradictory — and resolving them with a symbolic inference algorithm. Inspired by the maieutic method[2] of Socrates, MAIEUTIC PROMPTING induces the LM to generate abductive explanations for diverse hypotheses with deep recursive reasoning, then collectively eliminates the contradicting candidates, resulting in consistent answers.

Figure 2 shows the overview of MAIEUTIC PROMPTING. First, we prompt the LM to *abductively* (Peirce, 1974) rationalize both possible answers, *True* and *False*, rather than generating a single explanation and then connecting it to one of the answer choices. Moreover, we do not expect the 1-hop explanations to be always correct; thus, we further validate the LM's confidence in its explanations by *recursively* prompting the model with its own generation as the question. Our generation process derives a *tree structure* of generated propositions, where one proposition establishes a logical ground for the correctness of one another.

To infer the answer for the original question, we quantify the strength of the LM's *belief* in each proposition and the *logical relationships* between propositions in the maieutic tree. We then employ the weighted MAX-SAT (Battiti, 2009) solver to *collectively infer* the truth-values of all the propositions (including the original question) that best satisfy the set of observed relations. This way, we symbolically induce the subset of generations that makes the most probable and consistent inference. Our proposed method can run completely unsupervised with any few-shot promptable LM (e.g., GPT-3; Brown et al., 2020).

[2] *Maieutic method brings out definitions implicit in the interlocutor's beliefs, ... is a method of hypothesis elimination, steadily identifying and eliminating those that lead to contradictions* (Vlastos, 1991).

Our experiments show that the performance of MAIEUTIC PROMPTING exceeds that of all the few-shot prompting baselines (e.g., Chain of Thought; Wei et al., 2022) in three commonsense reasoning and fact verification benchmarks. MAIEUTIC PROMPTING performs up to 20% better than other prompting methods, and performs on par or even better than supervised models. Further analyses show that MAIEUTIC PROMPTING is robust to perturbations in both the questions and prompts, and offers an interpretable interface to understand the rationale behind the model's inference.

2 Problem Setup and Background

Our goal is to infer whether a given statement Q makes sense, i.e. inferring the truth value A of Q. Conventionally, this can be done through *prompting* an LM with the following two methods:

Standard Prompting Let Q be a statement we want to infer the truth value of (i.e., either *True* or *False*). In standard few-shot prompting, the model-inferred answer $\hat{A}$ is defined as:

$$\hat{A} = \underset{A \in \{T,F\}}{\text{argmax}}\, p_{LM}(A|Q,C), \quad (1)$$

where $C = \{(q_1, a_1), \cdots, (q_k, a_k)\}$ denotes the k examples for in-context learning.

Explanation-based Prompting In explanation-based prompting, the inference process is factorized into two steps:

$$\hat{A} = \underset{A \in \{T,F\}}{\text{argmax}} \int_E p_{LM}(A|Q,E,C)\, p_{LM}(E|Q,C) \quad (2)$$

Here, E denotes the explanation generated prior to inferring the answer label, and $C = \{(q_1, e_1, a_1), \cdots, (q_k, e_k, a_k)\}$ includes k examples of questions, explanations and answers. Since marginalizing over all E is intractable, prior works

resort to a sampling based approximation:

$$\hat{A} = \underset{A \in \{T,F\}}{\text{argmax}}\, p_{LM}(A|Q,E,C), \quad \text{where } E \sim p_{LM}(E|Q,C) \tag{3}$$

3 Maieutic Prompting

In this section, we introduce MAIEUTIC PROMPTING, which performs inference over a maieutic tree of generated explanations. First, we introduce *logical integrity*, a key concept that is used to determine the reliability of propositions.

Language models often generate logically inconsistent propositions; for instance, in Figure 1, the model infers *True* when prompted with either *"One is a number that comes before zero."* or *"One is a number that comes after zero."*. In this sense, $p(True|Q)$ does not provide a reliable value to determine whether Q is true or not. We formalize this idea as *logical integrity*: a proposition Q is *logically integral* when the LM consistently infers the truth value of Q and $\neg Q$ (i.e. Q as *True* and $\neg Q$ as *False*, or vice versa). Formally, we define a boolean function $\texttt{integral}(E)$ as follows:[3]

$$\begin{aligned} &1.\ \underset{A \in \{T,F\}}{\text{argmax}}\, p_{LM}(A|E,C) = T \text{ and} \\ &\quad \underset{A \in \{T,F\}}{\text{argmax}}\, p_{LM}(A|\neg E,C) = F \\ &2.\ \underset{A \in \{T,F\}}{\text{argmax}}\, p_{LM}(A|E,C) = F \text{ and} \\ &\quad \underset{A \in \{T,F\}}{\text{argmax}}\, p_{LM}(A|\neg E,C) = T \\ &\texttt{integral}(E) = \mathbb{1}_{\{1 \text{ or } 2 \text{ is satisfied}\}}. \end{aligned} \tag{4}$$

A statement is considered to be *logically integral / True* when condition 1 is met, and *logically integral / False* when condition 2 is met. Intuitively, the truth values of logically integral propositions are more credible than non-integral ones, to which LMs are inconsistent given a simple negation. For example, *"One is a number that comes before zero."* in Figure 1 would not be logically integral, as the model assigns same truth value to both Q and $\neg Q$.

For the rest of section, we first search for logically integral propositions by constructing the maieutic tree (Section 3.1), then quantify the relations between the propositions (Section 3.2), based on which we infer the final answer (Section 3.3).

[3]Given E, $\neg E$ can be automatically generated simply by inserting a prefix (e.g. *It is wrong to say that*), or prompting *LM* to negate the given sentence.

3.1 Maieutic Tree Generation

3.1.1 Abductive Explanation Generation

Given a question, we require the LM to post-hoc rationalize both *True* and *False* labels. This abductive explanation generation has several advantages over an ad-hoc approach that first generates an explanation, then predicts the label. First, in the ad-hoc setting, the model is required to generate a discriminative explanation that helps in choosing one label over the other. Abductive generation (Bhagavatula et al., 2019), on the contrary, exposes the model to consider different possible answers rather than discriminating one, which often reveals an explanation that otherwise would not have been generated. Second, the label information would intuitively help LM elicit more specific explanations, mitigating the issue of a bland and generic generation which does not help the inference, a well-known weakness of LMs (Adiwardana et al., 2020).

Concretely, we define a function `abduction` which gets the statement Q as the input and outputs a tuple of two abductive explanations with *True*, *False* given as the answer, respectively:

$$\texttt{abduction}(Q) = (E_T, E_F) \quad \text{where } E_{A \in \{T,F\}} \sim p_{LM}(E|Q,A,C). \tag{5}$$

Figure 2 shows a concrete example of generating E_T given Q. With Q, we prompt the model to rationalize *True* as the answer: *"War cannot have a tie? True, because"*, which then is completed by an explanation by LM *"In a context of war, there's always a victor and a loser."*.

3.1.2 Depth-wise Knowledge Spanning

As shown in Figure 1, LM-generated explanations are noisy and inaccurate by nature. Prior works indirectly compensate for the untrustworthy generations by independently sampling multiple generations then aggregating them at the answer level (e.g. through majority voting; Wang et al., 2022). Despite better performance, such an aggregation could still be brittle, as the inference fundamentally depends on the correctness of 1-hop explanations.

To enhance the robustness of reasoning, we hypothesize that the inference process should entail not only the *breadth* of reasoning, but also the *depth* of reasoning - whether the reasoning paths themselves are credible and consistent with each other. To do this, we require the LM itself to validate its own generations - by recursively prompting the

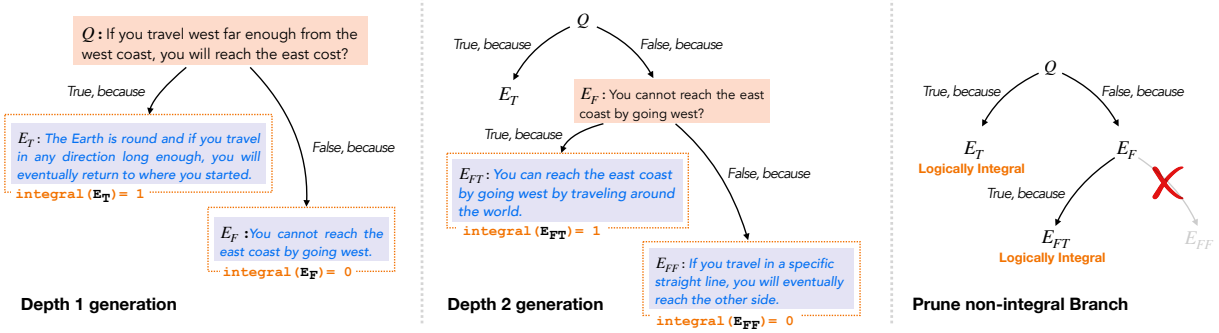


Figure 3: Illustrative example of *maieutic tree* generation, with the max tree depth set to 2. For visual clarity, we generate only 1 E_T and 1 E_F per question and omit the width-wise spanning of knowledge.

LM with the generated explanations. As Figure 2 shows, this corresponds to a depth-wise spanning of knowledge that induces a *maieutic tree*, a multi-depth structure of generated propositions and relations between them.

Let S_i denote the set of nodes at depth i in the maieutic tree $\mathcal{T}$. Each node in S_i is an explanation for an answer label (*True* or *False*), recursively generated given its parent node as the question:

$$S_i \subseteq \bigcup_{l \in \{T,F\}^{i-1}} \{E_{lT}, E_{lF}\}, \quad (E_{lT}, E_{lF}) = \texttt{abduction}(E_l). \tag{6}$$

Note that $\mathcal{T}$ is a full tree when the equality holds for all depths. For instance, in Figure 2, E_{TF} is generated by prompting the LM with its parent node E_T and *False*, i.e. $E_{TF} \sim p_{LM}(\cdot|E_T, F, C)$.

In practice, we sample multiple explanations with the same Q and A through nucleus sampling (Holtzman et al., 2019). This corresponds to the width-wise spanning of knowledge, enhancing the diversity and coverage of generated explanations.

3.1.3 When to Stop Generating

Generating a full tree could be computationally expensive, as the number of generation grows exponentially with the maximum tree depth. Therefore, in each branch, we stop generating further once we reach a logically integral proposition; intuitively, this aligns with our goal to identify propositions that can be validated by the LM with confidence.

Figure 3 illustrates an example of maieutic tree generation where the maximum depth of the tree is set to 2. For visual clarity, we only generate one explanation per Q and A. Given Q, we first generate E_T and E_F, then validate whether each of them is logically integral. Since E_T is logically integral, we stop generating in this branch, but continue generating from E_F which is not logically integral. After reaching the maximum depth, we prune the branches leading to leaf nodes that are still not logically integral. This way, the final tree *keeps only the generations that lead to a logically integral proposition.* We provide a formal description of the generation process in Appendix A.

3.2 Defining the Relations

Now that we have generated the maieutic tree, we seek to define the relations between propositions and quantify their strength into scalar weights. For illustration, assume that an LM has generated the following E_F for the given Q:

> Q: Captain Kirk is part of Star Wars?
> A: False, because ***Captain Kirk is a character in Star Trek.***

The generation can be logically interpreted as follows: (1) the LM believes that *Captain Kirk is a character in Star Trek*, (2) the LM believes that the proposition *Captain Kirk is a character in Star Trek* can be a reason to deny that *Captain Kirk is part of Star Wars*. Accordingly, we define *belief* and *consistency* to represent the two dimensions of the logical relationship.

Belief w_E corresponds to the LM's belief that the proposition E is true (and therefore, $\neg E$ is false). To quantify *belief*, we prompt the LM with E and $\neg E$ respectively as a question, then comparing the probability assigned to *True*:

$$w_E := \frac{p_{LM}(T|E,C) - p_{LM}(T|\neg E,C)}{p_{LM}(T|E,C) + p_{LM}(T|\neg E,C)}. \tag{7}$$

Note that calculating this does not require any additional prompting, as we already gained access to these values while checking for the logical integrity of each proposition.

Consistency $w_{E,Q,A}$ corresponds to the consistency of the generated E with the given Q and A. Intuitively, if the LM is logically consistent, the

likelihood of E being generated given an answer (e.g., E_F being generated given *False*) should be larger than its likelihood given the opposite answer (e.g., E_F being generated given *True*). Following this intuition, we compute the consistency as:

$$w_{E,Q,A} \coloneqq \frac{p_{LM}(E|Q,A,C)}{p_{LM}(E|Q,A,C) + p_{LM}(E|Q,\neg A,C)}. \quad (8)$$

3.3 Inference

The two types of relations formulate a set of unary and binary logical constraints, based on which we assign the truth values to all nodes in the maieutic tree $\mathcal{T}$, and in consequence, infer the answer to the original question. First, we represent $\mathcal{C}_{blf}$ as the set of unary constraints. For each leaf node E in $\mathcal{T}$,

$$c_{blf} = \begin{cases} E & \text{if } E \text{ is } \textit{logically integral / True} \\ \neg E & \text{if } E \text{ is } \textit{logically integral / False}. \end{cases} \quad (9)$$

Note that all the leaf nodes in $\mathcal{T}$ are logically integral, hence we can count on the credibility of *belief* for these nodes. We now define the set of all belief constraints $\mathcal{C}_{blf}$ as:

$$\mathcal{C}_{blf} = \{c_{blf} \text{ for } \forall E \in \text{leaf}(\mathcal{T})\}. \quad (10)$$

For example, the nodes E_F and E_{TF} in Figure 2 would have a belief constraint in $\mathcal{C}_{blf}$.

Likewise, for *consistency*, we define $\mathcal{C}_{con}$ as the set of binary constraints using logical implication. For each edge (E_l, E_{lA}) in $\mathcal{T}$,

$$c_{con} = \begin{cases} E_{lA} \rightarrow E_l & \text{if } A = \textit{True} \\ E_{lA} \rightarrow \neg E_l & \text{if } A = \textit{False} \end{cases} \quad (11)$$
$$\mathcal{C}_{con} = \{c_{con} \text{ for } \forall (E_l, E_{lA}) \in \text{edge}(\mathcal{T})\}.$$

Our objective is to assign the truth values for all Es and the root node Q in $\mathcal{T}$, such that we maximize

$$\sum_{c \in \mathcal{C}_{blf} \cup \mathcal{C}_{con}} w_c \cdot \mathbb{1}_{\{c=\text{True}\}}, \quad (12)$$

which sums up the weights of satisfied constraints.

This problem is naturally formulated as weighted MAX-SAT, which is a problem of determining truth values of variables that maximize the weight of satisfied clauses. The problem can be algorithmically solved using an off-the-shelf solver.

3.4 Verifier Model

One limitation of the consistency definition in Section 3.2 is that it only considers the relationship between a parent node and a child node. Since the definition builds upon the likelihood of each generation from an LM, we cannot take into account the relationships across branches, e.g. E_T and E_F in Figure 3. This motivates us to introduce a small NLI model as a verifier, which can infer the relationship between an arbitrary pair of nodes in $\mathcal{T}$. Following previous works (Minervini and Riedel, 2018; Wang et al., 2019), we convert the NLI labels into logical relations as following:

$$\begin{aligned} \texttt{Entail}(E_1, E_2) &: E_1 \rightarrow E_2 \\ \texttt{Contradict}(E_1, E_2) &: E_1 \rightarrow \neg E_2. \end{aligned} \quad (13)$$

For all pairs of nodes $(E_1, E_2) \in \text{node}(\mathcal{T})^2$, $E_1 \neq E_2$, we obtain either $E_1 \rightarrow E_2$ or $E_1 \rightarrow \neg E_2$ if E_1 entails or contradicts E_2. For NLI-based clauses, we fix the weights to 1.[4] While the objective function (Eq. 12) stays the same, $\mathcal{C}_{con}$ is now replaced with $\mathcal{C}_{NLI}$, a set of clauses induced by the verifier model.

4 Experiments

Datasets We evaluate MAIEUTIC PROMPTING on three commonsense reasoning and fact verification benchmarks in binary QA format: Com2Sense (Singh et al., 2021), CSQA 2.0 (Talmor et al., 2021), CREAK (Onoe et al., 2021). Despite the simple format, these datasets require a substantial amount of knowledge and robust reasoning, making them challenging even for the billion-scale fine-tuned LMs (Table 1).

Baselines We compare our method with both the few-shot prompting methods and supervised models. Along with the standard prompting, we include Chain of Thought (Wei et al., 2022), Self-Consistency (Wang et al., 2022) and Generated Knowledge Prompting (GKP) (Liu et al., 2021). For supervised models, we consider the strong baselines used for the respective dataset, such as T5 (Raffel et al., 2020), UnifiedQA (Khashabi et al., 2020) and Unicorn (Lourie et al., 2021).

Configuration Details For all prompting methods, we use the same set of 6 demonstration examples and the same version of GPT-3 (*text-davinci-001*) as the LM. We determine the hyperparameters of MAIEUTIC PROMPTING and baselines based on the dev set performance on the benchmarks. In maieutic tree generation, we set the maximum depth to 2. For depth 1, we use nucleus sampling ($p = 1.0$) (Holtzman et al., 2019) to generate 3 E_Ts

[4] We also tried using the label probability assigned by NLI model as weight, but fixing it to 1 yielded better results.

	Dataset	Com2Sense			CSQA 2.0		CREAK		
	Model	*dev*	*test*	*pairwise*	*dev*	*test*	*dev*	*test*	*contrast*
Supervised	RoBERTa-large (Liu et al., 2019)	62.8	59.4	33.3	-	-	80.6	80.3	61.5
	T5-large (Raffel et al., 2020)	62.8	60.6	41.8	53.8	54.6	-	-	-
	T5-3B (Raffel et al., 2020)	73.2	-	-	-	60.2	85.6	85.1	70.0
	UnifiedQA-3B (Khashabi et al., 2020)	75.1	**71.3**	**51.3**	-	-	-	-	-
	T5-11B (Raffel et al., 2020)	**77.2**	-	-	68.5	67.8	**89.5**	-	**75.2**
	Unicorn-11B (Lourie et al., 2021)	-	-	-	**69.9**	**70.2**	-	-	-
Prompting	Standard	58.1	-	-	54.1	-	60.3	-	55.2
	Chain of Thought (Wei et al., 2022)	61.6	-	-	59.6	-	64.8	-	59.4
	Self Consistency (Wang et al., 2022)	61.4	-	-	60.8	-	70.5	-	64.8
	GKP (Liu et al., 2021)	61.8	-	-	59.7	-	75.4	-	68.2
	MAIEUTIC PROMPTING (Ours)	**72.5**	**75.0**	**68.7**	**69.5**	**68.3**	**85.2**	**85.3**	**77.4**

Table 1: Experimental results of MAIEUTIC PROMPTING and baseline methods on three benchmark datasets. We differentiate supervised baselines (upper section) from prompting methods (lower section), and bold the best numbers for each section. MAIEUTIC PROMPTING with GPT-3 outperforms all prompting baselines with the same model, while being competitive against billion-scale supervised LMs.

and 3 E_Fs from Q. For depth 2, we use greedy decoding to generate 1 E_T and 1 E_F from each parent node. This constrains the generated tree to have at most 18 nodes excluding the original Q.[5] In Section 4.3, we conduct an ablation study on this depth-adaptive decoding scheme and analyze the effect of the tree size. For the main experiments, we use RoBERTa (Liu et al., 2019) fine-tuned on MNLI (Williams et al., 2018) as a verifier with 90.2% accuracy on MNLI dev set, and RC2 (Morgado et al., 2014) as a MAX-SAT solver.

4.1 Benchmark Performance

Table 1 presents overall evaluation results of MAIEUTIC PROMPTING along with the prompting and supervised baselines. MAIEUTIC PROMPTING significantly outperforms all prompting methods across all benchmarks. Notably, GKP and Self Consistency ensembled more 1-hop explanations than the maximal size of the maieutic tree; our superior performance compared to these methods confirms the sample efficiency of depth-wise knowledge spanning. Moreover, MAIEUTIC PROMPTING is the only prompting method that performs better than even the smallest supervised baseline (RoBERTa-large) in Com2Sense and CREAK. In fact, MAIEUTIC PROMPTING allows us to use an off-the-shelf LM to achieve comparable performance to a large *fine-tuned* LM by simply plugging in our inference algorithm. In Appendix C we also provide experiments on StrategyQA (Geva et al., 2021), to evaluate the generalizability of MAIEUTIC PROMPTING in multi-hop setting.

[5]Both GKP and Self Consistency employ an ensemble strategy, generating N different samples of explanations then aggregating their answers. For a fair comparison with ours, we set $N = 20$ for both methods, generating more explanations than the maximal possible size of the maieutic tree.

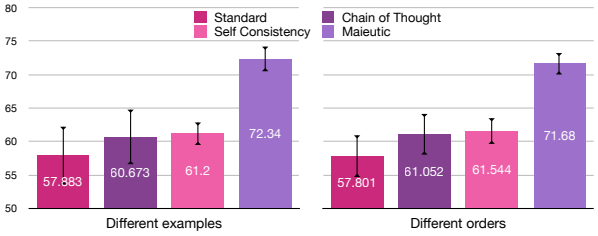


Figure 4: Robustness of prompting methods under different few-shot examples / different order of examples. We compare the mean and standard deviation of Com2Sense dev set accuracy.

4.2 Robustness Analysis

We perform additional analyses to understand the working of our method under semantic perturbations and different prompt formats.

Robustness to semantic perturbations In addition to the standard accuracy, we report two additional metrics called *pairwise accuracy* and *contrast set accuracy* in Table 1. In Com2Sense test set and CREAK contrast set, each question is paired with its complimentary counterpart, of which the surface form is similar but the answer should be the opposite (e.g. *"Barack Obama has daughters." vs "Barack Obama has no daughter."*), testing the models' robustness to semantic perturbations. In these metrics, the gap between MAIEUTIC PROMPTING and baselines widens substantially, indicating the robustness of our method against semantic perturbations.

Model	Accuracy
Non-abductive generation	68.4
All greedy decoding (no depth-adaptive)	67.2
All nucleus sampling (no depth-adaptive)	72.0
Likelihood-based consistency	65.6
Maieutic Prompting	**72.5**

Table 2: Ablation study on Com2Sense Dev set. The best configuration is with abductive generation, depth-adaptive decoding and verifier-based consistency.

Dimension	1	2	3	5	10
Depth	61.3	72.5	72.4	-	-
Width	62.4	66.5	72.5	71.5	72.1

Table 3: Performance of MAIEUTIC PROMPTING on Com2Sense with different maieutic tree sizes.

Robustness to different prompts Prior works revealed that prompting performance could be sensitive to few-shot examples and their order (Lu et al., 2021b; Zhao et al., 2021). We investigate whether this holds true for MAIEUTIC PROMPTING, as shown in Figure 4. We compare different prompting methods run with 3 different sets of few-shot examples (left), and 5 different permutations of the few-shot examples (right). In both settings, while Self Consistency and MAIEUTIC PROMPTING are much more stable then the other two, our method has slightly less variance.

4.3 Ablation Study

We ablate different components of MAIEUTIC PROMPTING to investigate their respective contributions as shown in Table 2.

Generation First, we consider MAIEUTIC PROMPTING without abductive generation — we generate each explanation without providing an answer label, i.e. in an identical fashion to Chain of Thought. In this setting, the performance of MAIEUTIC PROMPTING degrades by 4%, alluding to the importance of abductive generation in eliciting the latent knowledge from LM. Next, we ablate the depth-adaptive decoding mechanism (Section 4), by applying either greedy decoding or nucleus sampling for all depths of the maieutic tree. *All greedy decoding* restrains width-wise spanning of knowledge, hence leads to large degradation of performance. *All nucleus sampling* performs much more comparably with our best configuration, although the stochastic decoding produces slightly more errors in the explanations.

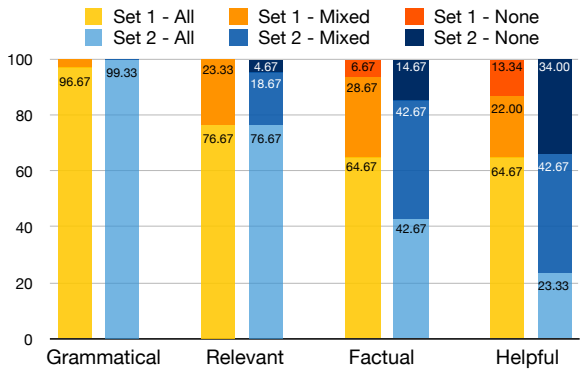


Figure 5: Human evaluation results (Krippendorff's alpha = 0.64; substantial inter-annotator agreement). To minimize subjectivity, we use a strict 3-level scale, where annotators choose *All* only when all the statements in the *true Es* are desirable (e.g. grammatical) on its own, *Mixed* when at least one E is undesirable, and *None* otherwise.

Consistency We ablate the NLI-based clauses and replace them with the original C_{con} discussed in Section 3.2. With the likelihood-based C_{con}, the accuracy reduces by about 7%, but still prevails over the prompting baselines in Table 1. The verifier model indeed benefits the inference process by providing more accurate relations between generated explanations, although our method performs competently even without the access to the verifier.

Effect of tree size We also investigate how the size of the maieutic tree influences the performance. In Table 3, we present the performance of MAIEUTIC PROMPTING on Com2Sense dev set with various values of maximal depth and width. In both dimensions, the accuracy saturates after a certain threshold. We attribute this to (1) the topic drift in generation which intensifies as the depth grows, (2) larger overlaps in generated knowledge as we sample more explanations width-wise.

4.4 Human Evaluation

We qualitatively analyze actual inference results of MAIEUTIC PROMPTING through human evaluation. For each sample, we first retrieve *true Es* (the set of generated Es that are inferred to be *True* by MAIEUTIC PROMPTING), then evaluate them over the four criteria from Liu et al. (2021): (1) *Grammaticality* of the explanations, (2) *Relevance* of the explanations to the question, (3) *Factuality*: whether the explanations states facts, and (4) *Helpfulness*: whether the explanation explicitly leads to the correct answer. Six NLP experts scored 100 examples sampled from CSQA 2.0 dev set, of which

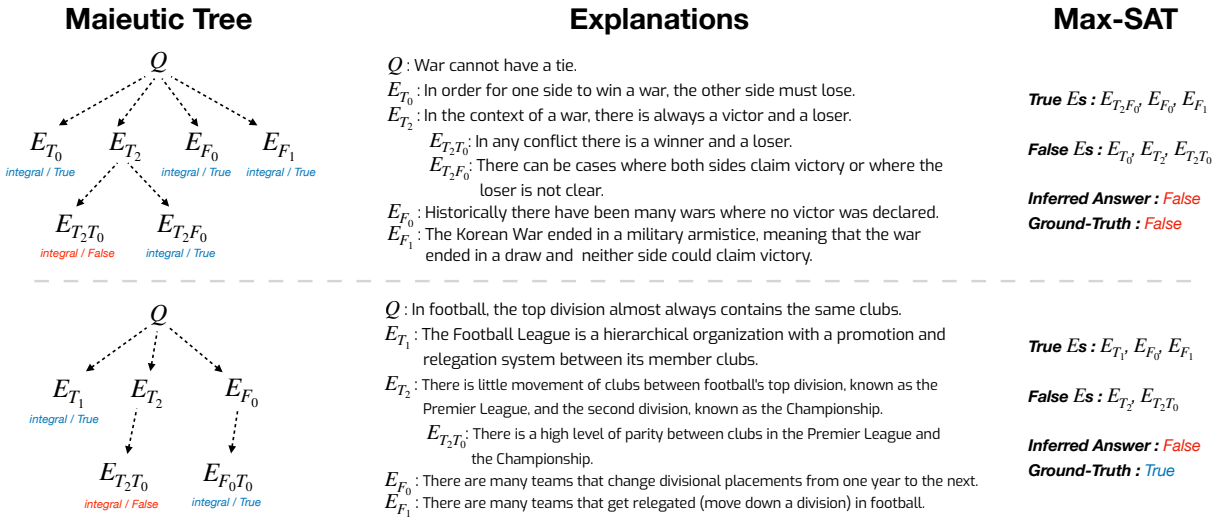


Figure 6: Examples of MAIEUTIC PROMPTING, each with correct and wrong answer. Even in the latter case, the generated explanations make sense toward the inferred answer. We provide more examples in Appendix D.

50 were answered correctly (Set 1) and 50 were answered wrongly by the model (Set 2).

Figure 5 presents the evaluation results. For both sets, over 99% of the *true Es* are grammatically perfect, and most of them provide relevant evidence to the question.[6] Surprisingly, the LM often generates both factual and helpful explanations even when its answer is different from the ground truth: 42% of the *true Es* for incorrectly answered examples are perfectly factual, and 23% of them are completely helpful in correctly answering the question. We find that in many of these cases, the questions did not have a clear-cut answer; as exemplified in Figure 6, the explanations generated and validated by MAIEUTIC PROMPTING are compelling enough as an alternative to the ground-truth answer.

5 Related Work

Prior works have leveraged natural language explanations (NLEs) to promote model reasoning, either by training a model to explain (Rajani et al., 2019; Camburu et al., 2018; Chen et al., 2022; Wiegreffe and Marasović, 2021), or generating answers to templated queries and distantly supervised rationales (Shwartz et al., 2020; Brahman et al., 2021). Incorporated with in-context learning (Brown et al., 2020; *inter alia*), these efforts have led to explanation-based prompting (Wei et al., 2022; Wang et al., 2022; Liu et al., 2021; Lampinen et al., 2022). Other works aim to improve model interpretability with NLEs, training a model that explains its inference post-hoc or in parallel with the answer (Camburu et al., 2018; Narang et al., 2020; Jacovi et al., 2021). Unlike these works, the explanations in our work are designed to be intrinsic (Du et al., 2019); the explanations themselves explicitly participate in the inference.

Meanwhile, recent observations reveal that LM explanations are unreliable, as they often lack logical consistency and are not factually grounded (Ye and Durrett, 2022; Kassner and Schütze, 2020). This is in part due to the broader limitations of generative LMs, which assign high probability to unlikely sentences (Welleck et al., 2020; Holtzman et al., 2021) and are sensitive to semantic perturbations (Elazar et al., 2021). MAIEUTIC PROMPTING overcomes these limitations by avoiding the use of explanations "as-is", and modeling the relationships between explanations.

Another line of works apply symbolic methods on top of LMs to improve their consistency, spanning from a lexical constraint on sequence decoding (Lu et al., 2021a) to a symbolic world model (Nye et al., 2021b) and discrete operations (Chen et al., 2019; Cobbe et al., 2021). Other works explore how to train a model that simulates the symbolic reasoning process, such as logical transformation (Bostrom et al., 2021) and consistent generation of beliefs (Kassner et al., 2021; Dalvi et al., 2022). However, these models require a curated set of human annotations that limits their application to specific domains. MAIEUTIC PROMPTING generalizes these neuro-symbolic approaches in an unsupervised setup, employing MAX-SAT algorithm to symbolically determine the true subset

[6] It is natural that some of the true Es are not directly relevant to Q, but still contribute to the inference by validating other Es.

from a noisy pool of neural generations.

6 Conclusion

In this work, we propose MAIEUTIC PROMPTING, a novel few-shot inference method inspired by the Socratic way of conversation. We systematically generate a tree of explanations that bear logical relations between each other, then find the truth values that max-satisfy these relations. Empirical results show that MAIEUTIC PROMPTING is both competitive and robust compared to diverse baselines, while providing intrinsic interpretations over its inference.

Limitations

Extension to different task formats In this work, we limit our experiments to validating a given statement. In future works, we aim to extend our method over a broader range of tasks, e.g. multiple-choice QA. A potential strategy could be binarizing multiple-choice options to respective statements and scoring them with MAIEUTIC PROMPTING, e.g. using the sum of weight of satisfied clauses from MAX-SAT.

Modeling relationships between trees MAIEUTIC PROMPTING models the relations between the nodes in each maieutic tree to infer a consistent answer. The scope of modeled relationships, however, could be further generalized beyond a single tree - a span of knowledge generated for one question could serve as the evidence for another question. Indeed, modeling the relationship between questions is an active area of research (Kossen et al., 2021). We envision that the knowledge elicited from MAIEUTIC PROMPTING could further be enriched through this type of generalization.

Acknowledgements

This work was funded in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) (funding reference number 401233309), DARPA MCS program through NIWC Pacific (N66001-19-2-4031), and the Allen Institute for AI. We also thank OpenAI for providing access to the GPT-3 API.

References

Daniel Adiwardana, Minh-Thang Luong, David R So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, et al. 2020. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*.

Roberto Battiti. 2009. *Maximum satisfiability problem*, pages 2035–2041. Springer US, Boston, MA.

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Scott Wen-tau Yih, and Yejin Choi. 2019. Abductive commonsense reasoning. *arXiv preprint arXiv:1908.05739*.

Kaj Bostrom, Xinyu Zhao, Swarat Chaudhuri, and Greg Durrett. 2021. Flexible generation of natural language deductions. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6266–6278.

Faeze Brahman, Vered Shwartz, Rachel Rudinger, and Yejin Choi. 2021. Learning to rationalize for nonmonotonic reasoning with distant supervision. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 12592–12601. AAAI Press.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris 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*, volume 33, pages 1877–1901. Curran Associates, Inc.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: Natural language inference with natural language explanations. *Advances in Neural Information Processing Systems*, 31.

Howard Chen, Jacqueline He, Karthik Narasimhan, and Danqi Chen. 2022. Can rationalization improve robustness? *arXiv preprint arXiv:2204.11790*.

Xinyun Chen, Chen Liang, Adams Wei Yu, Denny Zhou, Dawn Song, and Quoc V Le. 2019. Neural symbolic reader: Scalable integration of distributed and symbolic representations for reading comprehension. In *International Conference on Learning Representations*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Bhavana Dalvi, Oyvind Tafjord, and Peter Clark. 2022. Towards teachable reasoning systems. *arXiv preprint arXiv:2204.13074*.

Mengnan Du, Ninghao Liu, and Xia Hu. 2019. Techniques for interpretable machine learning. *Communications of the ACM*, 63(1):68–77.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. 2021. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031.

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. *Transactions of the Association for Computational Linguistics*, 9:346–361.

Robert G. M. Hausmann and Kurt VanLehn. 2007. Explaining self-explaining: A contrast between content and generation. In *AIED*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. 2021. Surface form competition: Why the highest probability answer isn't always right. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7038–7051.

Alon Jacovi, Swabha Swayamdipta, Shauli Ravfogel, Yanai Elazar, Yejin Choi, and Yoav Goldberg. 2021. Contrastive explanations for model interpretability. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1597–1611.

Nora Kassner and Hinrich Schütze. 2020. Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7811–7818, Online. Association for Computational Linguistics.

Nora Kassner, Oyvind Tafjord, Hinrich Schütze, and Peter Clark. 2021. Beliefbank: Adding memory to a pre-trained language model for a systematic notion of belief. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8849–8861.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. Unifiedqa: Crossing format boundaries with a single qa system. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1896–1907.

Jannik Kossen, Neil Band, Clare Lyle, Aidan N Gomez, Thomas Rainforth, and Yarin Gal. 2021. Self-attention between datapoints: Going beyond individual input-output pairs in deep learning. *Advances in Neural Information Processing Systems*, 34:28742–28756.

Klaus Krippendorff. 2007. Computing krippendorff's alpha-reliability. annenberg school for communication departmental paper 43.

Andrew K Lampinen, Ishita Dasgupta, Stephanie CY Chan, Kory Matthewson, Michael Henry Tessler, Antonia Creswell, James L McClelland, Jane X Wang, and Felix Hill. 2022. Can language models learn from explanations in context? *arXiv preprint arXiv:2204.02329*.

Jiacheng Liu, Alisa Liu, Ximing Lu, Sean Welleck, Peter West, Ronan Le Bras, Yejin Choi, and Hannaneh Hajishirzi. 2021. Generated knowledge prompting for commonsense reasoning. *arXiv preprint arXiv:2110.08387*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Nicholas Lourie, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Unicorn on rainbow: A universal commonsense reasoning model on a new multitask benchmark. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13480–13488.

Ximing Lu, Peter West, Rowan Zellers, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021a. Neurologic decoding:(un) supervised neural text generation with predicate logic constraints. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4288–4299.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2021b. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *arXiv preprint arXiv:2104.08786*.

Pasquale Minervini and Sebastian Riedel. 2018. Adversarially regularising neural nli models to integrate logical background knowledge. *arXiv preprint arXiv:1808.08609*.

António Morgado, Carmine Dodaro, and Joao Marques-Silva. 2014. Core-guided maxsat with soft cardinality constraints. In *International Conference on Principles and Practice of Constraint Programming*, pages 564–573. Springer.

Sharan Narang, Colin Raffel, Katherine Lee, Adam Roberts, Noah Fiedel, and Karishma Malkan. 2020. Wt5?! training text-to-text models to explain their predictions. *arXiv preprint arXiv:2004.14546*.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, et al. 2021a. Show your work: Scratchpads for intermediate computation with language models. *arXiv preprint arXiv:2112.00114*.

Maxwell Nye, Michael Tessler, Josh Tenenbaum, and Brenden M Lake. 2021b. Improving coherence and consistency in neural sequence models with dual-system, neuro-symbolic reasoning. *Advances in Neural Information Processing Systems*, 34.

Yasumasa Onoe, Michael J.Q. Zhang, Eunsol Choi, and Greg Durrett. 2021. Creak: A dataset for commonsense reasoning over entity knowledge. *OpenReview*.

Charles Sanders Peirce. 1974. *Collected papers of charles sanders peirce*, volume 5. Harvard University Press.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! leveraging language models for commonsense reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4932–4942.

Vered Shwartz, Peter West, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Unsupervised commonsense question answering with self-talk. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4615–4629.

Shikhar Singh, Nuan Wen, Yu Hou, Pegah Alipoormolabashi, Te-lin Wu, Xuezhe Ma, and Nanyun Peng. 2021. COM2SENSE: A commonsense reasoning benchmark with complementary sentences. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 883–898, Online. Association for Computational Linguistics.

Alon Talmor, Ori Yoran, Ronan Le Bras, Chandra Bhagavatula, Yoav Goldberg, Yejin Choi, and Jonathan Berant. 2021. CommonsenseQA 2.0: Exposing the limits of AI through gamification. In *Thirty-fifth Conference on Neural Information Processing Systems*.

Gregory Vlastos. 1991. *Socrates, ironist and moral philosopher*, volume 50. Cornell University Press.

Haohan Wang, Da Sun, and Eric P Xing. 2019. What if we simply swap the two text fragments? a straightforward yet effective way to test the robustness of methods to confounding signals in nature language inference tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7136–7143.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

Sean Welleck, Ilia Kulikov, Jaedeok Kim, Richard Yuanzhe Pang, and Kyunghyun Cho. 2020. Consistency of a recurrent language model with respect to incomplete decoding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5553–5568.

Sarah Wiegreffe and Ana Marasović. 2021. Teach me to explain: A review of datasets for explainable nlp. *arXiv preprint arXiv:2102.12060*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122.

Xi Ye and Greg Durrett. 2022. The unreliability of explanations in few-shot in-context learning. *arXiv preprint arxiv:2205.03401*.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*, pages 12697–12706. PMLR.

A Tree Generation Algorithm

Algorithm 1 Maieutic tree generation

Input: Question Q, Max tree depth D
Output: Maieutic tree $\mathcal{T}$

```
T ← init(Q)  // initialize the tree with Q
for d ∈ {1, ··· , D} do  // generate nodes
    S_i ← ∅
    for E ∈ S_{i−1} do
        if integral(E) = 1 then
            S_i ← S_i ∪ abductive(E)
        end if
    end for
    T.add(S_i)
end for
V ← {E; integral(E) = 0 for all E ∈ leaf(T)}  // set of non-integral leaf nodes
while V ≠ ∅ do
    T.remove(V)  // prune the non-integral leaf nodes
    V ← {E; integral(E) = 0 for all E ∈ leaf(T)}
end while
```

B Dataset Details

Dataset	Com2Sense	CSQA 2.0	CREAK
Train / Dev / Test split size	804 / 402 / 2779	9282 / 2544 / 2517	10176 / 1371 / 1371
Average # of tokens	21	11.3 (words)	10.8

Table 4: We evaluate MAIEUTIC PROMPTING in three commonsense reasoning and fact verification benchmarks - Com2Sense, CSQA 2.0 and CREAK. Com2Sense and CSQA 2.0 consist of adversarial commonsense questions generated to mislead a proxy model. CREAK tests for a combination of commonsense reasoning and accurate fact retrieval, consisting of long-tail questions such as *"Harry Potter can teach how to fly on a broomstick?"*. Table 4 presents key statistics of the three datasets.

C Multi-hop Reasoning on StrategyQA

Model	Standard	C-o-T	Maieutic	C-o-T (Multi-hop)	Maieutic (Multi-hop)
Accuracy	56.3	58.2	60.7	57.9	61.4

Table 5: Results on StrategyQA

To further evaluate the generalizability of MAIEUTIC PROMPTING, we conduct additional experiments on multi-hop reasoning over StrategyQA (Geva et al., 2021) dev split. Note that the original evaluation setting for StrategyQA presupposes access to Wikipedia articles, from which the gold knowledge could be retrieved from; hence the benchmark as-is may not represent the best evaluation setting for few-shot prompting methods.

To better address the multi-hop nature of the dataset, we add a straightforward adjustment to both C-o-T and Maieutic Prompting, to first decompose the original question into 2-3 minor questions and then generate the explanation and answer. We denote this as **Multi-hop** in Table 5.

Consistent with the original experimental results, Maieutic Prompting yields promising improvement compared to both the standard / C-o-T prompting. The result attests to both the generalizability of Maieutic Prompting to multi-hop setting and the importance of reasoning algorithm in challenging scenarios.

D Inference Examples

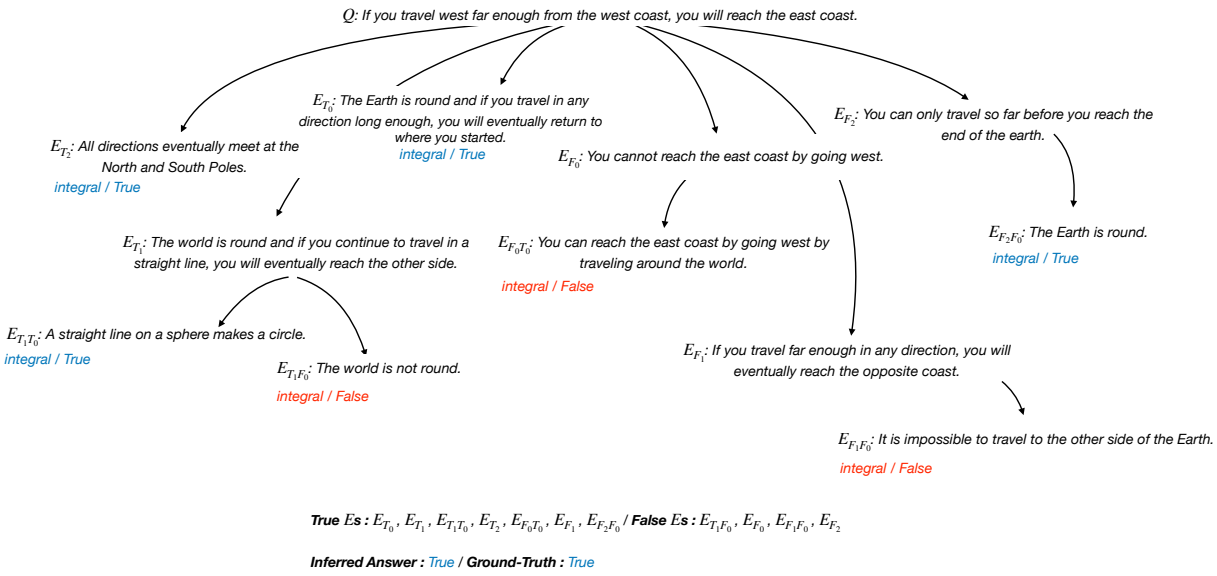


Figure 7: Example of correct inference by MAIEUTIC PROMPTING. We show the generated maieutic tree along with the assigned truth-values to each propositions.

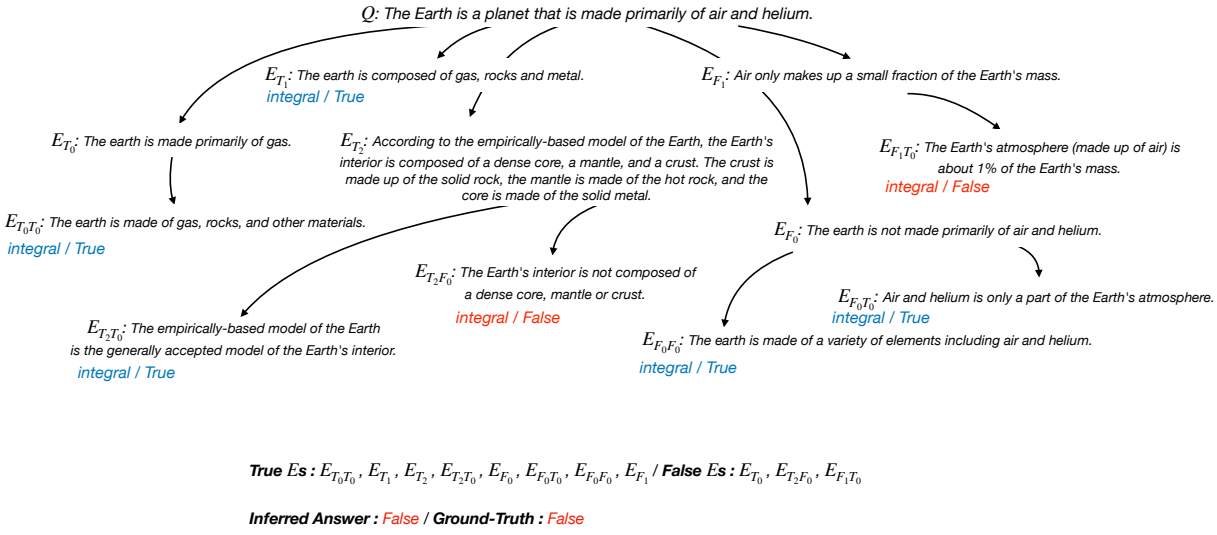
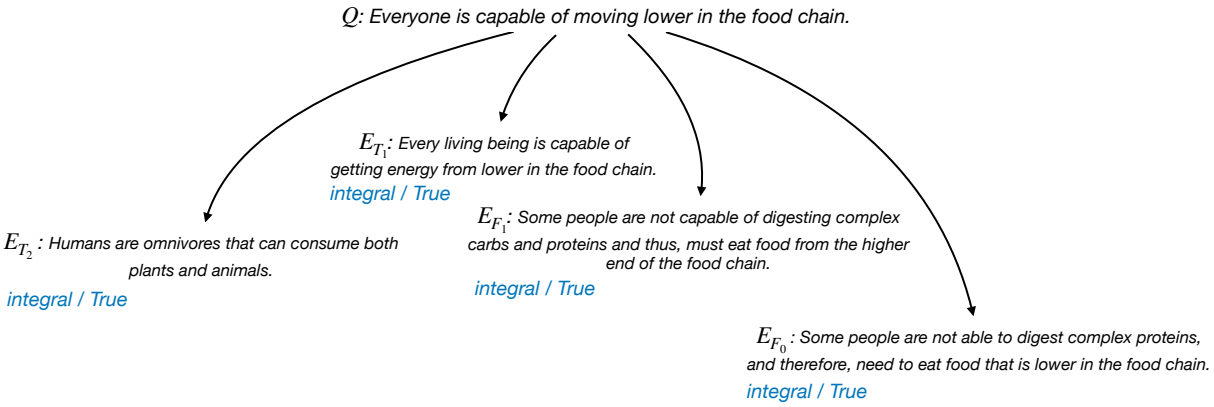


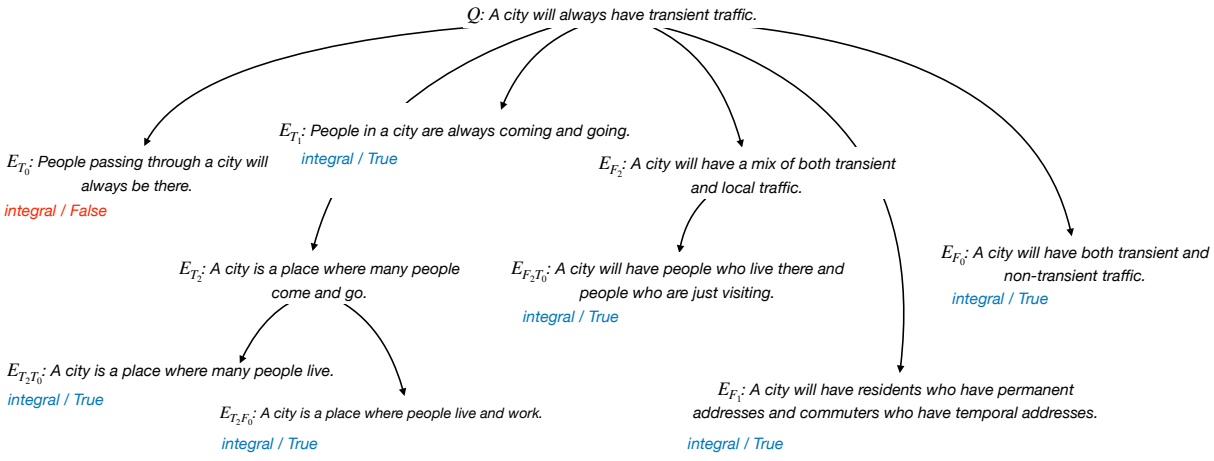
Figure 8: (continued) Example of correct inference by MAIEUTIC PROMPTING.



True Es : E_{T_1} ,E_{T_2}, E_{F_0}, E_{F_1}

Inferred Answer : *True* / ***Ground-Truth :*** *False*

Figure 9: Example of incorrect inference by MAIEUTIC PROMPTING.



True Es : E_{T_1} , E_{T_2} , $E_{T_2T_0}$, $E_{T_2F_0}$, E_{F_0} , E_{F_1} , E_{F_2} , $E_{F_2T_0}$ / ***False Es :*** E_{T_0}

Inferred Answer : *False* / ***Ground-Truth :*** *True*

Figure 10: (continued) Example of incorrect inference by MAIEUTIC PROMPTING.